100% EU PRODUCED RAN & OWNED

Fortnox API Data Model

for use with Invantive SQL

24.0

invantive the SQL company

Copyright

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

Despite all the care taken in the compilation of this text, neither the author nor the publisher can accept liability for any damage, which might result from any error, which might appear in this publication.

This manual is a reference guide intended to clarify usage. If data in the sample images match data in your system, the similarity is coincidental.

Important Safety and Usage Information

Intended Use and Limitations: This software, developed by Invantive, is designed to support a variety of business and information technology data processing functions, such as accounting, financial reporting an sales reporting. It is important to note that this software is not designed, tested, or approved for use in environments where malfunction or failure could lead to life-threatening situations or severe physical or environmental damage. This includes, but is not limited to:

- Nuclear facilities: The software should not be used for operations or functions related to the control, maintenance, or operation of nuclear facilities.
- Defense and Military Applications: This software is not suitable for use in defense-related applications, including but not limited to weaponry control, military strategy planning, or any other aspects of national defense.
- Aviation: The software is not intended for use in the operation, navigation, or communication systems of any aircraft or air traffic control environments.
- Healthcare and Medicine Production: This software should not be utilized for medical device operation, patient data analysis for critical health decisions, pharmaceutical production, or medical research where its failure or malfunction could impact patient health.
- Chemical and Hazardous Material Handling: This software is not intended for the management, control, or operational aspects of chemical plants or hazardous material handling facilities. Any malfunction in software used in these settings could result in dangerous chemical spills, explosions, or environmental disasters.
- Transportation and Traffic Control Systems: The software should not be used for the control, operation, or management of transportation systems, including railway signal controls, subway systems, or traffic light management. Malfunctions in such critical systems could lead to severe accidents and endanger public safety.
- Energy Grid and Utility Control Systems: This software is not designed for the control or operation of energy grid systems, including electrical substations, renewable energy control systems, or water utility control systems. The failure of software in these areas could lead to significant power outages, water supply disruptions, or other public utility failures, potentially endangering communities and causing extensive damage.
- Other High-Risk Environments: Any other critical infrastructure and environments where a failure of the software could result in significant harm to individuals or the environment.

User Responsibility: Users must ensure that they understand the intended use of the software and refrain from deploying it in any setting that falls outside of its designed purpose. It is the responsibility of the user to assess the suitability of the software for their intended application, especially in any scenarios that might pose a risk to life, health, or the environment.
Disclaimer of Liability: Invantive disclaims any responsibility for damage, injury, or legal consequences resulting from the use or misuse of this software in prohibited or unintended applications.

Contents

1 SQL Driver for Fortnox API 1

2 SQL Driver Attributes for Fortnox API 2

3 Catalog: Fortnox 17

3.1 Schemas ... **17**

3.1.1 Schema: AbsenceTransactions ... 17
3.1.2 Schema: AccountCharts ... 22
3.1.3 Schema: Accounts ... 23
3.1.4 Schema: Archive ... 28
3.1.5 Schema: ArticleFileConnections ... 34
3.1.6 Schema: Articles ... 36
3.1.7 Schema: ArticleUrlConnections ... 41
3.1.8 Schema: AssetFileConnection ... 44
3.1.9 Schema: Assets ... 45
3.1.10 Schema: AssetTypes ... 53
3.1.11 Schema: AttendanceTransactions ... 57
3.1.12 Schema: CompanyInformation ... 62
3.1.13 Schema: CompanySettings ... 63
3.1.14 Schema: ContractAccruals ... 66
3.1.15 Schema: Contracts ... 70
3.1.16 Schema: ContractTemplates ... 80
3.1.17 Schema: CostCenters ... 85
3.1.18 Schema: Currencies ... 88
3.1.19 Schema: CustomerReferences ... 90
3.1.20 Schema: Customers ... 95
3.1.21 Schema: Employees ... 101
3.1.22 Schema: EUVatLimitRegulation ... 137
3.1.23 Schema: Expenses ... 139
3.1.24 Schema: Fileattachments_Attachment ... 141
3.1.25 Schema: FinanceInvoices ... 144
3.1.26 Schema: FinancialYears ... 146
3.1.27 Schema: Inbox ... 149
3.1.28 Schema: IntegrationDeveloper_IntegrationRatings ... 154
3.1.29 Schema: IntegrationDeveloper_IntegrationSales ... 156
3.1.30 Schema: IntegrationPartner_IntegrationSales ... 157
3.1.31 Schema: InvoiceAccruals ... 170
3.1.32 Schema: InvoicePayments ... 175
3.1.33 Schema: Invoices ... 181
3.1.34 Schema: Labels ... 219
3.1.35 Schema: LockedPeriod ... 220
3.1.36 Schema: Me ... 221
3.1.37 Schema: ModesOfPayments ... 223
3.1.38 Schema: Native ... 225
3.1.39 Schema: Offers ... 227
3.1.40 Schema: Orders ... 241
3.1.41 Schema: PredefinedAccounts ... 256
3.1.42 Schema: PredefinedVoucherSeries ... 259
3.1.43 Schema: PriceLists ... 261
3.1.44 Schema: Prices ... 264
3.1.45 Schema: PrintTemplates ... 270
3.1.46 Schema: Projects ... 271

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

3.1.47 Schema: SalaryTransactions ... 274
3.1.48 Schema: ScheduleTimes ... 277
3.1.49 Schema: Sie ... 279
3.1.50 Schema: SupplierInvoiceAccruals ... 280
3.1.51 Schema: SupplierInvoiceExternalUrlConnections ... 285
3.1.52 Schema: SupplierInvoiceFileConnections ... 286
3.1.53 Schema: SupplierInvoicePayments ... 289
3.1.54 Schema: SupplierInvoices ... 295
3.1.55 Schema: Suppliers ... 308
3.1.56 Schema: TaxReductions ... 313
3.1.57 Schema: TermsOfDeliveries ... 318
3.1.58 Schema: TermsOfPayments ... 321
3.1.59 Schema: TimeReporting_Articles ... 323
3.1.60 Schema: TimeReporting_Registrations ... 341
3.1.61 Schema: TrustedEmailSenders ... 349
3.1.62 Schema: Units ... 353
3.1.63 Schema: VacationDebtBasis ... 355
3.1.64 Schema: Views ... 359
3.1.65 Schema: VoucherFileConnections ... 361
3.1.66 Schema: Vouchers ... 364
3.1.67 Schema: VoucherSeries ... 372
3.1.68 Schema: Warehouse_CustomDocumentType ... 375
3.1.69 Schema: Warehouse_CustomInboundDocument ... 377
3.1.70 Schema: Warehouse_CustomOutboundDocument ... 381
3.1.71 Schema: Warehouse_IncomingGoods ... 386
3.1.72 Schema: Warehouse_ManualDocument ... 392
3.1.73 Schema: Warehouse_ManualInboundDocument ... 394
3.1.74 Schema: Warehouse_ManualOutboundDocument ... 397
3.1.75 Schema: Warehouse_ProductionOrder ... 401
3.1.76 Schema: Warehouse_PurchaseOrder ... 410
3.1.77 Schema: Warehouse_StockPoint ... 437
3.1.78 Schema: Warehouse_StockStatus ... 449
3.1.79 Schema: Warehouse_StockTaking ... 450
3.1.80 Schema: Warehouse_StockTransfer ... 461
3.1.81 Schema: Warehouse_Tenant ... 464
3.1.82 Schema: WayOfDeliveries ... 465

4 Package: dcr_metadata 468

4.1 Procedures ... 468
4.1.1 dcr_metadata.get_partitions: Fortnox Data container metadata package ... 468

5 Package: fnx_fileattachments_attachment 468

5.1 Procedures ... 468
5.1.1 fnx_fileattachments_attachment.attach: Fortnox Package for fileattachments_Attachment operations that could not be mapped to a ta
5.1.2 fnx_fileattachments_attachment.detach: Fortnox Package for fileattachments_Attachment operations that could not be mapped to a ta
5.1.3 fnx_fileattachments_attachment.updateattachment: Fortnox Package for fileattachments_Attachment operations that could not be ma
5.1.4 fnx_fileattachments_attachment.validateincludedonsend: Fortnox Package for fileattachments_Attachment operations that could not b

6 Package: fnx_fortnox_archive 471

6.1 Procedures ... 471
6.1.1 fnx_fortnox_archive.removebyid: Fortnox Package for fortnox_Archive operations that could not be mapped to a table.
6.1.2 fnx_fortnox_archive.removebypath: Fortnox Package for fortnox_Archive operations that could not be mapped to a table.

7 Package: fnx_fortnox_assets 472

7.1 Procedures ... 472
7.1.1 fnx_fortnox_assets.changeob: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.
7.1.2 fnx_fortnox_assets.delete: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

7.1.3 fnx_fortnox_assets.depreciate: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.
7.1.4 fnx_fortnox_assets.scrap: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.
7.1.5 fnx_fortnox_assets.sell: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.
7.1.6 fnx_fortnox_assets.update: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.
7.1.7 fnx_fortnox_assets.w ritedow n: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.
7.1.8 fnx_fortnox_assets.w riteup: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.

8 Package: fnx_fortnox_contracts 477

8.1 Procedures 477
8.1.1 fnx_fortnox_contracts.createinvoice: Fortnox Package for fortnox_Contracts operations that could not be mapped to a table.
8.1.2 fnx_fortnox_contracts.finish: Fortnox Package for fortnox_Contracts operations that could not be mapped to a table.
8.1.3 fnx_fortnox_contracts.increaseinvoicecount: Fortnox Package for fortnox_Contracts operations that could not be mapped to a table.

9 Package: fnx_fortnox_contracttemplates 479

9.1 Procedures 479
9.1.1 fnx_fortnox_contracttemplates.update: Fortnox Package for fortnox_ContractTemplates operations that could not be mapped to a tab

10 Package: fnx_fortnox_customerreferences 480

10.1 Procedures 480
10.1.1 fnx_fortnox_customerreferences.remove: Fortnox Package for fortnox_CustomerReferences operations that could not be mapped to
10.1.2 fnx_fortnox_customerreferences.update: Fortnox Package for fortnox_CustomerReferences operations that could not be mapped to

11 Package: fnx_fortnox_financeinvoices 482

11.1 Procedures 482
11.1.1 fnx_fortnox_financeinvoices.create: Fortnox Package for fortnox_FinanceInvoices operations that could not be mapped to a table.
11.1.2 fnx_fortnox_financeinvoices.pause: Fortnox Package for fortnox_FinanceInvoices operations that could not be mapped to a table.
11.1.3 fnx_fortnox_financeinvoices.reportpayment: Fortnox Package for fortnox_FinanceInvoices operations that could not be mapped to a
11.1.4 fnx_fortnox_financeinvoices.stop: Fortnox Package for fortnox_FinanceInvoices operations that could not be mapped to a table.
11.1.5 fnx_fortnox_financeinvoices.takefees: Fortnox Package for fortnox_FinanceInvoices operations that could not be mapped to a table.
11.1.6 fnx_fortnox_financeinvoices.unpause: Fortnox Package for fortnox_FinanceInvoices operations that could not be mapped to a table.

12 Package: fnx_fortnox_inbox 486

12.1 Procedures 486
12.1.1 fnx_fortnox_inbox.remove: Fortnox Package for fortnox_Inbox operations that could not be mapped to a table.

13 Package: fnx_fortnox_invoicepayments 487

13.1 Procedures 487
13.1.1 fnx_fortnox_invoicepayments.bookkeep: Fortnox Package for fortnox_InvoicePayments operations that could not be mapped to a tab

14 Package: fnx_fortnox_invoices 488

14.1 Procedures 488
14.1.1 fnx_fortnox_invoices.bookkeep: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.
14.1.2 fnx_fortnox_invoices.cancel: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.
14.1.3 fnx_fortnox_invoices.credit: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.
14.1.4 fnx_fortnox_invoices.einvoice: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.
14.1.5 fnx_fortnox_invoices.eprint: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.
14.1.6 fnx_fortnox_invoices.externalprint: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.
14.1.7 fnx_fortnox_invoices.printreminder: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.
14.1.8 fnx_fortnox_invoices.w arehouseready: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.

15 Package: fnx_fortnox_offers 493

15.1 Procedures 493
15.1.1 fnx_fortnox_offers.cancel: Fortnox Package for fortnox_Offers operations that could not be mapped to a table.
15.1.2 fnx_fortnox_offers.createinvoice: Fortnox Package for fortnox_Offers operations that could not be mapped to a table.
15.1.3 fnx_fortnox_offers.createorder: Fortnox Package for fortnox_Offers operations that could not be mapped to a table.
15.1.4 fnx_fortnox_offers.email: Fortnox Package for fortnox_Offers operations that could not be mapped to a table.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

15.1.5 fnx_fortnox_offers.externalprint: Fortnox Package for fortnox_Offers operations that could not be mapped to a table.
15.1.6 fnx_fortnox_offers.preview : Fortnox Package for fortnox_Offers operations that could not be mapped to a table.
15.1.7 fnx_fortnox_offers.print: Fortnox Package for fortnox_Offers operations that could not be mapped to a table.

16 Package: fnx_fortnox_orders 497

16.1 Procedures 497
16.1.1 fnx_fortnox_orders.cancel: Fortnox Package for fortnox_Orders operations that could not be mapped to a table.
16.1.2 fnx_fortnox_orders.createinvoice: Fortnox Package for fortnox_Orders operations that could not be mapped to a table.
16.1.3 fnx_fortnox_orders.email: Fortnox Package for fortnox_Orders operations that could not be mapped to a table.
16.1.4 fnx_fortnox_orders.externalprint: Fortnox Package for fortnox_Orders operations that could not be mapped to a table.
16.1.5 fnx_fortnox_orders.preview : Fortnox Package for fortnox_Orders operations that could not be mapped to a table.
16.1.6 fnx_fortnox_orders.print: Fortnox Package for fortnox_Orders operations that could not be mapped to a table.

17 Package: fnx_fortnox_prices 501

17.1 Procedures 501
17.1.1 fnx_fortnox_prices.remove: Fortnox Package for fortnox_Prices operations that could not be mapped to a table.
17.1.2 fnx_fortnox_prices.update: Fortnox Package for fortnox_Prices operations that could not be mapped to a table.
17.1.3 fnx_fortnox_prices.update_2: Fortnox Package for fortnox_Prices operations that could not be mapped to a table.

18 Package: fnx_fortnox_scheduletimes 504

18.1 Procedures 504
18.1.1 fnx_fortnox_scheduletimes.reset: Fortnox Package for fortnox_ScheduleTimes operations that could not be mapped to a table.
18.1.2 fnx_fortnox_scheduletimes.update: Fortnox Package for fortnox_ScheduleTimes operations that could not be mapped to a table.

19 Package: fnx_fortnox_supplierinvoiceexternalurlconnections 505

19.1 Procedures 505
19.1.1 fnx_fortnox_supplierinvoiceexternalurlconnections.create: Fortnox Package for fortnox_SupplierInvoiceExternalUrlConnections opera
19.1.2 fnx_fortnox_supplierinvoiceexternalurlconnections.delete: Fortnox Package for fortnox_SupplierInvoiceExternalUrlConnections opera
19.1.3 fnx_fortnox_supplierinvoiceexternalurlconnections.update: Fortnox Package for fortnox_SupplierInvoiceExternalUrlConnections oper

20 Package: fnx_fortnox_supplierinvoicepayments 506

20.1 Procedures 506
20.1.1 fnx_fortnox_supplierinvoicepayments.bookkeep: Fortnox Package for fortnox_SupplierInvoicePayments operations that could not be

21 Package: fnx_fortnox_supplierinvoices 507

21.1 Procedures 507
21.1.1 fnx_fortnox_supplierinvoices.approvalbookkeep: Fortnox Package for fortnox_SupplierInvoices operations that could not be mapped
21.1.2 fnx_fortnox_supplierinvoices.approvalpayment: Fortnox Package for fortnox_SupplierInvoices operations that could not be mapped t
21.1.3 fnx_fortnox_supplierinvoices.bookkeep: Fortnox Package for fortnox_SupplierInvoices operations that could not be mapped to a table
21.1.4 fnx_fortnox_supplierinvoices.cancel: Fortnox Package for fortnox_SupplierInvoices operations that could not be mapped to a table.
21.1.5 fnx_fortnox_supplierinvoices.credit: Fortnox Package for fortnox_SupplierInvoices operations that could not be mapped to a table.

22 Package: fnx_fortnox_trustedemailsenders 509

22.1 Procedures 509
22.1.1 fnx_fortnox_trustedemailsenders.create: Fortnox Package for fortnox_TrustedEmailSenders operations that could not be mapped to a
22.1.2 fnx_fortnox_trustedemailsenders.remove: Fortnox Package for fortnox_TrustedEmailSenders operations that could not be mapped to

23 Package: fnx_warehouse_custominbounddocumen 510

23.1 Procedures 510
23.1.1 fnx_warehouse_custominbounddocument.release: Fortnox Package for warehouse_CustomInboundDocument operations that could
23.1.2 fnx_warehouse_custominbounddocument.save: Fortnox Package for warehouse_CustomInboundDocument operations that could no
23.1.3 fnx_warehouse_custominbounddocument.voiddocument: Fortnox Package for warehouse_CustomInboundDocument operations that

24 Package: fnx_warehouse_customoutbounddocume 513

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

24.1 Procedures 513
24.1.1 fnx_warehouse_customoutbounddocument.release: Fortnox Package for warehouse_CustomOutboundDocument operations that co
24.1.2 fnx_warehouse_customoutbounddocument.save: Fortnox Package for warehouse_CustomOutboundDocument operations that could
24.1.3 fnx_warehouse_customoutbounddocument.voiddocument: Fortnox Package for warehouse_CustomOutboundDocument operations

25 Package: fnx_warehouse_incominggoods 515

25.1 Procedures 515
25.1.1 fnx_warehouse_incominggoods.completed: Fortnox Package for warehouse_IncomingGoods operations that could not be mapped to
25.1.2 fnx_warehouse_incominggoods.patch: Fortnox Package for warehouse_IncomingGoods operations that could not be mapped to a ta
25.1.3 fnx_warehouse_incominggoods.release: Fortnox Package for warehouse_IncomingGoods operations that could not be mapped to a
25.1.4 fnx_warehouse_incominggoods.voiddocument: Fortnox Package for warehouse_IncomingGoods operations that could not be mappe

26 Package: fnx_warehouse_manualinbounddocumen 517

26.1 Procedures 517
26.1.1 fnx_warehouse_manualinbounddocument.create: Fortnox Package for warehouse_ManualInboundDocument operations that could n
26.1.2 fnx_warehouse_manualinbounddocument.release: Fortnox Package for warehouse_ManualInboundDocument operations that could
26.1.3 fnx_warehouse_manualinbounddocument.update: Fortnox Package for warehouse_ManualInboundDocument operations that could
26.1.4 fnx_warehouse_manualinbounddocument.updatenote: Fortnox Package for warehouse_ManualInboundDocument operations that co
26.1.5 fnx_warehouse_manualinbounddocument.voiddocument: Fortnox Package for warehouse_ManualInboundDocument operations that

27 Package: fnx_warehouse_manualoutbounddocume 520

27.1 Procedures 520
27.1.1 fnx_warehouse_manualoutbounddocument.create: Fortnox Package for warehouse_ManualOutboundDocument operations that coul
27.1.2 fnx_warehouse_manualoutbounddocument.release: Fortnox Package for warehouse_ManualOutboundDocument operations that cou
27.1.3 fnx_warehouse_manualoutbounddocument.update: Fortnox Package for warehouse_ManualOutboundDocument operations that cou
27.1.4 fnx_warehouse_manualoutbounddocument.updatenote: Fortnox Package for warehouse_ManualOutboundDocument operations that
27.1.5 fnx_warehouse_manualoutbounddocument.voiddocument: Fortnox Package for warehouse_ManualOutboundDocument operations th

28 Package: fnx_warehouse_productionorder 523

28.1 Procedures 523
28.1.1 fnx_warehouse_productionorder.release: Fortnox Package for warehouse_ProductionOrder operations that could not be mapped to
28.1.2 fnx_warehouse_productionorder.updatenote: Fortnox Package for warehouse_ProductionOrder operations that could not be mappe
28.1.3 fnx_warehouse_productionorder.voidproductionorder: Fortnox Package for warehouse_ProductionOrder operations that could not b

29 Package: fnx_warehouse_purchaseorder 525

29.1 Procedures 525
29.1.1 fnx_warehouse_purchaseorder.batchupdateresponsestate: Fortnox Package for warehouse_PurchaseOrder operations that could
29.1.2 fnx_warehouse_purchaseorder.sendpurchaseorder: Fortnox Package for warehouse_PurchaseOrder operations that could not be
29.1.3 fnx_warehouse_purchaseorder.sendpurchaseorders: Fortnox Package for warehouse_PurchaseOrder operations that could not be
29.1.4 fnx_warehouse_purchaseorder.setdropshipmanuallycompleted: Fortnox Package for warehouse_PurchaseOrder operations that co
29.1.5 fnx_warehouse_purchaseorder.setmanuallycompleted: Fortnox Package for warehouse_PurchaseOrder operations that could not b
29.1.6 fnx_warehouse_purchaseorder.updatepartial: Fortnox Package for warehouse_PurchaseOrder operations that could not be mapped
29.1.7 fnx_warehouse_purchaseorder.updateresponsestate: Fortnox Package for warehouse_PurchaseOrder operations that could not be
29.1.8 fnx_warehouse_purchaseorder.voiddocument: Fortnox Package for warehouse_PurchaseOrder operations that could not be mappe

30 Package: fnx_warehouse_stockpoint 529

30.1 Procedures 529
30.1.1 fnx_warehouse_stockpoint.appendstocklocations: Fortnox Package for warehouse_StockPoint operations that could not be mapped

31 Package: fnx_warehouse_stocktaking 530

31.1 Procedures 530
31.1.1 fnx_warehouse_stocktaking.addstocktakingrows: Fortnox Package for warehouse_StockTaking operations that could not be mapped
31.1.2 fnx_warehouse_stocktaking.addstocktakingrowsbyfilter: Fortnox Package for warehouse_StockTaking operations that could not be
31.1.3 fnx_warehouse_stocktaking.deletestocktakingrow : Fortnox Package for warehouse_StockTaking operations that could not be mappe
31.1.4 fnx_warehouse_stocktaking.deletestocktakingrowbyfilter: Fortnox Package for warehouse_StockTaking operations that could not be
31.1.5 fnx_warehouse_stocktaking.release: Fortnox Package for warehouse_StockTaking operations that could not be mapped to a table.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

31.1.6 fnx_warehouse_stocktaking.voidstocktaking: Fortnox Package for warehouse_StockTaking operations that could not be mapped to a

32 Package: fnx_warehouse_stocktransfer 534

32.1 Procedures 534

32.1.1 fnx_warehouse_stocktransfer.create: Fortnox Package for warehouse_StockTransfer operations that could not be mapped to a tabl
32.1.2 fnx_warehouse_stocktransfer.release: Fortnox Package for warehouse_StockTransfer operations that could not be mapped to a tab
32.1.3 fnx_warehouse_stocktransfer.update: Fortnox Package for warehouse_StockTransfer operations that could not be mapped to a tab
32.1.4 fnx_warehouse_stocktransfer.voidstocktransfer: Fortnox Package for warehouse_StockTransfer operations that could not be mapp

Index 537

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

1 SQL Driver for Fortnox API

Invantive UniversalSQL is the fastest, easiest and most reliable way to exchange data with the Fortnox API.

Use the "Search" option in the left menu to search for a specific term such as the table or column description. When you already know the term, please use the "Index" option. When you can't find the information needed, please click on the Chat button at the bottom or place your question in the user community. Invantive Support or other users will try to help you.

Fortnox ERP system provides cloud-based tools for accounting, invoicing, payroll, and inventory management, enabling small businesses to automate processes efficiently.

The Fortnox driver covers 222 tables and 4637 columns.

Fortnox API Clients

Invantive UniversalSQL is available on many user interfaces ("clients" in traditional server-client paradigma). All Invantive UniversalSQL statements can be exchanged with a close to 100% compatibility across all clients and operating systems (Windows, MacOS, Linux, iOS, Android).

The clients include Microsoft Excel, Microsoft Power BI, Microsoft Power Query, Microsoft Word and Microsoft Outlook. Web-based clients include Invantive Cloud, Invantive Bridge Online as OData proxy, Invantive App Online for interactive apps, Online SQL Editor for query execution and Invantive Data Access Point as extended proxy.

The Fortnox Power BI connector is based on the Invantive UniversalSQL driver for Fortnox, completed by a high-performance OData connector which works straight on Power BI without any add-on. The OData protocol is always version 4, independent whether the backing platform uses OData, SOAP or another protocol.

For technical users there are command-line editions of Invantive Data Hub running on iOS, Android, Windows, MacOS and Linux. Invantive Data Hub is also often used for enterprise server applications such as ETL. High-volume replication of data taken from the Fortnox API into traditional databases such as SQL Server (on-premises and Azure), MySQL, PostgreSQL and Oracle is possible using Invantive Data Replicator. Invantive Data Replicator automatically creates and maintains Fortnox datawarehouses, possibly in combination with data from over 75 other (cloud) platforms. Invantive Data Replicator supports data volumes up to over 1 TB and over 5.000 companies. The on-premise edition of Invantive Bridge offers an Fortnox ADO.net provider.

Finally, online web apps can be build for Fortnox using App Online of Invantive Cloud.

Monitor API Calls

When a query or DML-statement has been executed on Invantive UniversalSQL a developer can evaluate the actual calls made to the Fortnox API using a query on sessionios@DataDictionary. As an alternative, extensive request and response logging can be enabled by setting log-native-calls-to-disk to true. In the %USERPROFILE% \Invantive\NativeLog folder Invantive UniversalSQL will create log files per Fortnox API request and response.

Specifications

The SQL driver for Fortnox does not support partitioning. Define one data container in a database for each company in Fortnox to enable parallel access for data from multiple companies.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

An introduction into the concepts of Invantive UniversalSQL such as databases, data containers and partitioning can be found in the Invantive UniversalSQL grammar.

The configuration can be changed using various attributes from the database definition, on log on and during use. A full list of configuration options is listed in the driver attributes 2.

The catalog name is used to compose the full qualified name of an object like a table or view. The schema name is used to compose the full qualified name of an object like a table or view. On Fortnox the comparison of two texts is case sensitive by default.

Changes and bug fixes on the Fortnox SQL driver can be found in the release notes. Get access to the community through the Fortnox section of the Invantive forums.

Driver code for use in settings.xml: `Fortnox`

Alias: `fnx`

Recommended alias: `fnx`

Driver code for use in settings.xml

General documentation on Fortnox is available at https://www.fortnox.se/.

Updated 03-06-2025 18:15 using Invantive UniversalSQL version 24.1.110-BETA+5438.

2 SQL Driver Attributes for Fortnox API

The SQL driver for Fortnox has many attributes that can be finetuned to improve handling in scenarios with unreliable network connections to the API server of Fortnox or high volumes of data. Also, many drivers have driver-specific attributes to finetune actual behaviour or handle data not matching specifications.

The Fortnox driver attributes are assigned a default value which seldom requires change. However, changes can be applied when needed on four levels, which are reflected in the table below by separate checkmarks:

- Connection string: the connection string from the settings*.xml file and applied during log on.
- Set SQL statement: a set SQL-statement to be executed once connection has been established.
- Log on: value to be specified interactively by user during log on in a user interface.

The connection string for Fortnox can be found in the settings*.xml file used for the database. The reference manuals contain instructions how to relocate the settings*.xml files. Settings*.xml files are typically located in the `%USERPROFILE%\invantive` folder in most deployment scenarios. Each data container of a database in the connection string can have a `connectionString` element specifying the name and values of attributes. Both name and value must be properly escaped according to XML-semantics. Actual application of the value is solely done during log on. A new connection must be established to change the value of a driver attribute using a connection string.

The set SQL statement can be executed after log on. The syntax is: `set NAME VALUE`, or for a distributed database: `set NAME@ALIAS VALUE`. In some scenarios you may need to enclose the driver attribute name in square brackets to escape it from parsing, for instance when a reserved SQL keyword is part of the name. The new value takes effect straight after execution of the set-statement. The set-statement can be executed as often as needed during a session.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Driver attributes that can be interactively set to a value are typically presented in the log on window. Depending on the platform and design decisions of the user interface designer, some or all of the available driver attributes can have been made available.

The Fortnox driver can be configured using the following attributes:

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
add-odata-mandatory-filters	Whether to automatically add OData filters deemed necessary by the platform.	OData	False	✓	✓	✓	
analysis-enforce-row-uniqueness	Enforce rows to be unique for software analysis. A fingerprint is calculated from the whole row of data when the primary key column is unknown.	Shared	False	✓	✓	✓	
api-access-token	Access Token is a security token for multiple OAuth2 Flows. With an Access Token you can access protected resources. An Access Token must be stored securely since once compromised allows access to your protected resources.	OData		✓	✓	✓	✓
api-client-id	The client ID is a unique identifier of your application. It is generated by registering an application.	OData		✓		✓	✓
api-client-secret	The client secret is to be kept confidential. Such as a password for a logon code, the client secret is the confidential part of an app identified by a client ID. It is needed during the OAuth2 Code Grant Flow together with the refresh token to get access.	OData		✓		✓	✓
api-pre-expiry-refresh-sec	The number of seconds before the token expires to acquire a new token.	OData		✓	✓	✓	
api-redirect-url	The redirect URI is the website a browser session is redirected to after the OAuth2 authentication process has been completed.	OData		✓		✓	✓
api-refresh-token	Refresh Token is a security token for the OAuth2 Code Grant Flow. With a Refresh Token and client secret you can retrieve a renewed access token to access protected resources. A Refresh Token and client secret must be stored securely since once compromised allows access to your protected resources.	OData		✓		✓	✓
api-scope	The authorization scope(s) to request an OAuth token for.	OData		✓		✓	
api-token-url	The token URI is the OAuth2 endpoint to exchange tokens with.	OData		✓		✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
api-url	URL of web service.	OData		✓		✓	
bulk-delete-page-size-rows	Number of rows to delete per batch when bulk deleting.	Shared	10000	✓	✓	✓	
bulk-insert-page-size-bytes	Approximate maximum size in bytes of batch when bulk inserting.	Shared	10000000	✓	✓	✓	
bulk-insert-page-size-rows	Number of rows to insert per batch when bulk inserting.	Shared	250	✓	✓	✓	
download-error-400-bad-request-max-tries	Maximum number of tries when HTTP server reports bad format during retrieval of data.		3	✓	✓	✓	
download-error-400-bad-request-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		500	✓	✓	✓	
download-error-400-bad-request-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		5000	✓	✓	✓	
download-error-400-bad-request-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports that the API server is unavailable during retrieval of data.		2	✓	✓	✓	
download-error-403-forbidden-max-tries	Maximum number of tries when the website reports a HTTP status 403 (Forbidden).		2	✓	✓	✓	
download-error-403-forbidden-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 403 (Forbidden).		3000	✓	✓	✓	
download-error-403-forbidden-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 403 (Forbidden).		10000	✓	✓	✓	
download-error-403-forbidden-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 403 (Forbidden).		2	✓	✓	✓	
download-error-408-request-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 408.		10	✓	✓	✓	
download-error-408-request-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 408.		10000	✓	✓	✓	
download-error-408-request-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 408.		300000	✓	✓	✓	
download-error-408-request-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 408.		2	✓	✓	✓	
download-error-422-bad-request-	Maximum number of tries when HTTP server reports unprocessable		30	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
max-tries	entity during retrieval of data.						
download-error-422-bad-request-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports unprocessable entity during retrieval of data.		10000	✓	✓	✓	
download-error-422-bad-request-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports unprocessable entity during retrieval of data.		300000	✓	✓	✓	
download-error-422-bad-request-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports unprocessable entity during retrieval of data.		2	✓	✓	✓	
download-error-429-too-many-requests-max-tries	Maximum number of tries when the website reports that too many requests have been made during a timeslot of one minute or one day.		10	✓	✓	✓	
download-error-429-too-many-requests-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		10000	✓	✓	✓	
download-error-429-too-many-requests-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		300000	✓	✓	✓	
download-error-429-too-many-requests-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		2	✓	✓	✓	
download-error-500-internal-server-error-max-tries	{res:itgen_pae_download_error_500_internal_server_error_max_tries}		10	✓	✓	✓	
download-error-500-internal-server-error-sleep-initial-ms	{res:itgen_pae_download_error_500_internal_server_error_sleep_initial_ms}		10000	✓	✓	✓	
download-error-500-internal-server-error-sleep-max-ms	{res:itgen_pae_download_error_500_internal_server_error_sleep_max_ms}		300000	✓	✓	✓	
download-error-500-internal-server-error-sleep-multiplicator	{res:itgen_pae_download_error_500_internal_server_error_sleep_multiplicator}		2	✓	✓	✓	
download-error-502-server-unavailable-max-tries	Maximum number of tries when HTTP server reports a bad gateway during retrieval of data.		30	✓	✓	✓	
download-error-502-server-	Initial sleep in milliseconds between retries when HTTP server reports a		10000	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
unavailable-sleep-initial-ms	bad gateway during retrieval of data.						
download-error-502-server-unavailable-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports that a bad gateway during retrieval of data.		300000	✓	✓	✓	
download-error-502-server-unavailable-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports a bad gateway during retrieval of data.		2	✓	✓	✓	
download-error-503-server-unavailable-max-tries	Maximum number of tries when HTTP server reports that the API server is unavailable during retrieval of data.		30	✓	✓	✓	
download-error-503-server-unavailable-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		10000	✓	✓	✓	
download-error-503-server-unavailable-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		300000	✓	✓	✓	
download-error-503-server-unavailable-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports that the API server is unavailable during retrieval of data.		2	✓	✓	✓	
download-error-504-gateway-timeout-max-tries	Maximum number of tries when the website reports a gateway timeout.		10	✓	✓	✓	
download-error-504-gateway-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a gateway timeout.		10000	✓	✓	✓	
download-error-504-gateway-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a gateway timeout.		300000	✓	✓	✓	
download-error-504-gateway-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a gateway timeout.		2	✓	✓	✓	
download-error-590-network-connect-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 590.		10	✓	✓	✓	
download-error-590-network-connect-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 590.		10000	✓	✓	✓	
download-error-590-network-	Maximum sleep in milliseconds between retries when the website		300000	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
connect-timeout-sleep-max-ms	reports a HTTP status 590.						
download-error-590-network-connect-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 590.		2	✓	✓	✓	
download-error-599-network-connect-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 599.		10	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 599.		10000	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 599.		300000	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 599.		2	✓	✓	✓	
download-error-argument-exception-max-tries	Maximum number of tries when an argument exception is returned when downloading a blob.		10	✓	✓	✓	
download-error-argument-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an argument exception is returned when downloading a blob.		10000	✓	✓	✓	
download-error-argument-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an argument exception is returned when downloading a blob.		300000	✓	✓	✓	
download-error-argument-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an argument exception is returned when downloading a blob.		2	✓	✓	✓	
download-error-internet-down-max-tries	Maximum number of tries when the Internet connection seems down during retrieval of data.		10	✓	✓	✓	
download-error-internet-down-sleep-initial-ms	Initial sleep in milliseconds between retries when the Internet connection seems down during retrieval of data.		10000	✓	✓	✓	
download-error-internet-down-sleep-max-ms	Maximum sleep in milliseconds between retries when the Internet connection seems down during retrieval of data.		300000	✓	✓	✓	
download-error-internet-down-sleep-multiplicator	Multiplication factor for sleep between retries when the Internet connection seems down during retrieval of data.		2	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
download-error-io-exception-max-tries	Maximum number of tries when a network I/O connection failure occurs during retrieval of data.		10	✓	✓	✓	
download-error-io-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when a network I/O connection failure occurs during retrieval of data.		10000	✓	✓	✓	
download-error-io-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when a network I/O connection failure occurs during retrieval of data.		300000	✓	✓	✓	
download-error-io-exception-sleep-multiplicator	Multiplication factor for sleep between retries when a network I/O connection failure occurs during retrieval of data.		2	✓	✓	✓	
download-error-json-exception-max-tries	Maximum number of tries when an invalid JSON body is returned.		3	✓	✓	✓	
download-error-json-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an invalid JSON body is returned.		1000	✓	✓	✓	
download-error-json-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an invalid JSON body is returned.		10000	✓	✓	✓	
download-error-json-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an invalid JSON body is returned.		2	✓	✓	✓	
download-error-name-resolution-failure-max-tries	Maximum number of tries when the host name could not be resolved during retrieval of data.		5	✓	✓	✓	
download-error-name-resolution-failure-sleep-initial-ms	Initial sleep in milliseconds between retries when the host name could not be resolved during retrieval of data.		5000	✓	✓	✓	
download-error-name-resolution-failure-sleep-max-ms	Maximum sleep in milliseconds between retries when the host name could not be resolved during retrieval of data.		5000	✓	✓	✓	
download-error-name-resolution-failure-sleep-multiplicator	{res:itgen_pae_download_error_iname_resolution_failure_sleep_multiplicator}		1	✓	✓	✓	
download-error-other-exception-max-tries	Maximum number of tries when an unqualified error occurs during retrieval of data.		3	✓	✓	✓	
download-error-other-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an unqualified error ocurrs during retrieval of data.		10000	✓	✓	✓	
download-error-other-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an unqualified error ocurrs during retrieval of data.		300000	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
download-error-other-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an unqualified error ocurrs during retrieval of data.		2	✓	✓	✓	
download-error-socket-exception-max-tries	Maximum number of tries when the network connection is forcible dropped during retrieval of data.		10	✓	✓	✓	
download-error-socket-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when the network connection is forcible dropped during retrieval of data.		10000	✓	✓	✓	
download-error-socket-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when the network connection is forcible dropped during retrieval of data.		300000	✓	✓	✓	
download-error-socket-exception-sleep-multiplicator	Multiplication factor for sleep between retries when the network connection is forcible dropped during retrieval of data.		2	✓	✓	✓	
download-error-web-exception-max-tries	Maximum number of tries when a web connection failure occurs during retrieval of data.		10	✓	✓	✓	
download-error-web-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.		10000	✓	✓	✓	
download-error-web-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.		300000	✓	✓	✓	
download-error-web-exception-sleep-multiplicator	Multiplication factor for sleep between retries when a web connection failure occurs during retrieval of data.		2	✓	✓	✓	
download-error-web-not-found-max-tries	{res:itgen_pae_download_error_web_not_found_max_tries}		1	✓	✓	✓	
download-error-web-not-found-sleep-initial-ms	{res:itgen_pae_download_error_web_not_found_sleep_initial_ms}		10000	✓	✓	✓	
download-error-web-not-found-sleep-max-ms	{res:itgen_pae_download_error_web_not_found_sleep_max_ms}		300000	✓	✓	✓	
download-error-web-not-found-sleep-multiplicator	{res:itgen_pae_download_error_web_not_found_sleep_multiplicator}		2	✓	✓	✓	
download-error-web-not-implemented-max-tries	Maximum number of tries when the connection reports not implemented.		1	✓	✓	✓	
download-error-web-not-	Initial sleep in milliseconds between retries when the connection reports		10000	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
implemented-sleep-initial-ms	not implemented.						
download-error-web-not-implemented-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports not implemented.		300000	✓	✓	✓	
download-error-web-not-implemented-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports not implemented.		2	✓	✓	✓	
download-error-web-timeout-max-tries	Maximum number of tries when the connection reports a timeout.		10	✓	✓	✓	
download-error-web-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports a timeout.		1000	✓	✓	✓	
download-error-web-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports a timeout.		30000	✓	✓	✓	
download-error-web-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports a timeout.		2	✓	✓	✓	
download-error-web-unauthorized-max-tries	Maximum number of tries when the connection reports an unauthorized error.		1	✓	✓	✓	
download-error-web-unauthorized-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports an unauthorized error.		10000	✓	✓	✓	
download-error-web-unauthorized-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports an unauthorized error.		300000	✓	✓	✓	
download-error-web-unauthorized-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports an unauthorized error.		2	✓	✓	✓	
download-error-web-unknown-max-tries	{res:itgen_pae_download_error_web_unknown_max_tries}		10	✓	✓	✓	
download-error-web-unknown-sleep-initial-ms	{res:itgen_pae_download_error_web_unknown_sleep_initial_ms}		5000	✓	✓	✓	
download-error-web-unknown-sleep-max-ms	{res:itgen_pae_download_error_web_unknown_sleep_max_ms}		30000	✓	✓	✓	
download-error-web-unknown-sleep-multiplicator	{res:itgen_pae_download_error_web_unknown_sleep_multiplicator}		2	✓	✓	✓	
duplicate-row-mode	Configure behaviour when multiple rows are discovered during a limited			✓	✓	✓	✓

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
	search based upon a fingerprint generated from the primary key value, or all column values when no primary key is known.. The following values can be used: "error" to raise error on duplicate fingerprint values in an API-page of rows, "keep" to ignore duplicate fingerprint values silently, "skip" like "keep" but with logging to trace of duplicate values or "null" for default behaviour (typically "error")..						
force-case-sensitive-identifiers	Consider identifiers as case-sensitive independent of the platform capabilities.	Shared	False	✓	✓	✓	
forced-casing-identifiers	Forced casing of identifiers. Choose from: Unset, Lower, Upper and Mixed.	Shared		✓	✓	✓	
http-disk-cache-compression-level	Compression level for the HTTP disk cache, ranging from 1 (little) to 9 (intense). Default is 5.		5	✓	✓	✓	
http-disk-cache-directory	Directory where HTTP cache is stored.		C:\Users\guido\Invantive\Cache\http\guido\shared	✓	✓	✓	
http-disk-cache-ignore-write-errors	Whether to ignore write errors to disk cache.		False	✓	✓	✓	
http-disk-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP disk cache.		2592000	✓	✓	✓	
http-get-timeout-max-ms	HTTP GET maximum timeout on retry (ms).		24000	✓	✓	✓	
http-get-timeout-ms	HTTP GET timeout (ms).		56000	✓	✓	✓	
http-memory-cache-compression-level	Compression level for the HTTP memory cache, ranging from 1 (little) to 9 (intense). Default is 5.		5	✓	✓	✓	
http-memory-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP memory cache.		14400	✓	✓	✓	
http-post-timeout-max-ms	HTTP POST maximum timeout on retry (ms).		58000	✓	✓	✓	
http-post-timeout-ms	HTTP POST timeout (ms).		57000	✓	✓	✓	
ignore-http-400-errors	Ignore HTTP 400 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-401-errors	Ignore HTTP 401 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
ignore-http-402-errors	Ignore HTTP 402 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-403-errors	Ignore HTTP 403 errors when exchanging results with the HTTP endpoint.		True	✓	✓	✓	
ignore-http-404-errors	Ignore HTTP 404 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-422-errors	Ignore HTTP 422 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-429-errors	Ignore HTTP 429 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-500-errors	Ignore HTTP 500 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-502-errors	Ignore HTTP 502 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-503-errors	Ignore HTTP 503 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-unknown-path-type	Whether to ignore path types not yet supported. An error will be generated when an unsupported type occurs.		True	✓	✓	✓	✓
ignore-values-unknown-path	Whether to ignore values outside of processed paths. An error will be generated when a value occurs outside a path otherwise.		True	✓	✓	✓	✓
invalid-json-on-get-max-tries	Maximum number of tries when the JSON received on GET is invalid.		1	✓	✓	✓	
invalid-json-on-get-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on GET is invalid.		1000	✓	✓	✓	
invalid-json-on-get-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on GET is invalid.		10000	✓	✓	✓	
invalid-json-on-get-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on GET is invalid.		2	✓	✓	✓	
invalid-json-on-post-max-tries	Maximum number of tries when the JSON received on POST is invalid.		1	✓	✓	✓	
invalid-json-on-post-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on POST is invalid.		1000	✓	✓	✓	
invalid-json-on-post-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON		10000	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
	received on POST is invalid.						
invalid-json-on-post-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on POST is invalid.		2	✓	✓	✓	
invantive-sql-compress-sparse-arrays	Whether to compress sparse arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-correct-invalid-date	Whether to correct dates considered invalid since they are before 01-01-1753. When nullable, they are removed. Otherwise they are replaced by 01-01-1753.	SQL Engine V1	False	✓	✓	✓	
invantive-sql-execution-profile-disk-path	{res:itgen_pae_invantive_sql_execution_profile_disk_path}	SQL Engine V1		✓	✓	✓	
invantive-sql-execution-profile-to-disk	{res:itgen_pae_invantive_sql_execution_profile_to_disk}	SQL Engine V1	False	✓	✓	✓	
invantive-sql-forward-filters-to-data-containers	Whether to forward filters to data containers.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-share-byte-arrays	Whether to share the memory used by identical byte arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-share-strings	Whether to share the memory used by identical strings in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-shuffle-fetch-results-data-containers	Whether to shuffle results fetched from data containers.	SQL Engine V1	False	✓	✓	✓	
invantive-use-cache	Whether to cache the results of a query.	SQL Engine V1	True	✓	✓	✓	
join-set-points-per-request	Maximum number of values in a request when executing a join set.	OData	60	✓	✓	✓	
limit-partition-calls-left	Minimum number of remaining API calls on a partition towards a hard limit. When below, an error is raised.	OData	500	✓	✓	✓	
log-native-calls-to-disk-max-events	Maximum number of call events to register from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-max-seconds	Maximum number of seconds to register calls from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-on-error	Registers native calls to data container backend as disk files when the call raised an error.	Shared	False	✓	✓	✓	
log-native-calls-to-disk-on-success	Registers native calls to data container backend as disk files when the call raised no error.	Shared	False	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
log-native-calls-to-trace	Log native calls to data container backend on the trace.	Shared	False	✓	✓	✓	
maximum-discovered-column-count	Maximum number of discovered columns. An error will be generated when the column exceeds this value.		250	✓	✓	✓	✓
maximum-length-identifiers	Non-default maximum length in characters of identifier names.	Shared		✓	✓	✓	
max-odata-filters	Maximum number of OData filter elements.	OData	100	✓	✓	✓	
max-odata-rewrite-in-count	{res:itgen_pae_max_odata_rewrite_in_count}	OData	500	✓	✓	✓	
max-url-length-accepted	The maximum accepted URL length before raising an error.		8000	✓	✓	✓	
max-url-length-desired	The maximum desired URL length.		8000	✓	✓	✓	
metadata-cache-max-age-sec	Maximum acceptable age in seconds for re-use of metadata.	OData		✓	✓	✓	
oauth-unauthorized-max-tries	Maximum number of tries when an OAuth exception occurs.		2	✓	✓	✓	
oauth-unauthorized-sleep-initial-ms	Initial sleep in milliseconds between OAuth reauthentication tries when the OAuth authentication fails.		10000	✓	✓	✓	
oauth-unauthorized-sleep-max-ms	Maximum sleep in milliseconds between OAuth reauthentication tries when the OAuth authentication fails.		1000	✓	✓	✓	
oauth-unauthorized-sleep-multiplicator	Multiplication factor for sleep between OAuth reauthentication tries when the OAuth authentication fails.		2	✓	✓	✓	
partition-slot-based-rate-limit-length-ms	Total length in milliseconds across all slots of a partition-based rate limit.	Shared	60000	✓		✓	
partition-slot-based-rate-limit-slots	Number of slots per partition-based rate limit. Null means no slot-based rate limit.	Shared		✓		✓	
pre-request-delay-ms	Pre-request delay in milliseconds per request.	Shared	0	✓	✓	✓	
requested-page-size	Preferred number of rows to exchange per round trip; only effective on limited platforms such as AFAS Online.	Shared		✓	✓	✓	
requests-parallel-max	Maximum number of parallel data requests from individual partitions on the data container.	Shared	32	✓	✓	✓	
simulate-http-400-errors	Simulate HTTP 400 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
simulate-http-400-errors-percentage	Percentage of simulated HTTP 400 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-401-errors	Simulate HTTP 401 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-401-errors-percentage	Percentage of simulated HTTP 401 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-403-errors	Simulate HTTP 403 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-403-errors-percentage	Percentage of simulated HTTP 403 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-408-errors	Simulate HTTP 408 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-408-errors-percentage	Percentage of simulated HTTP 408 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-429-errors	Simulate HTTP 429 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-429-errors-percentage	Percentage of simulated HTTP 429 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-500-errors	Simulate HTTP 500 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-500-errors-percentage	Percentage of simulated HTTP 500 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-502-errors	Simulate HTTP 502 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-502-errors-percentage	Percentage of simulated HTTP 502 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-503-errors	Simulate HTTP 503 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-503-errors-percentage	Percentage of simulated HTTP 503 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-504-errors	{res:itgen_pae_simulate_http_504_errors}		False	✓	✓	✓	
simulate-http-504-errors-percentage	{res:itgen_pae_simulate_http_504_errors_percentage}		0	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
simulate-http-522-errors	{res:itgen_pae_simulate_http_522_errors}		False	✓	✓	✓	
simulate-http-522-errors-percentage	{res:itgen_pae_simulate_http_522_errors_percentage}		0	✓	✓	✓	
simulate-http-524-errors	{res:itgen_pae_simulate_http_524_errors}		False	✓	✓	✓	
simulate-http-524-errors-percentage	{res:itgen_pae_simulate_http_524_errors_percentage}		0	✓	✓	✓	
simulate-http-protocol-errors	Simulate HTTP protocol errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-protocol-errors-percentage	Percentage of simulated HTTP protocol errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-timeout-errors	Simulate HTTP timeout errors when exchanging results with the HTTP endpoint..		False	✓	✓	✓	
simulate-http-timeout-errors-percentage	Percentage of simulated HTTP timeout errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
slot-based-rate-limit-length-ms	Total length in milliseconds across all slots of a slot-based rate limit.	Shared	5000	✓		✓	
slot-based-rate-limit-slots	Number of slots of a slot-based rate limit. Null means no slot-based rate limit.	Shared	25	✓		✓	
standardize-identifiers	Rewrite all identifiers to the preferred standards as configured by standardize-identifiers-casing and maximum-length-identifiers.	Shared	True	✓	✓	✓	
standardize-identifiers-casing	Rewrite all identifiers to the recommended standard platform-specific casing when changing a data model on a case-dependent platform.	Shared	True	✓	✓	✓	
swagger-specification-allow-fallback	Whether to allow fallback in case of an Internet connection issue on a possibly bundled Swagger specification.		False	✓		✓	✓
swagger-specification-download-tries	Specification Download Tries		5	✓	✓	✓	
swagger-specification-file	The Swagger file path, such as C:\temp\swagger.json.			✓	✓	✓	✓
swagger-specification-http-disk-cache-max-age-sec	Maximum acceptable age in seconds for use of Swagger specification data in the HTTP disk cache.		86400	✓	✓	✓	
swagger-specification-url	The Swagger URL such as https://example.org/rest/swagger.json.			✓	✓	✓	✓

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
totp-secret	Shared secret key to generate one-time password using TOTP RFC 6238. For improved security, manually enter the one-time password asked during login.	OData		✓		✓	✓
use-batch-insert	Whether to use batch insert.	OData	True	✓	✓	✓	
use-http-disk-cache-read	Whether to use HTTP responses from previous queries stored on disk to answer the current query.		False	✓	✓	✓	
use-http-disk-cache-write	Whether to memorize HTTP responses on disk.		False	✓	✓	✓	
use-http-memory-cache-read	Whether to use HTTP responses from previous queries stored in memory that can answer the current query.		True	✓	✓	✓	
use-http-memory-cache-write	Whether to memorize HTTP responses from previous queries for use by future queries.		True	✓	✓	✓	

3 Catalog: Fortnox

3.1 Schemas

3.1.1 Schema: AbsenceTransactions

3.1.1.1 Tables

AbsenceTransactionById

Retrieve a specific absence transactionRetrieves a specific transaction

Catalog: Fortnox

Schema: AbsenceTransactions

Label: {res:itgen_fnx_absencetransactionbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AbsenceTransactionById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	guid	☑		identifies the transaction
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AbsenceTransactionById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AbsenceTransaction__url	string	{res:itgen_fnx_absencetransaction__url}	☐	
AbsenceTransaction_CauseCode	string(3)	{res:itgen_fnx_absencetransaction_causecode}	☐	
AbsenceTransaction_CostCenter	string	{res:itgen_fnx_absencetransaction_costcenter}	☐	
AbsenceTransaction_Date	datetime	{res:itgen_fnx_absencetransaction_date}	☐	
AbsenceTransaction_EmployeeId	string(15)	{res:itgen_fnx_absencetransaction_employeeid}	☐	
AbsenceTransaction_Extent	float	{res:itgen_fnx_absencetransaction_extent}	☐	
AbsenceTransaction_HolidayEntitling	boolean	{res:itgen_fnx_absencetransaction_holidayentitling}	☐	
AbsenceTransaction_Hours	float	{res:itgen_fnx_absencetransaction_hours}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
AbsenceTransaction_id	guid	{res:itgen_fnx_absencetransaction_id}	☐	
AbsenceTransaction_Project	string	{res:itgen_fnx_absencetransaction_project}	☐	

AbsenceTransactionsByEmployeeId

Lists all absence transactionsSupports query-string parameters employeeid and date for filtering the result.

Catalog: Fortnox

Schema: AbsenceTransactions

Primary Keys: id

Label: {res:itgen_fnx_absencetransactionsbyemployeeid_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AbsenceTransactionsByEmployeeId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
date	string	☐		filter by date
employeeid	string	☐		filter by employee id
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function AbsenceTransactionsByEmployeeId are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
CauseCode	string(3)	{res:itgen_fnx_causecode}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Date	datetime	{res:itgen_fnx_date}	☐	
EmployeeId	string(15)	{res:itgen_fnx_employeeid}	☐	
Extent	float	{res:itgen_fnx_extent}	☐	
HolidayEntitling	boolean	{res:itgen_fnx_holidayentitling}	☐	
Hours	float	{res:itgen_fnx_hours}	☐	
id	guid	{res:itgen_fnx_id}	☐	
Project	string	{res:itgen_fnx_project}	☐	

AbsenceTransactionsByEmployeeIdAndDateAndCode

Retrieve absence transactionsRetrieves a list of absence transactions for an employee on a specific date and cause code.

Catalog: Fortnox

Schema: AbsenceTransactions

Label: {res:itgen_fnx_absencetransactionsbyemployeeidanddateandcode_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AbsenceTransactionsByEmployeeIdAndDateAndCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
Code	string	☑		status code of the absence transaction (Values: ASK, FPE, FRA, HAV, KOM, MIL, NAR, OS1, OS2, OS3, OS4, OS5, PAP, PEM, PER, SEM, SJK, SMB, SVE, TJL, UTB, VAB)
Date	datetime	☑		of the absence transaction
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	string	☑		identifies the employee
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AbsenceTransactionsByEmployeeIdAndDateAndCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
CauseCode	string(3)	{res:itgen_fnx_causecode}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Date	datetime	{res:itgen_fnx_date}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
EmployeeId	string(15)	{res:itgen_fnx_employeeid}	☐	
Extent	float	{res:itgen_fnx_extent}	☐	
HolidayEntitling	boolean	{res:itgen_fnx_holidayentitling}	☐	
Hours	float	{res:itgen_fnx_hours}	☐	
id	guid	{res:itgen_fnx_id}	☐	
Project	string	{res:itgen_fnx_project}	☐	

3.1.2 Schema: AccountCharts

3.1.2.1 Tables

AccountCharts

List all account chartsRetrieves a list of all the available account charts.

Catalog: Fortnox

Schema: AccountCharts

Label: {res:itgen_fnx_accountcharts_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountCharts. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers,

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				vouchers and supplierinvoices (does not w ork on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function AccountCharts are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Name	string	{res:itgen_fnx_name}	☐	

3.1.3 Schema: Accounts

3.1.3.1 Tables

AccountByNumber

Retrieve an accountRetrieves the details of an account. You need to supply the unique account number that was returned when the account was created or retrieved from the list of accounts.

Catalog: Fortnox

Schema: Accounts

Label: {res:itgen_fnx_accountbynumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountByNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
Number	int32	☑		identifies the account
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AccountByNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Account__url	string	{res:itgen_fnx_account__url}	☐	
Account_Active	boolean	{res:itgen_fnx_account_active}	☐	
Account_BalanceBroughtForward	double	{res:itgen_fnx_account_balancebroughtforward}	☐	
Account_BalanceCarriedForward	double	{res:itgen_fnx_account_balancecarriedforward}	☐	
Account_CostCenter	string	{res:itgen_fnx_account_costcenter}	☐	
Account_CostCenterSettings	string	{res:itgen_fnx_account_costcentersettings}	☐	
Account_Description	string(200)	{res:itgen_fnx_account_description}	☐	
Account_Number	int32	{res:itgen_fnx_account_number}	☐	
Account_Project	string	{res:itgen_fnx_account_project}	☐	
Account_ProjectSettings	string	{res:itgen_fnx_account_projectsettings}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Account_QuantitySettings	string	{res:itgen_fnx_account_quantitysettings}	☐	
Account_QuantityUnit	string	{res:itgen_fnx_account_quantityunit}	☐	
Account_SRU	int32	{res:itgen_fnx_account_sru}	☐	
Account_TransactionInformation	string	{res:itgen_fnx_account_transactioninformation}	☐	
Account_TransactionInformationSettings	string	{res:itgen_fnx_account_transactioninformationsettings}	☐	
Account_VATCode	string	{res:itgen_fnx_account_vatcode}	☐	
Account_Year	int32	{res:itgen_fnx_account_year}	☐	

AccountOpeningQuantitiesByAccountNumber

Retrieve an accountRetrieves the details of an account. You need to supply the unique account number that was returned when the account was created or retrieved from the list of accounts.

Catalog: Fortnox

Schema: Accounts

Label: {res:itgen_fnx_accountopeningquantitiesbyaccountnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountOpeningQuantitiesByAccountNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
Number	int32	☑		identifies the account
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AccountOpeningQuantitiesByAccountNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Active	boolean	{res:itgen_fnx_active}	☐	
Balance	int32	{res:itgen_fnx_balance}	☐	
BalanceBroughtForward	double	{res:itgen_fnx_balancebroughtforward}	☐	
BalanceCarriedForward	double	{res:itgen_fnx_balancecarriedforward}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
CostCenterSettings	string	{res:itgen_fnx_costcentersettings}	☐	
Description	string(200)	{res:itgen_fnx_description}	☐	
Number	int32	{res:itgen_fnx_number}	☐	
Project_2	string	{res:itgen_fnx_project_2}	☐	
Project	string	{res:itgen_fnx_project}	☐	
ProjectSettings	string	{res:itgen_fnx_projectsettings}	☐	
QuantitySettings	string	{res:itgen_fnx_quantitysettings}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
QuantityUnit	string	{res:itgen_fnx_quantityunit}	☐	
SRU	int32	{res:itgen_fnx_sru}	☐	
TransactionInformation	string	{res:itgen_fnx_transactioninformation}	☐	
TransactionInformationSettings	string	{res:itgen_fnx_transactioninformationsettings}	☐	
VATCode	string	{res:itgen_fnx_vatcode}	☐	
Year	int32	{res:itgen_fnx_year}	☐	

Accounts

List all accountsThe accounts are returned sorted by account number with the lowest number appearing first.

Catalog: Fortnox

Schema: Accounts

Primary Keys: Number

Label: {res:itgen_fnx_accounts_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Accounts. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	string	☐		
sru	int32	☐		
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Accounts are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
Active	boolean	{res:itgen_fnx_active}	☐	
BalanceBroughtForward	double	{res:itgen_fnx_balancebroughtforward}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
CostCenterSettings	string	{res:itgen_fnx_costcentersettings}	☐	
Description	string(200)	{res:itgen_fnx_description}	☐	
Number	int32	{res:itgen_fnx_number}	☐	
Project	string	{res:itgen_fnx_project}	☐	
ProjectSettings	string	{res:itgen_fnx_projectsettings}	☐	
SRU	int32	{res:itgen_fnx_sru}	☐	
VATCode	string	{res:itgen_fnx_vatcode}	☐	
Year	int32	{res:itgen_fnx_year}	☐	

3.1.4 Schema: Archive

3.1.4.1 Tables

Archive

Retrieve folder or fileIf no path is provided the root will be returned.Providing fileId will return given file from fileattachments.

Catalog: Fortnox

Schema: Archive

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: {res:itgen_fnx_archive_plural}

Retrieve: true

Insert: true

Update: false

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Archive. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
fileid	string	☐		fileId from fileattachments
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
path	string	☐		name of folder
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Archive are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Folder__url	string	{res:itgen_fnx_folder__url}	☐	
Folder_Email	string	{res:itgen_fnx_folder_email}	☐	
Folder_Id	string	{res:itgen_fnx_folder_id}	☐	
Folder_Name	string(30)	{res:itgen_fnx_folder_name}	☐	

ArchiveById

Retrieve a single fileProviding fileId will return given file from fileattachments.

Catalog: Fortnox

Schema: Archive

Label: {res:itgen_fnx_archivebyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ArchiveById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
fileid	string	☐		fileId from fileattachments
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
id	string	☑		identifies the file
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ArchiveById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
TEXT	byte[]	{res:itgen_fnx_text}	☐	

ArchiveFiles

Retrieve folder or fileIf no path is provided the root will be returned.Providing fileId will return given file from fileattachments.

Catalog: Fortnox

Schema: Archive

Label: {res:itgen_fnx_archivefiles_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ArchiveFiles. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
fileid	string	☐		fileId from fileattachments

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
path	string	☐		name of folder
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ArchiveFiles are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
@url_2	string	{res:itgen_fnx_@url_2}	☐	
ArchiveFileId	string	{res:itgen_fnx_archivefileid}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Comments	string	{res:itgen_fnx_comments}	☐	
Email	string	{res:itgen_fnx_email}	☐	
Id_2	string	{res:itgen_fnx_id_2}	☐	
Id	string	{res:itgen_fnx_id}	☐	
Name_2	string(30)	{res:itgen_fnx_name_2}	☐	
Name	string	{res:itgen_fnx_name}	☐	
Path	string	{res:itgen_fnx_path}	☐	
Size	int32	{res:itgen_fnx_size}	☐	

ArchiveFolders

Retrieve folder or fileIf no path is provided the root will be returned.Providing fileId will return given file from fileattachments.

Catalog: Fortnox

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: Archive

Label: {res:itgen_fnx_archivefolders_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ArchiveFolders. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
fileid	string	☐		fileId from fileattachments
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
path	string	☐		name of folder
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ArchiveFolders are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
@url_2	string	{res:itgen_fnx_@url_2}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Email	string	{res:itgen_fnx_email}	☐	
Id_2	string	{res:itgen_fnx_id_2}	☐	
Id	string	{res:itgen_fnx_id}	☐	
Name_2	string(30)	{res:itgen_fnx_name_2}	☐	
Name	string	{res:itgen_fnx_name}	☐	

3.1.5 Schema: ArticleFileConnections

3.1.5.1 Tables

ArticleFileConnectionById

Retrieve a single article file connection

Catalog: Fortnox

Schema: ArticleFileConnections

Label: {res:itgen_fnx_articlefileconnectionbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ArticleFileConnectionById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
FileId	string	☑		identifies the article file connection
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ArticleFileConnectionById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ArticleFileConnection__url	string	{res:itgen_fnx_articlefileconnection__url}	☐	
ArticleFileConnection_ArticleNumber	string	{res:itgen_fnx_articlefileconnection_articlenumber}	☐	
ArticleFileConnection_FileId	string	{res:itgen_fnx_articlefileconnection_fileid}	☐	

ArticleFileConnections

Retrieve a list of article file connectionsThe article file connections register can return a list of records or a single record. By specifying a FileId in the URL, a single record will be returned. Not specifying a FileId will return a list of records.

Catalog: Fortnox

Schema: ArticleFileConnections

Primary Keys: FileId

Label: {res:itgen_fnx_articlefileconnections_plural}

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ArticleFileConnections. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ArticleFileConnections are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
ArticleNumber	string	{res:itgen_fnx_articlenumber}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
FileId	string	{res:itgen_fnx_fileid}	☐	

3.1.6 Schema: Articles

3.1.6.1 Tables

ArticleByNumber

Retrieve an articleRetrieves the details of an article. You need to supply the unique article number that was returned when the article was created or retrieved from the list of articles.

Catalog: Fortnox

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: Articles

Label: {res:itgen_fnx_articlebynumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ArticleByNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
ArticleNumber	int32	☑		identifies the article
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function ArticleByNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Article__url	string	{res:itgen_fnx_article_	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		_url}		
Article_Active	boolean	{res:itgen_fnx_article_active}	☐	
Article_ArticleNumber	string(50)	{res:itgen_fnx_article_articlenumber}	☐	
Article_Bulky	boolean	{res:itgen_fnx_article_bulky}	☐	
Article_ConstructionAccount	int32	{res:itgen_fnx_article_constructionaccount}	☐	
Article_CostCalculationMethod	string	{res:itgen_fnx_article_costcalculationmethod}	☐	
Article_DefaultStockLocation	string	{res:itgen_fnx_article_defaultstocklocation}	☐	
Article_DefaultStockPoint	string	{res:itgen_fnx_article_defaultstockpoint}	☐	
Article_Depth	int32	{res:itgen_fnx_article_depth}	☐	
Article_Description	string(200)	{res:itgen_fnx_article_description}	☐	
Article_DirectCost	double	{res:itgen_fnx_article_directcost}	☐	
Article_DisposableQuantity	double	{res:itgen_fnx_article_disposablequantity}	☐	
Article_EAN	string(30)	{res:itgen_fnx_article_ean}	☐	
Article_EUAccount	int32	{res:itgen_fnx_article_euaccount}	☐	
Article_EUVATAccount	int32	{res:itgen_fnx_article_euvataccount}	☐	
Article_Expired	boolean	{res:itgen_fnx_article_expired}	☐	
Article_ExportAccount	int32	{res:itgen_fnx_article_exportaccount}	☐	
Article_FreightCost	double	{res:itgen_fnx_article_freightcost}	☐	
Article_Height	int32	{res:itgen_fnx_article_height}	☐	
Article_Housework	boolean	{res:itgen_fnx_article_housework}	☐	
Article_HouseworkType	string	{res:itgen_fnx_article_houseworktype}	☐	
Article_Manufacturer	string(50)	{res:itgen_fnx_article_manufacturer}	☐	
Article_ManufacturerArticleNumber	string(50)	{res:itgen_fnx_article_manufacturerarticlenumber}	☐	
Article_Note	string(10000)	{res:itgen_fnx_article_note}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Article_OtherCost	double	{res:itgen_fnx_article_othercost}	☐	
Article_PurchaseAccount	int32	{res:itgen_fnx_article_purchaseaccount}	☐	
Article_PurchasePrice	double	{res:itgen_fnx_article_purchaseprice}	☐	
Article_QuantityInStock	double	{res:itgen_fnx_article_quantityinstock}	☐	
Article_ReservedQuantity	double	{res:itgen_fnx_article_reservedquantity}	☐	
Article_SalesAccount	int32	{res:itgen_fnx_article_salesaccount}	☐	
Article_SalesPrice	double	{res:itgen_fnx_article_salesprice}	☐	
Article_StockAccount	int32	{res:itgen_fnx_article_stockaccount}	☐	
Article_StockChangeAccount	int32	{res:itgen_fnx_article_stockchangeaccount}	☐	
Article_StockGoods	boolean	{res:itgen_fnx_article_stockgoods}	☐	
Article_StockPlace	string(100)	{res:itgen_fnx_article_stockplace}	☐	
Article_StockValue	double	{res:itgen_fnx_article_stockvalue}	☐	
Article_StockWarning	double	{res:itgen_fnx_article_stockw arning}	☐	
Article_SupplierName	string	{res:itgen_fnx_article_suppliername}	☐	
Article_SupplierNumber	string	{res:itgen_fnx_article_suppliernumber}	☐	
Article_Type	string	{res:itgen_fnx_article_type}	☐	
Article_Unit	string	{res:itgen_fnx_article_unit}	☐	
Article_VAT	double	{res:itgen_fnx_article_vat}	☐	
Article_WebshopArticle	boolean	{res:itgen_fnx_article_w ebshoparticle}	☐	
Article_Weight	int32	{res:itgen_fnx_article_w eight}	☐	
Article_Width	int32	{res:itgen_fnx_article_w idth}	☐	

Articles

Retrieve a list of articlesRetrieves a list of articles. The articles are returned sorted by article number with the lowest number appearing first.

Catalog: Fortnox

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: Articles

Primary Keys: ArticleNumber

Label: {res:itgen_fnx_articles_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Articles. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
articlenumber	string	☐		filter by article number
description	string	☐		filter by description
ean	string	☐		filter by ean
filter	string	☐		possibility to filter supplier invoices (Values: active, inactive)
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	string	☐		filter by lastmodified
manufacturer	string	☐		filter by manufacturer
manufacturerarticlenumber	string	☐		filter by manufacturerarticlenumber
suppliernumber	string	☐		filter by supplier number
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
webshop	string	☐		filter by web shop

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

The columns of the table function Articles are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
ArticleNumber	string(50)	{res:itgen_fnx_articlenumber}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Description	string(200)	{res:itgen_fnx_description}	☐	
DisposableQuantity	string	{res:itgen_fnx_disposablequantity}	☐	
EAN	string(30)	{res:itgen_fnx_ean}	☐	
Housework	boolean	{res:itgen_fnx_housework}	☐	
PurchasePrice	string	{res:itgen_fnx_purchaseprice}	☐	
QuantityInStock	double	{res:itgen_fnx_quantityinstock}	☐	
ReservedQuantity	string	{res:itgen_fnx_reservedquantity}	☐	
SalesPrice	string	{res:itgen_fnx_salesprice}	☐	
StockPlace	string(100)	{res:itgen_fnx_stockplace}	☐	
StockValue	string	{res:itgen_fnx_stockvalue}	☐	
Unit	string	{res:itgen_fnx_unit}	☐	
VAT	string	{res:itgen_fnx_vat}	☐	
WebshopArticle	boolean	{res:itgen_fnx_webshoparticle}	☐	

3.1.7 Schema: ArticleUrlConnections

3.1.7.1 Tables

ArticleUrlConnectionById

Retrieve a single article url connection

Catalog: Fortnox

Schema: ArticleUrlConnections

Label: {res:itgen_fnx_articleurlconnectionbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function ArticleUrlConnectionById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	string	☑		identifies the article url connection
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ArticleUrlConnectionById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ArticleUrlConnection__url	string	{res:itgen_fnx_articleurlconnection__url}	☐	
ArticleUrlConnection_ArticleNumber	string	{res:itgen_fnx_articleurlconnection_articlenumber}	☐	
ArticleUrlConnection_Id	int32	{res:itgen_fnx_articleurlconnection_id}	☐	
ArticleUrlConnection_URLConnection	string	{res:itgen_fnx_articleurlconnection_urlconnection}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ArticleUrlConnections

Retrieve a list of article url connectionsThe article url connections register can return a list of records or a single record. By specifying an id in the URL, a single record will be returned. Not specifying an id will return a list of records.

Catalog: Fortnox

Schema: ArticleUrlConnections

Primary Keys: id

Label: {res:itgen_fnx_articleurlconnections_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ArticleUrlConnections. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

The columns of the table function ArticleUrlConnections are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
ArticleNumber	string	{res:itgen_fnx_articlenumber}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Id	int32	{res:itgen_fnx_id}	☐	
URLConnection	string	{res:itgen_fnx_urlconnection}	☐	

3.1.8 Schema: AssetFileConnection

3.1.8.1 Tables

AssetFileConnectors

Retrieve a list of asset file connectionsThe asset register can return a list of assets or a single asset. By specifying a FileId in the URL, a single asset will be returned. Not specifying a FileId will return a list of records.

Catalog: Fortnox

Schema: AssetFileConnection

Primary Keys: fileId

Label: {res:itgen_fnx_assetfileconnectors_plural}

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AssetFileConnectors. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AssetFileConnectors are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
AssetId	string	{res:itgen_fnx_assetid}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
FileId	string	{res:itgen_fnx_fileid}	☐	
Name	string	{res:itgen_fnx_name}	☐	

3.1.9 Schema: Assets

3.1.9.1 Tables

AssetByGivenNumber

Retrieve a single asset

Catalog: Fortnox

Schema: Assets

Label: {res:itgen_fnx_assetbygivennumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AssetByGivenNumber. A value must be provided at all times for required parameters, but

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
GivenNumber	string	☑		Asset number
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AssetByGivenNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Assets__url	string	{res:itgen_fnx_assets__url}	☐	
Assets_AcquisitionDate	string	{res:itgen_fnx_assets_acquisitiondate}	☐	
Assets_AcquisitionStart	string	{res:itgen_fnx_assets_acquisitionstart}	☐	
Assets_AcquisitionValue	int32	{res:itgen_fnx_assets_acquisitionvalue}	☐	
Assets_Brand	string	{res:itgen_fnx_assets_brand}	☐	
Assets_CostCenter	string	{res:itgen_fnx_assets_costcenter}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Assets_Department	string	{res:itgen_fnx_assets_department}	☐	
Assets_DepreciatedTo	string	{res:itgen_fnx_assets_depreciatedto}	☐	
Assets_DepreciateToResidualValue	int32	{res:itgen_fnx_assets_depreciatetoresidualvalue}	☐	
Assets_DepreciationFinal	string	{res:itgen_fnx_assets_depreciationfinal}	☐	
Assets_DepreciationMethod	int32	{res:itgen_fnx_assets_depreciationmethod}	☐	
Assets_Description	string	{res:itgen_fnx_assets_description}	☐	
Assets_Group	string	{res:itgen_fnx_assets_group}	☐	
Assets_Id	int32	{res:itgen_fnx_assets_id}	☐	
Assets_InsuredNumber	string	{res:itgen_fnx_assets_insurednumber}	☐	
Assets_InsuredWith	string	{res:itgen_fnx_assets_insuredwith}	☐	
Assets_ManualOb	int32	{res:itgen_fnx_assets_manualob}	☐	
Assets_Notes	string	{res:itgen_fnx_assets_notes}	☐	
Assets_Number	string	{res:itgen_fnx_assets_number}	☐	
Assets_Placement	string	{res:itgen_fnx_assets_placement}	☐	
Assets_Project	string	{res:itgen_fnx_assets_project}	☐	
Assets_Reference	string	{res:itgen_fnx_assets_reference}	☐	
Assets_Room	string	{res:itgen_fnx_assets_room}	☐	
Assets_Status	string	{res:itgen_fnx_assets_status}	☐	
Assets_StatusId	string	{res:itgen_fnx_assets_statusid}	☐	
Assets_Type	string	{res:itgen_fnx_assets_type}	☐	
Assets_TypeId	int32	{res:itgen_fnx_assets_typeid}	☐	

AssetHistoryByGivenAssetNumber

Retrieve a single asset

Catalog: Fortnox

Schema: Assets

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: {res:itgen_fnx_assethistorybygivenassetnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AssetHistoryByGivenAssetNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
GivenNumber	string	☑		Asset number
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AssetHistoryByGivenAssetNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AcquisitionDate	string	{res:itgen_fnx_acquisitiondate}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
AcquisitionStart	string	{res:itgen_fnx_acquisitionstart}	☐	
AcquisitionValue	int32	{res:itgen_fnx_acquisitionvalue}	☐	
Amount	string	{res:itgen_fnx_amount}	☐	
Brand	string	{res:itgen_fnx_brand}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Date	string	{res:itgen_fnx_date}	☐	
Department	string	{res:itgen_fnx_department}	☐	
DepreciatedTo	string	{res:itgen_fnx_depreciatedto}	☐	
DepreciateToResidualValue	int32	{res:itgen_fnx_depreciatetoresidualvalue}	☐	
DepreciationFinal	string	{res:itgen_fnx_depreciationfinal}	☐	
DepreciationMethod	int32	{res:itgen_fnx_depreciationmethod}	☐	
Description	string	{res:itgen_fnx_description}	☐	
EventId	int32	{res:itgen_fnx_eventid}	☐	
Group	string	{res:itgen_fnx_group}	☐	
Id_2	int32	{res:itgen_fnx_id_2}	☐	
Id	int32	{res:itgen_fnx_id}	☐	
InsuredNumber	string	{res:itgen_fnx_insurednumber}	☐	
InsuredWith	string	{res:itgen_fnx_insuredwith}	☐	
ManualOb	int32	{res:itgen_fnx_manualob}	☐	
Notes_2	string	{res:itgen_fnx_notes_2}	☐	
Notes	string	{res:itgen_fnx_notes}	☐	
Number	string	{res:itgen_fnx_number}	☐	
Placement	string	{res:itgen_fnx_placement}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Reference	string	{res:itgen_fnx_reference}	☐	
Room	string	{res:itgen_fnx_room}	☐	
Status	string	{res:itgen_fnx_status}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
StatusId	string	{res:itgen_fnx_statusid}	☐	
SupplierInvoice	int32	{res:itgen_fnx_supplierinvoice}	☐	
Type	string	{res:itgen_fnx_type}	☐	
TypeId	int32	{res:itgen_fnx_typeid}	☐	
UserId	int32	{res:itgen_fnx_userid}	☐	
UserName	string	{res:itgen_fnx_username}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
VoucherYear	int32	{res:itgen_fnx_voucheryear}	☐	

Assets

Retrieve a list of assets

Catalog: Fortnox

Schema: Assets

Label: {res:itgen_fnx_assets_plural}

Retrieve: true

Insert: true

Update: false

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Assets. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Assets are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
AcquisitionDate	string	{res:itgen_fnx_acquisitiondate}	☐	
AcquisitionValue	int32	{res:itgen_fnx_acquisitionvalue}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
DepreciatedTo	string	{res:itgen_fnx_depreciatedto}	☐	
DepreciationFinal	string	{res:itgen_fnx_depreciationfinal}	☐	
Description	string	{res:itgen_fnx_description}	☐	
Id	int32	{res:itgen_fnx_id}	☐	
Number	string	{res:itgen_fnx_number}	☐	
Status	string	{res:itgen_fnx_status}	☐	
StatusId	string	{res:itgen_fnx_statusid}	☐	
Type	string	{res:itgen_fnx_type}	☐	
TypeId	int32	{res:itgen_fnx_typeid}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Deprecations

Assets depreciation listRetrieves a list of assets to depreciate.

Catalog: Fortnox

Schema: Assets

Label: {res:itgen_fnx_deprecations_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Deprecations. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
ToDate	string	☑		toDate
todate1	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the table function Deprecations are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AcquisitionDate	string	{res:itgen_fnx_acquisitiondate}	☐	
AcquisitionValue	int32	{res:itgen_fnx_acquisitionvalue}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
DepreciatedTo	string	{res:itgen_fnx_depreciatedto}	☐	
DepreciationFinal	string	{res:itgen_fnx_depreciationfinal}	☐	
Description	string	{res:itgen_fnx_description}	☐	
Id	int32	{res:itgen_fnx_id}	☐	
Number	string	{res:itgen_fnx_number}	☐	
Status	string	{res:itgen_fnx_status}	☐	
StatusId	string	{res:itgen_fnx_statusid}	☐	
Type	string	{res:itgen_fnx_type}	☐	
TypeId	int32	{res:itgen_fnx_typeid}	☐	

3.1.10 Schema: AssetTypes

3.1.10.1 Tables

AssetTypeById

Retrieve an asset type

Catalog: Fortnox

Schema: AssetTypes

Label: {res:itgen_fnx_assettypebyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AssetTypeById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int32	☑		id
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AssetTypeById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Type__url	string	{res:itgen_fnx_type__url}	☐	
Type_AccountAsset	int32	{res:itgen_fnx_type_accountasset}	☐	
Type_AccountAssetId	int32	{res:itgen_fnx_type_accountassetid}	☐	
Type_AccountDepreciation	int32	{res:itgen_fnx_type_accountdepreciation}	☐	
Type_AccountDepreciationId	int32	{res:itgen_fnx_type_accountdepreciationid}	☐	
Type_AccountRevaluation	int32	{res:itgen_fnx_type_accountrevaluation}	☐	
Type_AccountRevaluationId	int32	{res:itgen_fnx_type_accountrevaluationid}	☐	
Type_AccountSaleLoss	int32	{res:itgen_fnx_type_accountsaleloss}	☐	
Type_AccountSaleLossId	int32	{res:itgen_fnx_type_accountsalelossid}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Type_AccountSaleWin	int32	{res:itgen_fnx_type_accountsalewin}	☐	
Type_AccountSaleWinId	int32	{res:itgen_fnx_type_accountsalewinid}	☐	
Type_AccountValueLoss	int32	{res:itgen_fnx_type_accountvalueloss}	☐	
Type_AccountValueLossId	int32	{res:itgen_fnx_type_accountvaluelossid}	☐	
Type_AccountWriteDown	int32	{res:itgen_fnx_type_accountwritedown}	☐	
Type_AccountWriteDownAck	int32	{res:itgen_fnx_type_accountwritedownack}	☐	
Type_AccountWriteDownAckId	int32	{res:itgen_fnx_type_accountwritedownackid}	☐	
Type_AccountWriteDownId	int32	{res:itgen_fnx_type_accountwritedownid}	☐	
Type_Description	string	{res:itgen_fnx_type_description}	☐	
Type_Id	int32	{res:itgen_fnx_type_id}	☐	
Type_InUse	boolean	{res:itgen_fnx_type_inuse}	☐	
Type_Notes	string	{res:itgen_fnx_type_notes}	☐	
Type_Number	string	{res:itgen_fnx_type_number}	☐	
Type_Type	int32	{res:itgen_fnx_type_type}	☐	

AssetTypes

Retrieve a list of asset types

Catalog: Fortnox

Schema: AssetTypes

Primary Keys: id

Label: {res:itgen_fnx_assettypes_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AssetTypes. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AssetTypes are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
AccountAsset	int32	{res:itgen_fnx_accountasset}	☐	
AccountAssetId	int32	{res:itgen_fnx_accountassetid}	☐	
AccountDepreciation	int32	{res:itgen_fnx_accountdepreciation}	☐	
AccountDepreciationId	int32	{res:itgen_fnx_accountdepreciationid}	☐	
AccountRevaluation	int32	{res:itgen_fnx_accountrevaluation}	☐	
AccountRevaluationId	int32	{res:itgen_fnx_accountrevaluationid}	☐	
AccountSaleLoss	int32	{res:itgen_fnx_accountsaleloss}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
AccountSaleLossId	int32	{res:itgen_fnx_accountsalelossid}	☐	
AccountSaleWin	int32	{res:itgen_fnx_accountsalewin}	☐	
AccountSaleWinId	int32	{res:itgen_fnx_accountsalewinid}	☐	
AccountValueLoss	int32	{res:itgen_fnx_accountvalueloss}	☐	
AccountValueLossId	int32	{res:itgen_fnx_accountvaluelossid}	☐	
AccountWriteDown	int32	{res:itgen_fnx_accountwritedown}	☐	
AccountWriteDownAck	int32	{res:itgen_fnx_accountwritedownack}	☐	
AccountWriteDownAckId	int32	{res:itgen_fnx_accountwritedownackid}	☐	
AccountWriteDownId	int32	{res:itgen_fnx_accountwritedownid}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Description	string	{res:itgen_fnx_description}	☐	
Id	int32	{res:itgen_fnx_id}	☐	
InUse	boolean	{res:itgen_fnx_inuse}	☐	
Notes	string	{res:itgen_fnx_notes}	☐	
Number	string	{res:itgen_fnx_number}	☐	
Type	int32	{res:itgen_fnx_type}	☐	

3.1.11 Schema: AttendanceTransactions

3.1.11.1 Tables

ArticleUrlConnectionByEmployeeIdAndDateAndCode

Retrieve attendance transactionsRetrieves a list of attendance transaction for an employee on a specific date and cause code.

Catalog: Fortnox

Schema: AttendanceTransactions

Label: {res:itgen_fnx_articleurlconnectionbyemployeeidanddateandcode_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ArticleUrlConnectionByEmployeeIdAndDateAndCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
Code	string	☑		status code of the attendance transaction (Values: ARB, BE2, BER, FLX, HLG, JO2, JOR, MER, OB1, OB2, OB3, OB4, OB5, OK0, OK1, OK2, OK3, OK4, OK5, OT1, OT2, OT3, OT4, OT5, RES, TID)
Date	datetime	☑		date of the attendance
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
id	string	☑		identifies the employee
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function ArticleUrlConnectionByEmployeeIdAndDateAndCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
CauseCode	string(3)	{res:itgen_fnx_causecode}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Date	datetime	{res:itgen_fnx_date}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
EmployeeId	string(15)	{res:itgen_fnx_employeeid}	☐	
Hours	string	{res:itgen_fnx_hours}	☐	
id	guid	{res:itgen_fnx_id}	☐	
Project	string	{res:itgen_fnx_project}	☐	

AttendanceTransactionById

Retrieve a specific attendance transactionRetrieves a specific transaction

Catalog: Fortnox

Schema: AttendanceTransactions

Label: {res:itgen_fnx_attendancetransactionbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AttendanceTransactionById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	guid	☑		identifies the transaction
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function AttendanceTransactionById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AttendanceTransaction_CauseCode	string(3)	{res:itgen_fnx_attendancetransaction_causecode}	☐	
AttendanceTransaction_CostCenter	string	{res:itgen_fnx_attendancetransaction_costcenter}	☐	
AttendanceTransaction_Date	datetime	{res:itgen_fnx_attendancetransaction_date}	☐	
AttendanceTransaction_Employeeid	string(15)	{res:itgen_fnx_attendancetransaction_employeeid}	☐	
AttendanceTransaction_Hours	string	{res:itgen_fnx_attendancetransaction_hours}	☐	
AttendanceTransaction_Project	string	{res:itgen_fnx_attendancetransaction_project}	☐	

AttendanceTransactions

Lists all attendance transactionsSupports query-string parameters employeeid and date for filtering the result.

Catalog: Fortnox

Schema: AttendanceTransactions

Primary Keys: id

Label: {res:itgen_fnx_attendancetransactions_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AttendanceTransactions. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
date	string	☐		filter by date
employeeid	string	☐		filter by employee id
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AttendanceTransactions are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
CauseCode	string(3)	{res:itgen_fnx_causecode}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Date	datetime	{res:itgen_fnx_date}	☐	
EmployeeId	string(15)	{res:itgen_fnx_employeeid}	☐	
Hours	string	{res:itgen_fnx_hours}	☐	
id	guid	{res:itgen_fnx_id}	☐	
Project	string	{res:itgen_fnx_project}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

3.1.12 Schema: CompanyInformation
3.1.12.1 Tables
CompanyInformation

Retrieve the Company Information

Catalog: Fortnox

Schema: CompanyInformation

Label: {res:itgen_fnx_companyinformation_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CompanyInformation. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

The columns of the table function CompanyInformation are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CompanyInformation_Address	string	{res:itgen_fnx_companyinformation_address}	☐	
CompanyInformation_City	string	{res:itgen_fnx_companyinformation_city}	☐	
CompanyInformation_CompanyName	string	{res:itgen_fnx_companyinformation_companyname}	☐	
CompanyInformation_CountryCode	string	{res:itgen_fnx_companyinformation_countrycode}	☐	
CompanyInformation_DatabaseNumber	int32	{res:itgen_fnx_companyinformation_databasenumber}	☐	
CompanyInformation_OrganizationNumber	string	{res:itgen_fnx_companyinformation_organizationnumber}	☐	
CompanyInformation_VisitAddress	string	{res:itgen_fnx_companyinformation_visitaddress}	☐	
CompanyInformation_VisitCity	string	{res:itgen_fnx_companyinformation_visitcity}	☐	
CompanyInformation_VisitCountryCode	string	{res:itgen_fnx_companyinformation_visitcountrycode}	☐	
CompanyInformation_VisitZipCode	string	{res:itgen_fnx_companyinformation_visitzipcode}	☐	
CompanyInformation_ZipCode	string	{res:itgen_fnx_companyinformation_zipcode}	☐	

3.1.13 Schema: CompanySettings

3.1.13.1 Tables

CompanySettings

Retrieve the company settings

Catalog: Fortnox

Schema: CompanySettings

Label: {res:itgen_fnx_companysettings_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CompanySettings. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function CompanySettings are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CompanySettings_Address	string	{res:itgen_fnx_companysettings_address}	☐	
CompanySettings_BG	string	{res:itgen_fnx_companysettings_bg}	☐	
CompanySettings_BIC	string	{res:itgen_fnx_companysettings_bic}	☐	
CompanySettings_BranchCode	string	{res:itgen_fnx_companysettings_branchcode}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CompanySettings_City	string	{res:itgen_fnx_companysettings_city}	☐	
CompanySettings_ContactFirstName	string	{res:itgen_fnx_companysettings_contactfirstname}	☐	
CompanySettings_ContactLastName	string	{res:itgen_fnx_companysettings_contactlastname}	☐	
CompanySettings_Country	string	{res:itgen_fnx_companysettings_country}	☐	
CompanySettings_CountryCode	string	{res:itgen_fnx_companysettings_countrycode}	☐	
CompanySettings_DatabaseNumber	string	{res:itgen_fnx_companysettings_databasenumber}	☐	
CompanySettings_Domicile	string	{res:itgen_fnx_companysettings_domicile}	☐	
CompanySettings_Email	string	{res:itgen_fnx_companysettings_email}	☐	
CompanySettings_Fax	string	{res:itgen_fnx_companysettings_fax}	☐	
CompanySettings_IBAN	string	{res:itgen_fnx_companysettings_iban}	☐	
CompanySettings_Name	string	{res:itgen_fnx_companysettings_name}	☐	
CompanySettings_OrganizationNumber	string	{res:itgen_fnx_companysettings_organizationnumber}	☐	
CompanySettings_PG	string	{res:itgen_fnx_companysettings_pg}	☐	
CompanySettings_Phone1	string	{res:itgen_fnx_companysettings_phone1}	☐	
CompanySettings_Phone2	string	{res:itgen_fnx_companysettings_phone2}	☐	
CompanySettings_TaxEnabled	boolean	{res:itgen_fnx_companysettings_taxenabled}	☐	
CompanySettings_VATNumber	string	{res:itgen_fnx_companysettings_vatnumber}	☐	
CompanySettings_VisitAddress	string	{res:itgen_fnx_companysettings_visitaddress}	☐	
CompanySettings_VisitCity	string	{res:itgen_fnx_companysettings_visitcity}	☐	
CompanySettings_VisitCountry	string	{res:itgen_fnx_companysettings_visitcountry}	☐	
CompanySettings_VisitCountryCode	string	{res:itgen_fnx_companysettings_visitcountrycode}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CompanySettings_VisitName	string	{res:itgen_fnx_companysettings_visitname}	☐	
CompanySettings_VisitZipCode	string	{res:itgen_fnx_companysettings_visitzipcode}	☐	
CompanySettings_WWW	string	{res:itgen_fnx_companysettings_www}	☐	
CompanySettings_ZipCode	string	{res:itgen_fnx_companysettings_zipcode}	☐	

3.1.14 Schema: ContractAccruals

3.1.14.1 Tables

ContractAccrualByDocumentNumber

Retrieve a single contract accrual

Catalog: Fortnox

Schema: ContractAccruals

Label: {res:itgen_fnx_contractaccrualbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContractAccrualByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	int32	☑		identifies the contract accrual
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ContractAccrualByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ContractAccrual__url	string	{res:itgen_fnx_contractaccrual__url}	☐	
ContractAccrual_AccrualAccount	int32	{res:itgen_fnx_contractaccrual_accrualaccount}	☐	
ContractAccrual_CostAccount	int32	{res:itgen_fnx_contractaccrual_costaccount}	☐	
ContractAccrual_Description	string	{res:itgen_fnx_contractaccrual_description}	☐	
ContractAccrual_DocumentNumber	int32	{res:itgen_fnx_contractaccrual_documentnumber}	☐	
ContractAccrual_Period	string	{res:itgen_fnx_contractaccrual_period}	☐	
ContractAccrual_Times	int32	{res:itgen_fnx_contractaccrual_times}	☐	
ContractAccrual_Total	double	{res:itgen_fnx_contractaccrual_total}	☐	
ContractAccrual_VATIncluded	boolean	{res:itgen_fnx_contractaccrual_vatincluded}	☐	

ContractAccrualLinesByDocumentNumber

Retrieve a single contract accrual

Catalog: Fortnox

Schema: ContractAccruals

Label: {res:itgen_fnx_contractaccruallinesbydocumentnumber_plural}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContractAccrualLinesByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	int32	☑		identifies the contract accrual
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function ContractAccrualLinesByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Account	int32	{res:itgen_fnx_accoun t}	☐	
AccrualAccount	int32	{res:itgen_fnx_accrual account}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
CostAccount	int32	{res:itgen_fnx_costaccount}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Credit	double	{res:itgen_fnx_credit}	☐	
Debit	double	{res:itgen_fnx_debit}	☐	
Description	string	{res:itgen_fnx_description}	☐	
DocumentNumber	int32	{res:itgen_fnx_documentnumber}	☐	
Period	string	{res:itgen_fnx_period}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Times	int32	{res:itgen_fnx_times}	☐	
Total	double	{res:itgen_fnx_total}	☐	
TransactionInformation	string(100)	{res:itgen_fnx_transactioninformation}	☐	
VATIncluded	boolean	{res:itgen_fnx_vatincluded}	☐	

ContractAccruals

Retrieve a list of contract accrualsThe contract accruals register can return a list of records or a single record. By specifying a DocumentNumber in the URL, a single record will be returned. Not specifying a DocumentNumber will return a list of records.

Catalog: Fortnox

Schema: ContractAccruals

Primary Keys: DocumentNumber

Label: {res:itgen_fnx_contractaccruals_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContractAccruals. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ContractAccruals are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
Description	string	{res:itgen_fnx_description}	☐	
DocumentNumber	int32	{res:itgen_fnx_documentnumber}	☐	
Period	string	{res:itgen_fnx_period}	☐	

3.1.15 Schema: Contracts

3.1.15.1 Tables

ContractByDocumentNumber

Retrieve a single contract

Catalog: Fortnox

Schema: Contracts

Label: {res:itgen_fnx_contractbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContractByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the contract
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ContractByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Contract__url	string	{res:itgen_fnx_contract__url}	☐	
Contract__urlTaxReductionList	string	{res:itgen_fnx_contract__urltaxreductionlist}	☐	
Contract_Active	boolean	{res:itgen_fnx_contract_active}	☐	
Contract_AdministrationFee	double	{res:itgen_fnx_contract_administrationfee}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Contract_BasisTaxReduction	double	{res:itgen_fnx_contract_basistaxreduction}	☐	
Contract_Comments	string(1024)	{res:itgen_fnx_contract_comments}	☐	
Contract_Continuous	boolean	{res:itgen_fnx_contract_continuous}	☐	
Contract_ContractDate	datetime	{res:itgen_fnx_contract_contractdate}	☐	
Contract_ContractLength	int32	{res:itgen_fnx_contract_contractlength}	☐	
Contract_ContributionPercent	double	{res:itgen_fnx_contract_contributionpercent}	☐	
Contract_ContributionValue	double	{res:itgen_fnx_contract_contributionvalue}	☐	
Contract_CostCenter	string	{res:itgen_fnx_contract_costcenter}	☐	
Contract_Currency	string	{res:itgen_fnx_contract_currency}	☐	
Contract_CustomerName	string	{res:itgen_fnx_contract_customername}	☐	
Contract_CustomerNumber	string	{res:itgen_fnx_contract_customernumber}	☐	
Contract_DocumentNumber	string	{res:itgen_fnx_contract_documentnumber}	☐	
Contract_EmailInformation_EmailAddressBCC	string	{res:itgen_fnx_contract_emailinformation_emailaddressbcc}	☐	
Contract_EmailInformation_EmailAddressCC	string	{res:itgen_fnx_contract_emailinformation_emailaddresscc}	☐	
Contract_EmailInformation_EmailAddressFrom	string	{res:itgen_fnx_contract_emailinformation_emailaddressfrom}	☐	
Contract_EmailInformation_EmailAddressTo	string	{res:itgen_fnx_contract_emailinformation_emailaddressto}	☐	
Contract_EmailInformation_EmailBody	string	{res:itgen_fnx_contract_emailinformation_emailbody}	☐	
Contract_EmailInformation_EmailSubject	string	{res:itgen_fnx_contract_emailinformation_emailsubject}	☐	
Contract_ExternalInvoiceReference1	string	{res:itgen_fnx_contract_externalinvoicereference1}	☐	
Contract_ExternalInvoiceReference2	string	{res:itgen_fnx_contract_externalinvoicereference2}	☐	
Contract_Freight	double	{res:itgen_fnx_contract_freight}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Contract_Gross	double	{res:itgen_fnx_contract_gross}	☐	
Contract_HouseWork	boolean	{res:itgen_fnx_contract_housework}	☐	
Contract_InvoiceDiscount	double	{res:itgen_fnx_contract_invoicediscount}	☐	
Contract_InvoiceInterval	int32	{res:itgen_fnx_contract_invoiceinterval}	☐	
Contract_InvoicesRemaining	string	{res:itgen_fnx_contract_invoicesremaining}	☐	
Contract_Language	string	{res:itgen_fnx_contract_language}	☐	
Contract_LastInvoiceDate	string	{res:itgen_fnx_contract_lastinvoicedate}	☐	
Contract_Net	double	{res:itgen_fnx_contract_net}	☐	
Contract_OurReference	string(25)	{res:itgen_fnx_contract_ourreference}	☐	
Contract_PeriodEnd	datetime	{res:itgen_fnx_contract_periodend}	☐	
Contract_PeriodStart	datetime	{res:itgen_fnx_contract_periodstart}	☐	
Contract_PriceList	string	{res:itgen_fnx_contract_pricelist}	☐	
Contract_PrintTemplate	string	{res:itgen_fnx_contract_printtemplate}	☐	
Contract_Project	string	{res:itgen_fnx_contract_project}	☐	
Contract_Remarks	string(1024)	{res:itgen_fnx_contract_remarks}	☐	
Contract_RoundOff	double	{res:itgen_fnx_contract_roundoff}	☐	
Contract_TaxReduction	double	{res:itgen_fnx_contract_taxreduction}	☐	
Contract_TaxReductionType	string	{res:itgen_fnx_contract_taxreductiontype}	☐	
Contract_TemplateName	string	{res:itgen_fnx_contract_templatename}	☐	
Contract_TemplateNumber	int32	{res:itgen_fnx_contract_templatenumber}	☐	
Contract_TermsOfDelivery	string	{res:itgen_fnx_contract_termsofdelivery}	☐	
Contract_TermsOfPayment	string	{res:itgen_fnx_contract_termsofpayment}	☐	
Contract_Total	double	{res:itgen_fnx_contract_total}	☐	
Contract_TotalToPay	double	{res:itgen_fnx_contract_totaltopay}	☐	
Contract_TotalVAT	double	{res:itgen_fnx_contract_totalvat}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Contract_VatIncluded	boolean	{res:itgen_fnx_contract_vatincluded}	☐	
Contract_WayOfDelivery	string	{res:itgen_fnx_contract_wayofdelivery}	☐	
Contract_YourOrderNumber	string(75)	{res:itgen_fnx_contract_yourordernumber}	☐	
Contract_YourReference	string	{res:itgen_fnx_contract_yourreference}	☐	

ContractInvoiceLinesByDocumentNumber

Retrieve a single contract

Catalog: Fortnox

Schema: Contracts

Label: {res:itgen_fnx_contractinvoicelinesbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContractInvoiceLinesByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the contract
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ContractInvoiceLinesByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AccountNumber	int32	{res:itgen_fnx_accountnumber}	☐	
Active	boolean	{res:itgen_fnx_active}	☐	
AdministrationFee	double	{res:itgen_fnx_administrationfee}	☐	
ArticleNumber	string	{res:itgen_fnx_articlenumber}	☐	
BasisTaxReduction	double	{res:itgen_fnx_basistaxreduction}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
c_urlTaxReductionList	string	{res:itgen_fnx_c_urltaxreductionlist}	☐	
Comments	string(1024)	{res:itgen_fnx_comments}	☐	
Continuous	boolean	{res:itgen_fnx_continuous}	☐	
ContractDate	datetime	{res:itgen_fnx_contractdate}	☐	
ContractLength	int32	{res:itgen_fnx_contractlength}	☐	
ContributionPercent_2	double	{res:itgen_fnx_contributionpercent_2}	☐	
ContributionPercent	string	{res:itgen_fnx_contributionpercent}	☐	
ContributionValue_2	double	{res:itgen_fnx_contributionvalue_2}	☐	
ContributionValue	string	{res:itgen_fnx_contributionvalue}	☐	
CostCenter_2	string	{res:itgen_fnx_costcenter_2}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Currency	string	{res:itgen_fnx_currency}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CustomerName	string	{res:itgen_fnx_customername}	☐	
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
DeliveredQuantity	string	{res:itgen_fnx_deliveredquantity}	☐	
Description	string	{res:itgen_fnx_description}	☐	
Discount	double	{res:itgen_fnx_discount}	☐	
DiscountType	string	{res:itgen_fnx_discounttype}	☐	
DocumentNumber	string	{res:itgen_fnx_documentnumber}	☐	
EmailInformation_EmailAddressBCC	string	{res:itgen_fnx_emailinformation_emailaddressbcc}	☐	
EmailInformation_EmailAddressCC	string	{res:itgen_fnx_emailinformation_emailaddresscc}	☐	
EmailInformation_EmailAddressFrom	string	{res:itgen_fnx_emailinformation_emailaddressfrom}	☐	
EmailInformation_EmailAddressTo	string	{res:itgen_fnx_emailinformation_emailaddressto}	☐	
EmailInformation_EmailBody	string	{res:itgen_fnx_emailinformation_emailbody}	☐	
EmailInformation_EmailSubject	string	{res:itgen_fnx_emailinformation_emailsubject}	☐	
ExternalInvoiceReference1	string	{res:itgen_fnx_externalinvoicereference1}	☐	
ExternalInvoiceReference2	string	{res:itgen_fnx_externalinvoicereference2}	☐	
Freight	double	{res:itgen_fnx_freight}	☐	
Gross	double	{res:itgen_fnx_gross}	☐	
HouseWork_2	boolean	{res:itgen_fnx_housework_2}	☐	
HouseWork	boolean	{res:itgen_fnx_housework}	☐	
HouseWorkHoursToReport	int32	{res:itgen_fnx_houseworkhourstoreport}	☐	
HouseWorkType	string	{res:itgen_fnx_houseworktype}	☐	
InvoiceDiscount	double	{res:itgen_fnx_invoicediscount}	☐	
InvoiceInterval	int32	{res:itgen_fnx_invoiceinterval}	☐	
InvoicesRemaining	string	{res:itgen_fnx_invoicesremaining}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Language	string	{res:itgen_fnx_language}	☐	
LastInvoiceDate	string	{res:itgen_fnx_lastinvoicedate}	☐	
Net	double	{res:itgen_fnx_net}	☐	
OurReference	string(25)	{res:itgen_fnx_ourreference}	☐	
PeriodEnd	datetime	{res:itgen_fnx_periodend}	☐	
PeriodStart	datetime	{res:itgen_fnx_periodstart}	☐	
Price	double	{res:itgen_fnx_price}	☐	
PriceExcludingVAT	double	{res:itgen_fnx_priceexcludingvat}	☐	
PriceList	string	{res:itgen_fnx_pricelist}	☐	
PrintTemplate	string	{res:itgen_fnx_printtemplate}	☐	
Project_2	string	{res:itgen_fnx_project_2}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Remarks	string(1024)	{res:itgen_fnx_remarks}	☐	
RoundOff	double	{res:itgen_fnx_roundoff}	☐	
Row Id	int32	{res:itgen_fnx_row id}	☐	
TaxReduction	double	{res:itgen_fnx_taxreduction}	☐	
TaxReductionType	string	{res:itgen_fnx_taxreductiontype}	☐	
TemplateName	string	{res:itgen_fnx_templatename}	☐	
TemplateNumber	int32	{res:itgen_fnx_templatenumber}	☐	
TermsOfDelivery	string	{res:itgen_fnx_termsofdelivery}	☐	
TermsOfPayment	string	{res:itgen_fnx_termsofpayment}	☐	
Total_2	double	{res:itgen_fnx_total_2}	☐	
Total	double	{res:itgen_fnx_total}	☐	
TotalExcludingVAT	double	{res:itgen_fnx_totalexcludingvat}	☐	
TotalToPay	double	{res:itgen_fnx_totaltopay}	☐	
TotalVAT	double	{res:itgen_fnx_totalvat}	☐	
Unit	string	{res:itgen_fnx_unit}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
VAT	double	{res:itgen_fnx_vat}	☐	
VatIncluded	boolean	{res:itgen_fnx_vatincluded}	☐	
WayOfDelivery	string	{res:itgen_fnx_wayofdelivery}	☐	
YourOrderNumber	string(75)	{res:itgen_fnx_yourordernumber}	☐	
YourReference	string	{res:itgen_fnx_yourreference}	☐	

Contracts

Retrieve a list of contracts

Catalog: Fortnox

Schema: Contracts

Primary Keys: DocumentNumber

Label: {res:itgen_fnx_contracts_plural}

Retrieve: true

Insert: true

Update: true

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Contracts. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
filter	string	☐		possibility to filter contracts (Values: active, inactive, finished)
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Contracts are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
Continuous	boolean	{res:itgen_fnx_continuous}	☐	
ContractLength	int32	{res:itgen_fnx_contractlength}	☐	
Currency	string	{res:itgen_fnx_currency}	☐	
CustomerName	string	{res:itgen_fnx_customername}	☐	
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
DocumentNumber	string	{res:itgen_fnx_documentnumber}	☐	
Invoiceinterval	int32	{res:itgen_fnx_invoiceinterval}	☐	
InvoicesRemaining	int32	{res:itgen_fnx_invoicesremaining}	☐	
LastInvoiceDate	string	{res:itgen_fnx_lastinvoicedate}	☐	
PeriodEnd	datetime	{res:itgen_fnx_periodend}	☐	
PeriodStart	datetime	{res:itgen_fnx_periodstart}	☐	
Status	string	{res:itgen_fnx_status}	☐	
TemplateNumber	int32	{res:itgen_fnx_templatenumber}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Total	double	{res:itgen_fnx_total}	☐	

3.1.16 Schema: ContractTemplates

3.1.16.1 Tables

ContractTemplateByTemplateNumber

Retrieve a single contract template

Catalog: Fortnox

Schema: ContractTemplates

Label: {res:itgen_fnx_contracttemplatebytemplatenumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContractTemplateByTemplateNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
TemplateNumber	int32	☑		identifies the contract template
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers,

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function ContractTemplateByTemplateNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ContractTemplate__url	string	{res:itgen_fnx_contracttemplate__url}	☐	
ContractTemplate_AdministrationFee	double	{res:itgen_fnx_contracttemplate_administrationfee}	☐	
ContractTemplate_Continuous	boolean	{res:itgen_fnx_contracttemplate_continuous}	☐	
ContractTemplate_ContractLength	int32	{res:itgen_fnx_contracttemplate_contractlength}	☐	
ContractTemplate_Freight	double	{res:itgen_fnx_contracttemplate_freight}	☐	
ContractTemplate_InvoiceInterval	int32	{res:itgen_fnx_contracttemplate_invoiceinterval}	☐	
ContractTemplate_OurReference	string(25)	{res:itgen_fnx_contracttemplate_ourreference}	☐	
ContractTemplate_PrintTemplate	string	{res:itgen_fnx_contracttemplate_printtemplate}	☐	
ContractTemplate_Remarks	string(1024)	{res:itgen_fnx_contracttemplate_remarks}	☐	
ContractTemplate_TemplateName	string(1024)	{res:itgen_fnx_contracttemplate_templatename}	☐	
ContractTemplate_TemplateNumber	int32	{res:itgen_fnx_contracttemplate_templatenumber}	☐	
ContractTemplate_TermsOfDelivery	string	{res:itgen_fnx_contracttemplate_termsofdelivery}	☐	
ContractTemplate_TermsOfPayment	string	{res:itgen_fnx_contracttemplate_termsofpayment}	☐	
ContractTemplate_WayOfDelivery	string	{res:itgen_fnx_contracttemplate_w ayofdelivery}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ContractTemplateInvoiceLinesByTemplateNumber

Retrieve a single contract template

Catalog: Fortnox

Schema: ContractTemplates

Label: {res:itgen_fnx_contracttemplateinvoicelinesbytemplatenumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContractTemplateInvoiceLinesByTemplateNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
TemplateNumber	int32	☑		identifies the contract template
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the table function ContractTemplateInvoiceLinesByTemplateNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AccountNumber	int32	{res:itgen_fnx_accountnumber}	☐	
AdministrationFee	double	{res:itgen_fnx_administrationfee}	☐	
ArticleNumber	string	{res:itgen_fnx_articlenumber}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Continuous	boolean	{res:itgen_fnx_continuous}	☐	
ContractLength	int32	{res:itgen_fnx_contractlength}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
DeliveredQuantity	string	{res:itgen_fnx_deliveredquantity}	☐	
Description	string	{res:itgen_fnx_description}	☐	
Discount	double	{res:itgen_fnx_discount}	☐	
DiscountType	string	{res:itgen_fnx_discounttype}	☐	
Freight	double	{res:itgen_fnx_freight}	☐	
InvoiceInterval	int32	{res:itgen_fnx_invoiceinterval}	☐	
OurReference	string(25)	{res:itgen_fnx_ourreference}	☐	
Price	double	{res:itgen_fnx_price}	☐	
PrintTemplate	string	{res:itgen_fnx_printtemplate}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Remarks	string(1024)	{res:itgen_fnx_remarks}	☐	
TemplateName	string(1024)	{res:itgen_fnx_templatename}	☐	
TemplateNumber	int32	{res:itgen_fnx_templatenumber}	☐	
TermsOfDelivery	string	{res:itgen_fnx_termsofdelivery}	☐	
TermsOfPayment	string	{res:itgen_fnx_termsofpayment}	☐	
Unit	string	{res:itgen_fnx_unit}	☐	
WayOfDelivery	string	{res:itgen_fnx_wayofdelivery}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ContractTemplates

Retrieve a list of contract templatesThe contract template resource can return a list of records or a single record. By specifying a TemplateNumber in the URL, a single record will be returned. Not specifying a TemplateNumber will return a list of records.

Catalog: Fortnox

Schema: ContractTemplates

Label: {res:itgen_fnx_contracttemplates_plural}

Retrieve: true

Insert: true

Update: false

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContractTemplates. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

The columns of the table function ContractTemplates are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
ContractLength	int32	{res:itgen_fnx_contractlength}	☐	
ContractTemplate	int32	{res:itgen_fnx_contracttemplate}	☐	
ContractTemplateName	string	{res:itgen_fnx_contracttemplatename}	☐	
InvoiceInterval	int32	{res:itgen_fnx_invoiceinterval}	☐	

3.1.17 Schema: CostCenters

3.1.17.1 Tables

CostCenterByCode

Retrieve a single cost center

Catalog: Fortnox

Schema: CostCenters

Label: {res:itgen_fnx_costcenterbycode_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CostCenterByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
Code	string	☑		identifies the cost center
financialyear	int64	☐		Selects what financial year that should be used (does not work

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function CostCenterByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CostCenter__url	string	{res:itgen_fnx_costcenter__url}	☐	
CostCenter_Active	boolean	{res:itgen_fnx_costcenter_active}	☐	
CostCenter_Code	string(6)	{res:itgen_fnx_costcenter_code}	☐	
CostCenter_Description	string(2147483647)	{res:itgen_fnx_costcenter_description}	☐	
CostCenter_Note	string	{res:itgen_fnx_costcenter_note}	☐	

CostCenters

Retrieve a list of cost centersThe cost centers register can return a list of records or a single record. By specifying a Code in the URL, a single record will be returned. Not specifying a Code will return a list of records.

Catalog: Fortnox

Schema: CostCenters

Primary Keys: Code

Label: {res:itgen_fnx_costcenters_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function CostCenters. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function CostCenters are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
Active	boolean	{res:itgen_fnx_active}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Code	string(6)	{res:itgen_fnx_code}	☐	
Description	string(2147483647)	{res:itgen_fnx_description}	☐	
Note	string	{res:itgen_fnx_note}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

3.1.18 Schema: Currencies
3.1.18.1 Tables
Currencies

Retrieve a list of currenciesThe currency register can return a list of records or a single record. By specifying a Code in the URL, a single record will be returned. Not specifying a Code will return a list of records.

Catalog: Fortnox

Schema: Currencies

Primary Keys: Code

Label: {res:itgen_fnx_currencies_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Currencies. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the table function Currencies are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
BuyRate	double	{res:itgen_fnx_buyrate}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Code	string	{res:itgen_fnx_code}	☐	
Date	datetime	{res:itgen_fnx_date}	☐	
Description	string	{res:itgen_fnx_description}	☐	
IsAutomatic	boolean	{res:itgen_fnx_isautomatic}	☐	
SellRate	double	{res:itgen_fnx_sellrate}	☐	
Unit	double	{res:itgen_fnx_unit}	☐	

CurrencyByCode

Retrieve a single currency

Catalog: Fortnox

Schema: Currencies

Label: {res:itgen_fnx_currencybycode_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CurrencyByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
Code	string	☑		identifies currency
financialyear	int64	☐		Selects what financial year that should be used (does not work

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function CurrencyByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Currency__url	string	{res:itgen_fnx_currency__url}	☐	
Currency_BuyRate	double	{res:itgen_fnx_currency_buyrate}	☐	
Currency_Code	string	{res:itgen_fnx_currency_code}	☐	
Currency_Date	datetime	{res:itgen_fnx_currency_date}	☐	
Currency_Description	string	{res:itgen_fnx_currency_description}	☐	
Currency_IsAutomatic	boolean	{res:itgen_fnx_currency_isautomatic}	☐	
Currency_SellRate	double	{res:itgen_fnx_currency_sellrate}	☐	
Currency_Unit	double	{res:itgen_fnx_currency_unit}	☐	

3.1.19 Schema: CustomerReferences

3.1.19.1 Tables

CustomerReferenceById

Retrieve a customer reference rowYou need to supply the unique customer reference row id that was returned when the customer reference row was created or retrieved from the list of customer reference rows.

Catalog: Fortnox

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: CustomerReferences

Label: {res:itgen_fnx_customerreferencebyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomerReferenceById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
CustomerReferenceRowId	string	☑		identifies the customer reference row
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function CustomerReferenceById are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
DUMMY	string(1)	{res:itgen_fnx_dummy}	☑	Default column added since the specification specifies that no data is returned.

CustomerReferenceLines

Retrieve a list of customers reference rows

Catalog: Fortnox

Schema: CustomerReferences

Label: {res:itgen_fnx_customerreferencelines_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomerReferenceLines. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
customer	string	☐		possibility to filter by customer number
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function CustomerReferenceLines are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
Id	int32	{res:itgen_fnx_id}	☐	
Reference	string	{res:itgen_fnx_reference}	☐	

CustomerReferenceLinesByReferenceId

Retrieve a customer reference rowYou need to supply the unique customer reference row id that was returned when the customer reference row was created or retrieved from the list of customer reference rows.

Catalog: Fortnox

Schema: CustomerReferences

Label: {res:itgen_fnx_customerreferencelinesbyreferenceid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomerReferenceLinesByReferenceId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
CustomerReferenceRow Id	string	☑		identifies the customer reference row

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function CustomerReferenceLinesByReferenceId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
Id	int32	{res:itgen_fnx_id}	☐	
Reference	string	{res:itgen_fnx_reference}	☐	

CustomerReferences

Retrieve a list of customers reference rows

Catalog: Fortnox

Schema: CustomerReferences

Label: {res:itgen_fnx_customerreferences_plural}

Retrieve: true

Insert: true

Update: false

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomerReferences. A value must be provided at all times for required parameters, but

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
customer	string	☐		possibility to filter by customer number
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function CustomerReferences are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
DUMMY	string(1)	{res:itgen_fnx_dummy}	☑	Default column added since the specification specifies that no data is returned.

3.1.20 Schema: Customers
3.1.20.1 Tables
CustomerByNumber

Retrieve a customerYou need to supply the unique customer number that was returned when the customer was created or retrieved from the list of customers.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Catalog: Fortnox

Schema: Customers

Label: {res:itgen_fnx_customerbynumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomerByNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
CustomerNumber	string	☑		identifies the customer
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function CustomerByNumber are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Customer__url	string	{res:itgen_fnx_customer__url}	☐	
Customer_Active	boolean	{res:itgen_fnx_customer_active}	☐	
Customer_Address1	string(1024)	{res:itgen_fnx_customer_address1}	☐	
Customer_Address2	string(1024)	{res:itgen_fnx_customer_address2}	☐	
Customer_City	string(1024)	{res:itgen_fnx_customer_city}	☐	
Customer_Comments	string(1024)	{res:itgen_fnx_customer_comments}	☐	
Customer_CostCenter	string	{res:itgen_fnx_customer_costcenter}	☐	
Customer_Country	string	{res:itgen_fnx_customer_country}	☐	
Customer_CountryCode	string(2)	{res:itgen_fnx_customer_countrycode}	☐	
Customer_Currency	string(3)	{res:itgen_fnx_customer_currency}	☐	
Customer_CustomerNumber	string(1024)	{res:itgen_fnx_customer_customernumber}	☐	
Customer_DefaultDeliveryTypes_Invoice	string	{res:itgen_fnx_customer_defaultdeliverytypes_invoice}	☐	
Customer_DefaultDeliveryTypes_Offer	string	{res:itgen_fnx_customer_defaultdeliverytypes_offer}	☐	
Customer_DefaultDeliveryTypes_Order	string	{res:itgen_fnx_customer_defaultdeliverytypes_order}	☐	
Customer_DefaultTemplates_CashInvoice	string	{res:itgen_fnx_customer_defaulttemplates_cashinvoice}	☐	
Customer_DefaultTemplates_Invoice	string	{res:itgen_fnx_customer_defaulttemplates_invoice}	☐	
Customer_DefaultTemplates_Offer	string	{res:itgen_fnx_customer_defaulttemplates_offer}	☐	
Customer_DefaultTemplates_Order	string	{res:itgen_fnx_customer_defaulttemplates_order}	☐	
Customer_DeliveryAddress1	string(1024)	{res:itgen_fnx_customer_deliveryaddress1}	☐	
Customer_DeliveryAddress2	string(1024)	{res:itgen_fnx_customer_deliveryaddress2}	☐	
Customer_DeliveryCity	string(1024)	{res:itgen_fnx_customer_deliverycity}	☐	
Customer_DeliveryCountry	string(1024)	{res:itgen_fnx_customer_deliverycountry}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Customer_DeliveryCountryCode	string(2)	{res:itgen_fnx_customer_deliverycountrycode}	☐	
Customer_DeliveryFax	string(1024)	{res:itgen_fnx_customer_deliveryfax}	☐	
Customer_DeliveryName	string(1024)	{res:itgen_fnx_customer_deliveryname}	☐	
Customer_DeliveryPhone1	string(1024)	{res:itgen_fnx_customer_deliveryphone1}	☐	
Customer_DeliveryPhone2	string(1024)	{res:itgen_fnx_customer_deliveryphone2}	☐	
Customer_DeliveryZipCode	string(10)	{res:itgen_fnx_customer_deliveryzipcode}	☐	
Customer_Email	string(1024)	{res:itgen_fnx_customer_email}	☐	
Customer_EmailInvoice	string(1024)	{res:itgen_fnx_customer_emailinvoice}	☐	
Customer_EmailInvoiceBCC	string(1024)	{res:itgen_fnx_customer_emailinvoicebcc}	☐	
Customer_EmailInvoiceCC	string(1024)	{res:itgen_fnx_customer_emailinvoicecc}	☐	
Customer_EmailOffer	string(1024)	{res:itgen_fnx_customer_emailoffer}	☐	
Customer_EmailOfferBCC	string(1024)	{res:itgen_fnx_customer_emailofferbcc}	☐	
Customer_EmailOfferCC	string(1024)	{res:itgen_fnx_customer_emailoffercc}	☐	
Customer_EmailOrder	string(1024)	{res:itgen_fnx_customer_emailorder}	☐	
Customer_EmailOrderBCC	string(1024)	{res:itgen_fnx_customer_emailorderbcc}	☐	
Customer_EmailOrderCC	string(1024)	{res:itgen_fnx_customer_emailordercc}	☐	
Customer_ExternalReference	string(1024)	{res:itgen_fnx_customer_externalreference}	☐	
Customer_Fax	string(1024)	{res:itgen_fnx_customer_fax}	☐	
Customer_GLN	string(13)	{res:itgen_fnx_customer_gln}	☐	
Customer_GLNDelivery	string(13)	{res:itgen_fnx_customer_glndelivery}	☐	
Customer_InvoiceAdministrationFee	string	{res:itgen_fnx_customer_invoiceadministrationfee}	☐	
Customer_InvoiceDiscount	double	{res:itgen_fnx_customer_invoicediscount}	☐	
Customer_InvoiceFreight	string	{res:itgen_fnx_customer_invoicefreight}	☐	
Customer_InvoiceRemark	string(1024)	{res:itgen_fnx_customer_invoiceremark}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Customer_Name	string(1024)	{res:itgen_fnx_customer_name}	☐	
Customer_OrganisationNumber	string	{res:itgen_fnx_customer_organisationnumber}	☐	
Customer_OurReference	string(50)	{res:itgen_fnx_customer_ourreference}	☐	
Customer_Phone1	string(1024)	{res:itgen_fnx_customer_phone1}	☐	
Customer_Phone2	string(1024)	{res:itgen_fnx_customer_phone2}	☐	
Customer_PriceList	string	{res:itgen_fnx_customer_pricelist}	☐	
Customer_Project	string	{res:itgen_fnx_customer_project}	☐	
Customer_SalesAccount	string(4)	{res:itgen_fnx_customer_salesaccount}	☐	
Customer_Show PriceVATIncluded	boolean	{res:itgen_fnx_customer_show pricevatincluded}	☐	
Customer_TermsOfDelivery	string	{res:itgen_fnx_customer_termsofdelivery}	☐	
Customer_TermsOfPayment	string	{res:itgen_fnx_customer_termsofpayment}	☐	
Customer_Type	string	{res:itgen_fnx_customer_type}	☐	
Customer_VATNumber	string	{res:itgen_fnx_customer_vatnumber}	☐	
Customer_VATType	string	{res:itgen_fnx_customer_vattype}	☐	
Customer_VisitingAddress	string(128)	{res:itgen_fnx_customer_visitingaddress}	☐	
Customer_VisitingCity	string(128)	{res:itgen_fnx_customer_visitingcity}	☐	
Customer_VisitingCountry	string(128)	{res:itgen_fnx_customer_visitingcountry}	☐	
Customer_VisitingCountryCode	string(2)	{res:itgen_fnx_customer_visitingcountrycode}	☐	
Customer_VisitingZipCode	string(10)	{res:itgen_fnx_customer_visitingzipcode}	☐	
Customer_WayOfDelivery	string	{res:itgen_fnx_customer_w ayofdelivery}	☐	
Customer_WWW	string(128)	{res:itgen_fnx_customer_w w w}	☐	
Customer_YourReference	string(50)	{res:itgen_fnx_customer_yourreference}	☐	
Customer_ZipCode	string(10)	{res:itgen_fnx_customer_zipcode}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Customers

Retrieve a list of customersThe customers are returned sorted by customer number with the lowest number appearing first.

Catalog: Fortnox

Schema: Customers

Primary Keys: CustomerNumber

Label: {res:itgen_fnx_customers_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Customers. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
city	string	☐		filter by city
customernumber	string	☐		filter by customer number
email	string	☐		filter by email
filter	string	☐		possibility to filter customers (Values: active, inactive)
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
gln	string	☐		filter by gln
glndelivery	string	☐		filter by gln delivery
lastmodified	string	☐		filter by last modified
name	string	☐		filter by name
organisationnumber	string	☐		filter by organisation number
phone	string	☐		filter by phone

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
zipcode	string	☐		filter by zip code

Columns of Table Function

The columns of the table function Customers are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
Address1	string(1024)	{res:itgen_fnx_address1}	☐	
Address2	string(1024)	{res:itgen_fnx_address2}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
City	string(1024)	{res:itgen_fnx_city}	☐	
CustomerNumber	string(1024)	{res:itgen_fnx_customernumber}	☐	
Email	string(1024)	{res:itgen_fnx_email}	☐	
Name	string(1024)	{res:itgen_fnx_name}	☐	
OrganisationNumber	string(30)	{res:itgen_fnx_organisationnumber}	☐	
Phone	string(1024)	{res:itgen_fnx_phone}	☐	
ZipCode	string(10)	{res:itgen_fnx_zipcode}	☐	

3.1.21 Schema: Employees

3.1.21.1 Tables

EmployeeById

Retrieve a specific employeeScheduleId, MonthlySalary and HourlyPay reflect current values, allScheduleIds are returned in DatedSchedules and all MonthlySalary andHourlyPay pairs are returned in DatedWages.

Catalog: Fortnox

Schema: Employees

Label: {res:itgen_fnx_employeebyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function EmployeeById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
EmployeeId	string	☑		identifies the employee
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function EmployeeById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Employee_Address1	string	{res:itgen_fnx_employee_address1}	☐	
Employee_Address2	string	{res:itgen_fnx_employee_address2}	☐	
Employee_AutoNonRecurringTax	boolean	{res:itgen_fnx_employee_autononrecurringtax}	☐	
Employee_AverageHourlyWage	string	{res:itgen_fnx_employee_averagehourlywage}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Employee_AverageWeeklyHours	string	{res:itgen_fnx_employee_averageweeklyhours}	☐	
Employee_BankAccountNo	string	{res:itgen_fnx_employee_bankaccountno}	☐	
Employee_City	string	{res:itgen_fnx_employee_city}	☐	
Employee_ClearingNo	string	{res:itgen_fnx_employee_clearingno}	☐	
Employee_CostCenter	string	{res:itgen_fnx_employee_costcenter}	☐	
Employee_Country	string	{res:itgen_fnx_employee_country}	☐	
Employee_Email	string	{res:itgen_fnx_employee_email}	☐	
Employee_EmployedTo	datetime	{res:itgen_fnx_employee_employedto}	☐	
Employee_EmployeeId	string(15)	{res:itgen_fnx_employee_employeeid}	☐	
Employee_EmploymentDate	datetime	{res:itgen_fnx_employee_employmentdate}	☐	
Employee_EmploymentForm	string	{res:itgen_fnx_employee_employmentform}	☐	
Employee_FirstName	string	{res:itgen_fnx_employee_firstname}	☐	
Employee_ForaType	string	{res:itgen_fnx_employee_foratype}	☐	
Employee_FullName	string	{res:itgen_fnx_employee_fullname}	☐	
Employee_HourlyPay	string	{res:itgen_fnx_employee_hourlypay}	☐	
Employee_Inactive	boolean	{res:itgen_fnx_employee_inactive}	☐	
Employee_JobTitle	string(30)	{res:itgen_fnx_employee_jobtitle}	☐	
Employee_LastName	string	{res:itgen_fnx_employee_lastname}	☐	
Employee_MonthlySalary	string	{res:itgen_fnx_employee_monthlysalary}	☐	
Employee_NonRecurringTax	string	{res:itgen_fnx_employee_nonrecurringtax}	☐	
Employee_PersonalIdentityNumber	string	{res:itgen_fnx_employee_personalidentitynumber}	☐	
Employee_PersonelType	string	{res:itgen_fnx_employee_personeltype}	☐	
Employee_Phone1	string	{res:itgen_fnx_employee_phone1}	☐	
Employee_Phone2	string	{res:itgen_fnx_employee_phone2}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Employee_PostCode	string	{res:itgen_fnx_employee_postcode}	☐	
Employee_Project	string	{res:itgen_fnx_employee_project}	☐	
Employee_SalaryForm	string	{res:itgen_fnx_employee_salaryform}	☐	
Employee_ScheduleId	string	{res:itgen_fnx_employee_scheduleid}	☐	
Employee_TaxAllowance	string	{res:itgen_fnx_employee_taxallowance}	☐	
Employee_TaxColumn	int32	{res:itgen_fnx_employee_taxcolumn}	☐	
Employee_TaxTable	string	{res:itgen_fnx_employee_taxtable}	☐	
Employee_VacationDaysPaid	float	{res:itgen_fnx_employee_vacationdayspaid}	☐	
Employee_VacationDaysPendingPaid	float	{res:itgen_fnx_employee_vacationdayspendingpaid}	☐	
Employee_VacationDaysPendingPrepaid	float	{res:itgen_fnx_employee_vacationdayspendingprepaid}	☐	
Employee_VacationDaysPendingSaved	float	{res:itgen_fnx_employee_vacationdayspendingsaved}	☐	
Employee_VacationDaysPendingSavedYear1	float	{res:itgen_fnx_employee_vacationdayspendingsavedyear1}	☐	
Employee_VacationDaysPendingSavedYear2	float	{res:itgen_fnx_employee_vacationdayspendingsavedyear2}	☐	
Employee_VacationDaysPendingSavedYear3	float	{res:itgen_fnx_employee_vacationdayspendingsavedyear3}	☐	
Employee_VacationDaysPendingSavedYear4	float	{res:itgen_fnx_employee_vacationdayspendingsavedyear4}	☐	
Employee_VacationDaysPendingSavedYear5	float	{res:itgen_fnx_employee_vacationdayspendingsavedyear5}	☐	
Employee_VacationDaysPendingSavedYear6Plus	float	{res:itgen_fnx_employee_vacationdayspendingsavedyear6plus}	☐	
Employee_VacationDaysPendingUnpaid	float	{res:itgen_fnx_employee_vacationdayspendingunpaid}	☐	
Employee_VacationDaysPrepaid	float	{res:itgen_fnx_employee_vacationdaysprepaid}	☐	
Employee_VacationDaysRegisteredPaid	float	{res:itgen_fnx_employee_vacationdaysregisteredpaid}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Employee_VacationDaysRegisteredPrepaid	float	{res:itgen_fnx_employee_vacationdaysregisteredprepaid}	☐	
Employee_VacationDaysRegisteredSaved	float	{res:itgen_fnx_employee_vacationdaysregisteredsaved}	☐	
Employee_VacationDaysRegisteredSavedYear1	float	{res:itgen_fnx_employee_vacationdaysregisteredsavedyear1}	☐	
Employee_VacationDaysRegisteredSavedYear2	float	{res:itgen_fnx_employee_vacationdaysregisteredsavedyear2}	☐	
Employee_VacationDaysRegisteredSavedYear3	float	{res:itgen_fnx_employee_vacationdaysregisteredsavedyear3}	☐	
Employee_VacationDaysRegisteredSavedYear4	float	{res:itgen_fnx_employee_vacationdaysregisteredsavedyear4}	☐	
Employee_VacationDaysRegisteredSavedYear5	float	{res:itgen_fnx_employee_vacationdaysregisteredsavedyear5}	☐	
Employee_VacationDaysRegisteredSavedYear6Plus	float	{res:itgen_fnx_employee_vacationdaysregisteredsavedyear6plus}	☐	
Employee_VacationDaysRegisteredUnpaid	float	{res:itgen_fnx_employee_vacationdaysregisteredunpaid}	☐	
Employee_VacationDaysSaved	float	{res:itgen_fnx_employee_vacationdayssaved}	☐	
Employee_VacationDaysSavedEmploymentRateYear1	float	{res:itgen_fnx_employee_vacationdayssavedemploymentrateyear1}	☐	
Employee_VacationDaysSavedEmploymentRateYear2	float	{res:itgen_fnx_employee_vacationdayssavedemploymentrateyear2}	☐	
Employee_VacationDaysSavedEmploymentRateYear3	float	{res:itgen_fnx_employee_vacationdayssavedemploymentrateyear3}	☐	
Employee_VacationDaysSavedEmploymentRateYear4	float	{res:itgen_fnx_employee_vacationdayssavedemploymentrateyear4}	☐	
Employee_VacationDaysSavedEmploymentRateYear5	float	{res:itgen_fnx_employee_vacationdayssavedemploymentrateyear5}	☐	
Employee_VacationDaysSavedEmploymentRateYear6Plus	float	{res:itgen_fnx_employee_vacationdayssavedemploymentrateyear6plus}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Employee_VacationDaysSavedYear1	float	{res:itgen_fnx_employee_vacationdayssavedyear1}	☐	
Employee_VacationDaysSavedYear2	float	{res:itgen_fnx_employee_vacationdayssavedyear2}	☐	
Employee_VacationDaysSavedYear3	float	{res:itgen_fnx_employee_vacationdayssavedyear3}	☐	
Employee_VacationDaysSavedYear4	float	{res:itgen_fnx_employee_vacationdayssavedyear4}	☐	
Employee_VacationDaysSavedYear5	float	{res:itgen_fnx_employee_vacationdayssavedyear5}	☐	
Employee_VacationDaysSavedYear6Plus	float	{res:itgen_fnx_employee_vacationdayssavedyear6plus}	☐	
Employee_VacationDaysUnpaid	float	{res:itgen_fnx_employee_vacationdaysunpaid}	☐	

EmployeeChildren

Retrieve a list of employeesScheduleId, MonthlySalary and HourlyPay reflect current values, allScheduleIds are returned in DatedSchedules and all MonthlySalary andHourlyPay pairs are returned in DatedWages.

Catalog: Fortnox

Schema: Employees

Label: {res:itgen_fnx_employeechildren_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function EmployeeChildren. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function EmployeeChildren are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Address1	string	{res:itgen_fnx_address1}	☐	
Address2	string	{res:itgen_fnx_address2}	☐	
ApprovedDays	int32	{res:itgen_fnx_approveddays}	☐	
AutoNonRecurringTax	boolean	{res:itgen_fnx_autononrecurringtax}	☐	
AverageHourlyWage	string	{res:itgen_fnx_averagehourlywage}	☐	
AverageWeeklyHours	string	{res:itgen_fnx_averageweeklyhours}	☐	
BankAccountNo	string	{res:itgen_fnx_bankaccountno}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Child	string	{res:itgen_fnx_child}	☐	
City	string	{res:itgen_fnx_city}	☐	
ClearingNo	string	{res:itgen_fnx_clearingno}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	
Email	string	{res:itgen_fnx_email}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
EmployedTo	datetime	{res:itgen_fnx_employedto}	☐	
EmployeeId_2	string(15)	{res:itgen_fnx_employeeid_2}	☐	
EmployeeId	string(15)	{res:itgen_fnx_employeeid}	☐	
EmploymentDate	datetime	{res:itgen_fnx_employmentdate}	☐	
EmploymentForm	string	{res:itgen_fnx_employmentform}	☐	
FirstName	string	{res:itgen_fnx_firstname}	☐	
ForaType	string	{res:itgen_fnx_foratype}	☐	
FullName	string	{res:itgen_fnx_fullname}	☐	
HourlyPay	string	{res:itgen_fnx_hourlypay}	☐	
Id	string	{res:itgen_fnx_id}	☐	
Inactive	boolean	{res:itgen_fnx_inactive}	☐	
IngoingWithdraw nDays	int32	{res:itgen_fnx_ingoingw ithdraw ndays}	☐	
JobTitle	string(30)	{res:itgen_fnx_jobtitle}	☐	
LastName	string	{res:itgen_fnx_lastname}	☐	
MonthlySalary	string	{res:itgen_fnx_monthlysalary}	☐	
NonRecurringTax	string	{res:itgen_fnx_nonrecurringtax}	☐	
PersonalIdentityNumber	string	{res:itgen_fnx_personalidentitynumber}	☐	
PersonelType	string	{res:itgen_fnx_personeltype}	☐	
Phone1	string	{res:itgen_fnx_phone1}	☐	
Phone2	string	{res:itgen_fnx_phone2}	☐	
PostCode	string	{res:itgen_fnx_postcode}	☐	
Project	string	{res:itgen_fnx_project}	☐	
SalaryForm	string	{res:itgen_fnx_salaryform}	☐	
ScheduleId	string	{res:itgen_fnx_scheduleid}	☐	
TaxAllow ance	string	{res:itgen_fnx_taxallow ance}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TaxColumn	int32	{res:itgen_fnx_taxcolumn}	☐	
TaxTable	string	{res:itgen_fnx_taxtable}	☐	
VacationDaysPaid	float	{res:itgen_fnx_vacationdayspaid}	☐	
VacationDaysPendingPaid	float	{res:itgen_fnx_vacationdayspendingpaid}	☐	
VacationDaysPendingPrepaid	float	{res:itgen_fnx_vacationdayspendingprepaid}	☐	
VacationDaysPendingSaved	float	{res:itgen_fnx_vacationdayspendingsaved}	☐	
VacationDaysPendingSavedYear1	float	{res:itgen_fnx_vacationdayspendingsavedyear1}	☐	
VacationDaysPendingSavedYear2	float	{res:itgen_fnx_vacationdayspendingsavedyear2}	☐	
VacationDaysPendingSavedYear3	float	{res:itgen_fnx_vacationdayspendingsavedyear3}	☐	
VacationDaysPendingSavedYear4	float	{res:itgen_fnx_vacationdayspendingsavedyear4}	☐	
VacationDaysPendingSavedYear5	float	{res:itgen_fnx_vacationdayspendingsavedyear5}	☐	
VacationDaysPendingSavedYear6Plus	float	{res:itgen_fnx_vacationdayspendingsavedyear6plus}	☐	
VacationDaysPendingUnpaid	float	{res:itgen_fnx_vacationdayspendingunpaid}	☐	
VacationDaysPrepaid	float	{res:itgen_fnx_vacationdaysprepaid}	☐	
VacationDaysRegisteredPaid	float	{res:itgen_fnx_vacationdaysregisteredpaid}	☐	
VacationDaysRegisteredPrepaid	float	{res:itgen_fnx_vacationdaysregisteredprepaid}	☐	
VacationDaysRegisteredSaved	float	{res:itgen_fnx_vacationdaysregisteredsaved}	☐	
VacationDaysRegisteredSavedYear1	float	{res:itgen_fnx_vacationdaysregisteredsavedyear1}	☐	
VacationDaysRegisteredSavedYear2	float	{res:itgen_fnx_vacationdaysregisteredsavedyear2}	☐	
VacationDaysRegisteredSavedYear3	float	{res:itgen_fnx_vacationdaysregisteredsavedyear3}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
VacationDaysRegisteredSavedYear4	float	{res:itgen_fnx_vacationdaysregisteredsavedyear4}	☐	
VacationDaysRegisteredSavedYear5	float	{res:itgen_fnx_vacationdaysregisteredsavedyear5}	☐	
VacationDaysRegisteredSavedYear6Plus	float	{res:itgen_fnx_vacationdaysregisteredsavedyear6plus}	☐	
VacationDaysRegisteredUnpaid	float	{res:itgen_fnx_vacationdaysregisteredunpaid}	☐	
VacationDaysSaved	float	{res:itgen_fnx_vacationdayssaved}	☐	
VacationDaysSavedEmploymentRateYear1	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear1}	☐	
VacationDaysSavedEmploymentRateYear2	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear2}	☐	
VacationDaysSavedEmploymentRateYear3	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear3}	☐	
VacationDaysSavedEmploymentRateYear4	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear4}	☐	
VacationDaysSavedEmploymentRateYear5	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear5}	☐	
VacationDaysSavedEmploymentRateYear6Plus	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear6plus}	☐	
VacationDaysSavedYear1	float	{res:itgen_fnx_vacationdayssavedyear1}	☐	
VacationDaysSavedYear2	float	{res:itgen_fnx_vacationdayssavedyear2}	☐	
VacationDaysSavedYear3	float	{res:itgen_fnx_vacationdayssavedyear3}	☐	
VacationDaysSavedYear4	float	{res:itgen_fnx_vacationdayssavedyear4}	☐	
VacationDaysSavedYear5	float	{res:itgen_fnx_vacationdayssavedyear5}	☐	
VacationDaysSavedYear6Plus	float	{res:itgen_fnx_vacationdayssavedyear6plus}	☐	
VacationDaysUnpaid	float	{res:itgen_fnx_vacationdaysunpaid}	☐	
Withdraw nDays	float	{res:itgen_fnx_withdraw ndays}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

EmployeeChildrenByEmployeeId

Retrieve a specific employeeScheduleId, MonthlySalary and HourlyPay reflect current values, allScheduleIds are returned in DatedSchedules and all MonthlySalary andHourlyPay pairs are returned in DatedWages.

Catalog: Fortnox

Schema: Employees

Label: {res:itgen_fnx_employeechildrenbyemployeeid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function EmployeeChildrenByEmployeeId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
EmployeeId	string	☑		identifies the employee
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the table function EmployeeChildrenByEmployeeId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Address1	string	{res:itgen_fnx_address1}	☐	
Address2	string	{res:itgen_fnx_address2}	☐	
ApprovedDays	int32	{res:itgen_fnx_approveddays}	☐	
AutoNonRecurringTax	boolean	{res:itgen_fnx_autononrecurringtax}	☐	
AverageHourlyWage	string	{res:itgen_fnx_averagehourlyw age}	☐	
AverageWeeklyHours	string	{res:itgen_fnx_averagew eeklyhours}	☐	
BankAccountNo	string	{res:itgen_fnx_bankaccountno}	☐	
Child	string	{res:itgen_fnx_child}	☐	
City	string	{res:itgen_fnx_city}	☐	
ClearingNo	string	{res:itgen_fnx_clearingno}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	
Email	string	{res:itgen_fnx_email}	☐	
EmployedTo	datetime	{res:itgen_fnx_employedto}	☐	
EmployeeId_2	string(15)	{res:itgen_fnx_employeeid_2}	☐	
EmployeeId	string(15)	{res:itgen_fnx_employeeid}	☐	
EmploymentDate	datetime	{res:itgen_fnx_employmentdate}	☐	
EmploymentForm	string	{res:itgen_fnx_employmentform}	☐	
FirstName	string	{res:itgen_fnx_firstname}	☐	
ForaType	string	{res:itgen_fnx_foratype}	☐	
FullName	string	{res:itgen_fnx_fullname}	☐	
HourlyPay	string	{res:itgen_fnx_hourlypay}	☐	
Id	string	{res:itgen_fnx_id}	☐	
Inactive	boolean	{res:itgen_fnx_inactive}	☐	
IngoingWithdraw nDays	int32	{res:itgen_fnx_ingoing w ithdraw ndays}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
JobTitle	string(30)	{res:itgen_fnx_jobtitle}	☐	
LastName	string	{res:itgen_fnx_lastname}	☐	
MonthlySalary	string	{res:itgen_fnx_monthlysalary}	☐	
NonRecurringTax	string	{res:itgen_fnx_nonrecurringtax}	☐	
PersonalIdentityNumber	string	{res:itgen_fnx_personalidentitynumber}	☐	
PersonelType	string	{res:itgen_fnx_personeltype}	☐	
Phone1	string	{res:itgen_fnx_phone1}	☐	
Phone2	string	{res:itgen_fnx_phone2}	☐	
PostCode	string	{res:itgen_fnx_postcode}	☐	
Project	string	{res:itgen_fnx_project}	☐	
SalaryForm	string	{res:itgen_fnx_salaryform}	☐	
ScheduleId	string	{res:itgen_fnx_scheduleid}	☐	
TaxAllow ance	string	{res:itgen_fnx_taxallow ance}	☐	
TaxColumn	int32	{res:itgen_fnx_taxcolumn}	☐	
TaxTable	string	{res:itgen_fnx_taxtable}	☐	
VacationDaysPaid	float	{res:itgen_fnx_vacationdayspaid}	☐	
VacationDaysPendingPaid	float	{res:itgen_fnx_vacationdayspendingpaid}	☐	
VacationDaysPendingPrepaid	float	{res:itgen_fnx_vacationdayspendingprepaid}	☐	
VacationDaysPendingSaved	float	{res:itgen_fnx_vacationdayspendingsaved}	☐	
VacationDaysPendingSavedYear1	float	{res:itgen_fnx_vacationdayspendingsavedyear1}	☐	
VacationDaysPendingSavedYear2	float	{res:itgen_fnx_vacationdayspendingsavedyear2}	☐	
VacationDaysPendingSavedYear3	float	{res:itgen_fnx_vacationdayspendingsavedyear3}	☐	
VacationDaysPendingSavedYear4	float	{res:itgen_fnx_vacationdayspendingsavedyear4}	☐	
VacationDaysPendingSavedYear5	float	{res:itgen_fnx_vacationdayspendingsavedye	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		ar5}		
VacationDaysPendingSavedYear6Plus	float	{res:itgen_fnx_vacationdayspendingsavedyear6plus}	☐	
VacationDaysPendingUnpaid	float	{res:itgen_fnx_vacationdayspendingunpaid}	☐	
VacationDaysPrepaid	float	{res:itgen_fnx_vacationdaysprepaid}	☐	
VacationDaysRegisteredPaid	float	{res:itgen_fnx_vacationdaysregisteredpaid}	☐	
VacationDaysRegisteredPrepaid	float	{res:itgen_fnx_vacationdaysregisteredprepaid}	☐	
VacationDaysRegisteredSaved	float	{res:itgen_fnx_vacationdaysregisteredsaved}	☐	
VacationDaysRegisteredSavedYear1	float	{res:itgen_fnx_vacationdaysregisteredsavedyear1}	☐	
VacationDaysRegisteredSavedYear2	float	{res:itgen_fnx_vacationdaysregisteredsavedyear2}	☐	
VacationDaysRegisteredSavedYear3	float	{res:itgen_fnx_vacationdaysregisteredsavedyear3}	☐	
VacationDaysRegisteredSavedYear4	float	{res:itgen_fnx_vacationdaysregisteredsavedyear4}	☐	
VacationDaysRegisteredSavedYear5	float	{res:itgen_fnx_vacationdaysregisteredsavedyear5}	☐	
VacationDaysRegisteredSavedYear6Plus	float	{res:itgen_fnx_vacationdaysregisteredsavedyear6plus}	☐	
VacationDaysRegisteredUnpaid	float	{res:itgen_fnx_vacationdaysregisteredunpaid}	☐	
VacationDaysSaved	float	{res:itgen_fnx_vacationdayssaved}	☐	
VacationDaysSavedEmploymentRateYear1	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear1}	☐	
VacationDaysSavedEmploymentRateYear2	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear2}	☐	
VacationDaysSavedEmploymentRateYear3	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear3}	☐	
VacationDaysSavedEmploymentRateYear4	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear4}	☐	
VacationDaysSavedEmploymentRateYear5	float	{res:itgen_fnx_vacationdayssavedemployme	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		ntrateyear5}		
VacationDaysSavedEmploymentRateYear6Plus	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear6plus}	☐	
VacationDaysSavedYear1	float	{res:itgen_fnx_vacationdayssavedyear1}	☐	
VacationDaysSavedYear2	float	{res:itgen_fnx_vacationdayssavedyear2}	☐	
VacationDaysSavedYear3	float	{res:itgen_fnx_vacationdayssavedyear3}	☐	
VacationDaysSavedYear4	float	{res:itgen_fnx_vacationdayssavedyear4}	☐	
VacationDaysSavedYear5	float	{res:itgen_fnx_vacationdayssavedyear5}	☐	
VacationDaysSavedYear6Plus	float	{res:itgen_fnx_vacationdayssavedyear6plus}	☐	
VacationDaysUnpaid	float	{res:itgen_fnx_vacationdaysunpaid}	☐	
WithdrawnDays	float	{res:itgen_fnx_withdrawndays}	☐	

EmployeeDatedSchedules

Retrieve a list of employeesScheduleId, MonthlySalary and HourlyPay reflect current values, allScheduleIds are returned in DatedSchedules and all MonthlySalary andHourlyPay pairs are returned in DatedWages.

Catalog: Fortnox

Schema: Employees

Label: {res:itgen_fnx_employeedatedschedules_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function EmployeeDatedSchedules. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function EmployeeDatedSchedules are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Address1	string	{res:itgen_fnx_address1}	☐	
Address2	string	{res:itgen_fnx_address2}	☐	
AutoNonRecurringTax	boolean	{res:itgen_fnx_autononrecurringtax}	☐	
AverageHourlyWage	string	{res:itgen_fnx_averagehourlywage}	☐	
AverageWeeklyHours	string	{res:itgen_fnx_averageweeklyhours}	☐	
BankAccountNo	string	{res:itgen_fnx_bankaccountno}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
City	string	{res:itgen_fnx_city}	☐	
ClearingNo	string	{res:itgen_fnx_clearingno}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	
Email	string	{res:itgen_fnx_email}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
EmployedTo	datetime	{res:itgen_fnx_employedto}	☐	
EmployeeId_2	string(15)	{res:itgen_fnx_employeeid_2}	☐	
EmployeeId	string(15)	{res:itgen_fnx_employeeid}	☐	
EmploymentDate	datetime	{res:itgen_fnx_employmentdate}	☐	
EmploymentForm	string	{res:itgen_fnx_employmentform}	☐	
FirstDay	datetime	{res:itgen_fnx_firstday}	☐	
FirstName	string	{res:itgen_fnx_firstname}	☐	
ForaType	string	{res:itgen_fnx_foratype}	☐	
FullName	string	{res:itgen_fnx_fullname}	☐	
HourlyPay	string	{res:itgen_fnx_hourlypay}	☐	
Inactive	boolean	{res:itgen_fnx_inactive}	☐	
JobTitle	string(30)	{res:itgen_fnx_jobtitle}	☐	
LastName	string	{res:itgen_fnx_lastname}	☐	
MonthlySalary	string	{res:itgen_fnx_monthlysalary}	☐	
NonRecurringTax	string	{res:itgen_fnx_nonrecurringtax}	☐	
PersonalIdentityNumber	string	{res:itgen_fnx_personalidentitynumber}	☐	
PersonelType	string	{res:itgen_fnx_personeltype}	☐	
Phone1	string	{res:itgen_fnx_phone1}	☐	
Phone2	string	{res:itgen_fnx_phone2}	☐	
PostCode	string	{res:itgen_fnx_postcode}	☐	
Project	string	{res:itgen_fnx_project}	☐	
SalaryForm	string	{res:itgen_fnx_salaryform}	☐	
ScheduleId_2	string	{res:itgen_fnx_scheduleid_2}	☐	
ScheduleId	string	{res:itgen_fnx_scheduleid}	☐	
TaxAllow ance	string	{res:itgen_fnx_taxallow ance}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TaxColumn	int32	{res:itgen_fnx_taxcolumn}	☐	
TaxTable	string	{res:itgen_fnx_taxtable}	☐	
VacationDaysPaid	float	{res:itgen_fnx_vacationdayspaid}	☐	
VacationDaysPendingPaid	float	{res:itgen_fnx_vacationdayspendingpaid}	☐	
VacationDaysPendingPrepaid	float	{res:itgen_fnx_vacationdayspendingprepaid}	☐	
VacationDaysPendingSaved	float	{res:itgen_fnx_vacationdayspendingsaved}	☐	
VacationDaysPendingSavedYear1	float	{res:itgen_fnx_vacationdayspendingsavedyear1}	☐	
VacationDaysPendingSavedYear2	float	{res:itgen_fnx_vacationdayspendingsavedyear2}	☐	
VacationDaysPendingSavedYear3	float	{res:itgen_fnx_vacationdayspendingsavedyear3}	☐	
VacationDaysPendingSavedYear4	float	{res:itgen_fnx_vacationdayspendingsavedyear4}	☐	
VacationDaysPendingSavedYear5	float	{res:itgen_fnx_vacationdayspendingsavedyear5}	☐	
VacationDaysPendingSavedYear6Plus	float	{res:itgen_fnx_vacationdayspendingsavedyear6plus}	☐	
VacationDaysPendingUnpaid	float	{res:itgen_fnx_vacationdayspendingunpaid}	☐	
VacationDaysPrepaid	float	{res:itgen_fnx_vacationdaysprepaid}	☐	
VacationDaysRegisteredPaid	float	{res:itgen_fnx_vacationdaysregisteredpaid}	☐	
VacationDaysRegisteredPrepaid	float	{res:itgen_fnx_vacationdaysregisteredprepaid}	☐	
VacationDaysRegisteredSaved	float	{res:itgen_fnx_vacationdaysregisteredsaved}	☐	
VacationDaysRegisteredSavedYear1	float	{res:itgen_fnx_vacationdaysregisteredsavedyear1}	☐	
VacationDaysRegisteredSavedYear2	float	{res:itgen_fnx_vacationdaysregisteredsavedyear2}	☐	
VacationDaysRegisteredSavedYear3	float	{res:itgen_fnx_vacationdaysregisteredsavedyear3}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
VacationDaysRegisteredSavedYear4	float	{res:itgen_fnx_vacationdaysregisteredsavedyear4}	☐	
VacationDaysRegisteredSavedYear5	float	{res:itgen_fnx_vacationdaysregisteredsavedyear5}	☐	
VacationDaysRegisteredSavedYear6Plus	float	{res:itgen_fnx_vacationdaysregisteredsavedyear6plus}	☐	
VacationDaysRegisteredUnpaid	float	{res:itgen_fnx_vacationdaysregisteredunpaid}	☐	
VacationDaysSaved	float	{res:itgen_fnx_vacationdayssaved}	☐	
VacationDaysSavedEmploymentRateYear1	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear1}	☐	
VacationDaysSavedEmploymentRateYear2	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear2}	☐	
VacationDaysSavedEmploymentRateYear3	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear3}	☐	
VacationDaysSavedEmploymentRateYear4	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear4}	☐	
VacationDaysSavedEmploymentRateYear5	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear5}	☐	
VacationDaysSavedEmploymentRateYear6Plus	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear6plus}	☐	
VacationDaysSavedYear1	float	{res:itgen_fnx_vacationdayssavedyear1}	☐	
VacationDaysSavedYear2	float	{res:itgen_fnx_vacationdayssavedyear2}	☐	
VacationDaysSavedYear3	float	{res:itgen_fnx_vacationdayssavedyear3}	☐	
VacationDaysSavedYear4	float	{res:itgen_fnx_vacationdayssavedyear4}	☐	
VacationDaysSavedYear5	float	{res:itgen_fnx_vacationdayssavedyear5}	☐	
VacationDaysSavedYear6Plus	float	{res:itgen_fnx_vacationdayssavedyear6plus}	☐	
VacationDaysUnpaid	float	{res:itgen_fnx_vacationdaysunpaid}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

EmployeeDatedSchedulesByEmployeeId

Retrieve a specific employeeScheduleId, MonthlySalary and HourlyPay reflect current values, allScheduleIds are returned in DatedSchedules and all MonthlySalary andHourlyPay pairs are returned in DatedWages.

Catalog: Fortnox

Schema: Employees

Label: {res:itgen_fnx_employeedatedschedulesbyemployeeid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function EmployeeDatedSchedulesByEmployeeId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
EmployeeId	string	☑		identifies the employee
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the table function EmployeeDatedSchedulesByEmployeeId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Address1	string	{res:itgen_fnx_address1}	☐	
Address2	string	{res:itgen_fnx_address2}	☐	
AutoNonRecurringTax	boolean	{res:itgen_fnx_autononrecurringtax}	☐	
AverageHourlyWage	string	{res:itgen_fnx_averagehourlywage}	☐	
AverageWeeklyHours	string	{res:itgen_fnx_averageweeklyhours}	☐	
BankAccountNo	string	{res:itgen_fnx_bankaccountno}	☐	
City	string	{res:itgen_fnx_city}	☐	
ClearingNo	string	{res:itgen_fnx_clearingno}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	
Email	string	{res:itgen_fnx_email}	☐	
EmployedTo	datetime	{res:itgen_fnx_employedto}	☐	
EmployeeId_2	string(15)	{res:itgen_fnx_employeeid_2}	☐	
EmployeeId	string(15)	{res:itgen_fnx_employeeid}	☐	
EmploymentDate	datetime	{res:itgen_fnx_employmentdate}	☐	
EmploymentForm	string	{res:itgen_fnx_employmentform}	☐	
FirstDay	datetime	{res:itgen_fnx_firstday}	☐	
FirstName	string	{res:itgen_fnx_firstname}	☐	
ForaType	string	{res:itgen_fnx_foratype}	☐	
FullName	string	{res:itgen_fnx_fullname}	☐	
HourlyPay	string	{res:itgen_fnx_hourlypay}	☐	
Inactive	boolean	{res:itgen_fnx_inactive}	☐	
JobTitle	string(30)	{res:itgen_fnx_jobtitle}	☐	
LastName	string	{res:itgen_fnx_lastname}	☐	
MonthlySalary	string	{res:itgen_fnx_monthlysalary}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
NonRecurringTax	string	{res:itgen_fnx_nonrecurringtax}	☐	
PersonalIdentityNumber	string	{res:itgen_fnx_personalidentitynumber}	☐	
PersonelType	string	{res:itgen_fnx_personeltype}	☐	
Phone1	string	{res:itgen_fnx_phone1}	☐	
Phone2	string	{res:itgen_fnx_phone2}	☐	
PostCode	string	{res:itgen_fnx_postcode}	☐	
Project	string	{res:itgen_fnx_project}	☐	
SalaryForm	string	{res:itgen_fnx_salaryform}	☐	
ScheduleId_2	string	{res:itgen_fnx_scheduleid_2}	☐	
ScheduleId	string	{res:itgen_fnx_scheduleid}	☐	
TaxAllow ance	string	{res:itgen_fnx_taxallow ance}	☐	
TaxColumn	int32	{res:itgen_fnx_taxcolumn}	☐	
TaxTable	string	{res:itgen_fnx_taxtable}	☐	
VacationDaysPaid	float	{res:itgen_fnx_vacationdayspaid}	☐	
VacationDaysPendingPaid	float	{res:itgen_fnx_vacationdayspendingpaid}	☐	
VacationDaysPendingPrepaid	float	{res:itgen_fnx_vacationdayspendingprepaid}	☐	
VacationDaysPendingSaved	float	{res:itgen_fnx_vacationdayspendingsaved}	☐	
VacationDaysPendingSavedYear1	float	{res:itgen_fnx_vacationdayspendingsavedyear1}	☐	
VacationDaysPendingSavedYear2	float	{res:itgen_fnx_vacationdayspendingsavedyear2}	☐	
VacationDaysPendingSavedYear3	float	{res:itgen_fnx_vacationdayspendingsavedyear3}	☐	
VacationDaysPendingSavedYear4	float	{res:itgen_fnx_vacationdayspendingsavedyear4}	☐	
VacationDaysPendingSavedYear5	float	{res:itgen_fnx_vacationdayspendingsavedyear5}	☐	
VacationDaysPendingSavedYear6Plus	float	{res:itgen_fnx_vacationdayspendingsavedye	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		ar6plus}		
VacationDaysPendingUnpaid	float	{res:itgen_fnx_vacationdayspendingunpaid}	☐	
VacationDaysPrepaid	float	{res:itgen_fnx_vacationdaysprepaid}	☐	
VacationDaysRegisteredPaid	float	{res:itgen_fnx_vacationdaysregisteredpaid}	☐	
VacationDaysRegisteredPrepaid	float	{res:itgen_fnx_vacationdaysregisteredprepaid}	☐	
VacationDaysRegisteredSaved	float	{res:itgen_fnx_vacationdaysregisteredsaved}	☐	
VacationDaysRegisteredSavedYear1	float	{res:itgen_fnx_vacationdaysregisteredsavedyear1}	☐	
VacationDaysRegisteredSavedYear2	float	{res:itgen_fnx_vacationdaysregisteredsavedyear2}	☐	
VacationDaysRegisteredSavedYear3	float	{res:itgen_fnx_vacationdaysregisteredsavedyear3}	☐	
VacationDaysRegisteredSavedYear4	float	{res:itgen_fnx_vacationdaysregisteredsavedyear4}	☐	
VacationDaysRegisteredSavedYear5	float	{res:itgen_fnx_vacationdaysregisteredsavedyear5}	☐	
VacationDaysRegisteredSavedYear6Plus	float	{res:itgen_fnx_vacationdaysregisteredsavedyear6plus}	☐	
VacationDaysRegisteredUnpaid	float	{res:itgen_fnx_vacationdaysregisteredunpaid}	☐	
VacationDaysSaved	float	{res:itgen_fnx_vacationdayssaved}	☐	
VacationDaysSavedEmploymentRateYear1	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear1}	☐	
VacationDaysSavedEmploymentRateYear2	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear2}	☐	
VacationDaysSavedEmploymentRateYear3	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear3}	☐	
VacationDaysSavedEmploymentRateYear4	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear4}	☐	
VacationDaysSavedEmploymentRateYear5	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear5}	☐	
VacationDaysSavedEmploymentRateYear6Plus	float	{res:itgen_fnx_vacationdayssavedemployme	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		ntrateyear6plus}		
VacationDaysSavedYear1	float	{res:itgen_fnx_vacationdayssavedyear1}	☐	
VacationDaysSavedYear2	float	{res:itgen_fnx_vacationdayssavedyear2}	☐	
VacationDaysSavedYear3	float	{res:itgen_fnx_vacationdayssavedyear3}	☐	
VacationDaysSavedYear4	float	{res:itgen_fnx_vacationdayssavedyear4}	☐	
VacationDaysSavedYear5	float	{res:itgen_fnx_vacationdayssavedyear5}	☐	
VacationDaysSavedYear6Plus	float	{res:itgen_fnx_vacationdayssavedyear6plus}	☐	
VacationDaysUnpaid	float	{res:itgen_fnx_vacationdaysunpaid}	☐	

EmployeeDatedWages

Retrieve a list of employeesScheduleId, MonthlySalary and HourlyPay reflect current values, allScheduleIds are returned in DatedSchedules and all MonthlySalary andHourlyPay pairs are returned in DatedWages.

Catalog: Fortnox

Schema: Employees

Label: {res:itgen_fnx_employeedatedwages_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function EmployeeDatedWages. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function EmployeeDatedWages are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Address1	string	{res:itgen_fnx_address1}	☐	
Address2	string	{res:itgen_fnx_address2}	☐	
AutoNonRecurringTax	boolean	{res:itgen_fnx_autononrecurringtax}	☐	
AverageHourlyWage	string	{res:itgen_fnx_averagehourlywage}	☐	
AverageWeeklyHours	string	{res:itgen_fnx_averageweeklyhours}	☐	
BankAccountNo	string	{res:itgen_fnx_bankaccountno}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
City	string	{res:itgen_fnx_city}	☐	
ClearingNo	string	{res:itgen_fnx_clearingno}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	
Email	string	{res:itgen_fnx_email}	☐	
EmployedTo	datetime	{res:itgen_fnx_employedto}	☐	
EmployeeId_2	string(15)	{res:itgen_fnx_employeeid_2}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
EmployeeId	string(15)	{res:itgen_fnx_employeeid}	☐	
EmploymentDate	datetime	{res:itgen_fnx_employmentdate}	☐	
EmploymentForm	string	{res:itgen_fnx_employmentform}	☐	
FirstDay	datetime	{res:itgen_fnx_firstday}	☐	
FirstName	string	{res:itgen_fnx_firstname}	☐	
ForaType	string	{res:itgen_fnx_foratype}	☐	
FullName	string	{res:itgen_fnx_fullname}	☐	
HourlyPay_2	string	{res:itgen_fnx_hourlypay_2}	☐	
HourlyPay	string	{res:itgen_fnx_hourlypay}	☐	
Inactive	boolean	{res:itgen_fnx_inactive}	☐	
JobTitle	string(30)	{res:itgen_fnx_jobtitle}	☐	
LastName	string	{res:itgen_fnx_lastname}	☐	
MonthlySalary_2	string	{res:itgen_fnx_monthlysalary_2}	☐	
MonthlySalary	string	{res:itgen_fnx_monthlysalary}	☐	
NonRecurringTax	string	{res:itgen_fnx_nonrecurringtax}	☐	
PersonalIdentityNumber	string	{res:itgen_fnx_personalidentitynumber}	☐	
PersonelType	string	{res:itgen_fnx_personeltype}	☐	
Phone1	string	{res:itgen_fnx_phone1}	☐	
Phone2	string	{res:itgen_fnx_phone2}	☐	
PostCode	string	{res:itgen_fnx_postcode}	☐	
Project	string	{res:itgen_fnx_project}	☐	
SalaryForm	string	{res:itgen_fnx_salaryform}	☐	
ScheduleId	string	{res:itgen_fnx_scheduleid}	☐	
TaxAllowance	string	{res:itgen_fnx_taxallowance}	☐	
TaxColumn	int32	{res:itgen_fnx_taxcolumn}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TaxTable	string	{res:itgen_fnx_taxtable}	☐	
VacationDaysPaid	float	{res:itgen_fnx_vacationdayspaid}	☐	
VacationDaysPendingPaid	float	{res:itgen_fnx_vacationdayspendingpaid}	☐	
VacationDaysPendingPrepaid	float	{res:itgen_fnx_vacationdayspendingprepaid}	☐	
VacationDaysPendingSaved	float	{res:itgen_fnx_vacationdayspendingsaved}	☐	
VacationDaysPendingSavedYear1	float	{res:itgen_fnx_vacationdayspendingsavedyear1}	☐	
VacationDaysPendingSavedYear2	float	{res:itgen_fnx_vacationdayspendingsavedyear2}	☐	
VacationDaysPendingSavedYear3	float	{res:itgen_fnx_vacationdayspendingsavedyear3}	☐	
VacationDaysPendingSavedYear4	float	{res:itgen_fnx_vacationdayspendingsavedyear4}	☐	
VacationDaysPendingSavedYear5	float	{res:itgen_fnx_vacationdayspendingsavedyear5}	☐	
VacationDaysPendingSavedYear6Plus	float	{res:itgen_fnx_vacationdayspendingsavedyear6plus}	☐	
VacationDaysPendingUnpaid	float	{res:itgen_fnx_vacationdayspendingunpaid}	☐	
VacationDaysPrepaid	float	{res:itgen_fnx_vacationdaysprepaid}	☐	
VacationDaysRegisteredPaid	float	{res:itgen_fnx_vacationdaysregisteredpaid}	☐	
VacationDaysRegisteredPrepaid	float	{res:itgen_fnx_vacationdaysregisteredprepaid}	☐	
VacationDaysRegisteredSaved	float	{res:itgen_fnx_vacationdaysregisteredsaved}	☐	
VacationDaysRegisteredSavedYear1	float	{res:itgen_fnx_vacationdaysregisteredsavedyear1}	☐	
VacationDaysRegisteredSavedYear2	float	{res:itgen_fnx_vacationdaysregisteredsavedyear2}	☐	
VacationDaysRegisteredSavedYear3	float	{res:itgen_fnx_vacationdaysregisteredsavedyear3}	☐	
VacationDaysRegisteredSavedYear4	float	{res:itgen_fnx_vacationdaysregisteredsavedyear4}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
VacationDaysRegisteredSavedYear5	float	{res:itgen_fnx_vacationdaysregisteredsavedyear5}	☐	
VacationDaysRegisteredSavedYear6Plus	float	{res:itgen_fnx_vacationdaysregisteredsavedyear6plus}	☐	
VacationDaysRegisteredUnpaid	float	{res:itgen_fnx_vacationdaysregisteredunpaid}	☐	
VacationDaysSaved	float	{res:itgen_fnx_vacationdayssaved}	☐	
VacationDaysSavedEmploymentRateYear1	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear1}	☐	
VacationDaysSavedEmploymentRateYear2	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear2}	☐	
VacationDaysSavedEmploymentRateYear3	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear3}	☐	
VacationDaysSavedEmploymentRateYear4	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear4}	☐	
VacationDaysSavedEmploymentRateYear5	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear5}	☐	
VacationDaysSavedEmploymentRateYear6Plus	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear6plus}	☐	
VacationDaysSavedYear1	float	{res:itgen_fnx_vacationdayssavedyear1}	☐	
VacationDaysSavedYear2	float	{res:itgen_fnx_vacationdayssavedyear2}	☐	
VacationDaysSavedYear3	float	{res:itgen_fnx_vacationdayssavedyear3}	☐	
VacationDaysSavedYear4	float	{res:itgen_fnx_vacationdayssavedyear4}	☐	
VacationDaysSavedYear5	float	{res:itgen_fnx_vacationdayssavedyear5}	☐	
VacationDaysSavedYear6Plus	float	{res:itgen_fnx_vacationdayssavedyear6plus}	☐	
VacationDaysUnpaid	float	{res:itgen_fnx_vacationdaysunpaid}	☐	

EmployeeDatedWagesByEmployeeId

Retrieve a specific employeeScheduleId, MonthlySalary and HourlyPay reflect current values, allScheduleIds are returned in DatedSchedules and all MonthlySalary andHourlyPay pairs are returned in DatedWages.

Catalog: Fortnox

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: Employees

Label: {res:itgen_fnx_employeedatedwagesbyemployeeid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function EmployeeDatedWagesByEmployeeId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
EmployeeId	string	☑		identifies the employee
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function EmployeeDatedWagesByEmployeeId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Address1	string	{res:itgen_fnx_addres	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		s1}		
Address2	string	{res:itgen_fnx_address2}	☐	
AutoNonRecurringTax	boolean	{res:itgen_fnx_autononrecurringtax}	☐	
AverageHourlyWage	string	{res:itgen_fnx_averagehourlywage}	☐	
AverageWeeklyHours	string	{res:itgen_fnx_averageweeklyhours}	☐	
BankAccountNo	string	{res:itgen_fnx_bankaccountno}	☐	
City	string	{res:itgen_fnx_city}	☐	
ClearingNo	string	{res:itgen_fnx_clearingno}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	
Email	string	{res:itgen_fnx_email}	☐	
EmployedTo	datetime	{res:itgen_fnx_employedto}	☐	
EmployeeId_2	string(15)	{res:itgen_fnx_employeeid_2}	☐	
EmployeeId	string(15)	{res:itgen_fnx_employeeid}	☐	
EmploymentDate	datetime	{res:itgen_fnx_employmentdate}	☐	
EmploymentForm	string	{res:itgen_fnx_employmentform}	☐	
FirstDay	datetime	{res:itgen_fnx_firstday}	☐	
FirstName	string	{res:itgen_fnx_firstname}	☐	
ForaType	string	{res:itgen_fnx_foratype}	☐	
FullName	string	{res:itgen_fnx_fullname}	☐	
HourlyPay_2	string	{res:itgen_fnx_hourlypay_2}	☐	
HourlyPay	string	{res:itgen_fnx_hourlypay}	☐	
Inactive	boolean	{res:itgen_fnx_inactive}	☐	
JobTitle	string(30)	{res:itgen_fnx_jobtitle}	☐	
LastName	string	{res:itgen_fnx_lastname}	☐	
MonthlySalary_2	string	{res:itgen_fnx_monthlysalary_2}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
MonthlySalary	string	{res:itgen_fnx_monthlysalary}	☐	
NonRecurringTax	string	{res:itgen_fnx_nonrecurringtax}	☐	
PersonalIdentityNumber	string	{res:itgen_fnx_personalidentitynumber}	☐	
PersonelType	string	{res:itgen_fnx_personeltype}	☐	
Phone1	string	{res:itgen_fnx_phone1}	☐	
Phone2	string	{res:itgen_fnx_phone2}	☐	
PostCode	string	{res:itgen_fnx_postcode}	☐	
Project	string	{res:itgen_fnx_project}	☐	
SalaryForm	string	{res:itgen_fnx_salaryform}	☐	
ScheduleId	string	{res:itgen_fnx_scheduleid}	☐	
TaxAllow ance	string	{res:itgen_fnx_taxallow ance}	☐	
TaxColumn	int32	{res:itgen_fnx_taxcolumn}	☐	
TaxTable	string	{res:itgen_fnx_taxtable}	☐	
VacationDaysPaid	float	{res:itgen_fnx_vacationdayspaid}	☐	
VacationDaysPendingPaid	float	{res:itgen_fnx_vacationdayspendingpaid}	☐	
VacationDaysPendingPrepaid	float	{res:itgen_fnx_vacationdayspendingprepaid}	☐	
VacationDaysPendingSaved	float	{res:itgen_fnx_vacationdayspendingsaved}	☐	
VacationDaysPendingSavedYear1	float	{res:itgen_fnx_vacationdayspendingsavedyear1}	☐	
VacationDaysPendingSavedYear2	float	{res:itgen_fnx_vacationdayspendingsavedyear2}	☐	
VacationDaysPendingSavedYear3	float	{res:itgen_fnx_vacationdayspendingsavedyear3}	☐	
VacationDaysPendingSavedYear4	float	{res:itgen_fnx_vacationdayspendingsavedyear4}	☐	
VacationDaysPendingSavedYear5	float	{res:itgen_fnx_vacationdayspendingsavedyear5}	☐	
VacationDaysPendingSavedYear6Plus	float	{res:itgen_fnx_vacationdayspendingsavedye	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		ar6plus}		
VacationDaysPendingUnpaid	float	{res:itgen_fnx_vacationdayspendingunpaid}	☐	
VacationDaysPrepaid	float	{res:itgen_fnx_vacationdaysprepaid}	☐	
VacationDaysRegisteredPaid	float	{res:itgen_fnx_vacationdaysregisteredpaid}	☐	
VacationDaysRegisteredPrepaid	float	{res:itgen_fnx_vacationdaysregisteredprepaid}	☐	
VacationDaysRegisteredSaved	float	{res:itgen_fnx_vacationdaysregisteredsaved}	☐	
VacationDaysRegisteredSavedYear1	float	{res:itgen_fnx_vacationdaysregisteredsavedyear1}	☐	
VacationDaysRegisteredSavedYear2	float	{res:itgen_fnx_vacationdaysregisteredsavedyear2}	☐	
VacationDaysRegisteredSavedYear3	float	{res:itgen_fnx_vacationdaysregisteredsavedyear3}	☐	
VacationDaysRegisteredSavedYear4	float	{res:itgen_fnx_vacationdaysregisteredsavedyear4}	☐	
VacationDaysRegisteredSavedYear5	float	{res:itgen_fnx_vacationdaysregisteredsavedyear5}	☐	
VacationDaysRegisteredSavedYear6Plus	float	{res:itgen_fnx_vacationdaysregisteredsavedyear6plus}	☐	
VacationDaysRegisteredUnpaid	float	{res:itgen_fnx_vacationdaysregisteredunpaid}	☐	
VacationDaysSaved	float	{res:itgen_fnx_vacationdayssaved}	☐	
VacationDaysSavedEmploymentRateYear1	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear1}	☐	
VacationDaysSavedEmploymentRateYear2	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear2}	☐	
VacationDaysSavedEmploymentRateYear3	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear3}	☐	
VacationDaysSavedEmploymentRateYear4	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear4}	☐	
VacationDaysSavedEmploymentRateYear5	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear5}	☐	
VacationDaysSavedEmploymentRateYear6Plus	float	{res:itgen_fnx_vacationdayssavedemployme	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		ntrateyear6plus}		
VacationDaysSavedYear1	float	{res:itgen_fnx_vacationdayssavedyear1}	☐	
VacationDaysSavedYear2	float	{res:itgen_fnx_vacationdayssavedyear2}	☐	
VacationDaysSavedYear3	float	{res:itgen_fnx_vacationdayssavedyear3}	☐	
VacationDaysSavedYear4	float	{res:itgen_fnx_vacationdayssavedyear4}	☐	
VacationDaysSavedYear5	float	{res:itgen_fnx_vacationdayssavedyear5}	☐	
VacationDaysSavedYear6Plus	float	{res:itgen_fnx_vacationdayssavedyear6plus}	☐	
VacationDaysUnpaid	float	{res:itgen_fnx_vacationdaysunpaid}	☐	

Employees

Retrieve a list of employeesScheduleId, MonthlySalary and HourlyPay reflect current values, allScheduleIds are returned in DatedSchedules and all MonthlySalary andHourlyPay pairs are returned in DatedWages.

Catalog: Fortnox

Schema: Employees

Primary Keys: EmployeeId

Label: {res:itgen_fnx_employees_plural}

Retrieve: true

Insert: true

Update: true

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Employees. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Employees are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
Address1	string	{res:itgen_fnx_address1}	☐	
Address2	string	{res:itgen_fnx_address2}	☐	
AutoNonRecurringTax	boolean	{res:itgen_fnx_autononrecurringtax}	☐	
AverageHourlyWage	string	{res:itgen_fnx_averagehourlywage}	☐	
AverageWeeklyHours	string	{res:itgen_fnx_averageweeklyhours}	☐	
BankAccountNo	string	{res:itgen_fnx_bankaccountno}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
City	string	{res:itgen_fnx_city}	☐	
ClearingNo	string	{res:itgen_fnx_clearingno}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	
Email	string	{res:itgen_fnx_email}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
EmployedTo	datetime	{res:itgen_fnx_employedto}	☐	
EmployeeId	string(15)	{res:itgen_fnx_employeeid}	☐	
EmploymentDate	datetime	{res:itgen_fnx_employmentdate}	☐	
EmploymentForm	string	{res:itgen_fnx_employmentform}	☐	
FirstName	string	{res:itgen_fnx_firstname}	☐	
ForaType	string	{res:itgen_fnx_foratype}	☐	
FullName	string	{res:itgen_fnx_fullname}	☐	
HourlyPay	string	{res:itgen_fnx_hourlypay}	☐	
Inactive	boolean	{res:itgen_fnx_inactive}	☐	
JobTitle	string(30)	{res:itgen_fnx_jobtitle}	☐	
LastName	string	{res:itgen_fnx_lastname}	☐	
MonthlySalary	string	{res:itgen_fnx_monthlysalary}	☐	
NonRecurringTax	string	{res:itgen_fnx_nonrecurringtax}	☐	
PersonalIdentityNumber	string	{res:itgen_fnx_personalidentitynumber}	☐	
PersonelType	string	{res:itgen_fnx_personeltype}	☐	
Phone1	string	{res:itgen_fnx_phone1}	☐	
Phone2	string	{res:itgen_fnx_phone2}	☐	
PostCode	string	{res:itgen_fnx_postcode}	☐	
Project	string	{res:itgen_fnx_project}	☐	
SalaryForm	string	{res:itgen_fnx_salaryform}	☐	
ScheduleId	string	{res:itgen_fnx_scheduleid}	☐	
TaxAllowance	string	{res:itgen_fnx_taxallowance}	☐	
TaxColumn	int32	{res:itgen_fnx_taxcolumn}	☐	
TaxTable	string	{res:itgen_fnx_taxtable}	☐	
VacationDaysPaid	float	{res:itgen_fnx_vacationdayspaid}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
VacationDaysPendingPaid	float	{res:itgen_fnx_vacationdayspendingpaid}	☐	
VacationDaysPendingPrepaid	float	{res:itgen_fnx_vacationdayspendingprepaid}	☐	
VacationDaysPendingSaved	float	{res:itgen_fnx_vacationdayspendingsaved}	☐	
VacationDaysPendingSavedYear1	float	{res:itgen_fnx_vacationdayspendingsavedyear1}	☐	
VacationDaysPendingSavedYear2	float	{res:itgen_fnx_vacationdayspendingsavedyear2}	☐	
VacationDaysPendingSavedYear3	float	{res:itgen_fnx_vacationdayspendingsavedyear3}	☐	
VacationDaysPendingSavedYear4	float	{res:itgen_fnx_vacationdayspendingsavedyear4}	☐	
VacationDaysPendingSavedYear5	float	{res:itgen_fnx_vacationdayspendingsavedyear5}	☐	
VacationDaysPendingSavedYear6Plus	float	{res:itgen_fnx_vacationdayspendingsavedyear6plus}	☐	
VacationDaysPendingUnpaid	float	{res:itgen_fnx_vacationdayspendingunpaid}	☐	
VacationDaysPrepaid	float	{res:itgen_fnx_vacationdaysprepaid}	☐	
VacationDaysRegisteredPaid	float	{res:itgen_fnx_vacationdaysregisteredpaid}	☐	
VacationDaysRegisteredPrepaid	float	{res:itgen_fnx_vacationdaysregisteredprepaid}	☐	
VacationDaysRegisteredSaved	float	{res:itgen_fnx_vacationdaysregisteredsaved}	☐	
VacationDaysRegisteredSavedYear1	float	{res:itgen_fnx_vacationdaysregisteredsavedyear1}	☐	
VacationDaysRegisteredSavedYear2	float	{res:itgen_fnx_vacationdaysregisteredsavedyear2}	☐	
VacationDaysRegisteredSavedYear3	float	{res:itgen_fnx_vacationdaysregisteredsavedyear3}	☐	
VacationDaysRegisteredSavedYear4	float	{res:itgen_fnx_vacationdaysregisteredsavedyear4}	☐	
VacationDaysRegisteredSavedYear5	float	{res:itgen_fnx_vacationdaysregisteredsavedyear5}	☐	
VacationDaysRegisteredSavedYear6Plus	float	{res:itgen_fnx_vacationdaysregisteredsaved	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		year6plus}		
VacationDaysRegisteredUnpaid	float	{res:itgen_fnx_vacationdaysregisteredunpaid}	☐	
VacationDaysSaved	float	{res:itgen_fnx_vacationdayssaved}	☐	
VacationDaysSavedEmploymentRateYear1	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear1}	☐	
VacationDaysSavedEmploymentRateYear2	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear2}	☐	
VacationDaysSavedEmploymentRateYear3	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear3}	☐	
VacationDaysSavedEmploymentRateYear4	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear4}	☐	
VacationDaysSavedEmploymentRateYear5	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear5}	☐	
VacationDaysSavedEmploymentRateYear6Plus	float	{res:itgen_fnx_vacationdayssavedemploymentrateyear6plus}	☐	
VacationDaysSavedYear1	float	{res:itgen_fnx_vacationdayssavedyear1}	☐	
VacationDaysSavedYear2	float	{res:itgen_fnx_vacationdayssavedyear2}	☐	
VacationDaysSavedYear3	float	{res:itgen_fnx_vacationdayssavedyear3}	☐	
VacationDaysSavedYear4	float	{res:itgen_fnx_vacationdayssavedyear4}	☐	
VacationDaysSavedYear5	float	{res:itgen_fnx_vacationdayssavedyear5}	☐	
VacationDaysSavedYear6Plus	float	{res:itgen_fnx_vacationdayssavedyear6plus}	☐	
VacationDaysUnpaid	float	{res:itgen_fnx_vacationdaysunpaid}	☐	

3.1.22 Schema: EUVatLimitRegulation

3.1.22.1 Tables

EUVatLimitRegulationByYear

Retrieve details about eu vat limit

Catalog: Fortnox

Schema: EUVatLimitRegulation

Label: {res:itgen_fnx_euvatlimitregulationbyyear_plural}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function EUVatLimitRegulationByYear. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
year	int32	☐		eu vat limit regulation for year, if not provided than this will be set to current year

Columns of Table Function

The columns of the table function EUVatLimitRegulationByYear are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
EUVatLimitRegulation_IsOverLimit	boolean	{res:itgen_fnx_euvatlimitregulation_isoverlimit}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
EUVatLimitRegulation_Limit	int32	{res:itgen_fnx_euvatlimitregulation_limit}	☐	
EUVatLimitRegulation_TotalExclVat	double	{res:itgen_fnx_euvatlimitregulation_totalexclvat}	☐	
EUVatLimitRegulation_Year	string	{res:itgen_fnx_euvatlimitregulation_year}	☐	

3.1.23 Schema: Expenses

3.1.23.1 Tables

ExpenseByCode

Retrieve an expenseRetrieves expense information for specified expense.

Catalog: Fortnox

Schema: Expenses

Label: {res:itgen_fnx_expensebycode_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ExpenseByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
ExpenseCode	string	☑		expenseCode
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ExpenseByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Expense_Account	int32	{res:itgen_fnx_expense_account}	☐	
Expense_Code	string	{res:itgen_fnx_expense_code}	☐	
Expense_Text	string	{res:itgen_fnx_expense_text}	☐	

Expenses

Retrieve expensesRetrieve expense codes.

Catalog: Fortnox

Schema: Expenses

Label: {res:itgen_fnx_expenses_plural}

Retrieve: true

Insert: true

Update: false

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Expenses. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Expenses are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
Account	int32	{res:itgen_fnx_account}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Code	string	{res:itgen_fnx_code}	☐	
Text	string	{res:itgen_fnx_text}	☐	

3.1.24 Schema: Fileattachments_Attachment

3.1.24.1 Tables

AttachmentCountByEntityType

List number of attachments

Catalog: Fortnox

Schema: Fileattachments_Attachment

Label: {res:itgen_fnx_attachmentcountbyentitytype_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function AttachmentCountByEntityType. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
entitytype	string	☑		type of the entities to look for number of attachments on (Values: OF, O, F, C, LGR_IO, LGR_IG)
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AttachmentCountByEntityType are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
entityId	int64	{res:itgen_fnx_entityid}	☐	
numberOfAttachments	int64	{res:itgen_fnx_numberofattachments}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

AttachmentsByEntityType

Get attached files on an entity

Catalog: Fortnox

Schema: Fileattachments_Attachment

Label: {res:itgen_fnx_attachmentsbyentitytype_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AttachmentsByEntityType. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
entitytype	string	☑		type of the entities whose attachments should be fetched (Values: OF, O, F, C, LGR_IO, LGR_IG)
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

The columns of the table function AttachmentsByEntityType are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
entityId	int64	{res:itgen_fnx_entityid}	☐	
entityType	string	{res:itgen_fnx_entitytype}	☐	
fileId	string	{res:itgen_fnx_fileid}	☐	
id	guid	{res:itgen_fnx_id}	☐	
includeOnSend	boolean	{res:itgen_fnx_includeonsend}	☐	

3.1.25 Schema: FinanceInvoices

3.1.25.1 Tables

FinancialInvoiceByNumber

Retrieve a single invoice paymentRetrieves the status and balance of an invoice sent to Fortnox Finans.You need to supply the invoice number in Fortox to retrieve the invoice.Note that invoices sent with the old "Noxbox" platform will not have the "ServiceName"property in the response. This new property is added to the response if the invoice issent with the new finance service.Response explanation for Service and ServiceNameService:LEDGERBASE: if the invoice is sent by using the old "Noxbox" platform, or the new finance service with the subtypes "Service Full" or "Service Light". These services are explained above in the "Fortnox Finans services" sectionREMINDER: If the invoice is sent by the new finance service, with the service Reminder ServiceServiceName (only provided for new finance service invoices):SERVICE_FULL: Ledgerbase service with automatic reminders is usedSERVICE_LIGHT: Ledgerbase service without automatic reminders is used.REMINDER_SERVICE: Reminder service is used

Catalog: Fortnox

Schema: FinanceInvoices

Label: {res:itgen_fnx_financialinvoicebynumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FinancialInvoiceByNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
Number	string	☑		The Fortnox invoice number
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function FinancialInvoiceByNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
NoxFinansInvoice__url	string	{res:itgen_fnx_noxfinansinvoice__url}	☐	
NoxFinansInvoice_BalanceIncludeFees	float	{res:itgen_fnx_noxfinansinvoice_balanceincludefees}	☐	
NoxFinansInvoice_BalanceIncludeFeesCurrency	float	{res:itgen_fnx_noxfinansinvoice_balanceincludefeescurrency}	☐	
NoxFinansInvoice_CurrentCapitalBalance	float	{res:itgen_fnx_noxfinansinvoice_currentcapitalbalance}	☐	
NoxFinansInvoice_CurrentCapitalBalanceCurrency	float	{res:itgen_fnx_noxfinansinvoice_currentcapitalbalancecurrency}	☐	
NoxFinansInvoice_InvoiceDocumentURL	string	{res:itgen_fnx_noxfinansinvoice_invoicedocumenturl}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
NoxFinansInvoice_InvoiceNumber	int32	{res:itgen_fnx_noxfinansinvoice_invoicenumber}	☐	
NoxFinansInvoice_NextEvent	string	{res:itgen_fnx_noxfinansinvoice_nextevent}	☐	
NoxFinansInvoice_NextEventDate	datetime	{res:itgen_fnx_noxfinansinvoice_nexteventdate}	☐	
NoxFinansInvoice_OCRNumber	string	{res:itgen_fnx_noxfinansinvoice_ocrnumber}	☐	
NoxFinansInvoice_Service	string	{res:itgen_fnx_noxfinansinvoice_service}	☐	
NoxFinansInvoice_ServiceName	string	{res:itgen_fnx_noxfinansinvoice_servicename}	☐	
NoxFinansInvoice_Status	string	{res:itgen_fnx_noxfinansinvoice_status}	☐	

3.1.26 Schema: FinancialYears

3.1.26.1 Tables

FinancialYearById

Retrieve financial year by id

Catalog: Fortnox

Schema: FinancialYears

Label: {res:itgen_fnx_financialyearbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FinancialYearById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
Id	int32	☑		identifies the year
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function FinancialYearById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
FinancialYear__url	string	{res:itgen_fnx_financialyear__url}	☐	
FinancialYear_AccountChartType	string	{res:itgen_fnx_financialyear_accountcharttype}	☐	
FinancialYear_AccountingMethod	string	{res:itgen_fnx_financialyear_accountingmethod}	☐	
FinancialYear_FromDate	datetime	{res:itgen_fnx_financialyear_fromdate}	☐	
FinancialYear_Id	int32	{res:itgen_fnx_financialyear_id}	☐	
FinancialYear_ToDate	datetime	{res:itgen_fnx_financialyear_todate}	☐	

FinancialYears

Retrieve a list of financial yearsAdd the query param to filter on specific date.

Catalog: Fortnox

Schema: FinancialYears

Label: {res:itgen_fnx_financialyears_plural}

Retrieve: true

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Insert: true

Update: false

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FinancialYears. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
Date	datetime	☐		date to filter on, for example 2020-06-30
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function FinancialYears are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
AccountChartType	string	{res:itgen_fnx_accountcharttype}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
AccountingMethod	string	{res:itgen_fnx_accountingmethod}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
FromDate	datetime	{res:itgen_fnx_fromdate}	☐	
Id	int32	{res:itgen_fnx_id}	☐	
ToDate	datetime	{res:itgen_fnx_todate}	☐	

3.1.27 Schema: Inbox

3.1.27.1 Tables

Inbox

Retrieve the root folder containing files and folders

Catalog: Fortnox

Schema: Inbox

Label: {res:itgen_fnx_inbox_plural}

Retrieve: true

Insert: true

Update: false

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Inbox. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Inbox are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
Folder__url	string	{res:itgen_fnx_folder__url}	☐	
Folder_Email	string	{res:itgen_fnx_folder_email}	☐	
Folder_Id	string	{res:itgen_fnx_folder_id}	☐	
Folder_Name	string(30)	{res:itgen_fnx_folder_name}	☐	

InboxById

Retrieve a single file

Catalog: Fortnox

Schema: Inbox

Label: {res:itgen_fnx_inboxbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InboxById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
Id	string	☑		identifies the folder
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function InboxById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
TEXT	byte[]	{res:itgen_fnx_text}	☐	

InboxFiles

Retrieve the root folder containing files and folders

Catalog: Fortnox

Schema: Inbox

Label: {res:itgen_fnx_inboxfiles_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function InboxFiles. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function InboxFiles are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
@url_2	string	{res:itgen_fnx_@url_2}	☐	
ArchiveFileId	string	{res:itgen_fnx_archivefileid}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Comments	string	{res:itgen_fnx_comments}	☐	
Email	string	{res:itgen_fnx_email}	☐	
Id_2	string	{res:itgen_fnx_id_2}	☐	
Id	string	{res:itgen_fnx_id}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Name_2	string(30)	{res:itgen_fnx_name_2}	☐	
Name	string	{res:itgen_fnx_name}	☐	
Path	string	{res:itgen_fnx_path}	☐	
Size	int32	{res:itgen_fnx_size}	☐	

InboxFolders

Retrieve the root folder containing files and folders

Catalog: Fortnox

Schema: Inbox

Label: {res:itgen_fnx_inboxfolders_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InboxFolders. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function InboxFolders are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
@url_2	string	{res:itgen_fnx_@url_2}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Email	string	{res:itgen_fnx_email}	☐	
Id_2	string	{res:itgen_fnx_id_2}	☐	
Id	string	{res:itgen_fnx_id}	☐	
Name_2	string(30)	{res:itgen_fnx_name_2}	☐	
Name	string	{res:itgen_fnx_name}	☐	

3.1.28 Schema: IntegrationDeveloper_IntegrationRatings

3.1.28.1 Tables

Ratings

List rating and reviews for integrations that you own

Catalog: Fortnox

Schema: IntegrationDeveloper_IntegrationRatings

Label: {res:itgen_fnx_ratings_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Ratings. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Ratings are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
comment	string	{res:itgen_fnx_comment}	☐	
companyEmployeeRange	string	{res:itgen_fnx_companyemployeerange}	☐	
created	datetime	{res:itgen_fnx_created}	☐	
integrationId	int64	{res:itgen_fnx_integrationid}	☐	
isMyRating	boolean	{res:itgen_fnx_ismyrating}	☐	
rating	int32	{res:itgen_fnx_rating}	☐	
ratingId	int32	{res:itgen_fnx_ratingid}	☐	
response_comment	string	{res:itgen_fnx_response_comment}	☐	
response_created	datetime	{res:itgen_fnx_response_created}	☐	
response_updated	datetime	{res:itgen_fnx_response_updated}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
updated	datetime	{res:itgen_fnx_updated}	☐	

3.1.29 Schema: IntegrationDeveloper_IntegrationSales

3.1.29.1 Tables

IntegrationSalesByIntegrationId

Resolves sales information of an integrationPrerequisitesThe partner has an active developer account and a published integration that is purchased through Fortnox.

Catalog: Fortnox

Schema: IntegrationDeveloper_IntegrationSales

Label: {res:itgen_fnx_integrationsalesbyintegrationid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function IntegrationSalesByIntegrationId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
integrationId	int64	☑		of the integration to look up sales information for
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function IntegrationSalesByIntegrationId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
amount	int32	{res:itgen_fnx_amount}	☐	
date	datetime	{res:itgen_fnx_date}	☐	
orderId	string	{res:itgen_fnx_orderid}	☐	
tenantId	string	{res:itgen_fnx_tenantid}	☐	
type	string	{res:itgen_fnx_type}	☐	

3.1.30 Schema: IntegrationPartner_IntegrationSales

3.1.30.1 Tables

AppSalesByAppId

Resolves sales information and active users of an integrationPrerequisitesThe partner has an active developer account and a published integration that is purchased through Fortnox.

Catalog: Fortnox

Schema: IntegrationPartner_IntegrationSales

Label: {res:itgen_fnx_appsalesbyappid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AppSalesByAppId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
appId	string	☑		clientId of the integration to look up sales information for
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AppSalesByAppId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
appId	string	{res:itgen_fnx_appid}	☐	

AppSalesByAppIdAndTenantId

Resolves sales information and active users of an integrationPrerequisitesThe partner has an active developer account and a published integration that is purchased through Fortnox.

Catalog: Fortnox

Schema: IntegrationPartner_IntegrationSales

Label: {res:itgen_fnx_appsalesbyappidandtenantid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AppSalesByAppIdAndTenantId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
appId	string	☑		clientId of the integration to look up sales information for
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
tenantId	int64	☑		tenantId of the tenant to look up sales information for
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AppSalesByAppIdAndTenantId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
appId	string	{res:itgen_fnx_appid}	☐	

AppSalesTenantPurchasesByAppId

Resolves sales information and active users of an integrationPrerequisitesThe partner has an active developer account and a published integration that is purchased through Fortnox.

Catalog: Fortnox

Schema: IntegrationPartner_IntegrationSales

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: {res:itgen_fnx_appsalestenantpurchasesbyappid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AppSalesTenantPurchasesByAppId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
appId	string	☑		clientId of the integration to look up sales information for
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AppSalesTenantPurchasesByAppId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
amount	int64	{res:itgen_fnx_amount}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
appId	string	{res:itgen_fnx_appid}	☐	
identifier	string	{res:itgen_fnx_identifier}	☐	
purchaseTime	string	{res:itgen_fnx_purchasetime}	☐	
tenantId	string	{res:itgen_fnx_tenantid}	☐	

AppSalesTenantsByAppId

Resolves sales information and active users of an integrationPrerequisitesThe partner has an active developer account and a published integration that is purchased through Fortnox.

Catalog: Fortnox

Schema: IntegrationPartner_IntegrationSales

Label: {res:itgen_fnx_appsalestenantsbyappid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AppSalesTenantsByAppId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
appId	string	☑		clientId of the integration to look up sales information for
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AppSalesTenantsByAppId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
appId	string	{res:itgen_fnx_appid}	☐	
tenantId	string	{res:itgen_fnx_tenantid}	☐	

AppSalesTenantTerminationsByAppId

Resolves sales information and active users of an integrationPrerequisitesThe partner has an active developer account and a published integration that is purchased through Fortnox.

Catalog: Fortnox

Schema: IntegrationPartner_IntegrationSales

Label: {res:itgen_fnx_appsalestenantterminationsbyappid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AppSalesTenantTerminationsByAppId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
appId	string	☑		clientId of the integration to look up sales information for
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AppSalesTenantTerminationsByAppId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
amount	int64	{res:itgen_fnx_amount}	☐	
appId	string	{res:itgen_fnx_appid}	☐	
identifier	string	{res:itgen_fnx_identifier}	☐	
onCancellationTime	string	{res:itgen_fnx_oncancellationtime}	☐	
tenantId	string	{res:itgen_fnx_tenantid}	☐	

AppSalesTenantUsersByAppId

Resolves sales information and active users of an integrationPrerequisitesThe partner has an active developer account and a published integration that is purchased through Fortnox.

Catalog: Fortnox

Schema: IntegrationPartner_IntegrationSales

Label: {res:itgen_fnx_appsalestenantusersbyappid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AppSalesTenantUsersByAppId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
appId	string	☑		clientId of the integration to look up sales information for
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AppSalesTenantUsersByAppId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
activationTime	string	{res:itgen_fnx_activationtime}	☐	
appId	string	{res:itgen_fnx_appid}	☐	
email	string	{res:itgen_fnx_email}	☐	
externalSourceId	string	{res:itgen_fnx_externalsourceid}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
name	string	{res:itgen_fnx_name}	☐	
tenantId	string	{res:itgen_fnx_tenantid}	☐	
userType	string	{res:itgen_fnx_usertype}	☐	

AppSaleTenantPurchasesByAppIdAndTenantId

Resolves sales information and active users of an integrationPrerequisitesThe partner has an active developer account and a published integration that is purchased through Fortnox.

Catalog: Fortnox

Schema: IntegrationPartner_IntegrationSales

Label: {res:itgen_fnx_appsaletenantpurchasesbyappidandtenantid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AppSaleTenantPurchasesByAppIdAndTenantId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
appId	string	☑		clientId of the integration to look up sales information for
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
tenantId	int64	☑		tenantId of the tenant to look up sales information for
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AppSaleTenantPurchasesByAppIdAndTenantId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
amount	int64	{res:itgen_fnx_amount}	☐	
appId	string	{res:itgen_fnx_appid}	☐	
identifier	string	{res:itgen_fnx_identifier}	☐	
purchaseTime	string	{res:itgen_fnx_purchasetime}	☐	
tenantId	string	{res:itgen_fnx_tenantid}	☐	

AppSaleTenantsByAppIdAndTenantId

Resolves sales information and active users of an integrationPrerequisitesThe partner has an active developer account and a published integration that is purchased through Fortnox.

Catalog: Fortnox

Schema: IntegrationPartner_IntegrationSales

Label: {res:itgen_fnx_appsaletenantsbyappidandtenantid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AppSaleTenantsByAppIdAndTenantId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
appId	string	☑		clientId of the integration to look up sales information for
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
tenantId	int64	☑		tenantId of the tenant to look up sales information for
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AppSaleTenantsByAppIdAndTenantId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
appId	string	{res:itgen_fnx_appid}	☐	
tenantId	string	{res:itgen_fnx_tenantid}	☐	

AppSaleTenantTerminationsByAppIdAndTenantId

Resolves sales information and active users of an integrationPrerequisitesThe partner has an active developer account and a published integration that is purchased through Fortnox.

Catalog: Fortnox

Schema: IntegrationPartner_IntegrationSales

Label: {res:itgen_fnx_appsaletenantterminationsbyappidandtenantid_plural}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AppSaleTenantTerminationsByAppIdAndTenantId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
appId	string	☑		clientId of the integration to look up sales information for
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
tenantId	int64	☑		tenantId of the tenant to look up sales information for
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AppSaleTenantTerminationsByAppIdAndTenantId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
amount	int64	{res:itgen_fnx_amount}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
appId	string	{res:itgen_fnx_appid}	☐	
identifier	string	{res:itgen_fnx_identifier}	☐	
onCancellationTime	string	{res:itgen_fnx_oncancellationtime}	☐	
tenantId	string	{res:itgen_fnx_tenantid}	☐	

AppSaleTenantUsersByAppIdAndTenantId

Resolves sales information and active users of an integrationPrerequisitesThe partner has an active developer account and a published integration that is purchased through Fortnox.

Catalog: Fortnox

Schema: IntegrationPartner_IntegrationSales

Label: {res:itgen_fnx_appsaletenantusersbyappidandtenantid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AppSaleTenantUsersByAppIdAndTenantId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
appId	string	☑		clientId of the integration to look up sales information for
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
tenantId	int64	☑		tenantId of the tenant to look up sales information for
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function AppSaleTenantUsersByAppIdAndTenantId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
activationTime	string	{res:itgen_fnx_activationtime}	☐	
appId	string	{res:itgen_fnx_appid}	☐	
email	string	{res:itgen_fnx_email}	☐	
externalSourceId	string	{res:itgen_fnx_externalsourceid}	☐	
name	string	{res:itgen_fnx_name}	☐	
tenantId	string	{res:itgen_fnx_tenantid}	☐	
userType	string	{res:itgen_fnx_usertype}	☐	

3.1.31 Schema: InvoiceAccruals

3.1.31.1 Tables

InvoiceAccrualLinesByInvoiceNumber

Retrieve a single invoice accrual

Catalog: Fortnox

Schema: InvoiceAccruals

Label: {res:itgen_fnx_invoiceaccruallinesbyinvoicenumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceAccrualLinesByInvoiceNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
InvoiceNumber	int32	☑		identifies the invoice accrual
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function InvoiceAccrualLinesByInvoiceNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Account	int32	{res:itgen_fnx_account}	☐	
AccrualAccount	int32	{res:itgen_fnx_accrualaccount}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Credit	double	{res:itgen_fnx_credit}	☐	
Debit	double	{res:itgen_fnx_debit}	☐	
Description	string	{res:itgen_fnx_description}	☐	
EndDate	datetime	{res:itgen_fnx_enddate}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
InvoiceNumber	int32	{res:itgen_fnx_invoicenumber}	☐	
Period	string	{res:itgen_fnx_period}	☐	
Project	string	{res:itgen_fnx_project}	☐	
RevenueAccount	int32	{res:itgen_fnx_revenueaccount}	☐	
StartDate	datetime	{res:itgen_fnx_startdate}	☐	
Times	int32	{res:itgen_fnx_times}	☐	
Total	double	{res:itgen_fnx_total}	☐	
TransactionInformation	string	{res:itgen_fnx_transactioninformation}	☐	
VATIncluded	boolean	{res:itgen_fnx_vatincluded}	☐	

InvoiceAccruals

Retrieve a list of invoice accrualsThe invoice accruals register can return a list of records or a single record. By specifying a InvoiceNumber in the URL, a single record will be returned. Not specifying a InvoiceNumber will return a list of records.

Catalog: Fortnox

Schema: InvoiceAccruals

Primary Keys: InvoiceNumber

Label: {res:itgen_fnx_invoiceaccruals_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceAccruals. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function InvoiceAccruals are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
Description	string	{res:itgen_fnx_description}	☐	
InvoiceNumber	int32	{res:itgen_fnx_invoicenumber}	☐	
Period	string	{res:itgen_fnx_period}	☐	

InvoiceAccrualsByInvoiceNumber

Retrieve a single invoice accrual

Catalog: Fortnox

Schema: InvoiceAccruals

Label: {res:itgen_fnx_invoiceaccrualsbyinvoicenumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceAccrualsByInvoiceNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
InvoiceNumber	int32	☑		identifies the invoice accrual
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function InvoiceAccrualsByInvoiceNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
InvoiceAccrual__url	string	{res:itgen_fnx_invoiceaccrual__url}	☐	
InvoiceAccrual_AccrualAccount	int32	{res:itgen_fnx_invoiceaccrual_accrualaccount}	☐	
InvoiceAccrual_Description	string	{res:itgen_fnx_invoiceaccrual_description}	☐	
InvoiceAccrual_EndDate	datetime	{res:itgen_fnx_invoiceaccrual_enddate}	☐	
InvoiceAccrual_InvoiceNumber	int32	{res:itgen_fnx_invoiceaccrual_invoicenumber}	☐	
InvoiceAccrual_Period	string	{res:itgen_fnx_invoiceaccrual_period}	☐	
InvoiceAccrual_RevenueAccount	int32	{res:itgen_fnx_invoiceaccrual_revenueacco	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		unt}		
InvoiceAccrual_StartDate	datetime	{res:itgen_fnx_invoiceaccrual_startdate}	☐	
InvoiceAccrual_Times	int32	{res:itgen_fnx_invoiceaccrual_times}	☐	
InvoiceAccrual_Total	double	{res:itgen_fnx_invoiceaccrual_total}	☐	
InvoiceAccrual_VATIncluded	boolean	{res:itgen_fnx_invoiceaccrual_vatincluded}	☐	

3.1.32 Schema: InvoicePayments

3.1.32.1 Tables

InvoicePaymentByNumber

Retrieve a single invoice payment

Catalog: Fortnox

Schema: InvoicePayments

Label: {res:itgen_fnx_invoicepaymentbynumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicePaymentByNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers,

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
Number	string	☑		identifies the invoice payment
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function InvoicePaymentByNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
InvoicePayment__url	string	{res:itgen_fnx_invoicepayment__url}	☐	
InvoicePayment_Amount	double	{res:itgen_fnx_invoicepayment_amount}	☐	
InvoicePayment_AmountCurrency	double	{res:itgen_fnx_invoicepayment_amountcurrency}	☐	
InvoicePayment_Booked	boolean	{res:itgen_fnx_invoicepayment_booked}	☐	
InvoicePayment_Currency	string(3)	{res:itgen_fnx_invoicepayment_currency}	☐	
InvoicePayment_CurrencyRate	double	{res:itgen_fnx_invoicepayment_currencyrate}	☐	
InvoicePayment_CurrencyUnit	double	{res:itgen_fnx_invoicepayment_currencyunit}	☐	
InvoicePayment_ExternalInvoiceReference1	string	{res:itgen_fnx_invoicepayment_externalinvoicereference1}	☐	
InvoicePayment_ExternalInvoiceReference2	string	{res:itgen_fnx_invoicepayment_externalinvoicereference2}	☐	
InvoicePayment_InvoiceCustomerName	string	{res:itgen_fnx_invoicepayment_invoicecustomername}	☐	
InvoicePayment_InvoiceCustomerNumber	string	{res:itgen_fnx_invoicepayment_invoicecustomernumber}	☐	
InvoicePayment_InvoiceDueDate	datetime	{res:itgen_fnx_invoicepayment_invoiceduedate}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
InvoicePayment_InvoiceNumber	int32	{res:itgen_fnx_invoicepayment_invoicenumber}	☐	
InvoicePayment_InvoiceOCR	string	{res:itgen_fnx_invoicepayment_invoiceocr}	☐	
InvoicePayment_InvoiceTotal	string	{res:itgen_fnx_invoicepayment_invoicetotal}	☐	
InvoicePayment_ModeOfPayment	string	{res:itgen_fnx_invoicepayment_modeofpayment}	☐	
InvoicePayment_ModeOfPaymentAccount	int32	{res:itgen_fnx_invoicepayment_modeofpaymentaccount}	☐	
InvoicePayment_Number	string	{res:itgen_fnx_invoicepayment_number}	☐	
InvoicePayment_PaymentDate	datetime	{res:itgen_fnx_invoicepayment_paymentdate}	☐	
InvoicePayment_Source	string	{res:itgen_fnx_invoicepayment_source}	☐	
InvoicePayment_VoucherNumber	int32	{res:itgen_fnx_invoicepayment_vouchernumber}	☐	
InvoicePayment_VoucherSeries	string	{res:itgen_fnx_invoicepayment_voucherseries}	☐	
InvoicePayment_VoucherYear	int32	{res:itgen_fnx_invoicepayment_voucheryear}	☐	

InvoicePayments

Retrieve a list of invoice payments

Catalog: Fortnox

Schema: InvoicePayments

Primary Keys: Number

Label: {res:itgen_fnx_invoicepayments_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicePayments. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
invoicenumber	int32	☐		filter by invoice number
lastmodified	string	☐		filter by last modified
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function InvoicePayments are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
Amount	double	{res:itgen_fnx_amount}	☐	
Booked	boolean	{res:itgen_fnx_booked}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Currency	string(3)	{res:itgen_fnx_currency}	☐	
CurrencyRate	double	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	double	{res:itgen_fnx_currencyunit}	☐	
InvoiceNumber	int32	{res:itgen_fnx_invoicenumber}	☐	
Number	string	{res:itgen_fnx_number}	☐	
PaymentDate	datetime	{res:itgen_fnx_paymentdate}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Source	string	{res:itgen_fnx_source}	☐	
WriteOffExist	boolean	{res:itgen_fnx_writeoffexist}	☐	

InvoicePaymentWriteOffsByPaymentNumber

Retrieve a single invoice payment

Catalog: Fortnox

Schema: InvoicePayments

Label: {res:itgen_fnx_invoicepaymentwriteoffsbypaymentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicePaymentWriteOffsByPaymentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
Number	string	☑		identifies the invoice payment
todate	datetime	☐		Defines a selection based on an end date. Only available for

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function InvoicePaymentWriteOffsByPaymentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AccountNumber	int32	{res:itgen_fnx_accountnumber}	☐	
Amount_2	double	{res:itgen_fnx_amount_2}	☐	
Amount	double	{res:itgen_fnx_amount}	☐	
AmountCurrency	double	{res:itgen_fnx_amountcurrency}	☐	
Booked	boolean	{res:itgen_fnx_booked}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Currency_2	string(3)	{res:itgen_fnx_currency_2}	☐	
Currency	string(3)	{res:itgen_fnx_currency}	☐	
CurrencyRate	double	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	double	{res:itgen_fnx_currencyunit}	☐	
Description	string	{res:itgen_fnx_description}	☐	
ExternalInvoiceReference1	string	{res:itgen_fnx_externalinvoicereference1}	☐	
ExternalInvoiceReference2	string	{res:itgen_fnx_externalinvoicereference2}	☐	
InvoiceCustomerName	string	{res:itgen_fnx_invoicecustomername}	☐	
InvoiceCustomerNumber	string	{res:itgen_fnx_invoicecustomernumber}	☐	
InvoiceDueDate	datetime	{res:itgen_fnx_invoiceduedate}	☐	
InvoiceNumber	int32	{res:itgen_fnx_invoicenumber}	☐	
InvoiceOCR	string	{res:itgen_fnx_invoiceocr}	☐	
InvoiceTotal	string	{res:itgen_fnx_invoicetotal}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ModeOfPayment	string	{res:itgen_fnx_modeofpayment}	☐	
ModeOfPaymentAccount	int32	{res:itgen_fnx_modeofpaymentaccount}	☐	
Number	string	{res:itgen_fnx_number}	☐	
PaymentDate	datetime	{res:itgen_fnx_paymentdate}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Source	string	{res:itgen_fnx_source}	☐	
TransactionInformation	string	{res:itgen_fnx_transactioninformation}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
VoucherYear	int32	{res:itgen_fnx_voucheryear}	☐	

3.1.33 Schema: Invoices

3.1.33.1 Tables

InvoiceByDocumentNumber

Retrieve a single invoice

Catalog: Fortnox

Schema: Invoices

Label: {res:itgen_fnx_invoicebydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the invoice
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function InvoiceByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Invoice__url	string	{res:itgen_fnx_invoice__url}	☐	
Invoice__urlTaxReductionList	string	{res:itgen_fnx_invoice__urltaxreductionlist}	☐	
Invoice_AccountingMethod	string	{res:itgen_fnx_invoice_accountingmethod}	☐	
Invoice_Address1	string	{res:itgen_fnx_invoice_address1}	☐	
Invoice_Address2	string	{res:itgen_fnx_invoice_address2}	☐	
Invoice_AdministrationFee	double	{res:itgen_fnx_invoice_administrationfee}	☐	
Invoice_AdministrationFeeVAT	double	{res:itgen_fnx_invoice_administrationfeevat}	☐	
Invoice_Balance	double	{res:itgen_fnx_invoice_balance}	☐	
Invoice_BasisTaxReduction	double	{res:itgen_fnx_invoice_basistaxreduction}	☐	
Invoice_Booked	boolean	{res:itgen_fnx_invoice_booked}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Invoice_Cancelled	boolean	{res:itgen_fnx_invoice_cancelled}	☐	
Invoice_City	string(1024)	{res:itgen_fnx_invoice_city}	☐	
Invoice_Comments	string(1024)	{res:itgen_fnx_invoice_comments}	☐	
Invoice_ContractReference	int32	{res:itgen_fnx_invoice_contractreference}	☐	
Invoice_ContributionPercent	double	{res:itgen_fnx_invoice_contributionpercent}	☐	
Invoice_ContributionValue	double	{res:itgen_fnx_invoice_contributionvalue}	☐	
Invoice_CostCenter	string	{res:itgen_fnx_invoice_costcenter}	☐	
Invoice_Country	string	{res:itgen_fnx_invoice_country}	☐	
Invoice_Credit	string	{res:itgen_fnx_invoice_credit}	☐	
Invoice_CreditInvoiceReference	string	{res:itgen_fnx_invoice_creditinvoicereference}	☐	
Invoice_Currency	string	{res:itgen_fnx_invoice_currency}	☐	
Invoice_CurrencyRate	double	{res:itgen_fnx_invoice_currencyrate}	☐	
Invoice_CurrencyUnit	int32	{res:itgen_fnx_invoice_currencyunit}	☐	
Invoice_CustomerName	string	{res:itgen_fnx_invoice_customername}	☐	
Invoice_CustomerNumber	string	{res:itgen_fnx_invoice_customernumber}	☐	
Invoice_DeliveryAddress1	string(1024)	{res:itgen_fnx_invoice_deliveryaddress1}	☐	
Invoice_DeliveryAddress2	string(1024)	{res:itgen_fnx_invoice_deliveryaddress2}	☐	
Invoice_DeliveryCity	string(1024)	{res:itgen_fnx_invoice_deliverycity}	☐	
Invoice_DeliveryCountry	string	{res:itgen_fnx_invoice_deliverycountry}	☐	
Invoice_DeliveryDate	datetime	{res:itgen_fnx_invoice_deliverydate}	☐	
Invoice_DeliveryName	string(1024)	{res:itgen_fnx_invoice_deliveryname}	☐	
Invoice_DeliveryZipCode	string(1024)	{res:itgen_fnx_invoice_deliveryzipcode}	☐	
Invoice_DocumentNumber	string	{res:itgen_fnx_invoice_documentnumber}	☐	
Invoice_DueDate	datetime	{res:itgen_fnx_invoice_duedate}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Invoice_EDIInformation_EDIGlobalLocationNumber	string(13)	{res:itgen_fnx_invoice_ediinformation_edigloballocationnumber}	☐	
Invoice_EDIInformation_EDIGlobalLocationNumberDelivery	string(13)	{res:itgen_fnx_invoice_ediinformation_edigloballocationnumberdelivery}	☐	
Invoice_EDIInformation_EDIInvoiceExtra1	string	{res:itgen_fnx_invoice_ediinformation_ediinvoiceextra1}	☐	
Invoice_EDIInformation_EDIInvoiceExtra2	string	{res:itgen_fnx_invoice_ediinformation_ediinvoiceextra2}	☐	
Invoice_EDIInformation_EDIOurElectronicReference	string	{res:itgen_fnx_invoice_ediinformation_ediourelectronicreference}	☐	
Invoice_EDIInformation_EDIStatus	string	{res:itgen_fnx_invoice_ediinformation_edistatus}	☐	
Invoice_EDIInformation_EDIYourElectronicReference	string	{res:itgen_fnx_invoice_ediinformation_ediyourelectronicreference}	☐	
Invoice_EmailInformation_EmailAddressBCC	string	{res:itgen_fnx_invoice_emailinformation_emailaddressbcc}	☐	
Invoice_EmailInformation_EmailAddressCC	string	{res:itgen_fnx_invoice_emailinformation_emailaddresscc}	☐	
Invoice_EmailInformation_EmailAddressFrom	string	{res:itgen_fnx_invoice_emailinformation_emailaddressfrom}	☐	
Invoice_EmailInformation_EmailAddressTo	string	{res:itgen_fnx_invoice_emailinformation_emailaddressto}	☐	
Invoice_EmailInformation_EmailBody	string(20000)	{res:itgen_fnx_invoice_emailinformation_emailbody}	☐	
Invoice_EmailInformation_EmailSubject	string(100)	{res:itgen_fnx_invoice_emailinformation_emailsubject}	☐	
Invoice_EUQuarterlyReport	boolean	{res:itgen_fnx_invoice_euquarterlyreport}	☐	
Invoice_ExternalInvoiceReference1	string(80)	{res:itgen_fnx_invoice_externalinvoicereference1}	☐	
Invoice_ExternalInvoiceReference2	string(80)	{res:itgen_fnx_invoice_externalinvoicereference2}	☐	
Invoice_FinalPayDate	datetime	{res:itgen_fnx_invoice_finalpaydate}	☐	
Invoice_Freight	double	{res:itgen_fnx_invoice_freight}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Invoice_FreightVAT	double	{res:itgen_fnx_invoice_freightvat}	☐	
Invoice_Gross	double	{res:itgen_fnx_invoice_gross}	☐	
Invoice_HouseWork	boolean	{res:itgen_fnx_invoice_housework}	☐	
Invoice_InvoiceDate	datetime	{res:itgen_fnx_invoice_invoicedate}	☐	
Invoice_InvoicePeriodEnd	datetime	{res:itgen_fnx_invoice_invoiceperiodend}	☐	
Invoice_InvoicePeriodReference	string	{res:itgen_fnx_invoice_invoiceperiodreference}	☐	
Invoice_InvoicePeriodStart	datetime	{res:itgen_fnx_invoice_invoiceperiodstart}	☐	
Invoice_InvoiceType	string	{res:itgen_fnx_invoice_invoicetype}	☐	
Invoice_Language	string	{res:itgen_fnx_invoice_language}	☐	
Invoice_LastRemindDate	datetime	{res:itgen_fnx_invoice_lastreminddate}	☐	
Invoice_Net	double	{res:itgen_fnx_invoice_net}	☐	
Invoice_NotCompleted	boolean	{res:itgen_fnx_invoice_notcompleted}	☐	
Invoice_NoxFinans	boolean	{res:itgen_fnx_invoice_noxfinans}	☐	
Invoice_OCR	string	{res:itgen_fnx_invoice_ocr}	☐	
Invoice_OfferReference	string	{res:itgen_fnx_invoice_offerreference}	☐	
Invoice_OrderReference	string	{res:itgen_fnx_invoice_orderreference}	☐	
Invoice_OrganisationNumber	string	{res:itgen_fnx_invoice_organisationnumber}	☐	
Invoice_OurReference	string(50)	{res:itgen_fnx_invoice_ourreference}	☐	
Invoice_OutboundDate	datetime	{res:itgen_fnx_invoice_outbounddate}	☐	
Invoice_PaymentWay	string	{res:itgen_fnx_invoice_paymentway}	☐	
Invoice_Phone1	string(1024)	{res:itgen_fnx_invoice_phone1}	☐	
Invoice_Phone2	string(1024)	{res:itgen_fnx_invoice_phone2}	☐	
Invoice_PriceList	string	{res:itgen_fnx_invoice_pricelist}	☐	
Invoice_PrintTemplate	string	{res:itgen_fnx_invoice_printtemplate}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Invoice_Project	string	{res:itgen_fnx_invoice_project}	☐	
Invoice_Remarks	string(1024)	{res:itgen_fnx_invoice_remarks}	☐	
Invoice_Reminders	int32	{res:itgen_fnx_invoice_reminders}	☐	
Invoice_RoundOff	double	{res:itgen_fnx_invoice_roundoff}	☐	
Invoice_Sent	boolean	{res:itgen_fnx_invoice_sent}	☐	
Invoice_TaxReduction	int32	{res:itgen_fnx_invoice_taxreduction}	☐	
Invoice_TaxReductionType	string	{res:itgen_fnx_invoice_taxreductiontype}	☐	
Invoice_TermsOfDelivery	string	{res:itgen_fnx_invoice_termsofdelivery}	☐	
Invoice_TermsOfPayment	string	{res:itgen_fnx_invoice_termsofpayment}	☐	
Invoice_TimeBasisReference	int32	{res:itgen_fnx_invoice_timebasisreference}	☐	
Invoice_Total	double	{res:itgen_fnx_invoice_total}	☐	
Invoice_TotalToPay	double	{res:itgen_fnx_invoice_totaltopay}	☐	
Invoice_TotalVAT	double	{res:itgen_fnx_invoice_totalvat}	☐	
Invoice_VATIncluded	boolean	{res:itgen_fnx_invoice_vatincluded}	☐	
Invoice_VoucherNumber	int32	{res:itgen_fnx_invoice_vouchernumber}	☐	
Invoice_VoucherSeries	string	{res:itgen_fnx_invoice_voucherseries}	☐	
Invoice_VoucherYear	int32	{res:itgen_fnx_invoice_voucheryear}	☐	
Invoice_WarehouseReady	boolean	{res:itgen_fnx_invoice_warehouseready}	☐	
Invoice_WayOfDelivery	string	{res:itgen_fnx_invoice_wayofdelivery}	☐	
Invoice_YourOrderNumber	string(75)	{res:itgen_fnx_invoice_yourordernumber}	☐	
Invoice_YourReference	string(50)	{res:itgen_fnx_invoice_yourreference}	☐	
Invoice_ZipCode	string(1024)	{res:itgen_fnx_invoice_zipcode}	☐	

InvoiceEmailByDocumentNumber

Send an invoice as emailYou can use the properties in the EmailInformation to customize the e-mail message on each invoice.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Catalog: Fortnox

Schema: Invoices

Label: {res:itgen_fnx_invoiceemailbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceEmailByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the invoice
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function InvoiceEmailByDocumentNumber are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Invoice__url	string	{res:itgen_fnx_invoice__url}	☐	
Invoice__urlTaxReductionList	string	{res:itgen_fnx_invoice__urltaxreductionlist}	☐	
Invoice_AccountingMethod	string	{res:itgen_fnx_invoice_accountingmethod}	☐	
Invoice_Address1	string	{res:itgen_fnx_invoice_address1}	☐	
Invoice_Address2	string	{res:itgen_fnx_invoice_address2}	☐	
Invoice_AdministrationFee	double	{res:itgen_fnx_invoice_administrationfee}	☐	
Invoice_AdministrationFeeVAT	double	{res:itgen_fnx_invoice_administrationfeevat}	☐	
Invoice_Balance	double	{res:itgen_fnx_invoice_balance}	☐	
Invoice_BasisTaxReduction	double	{res:itgen_fnx_invoice_basistaxreduction}	☐	
Invoice_Booked	boolean	{res:itgen_fnx_invoice_booked}	☐	
Invoice_Cancelled	boolean	{res:itgen_fnx_invoice_cancelled}	☐	
Invoice_City	string(1024)	{res:itgen_fnx_invoice_city}	☐	
Invoice_Comments	string(1024)	{res:itgen_fnx_invoice_comments}	☐	
Invoice_ContractReference	int32	{res:itgen_fnx_invoice_contractreference}	☐	
Invoice_ContributionPercent	double	{res:itgen_fnx_invoice_contributionpercent}	☐	
Invoice_ContributionValue	double	{res:itgen_fnx_invoice_contributionvalue}	☐	
Invoice_CostCenter	string	{res:itgen_fnx_invoice_costcenter}	☐	
Invoice_Country	string	{res:itgen_fnx_invoice_country}	☐	
Invoice_Credit	string	{res:itgen_fnx_invoice_credit}	☐	
Invoice_CreditInvoiceReference	string	{res:itgen_fnx_invoice_creditinvoicereference}	☐	
Invoice_Currency	string	{res:itgen_fnx_invoice_currency}	☐	
Invoice_CurrencyRate	double	{res:itgen_fnx_invoice_currencyrate}	☐	
Invoice_CurrencyUnit	int32	{res:itgen_fnx_invoice_currencyunit}	☐	
Invoice_CustomerName	string	{res:itgen_fnx_invoice_customername}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Invoice_CustomerNumber	string	{res:itgen_fnx_invoice_customernumber}	☐	
Invoice_DeliveryAddress1	string(1024)	{res:itgen_fnx_invoice_deliveryaddress1}	☐	
Invoice_DeliveryAddress2	string(1024)	{res:itgen_fnx_invoice_deliveryaddress2}	☐	
Invoice_DeliveryCity	string(1024)	{res:itgen_fnx_invoice_deliverycity}	☐	
Invoice_DeliveryCountry	string	{res:itgen_fnx_invoice_deliverycountry}	☐	
Invoice_DeliveryDate	datetime	{res:itgen_fnx_invoice_deliverydate}	☐	
Invoice_DeliveryName	string(1024)	{res:itgen_fnx_invoice_deliveryname}	☐	
Invoice_DeliveryZipCode	string(1024)	{res:itgen_fnx_invoice_deliveryzipcode}	☐	
Invoice_DocumentNumber	string	{res:itgen_fnx_invoice_documentnumber}	☐	
Invoice_DueDate	datetime	{res:itgen_fnx_invoice_duedate}	☐	
Invoice_EDIInformation_EDIGlobalLocationNumber	string(13)	{res:itgen_fnx_invoice_ediinformation_edigloballocationnumber}	☐	
Invoice_EDIInformation_EDIGlobalLocationNumberDelivery	string(13)	{res:itgen_fnx_invoice_ediinformation_edigloballocationnumberdelivery}	☐	
Invoice_EDIInformation_EDIInvoiceExtra1	string	{res:itgen_fnx_invoice_ediinformation_ediinvoiceextra1}	☐	
Invoice_EDIInformation_EDIInvoiceExtra2	string	{res:itgen_fnx_invoice_ediinformation_ediinvoiceextra2}	☐	
Invoice_EDIInformation_EDIOurElectronicReference	string	{res:itgen_fnx_invoice_ediinformation_ediourelectronicreference}	☐	
Invoice_EDIInformation_EDIStatus	string	{res:itgen_fnx_invoice_ediinformation_edistatus}	☐	
Invoice_EDIInformation_EDIYourElectronicReference	string	{res:itgen_fnx_invoice_ediinformation_ediyourelectronicreference}	☐	
Invoice_EmailInformation_EmailAddressBCC	string	{res:itgen_fnx_invoice_emailinformation_emailaddressbcc}	☐	
Invoice_EmailInformation_EmailAddressCC	string	{res:itgen_fnx_invoice_emailinformation_emailaddresscc}	☐	
Invoice_EmailInformation_EmailAddressFrom	string	{res:itgen_fnx_invoice_emailinformation_emailaddressfrom}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Invoice_EmailInformation_EmailAddressTo	string	{res:itgen_fnx_invoice_emailinformation_emailaddressto}	☐	
Invoice_EmailInformation_EmailBody	string(20000)	{res:itgen_fnx_invoice_emailinformation_emailbody}	☐	
Invoice_EmailInformation_EmailSubject	string(100)	{res:itgen_fnx_invoice_emailinformation_emailsubject}	☐	
Invoice_EUQuarterlyReport	boolean	{res:itgen_fnx_invoice_euquarterlyreport}	☐	
Invoice_ExternalInvoiceReference1	string(80)	{res:itgen_fnx_invoice_externalinvoicereference1}	☐	
Invoice_ExternalInvoiceReference2	string(80)	{res:itgen_fnx_invoice_externalinvoicereference2}	☐	
Invoice_FinalPayDate	datetime	{res:itgen_fnx_invoice_finalpaydate}	☐	
Invoice_Freight	double	{res:itgen_fnx_invoice_freight}	☐	
Invoice_FreightVAT	double	{res:itgen_fnx_invoice_freightvat}	☐	
Invoice_Gross	double	{res:itgen_fnx_invoice_gross}	☐	
Invoice_HouseWork	boolean	{res:itgen_fnx_invoice_housework}	☐	
Invoice_InvoiceDate	datetime	{res:itgen_fnx_invoice_invoicedate}	☐	
Invoice_InvoicePeriodEnd	datetime	{res:itgen_fnx_invoice_invoiceperiodend}	☐	
Invoice_InvoicePeriodReference	string	{res:itgen_fnx_invoice_invoiceperiodreference}	☐	
Invoice_InvoicePeriodStart	datetime	{res:itgen_fnx_invoice_invoiceperiodstart}	☐	
Invoice_InvoiceType	string	{res:itgen_fnx_invoice_invoicetype}	☐	
Invoice_Language	string	{res:itgen_fnx_invoice_language}	☐	
Invoice_LastRemindDate	datetime	{res:itgen_fnx_invoice_lastreminddate}	☐	
Invoice_Net	double	{res:itgen_fnx_invoice_net}	☐	
Invoice_NotCompleted	boolean	{res:itgen_fnx_invoice_notcompleted}	☐	
Invoice_NoxFinans	boolean	{res:itgen_fnx_invoice_noxfinans}	☐	
Invoice_OCR	string	{res:itgen_fnx_invoice_ocr}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Invoice_OfferReference	string	{res:itgen_fnx_invoice_offerreference}	☐	
Invoice_OrderReference	string	{res:itgen_fnx_invoice_orderreference}	☐	
Invoice_OrganisationNumber	string	{res:itgen_fnx_invoice_organisationnumber}	☐	
Invoice_OurReference	string(50)	{res:itgen_fnx_invoice_ourreference}	☐	
Invoice_OutboundDate	datetime	{res:itgen_fnx_invoice_outbounddate}	☐	
Invoice_PaymentWay	string	{res:itgen_fnx_invoice_paymentway}	☐	
Invoice_Phone1	string(1024)	{res:itgen_fnx_invoice_phone1}	☐	
Invoice_Phone2	string(1024)	{res:itgen_fnx_invoice_phone2}	☐	
Invoice_PriceList	string	{res:itgen_fnx_invoice_pricelist}	☐	
Invoice_PrintTemplate	string	{res:itgen_fnx_invoice_printtemplate}	☐	
Invoice_Project	string	{res:itgen_fnx_invoice_project}	☐	
Invoice_Remarks	string(1024)	{res:itgen_fnx_invoice_remarks}	☐	
Invoice_Reminders	int32	{res:itgen_fnx_invoice_reminders}	☐	
Invoice_RoundOff	double	{res:itgen_fnx_invoice_roundoff}	☐	
Invoice_Sent	boolean	{res:itgen_fnx_invoice_sent}	☐	
Invoice_TaxReduction	int32	{res:itgen_fnx_invoice_taxreduction}	☐	
Invoice_TaxReductionType	string	{res:itgen_fnx_invoice_taxreductiontype}	☐	
Invoice_TermsOfDelivery	string	{res:itgen_fnx_invoice_termsofdelivery}	☐	
Invoice_TermsOfPayment	string	{res:itgen_fnx_invoice_termsofpayment}	☐	
Invoice_TimeBasisReference	int32	{res:itgen_fnx_invoice_timebasisreference}	☐	
Invoice_Total	double	{res:itgen_fnx_invoice_total}	☐	
Invoice_TotalToPay	double	{res:itgen_fnx_invoice_totaltopay}	☐	
Invoice_TotalVAT	double	{res:itgen_fnx_invoice_totalvat}	☐	
Invoice_VATIncluded	boolean	{res:itgen_fnx_invoice_vatincluded}	☐	
Invoice_VoucherNumber	int32	{res:itgen_fnx_invoice_vouchernumber}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Invoice_VoucherSeries	string	{res:itgen_fnx_invoice_voucherseries}	☐	
Invoice_VoucherYear	int32	{res:itgen_fnx_invoice_voucheryear}	☐	
Invoice_WarehouseReady	boolean	{res:itgen_fnx_invoice_warehouseready}	☐	
Invoice_WayOfDelivery	string	{res:itgen_fnx_invoice_wayofdelivery}	☐	
Invoice_YourOrderNumber	string(75)	{res:itgen_fnx_invoice_yourordernumber}	☐	
Invoice_YourReference	string(50)	{res:itgen_fnx_invoice_yourreference}	☐	
Invoice_ZipCode	string(1024)	{res:itgen_fnx_invoice_zipcode}	☐	

InvoiceEmailInvoiceLabelsByDocumentNumber

Send an invoice as emailYou can use the properties in the EmailInformation to customize the e-mail message on each invoice.

Catalog: Fortnox

Schema: Invoices

Label: {res:itgen_fnx_invoiceemailinvoicelabelsbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceEmailInvoiceLabelsByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the invoice
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function InvoiceEmailInvoiceLabelsByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AccountingMethod	string	{res:itgen_fnx_accountingmethod}	☐	
Address1	string	{res:itgen_fnx_address1}	☐	
Address2	string	{res:itgen_fnx_address2}	☐	
AdministrationFee	double	{res:itgen_fnx_administrationfee}	☐	
AdministrationFeeVAT	double	{res:itgen_fnx_administrationfeevat}	☐	
Balance	double	{res:itgen_fnx_balance}	☐	
BasisTaxReduction	double	{res:itgen_fnx_basistaxreduction}	☐	
Booked	boolean	{res:itgen_fnx_booked}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
c_urlTaxReductionList	string	{res:itgen_fnx_c_urltaxreductionlist}	☐	
Cancelled	boolean	{res:itgen_fnx_cancelled}	☐	
City	string(1024)	{res:itgen_fnx_city}	☐	
Comments	string(1024)	{res:itgen_fnx_comments}	☐	
ContractReference	int32	{res:itgen_fnx_contractreference}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ContributionPercent	double	{res:itgen_fnx_contributionpercent}	☐	
ContributionValue	double	{res:itgen_fnx_contributionvalue}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	
Credit	string	{res:itgen_fnx_credit}	☐	
CreditInvoiceReference	string	{res:itgen_fnx_creditinvoicereference}	☐	
Currency	string	{res:itgen_fnx_currency}	☐	
CurrencyRate	double	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	int32	{res:itgen_fnx_currencyunit}	☐	
CustomerName	string	{res:itgen_fnx_customername}	☐	
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
DeliveryAddress1	string(1024)	{res:itgen_fnx_deliveryaddress1}	☐	
DeliveryAddress2	string(1024)	{res:itgen_fnx_deliveryaddress2}	☐	
DeliveryCity	string(1024)	{res:itgen_fnx_deliverycity}	☐	
DeliveryCountry	string	{res:itgen_fnx_deliverycountry}	☐	
DeliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	
DeliveryName	string(1024)	{res:itgen_fnx_deliveryname}	☐	
DeliveryZipCode	string(1024)	{res:itgen_fnx_deliveryzipcode}	☐	
DocumentNumber	string	{res:itgen_fnx_documentnumber}	☐	
DueDate	datetime	{res:itgen_fnx_duedate}	☐	
EDIInformation_EDIGlobalLocationNumber	string(13)	{res:itgen_fnx_ediinformation_ediglobalocationnumber}	☐	
EDIInformation_EDIGlobalLocationNumberDelivery	string(13)	{res:itgen_fnx_ediinformation_ediglobalocationnumberdelivery}	☐	
EDIInformation_EDIInvoiceExtra1	string	{res:itgen_fnx_ediinformation_ediinvoiceextra1}	☐	
EDIInformation_EDIInvoiceExtra2	string	{res:itgen_fnx_ediinformation_ediinvoiceextra2}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
EDIInformation_EDIOurElectronicReference	string	{res:itgen_fnx_ediinformation_ediourelectronicreference}	☐	
EDIInformation_EDIStatus	string	{res:itgen_fnx_ediinformation_edistatus}	☐	
EDIInformation_EDIYourElectronicReference	string	{res:itgen_fnx_ediinformation_ediyourelectronicreference}	☐	
EmailInformation_EmailAddressBCC	string	{res:itgen_fnx_emailinformation_emailaddressbcc}	☐	
EmailInformation_EmailAddressCC	string	{res:itgen_fnx_emailinformation_emailaddresscc}	☐	
EmailInformation_EmailAddressFrom	string	{res:itgen_fnx_emailinformation_emailaddressfrom}	☐	
EmailInformation_EmailAddressTo	string	{res:itgen_fnx_emailinformation_emailaddressto}	☐	
EmailInformation_EmailBody	string(20000)	{res:itgen_fnx_emailinformation_emailbody}	☐	
EmailInformation_EmailSubject	string(100)	{res:itgen_fnx_emailinformation_emailsubject}	☐	
EUQuarterlyReport	boolean	{res:itgen_fnx_euquarterlyreport}	☐	
ExternalInvoiceReference1	string(80)	{res:itgen_fnx_externalinvoicereference1}	☐	
ExternalInvoiceReference2	string(80)	{res:itgen_fnx_externalinvoicereference2}	☐	
FinalPayDate	datetime	{res:itgen_fnx_finalpaydate}	☐	
Freight	double	{res:itgen_fnx_freight}	☐	
FreightVAT	double	{res:itgen_fnx_freightvat}	☐	
Gross	double	{res:itgen_fnx_gross}	☐	
HouseWork	boolean	{res:itgen_fnx_housework}	☐	
Id	int32	{res:itgen_fnx_id}	☐	
InvoiceDate	datetime	{res:itgen_fnx_invoicedate}	☐	
InvoicePeriodEnd	datetime	{res:itgen_fnx_invoiceperiodend}	☐	
InvoicePeriodReference	string	{res:itgen_fnx_invoiceperiodreference}	☐	
InvoicePeriodStart	datetime	{res:itgen_fnx_invoiceperiodstart}	☐	
InvoiceType	string	{res:itgen_fnx_invoicetype}	☐	
Language	string	{res:itgen_fnx_language}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
LastRemindDate	datetime	{res:itgen_fnx_lastreminddate}	☐	
Net	double	{res:itgen_fnx_net}	☐	
NotCompleted	boolean	{res:itgen_fnx_notcompleted}	☐	
NoxFinans	boolean	{res:itgen_fnx_noxfinans}	☐	
OCR	string	{res:itgen_fnx_ocr}	☐	
OfferReference	string	{res:itgen_fnx_offerreference}	☐	
OrderReference	string	{res:itgen_fnx_orderreference}	☐	
OrganisationNumber	string	{res:itgen_fnx_organisationnumber}	☐	
OurReference	string(50)	{res:itgen_fnx_ourreference}	☐	
OutboundDate	datetime	{res:itgen_fnx_outbounddate}	☐	
PaymentWay	string	{res:itgen_fnx_paymentway}	☐	
Phone1	string(1024)	{res:itgen_fnx_phone1}	☐	
Phone2	string(1024)	{res:itgen_fnx_phone2}	☐	
PriceList	string	{res:itgen_fnx_pricelist}	☐	
PrintTemplate	string	{res:itgen_fnx_printtemplate}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Remarks	string(1024)	{res:itgen_fnx_remarks}	☐	
Reminders	int32	{res:itgen_fnx_reminders}	☐	
RoundOff	double	{res:itgen_fnx_roundoff}	☐	
Sent	boolean	{res:itgen_fnx_sent}	☐	
TaxReduction	int32	{res:itgen_fnx_taxreduction}	☐	
TaxReductionType	string	{res:itgen_fnx_taxreductiontype}	☐	
TermsOfDelivery	string	{res:itgen_fnx_termsofdelivery}	☐	
TermsOfPayment	string	{res:itgen_fnx_termsofpayment}	☐	
TimeBasisReference	int32	{res:itgen_fnx_timebasisreference}	☐	
Total	double	{res:itgen_fnx_total}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TotalToPay	double	{res:itgen_fnx_totaltopay}	☐	
TotalVAT	double	{res:itgen_fnx_totalvat}	☐	
VATIncluded	boolean	{res:itgen_fnx_vatincluded}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
VoucherYear	int32	{res:itgen_fnx_voucheryear}	☐	
WarehouseReady	boolean	{res:itgen_fnx_warehouseready}	☐	
WayOfDelivery	string	{res:itgen_fnx_wayofdelivery}	☐	
YourOrderNumber	string(75)	{res:itgen_fnx_yourordernumber}	☐	
YourReference	string(50)	{res:itgen_fnx_yourreference}	☐	
ZipCode	string(1024)	{res:itgen_fnx_zipcode}	☐	

InvoiceEmailInvoiceLinesByDocumentNumber

Send an invoice as emailYou can use the properties in the EmailInformation to customize the e-mail message on each invoice.

Catalog: Fortnox

Schema: Invoices

Label: {res:itgen_fnx_invoiceemailinvoicelinesbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceEmailInvoiceLinesByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the invoice
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function InvoiceEmailInvoiceLinesByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AccountingMethod	string	{res:itgen_fnx_accountingmethod}	☐	
AccountNumber	int32	{res:itgen_fnx_accountnumber}	☐	
Address1	string	{res:itgen_fnx_address1}	☐	
Address2	string	{res:itgen_fnx_address2}	☐	
AdministrationFee	double	{res:itgen_fnx_administrationfee}	☐	
AdministrationFeeVAT	double	{res:itgen_fnx_administrationfeevat}	☐	
ArticleNumber	string	{res:itgen_fnx_articlenumber}	☐	
Balance	double	{res:itgen_fnx_balance}	☐	
BasisTaxReduction	double	{res:itgen_fnx_basistaxreduction}	☐	
Booked	boolean	{res:itgen_fnx_booked}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
c_urlTaxReductionList	string	{res:itgen_fnx_c_urltaxreductionlist}	☐	
Cancelled	boolean	{res:itgen_fnx_cancelled}	☐	
City	string(1024)	{res:itgen_fnx_city}	☐	
Comments	string(1024)	{res:itgen_fnx_comments}	☐	
ContractReference	int32	{res:itgen_fnx_contractreference}	☐	
ContributionPercent_2	double	{res:itgen_fnx_contributionpercent_2}	☐	
ContributionPercent	string	{res:itgen_fnx_contributionpercent}	☐	
ContributionValue_2	double	{res:itgen_fnx_contributionvalue_2}	☐	
ContributionValue	string	{res:itgen_fnx_contributionvalue}	☐	
CostCenter_2	string	{res:itgen_fnx_costcenter_2}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	
Credit	string	{res:itgen_fnx_credit}	☐	
CreditInvoiceReference	string	{res:itgen_fnx_creditinvoicereference}	☐	
Currency	string	{res:itgen_fnx_currency}	☐	
CurrencyRate	double	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	int32	{res:itgen_fnx_currencyunit}	☐	
CustomerName	string	{res:itgen_fnx_customername}	☐	
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
DeliveredQuantity	string	{res:itgen_fnx_deliveredquantity}	☐	
DeliveryAddress1	string(1024)	{res:itgen_fnx_deliveryaddress1}	☐	
DeliveryAddress2	string(1024)	{res:itgen_fnx_deliveryaddress2}	☐	
DeliveryCity	string(1024)	{res:itgen_fnx_deliverycity}	☐	
DeliveryCountry	string	{res:itgen_fnx_deliverycountry}	☐	
DeliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
DeliveryName	string(1024)	{res:itgen_fnx_deliveryname}	☐	
DeliveryZipCode	string(1024)	{res:itgen_fnx_deliveryzipcode}	☐	
Description	string(255)	{res:itgen_fnx_description}	☐	
Discount	double	{res:itgen_fnx_discount}	☐	
DiscountType	string	{res:itgen_fnx_discounttype}	☐	
DocumentNumber	string	{res:itgen_fnx_documentnumber}	☐	
DueDate	datetime	{res:itgen_fnx_duedate}	☐	
EDIInformation_EDIGlobalLocationNumber	string(13)	{res:itgen_fnx_ediinformation_edigloballocationnumber}	☐	
EDIInformation_EDIGlobalLocationNumberDelivery	string(13)	{res:itgen_fnx_ediinformation_edigloballocationnumberdelivery}	☐	
EDIInformation_EDIInvoiceExtra1	string	{res:itgen_fnx_ediinformation_ediinvoiceextra1}	☐	
EDIInformation_EDIInvoiceExtra2	string	{res:itgen_fnx_ediinformation_ediinvoiceextra2}	☐	
EDIInformation_EDIOurElectronicReference	string	{res:itgen_fnx_ediinformation_ediourelectronicreference}	☐	
EDIInformation_EDIStatus	string	{res:itgen_fnx_ediinformation_edistatus}	☐	
EDIInformation_EDIYourElectronicReference	string	{res:itgen_fnx_ediinformation_ediyourelectronicreference}	☐	
EmailInformation_EmailAddressBCC	string	{res:itgen_fnx_emailinformation_emailaddressbcc}	☐	
EmailInformation_EmailAddressCC	string	{res:itgen_fnx_emailinformation_emailaddresscc}	☐	
EmailInformation_EmailAddressFrom	string	{res:itgen_fnx_emailinformation_emailaddressfrom}	☐	
EmailInformation_EmailAddressTo	string	{res:itgen_fnx_emailinformation_emailaddressto}	☐	
EmailInformation_EmailBody	string(20000)	{res:itgen_fnx_emailinformation_emailbody}	☐	
EmailInformation_EmailSubject	string(100)	{res:itgen_fnx_emailinformation_emailsubject}	☐	
EUQuarterlyReport	boolean	{res:itgen_fnx_euquarterlyreport}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ExternalInvoiceReference1	string(80)	{res:itgen_fnx_externalinvoicereference1}	☐	
ExternalInvoiceReference2	string(80)	{res:itgen_fnx_externalinvoicereference2}	☐	
FinalPayDate	datetime	{res:itgen_fnx_finalpaydate}	☐	
Freight	double	{res:itgen_fnx_freight}	☐	
FreightVAT	double	{res:itgen_fnx_freightvat}	☐	
Gross	double	{res:itgen_fnx_gross}	☐	
HouseWork_2	boolean	{res:itgen_fnx_housework_2}	☐	
HouseWork	boolean	{res:itgen_fnx_housework}	☐	
HouseWorkHoursToReport	int32	{res:itgen_fnx_houseworkhourstoreport}	☐	
HouseWorkType	string	{res:itgen_fnx_houseworktype}	☐	
InvoiceDate	datetime	{res:itgen_fnx_invoicedate}	☐	
InvoicePeriodEnd	datetime	{res:itgen_fnx_invoiceperiodend}	☐	
InvoicePeriodReference	string	{res:itgen_fnx_invoiceperiodreference}	☐	
InvoicePeriodStart	datetime	{res:itgen_fnx_invoiceperiodstart}	☐	
InvoiceType	string	{res:itgen_fnx_invoicetype}	☐	
Language	string	{res:itgen_fnx_language}	☐	
LastRemindDate	datetime	{res:itgen_fnx_lastreminddate}	☐	
Net	double	{res:itgen_fnx_net}	☐	
NotCompleted	boolean	{res:itgen_fnx_notcompleted}	☐	
NoxFinans	boolean	{res:itgen_fnx_noxfinans}	☐	
OCR	string	{res:itgen_fnx_ocr}	☐	
OfferReference	string	{res:itgen_fnx_offerreference}	☐	
OrderReference	string	{res:itgen_fnx_orderreference}	☐	
OrganisationNumber	string	{res:itgen_fnx_organisationnumber}	☐	
OurReference	string(50)	{res:itgen_fnx_ourreference}	☐	
OutboundDate	datetime	{res:itgen_fnx_outbounddate}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
PaymentWay	string	{res:itgen_fnx_paymentway}	☐	
Phone1	string(1024)	{res:itgen_fnx_phone1}	☐	
Phone2	string(1024)	{res:itgen_fnx_phone2}	☐	
Price	double	{res:itgen_fnx_price}	☐	
PriceExcludingVAT	double	{res:itgen_fnx_priceexcludingvat}	☐	
PriceList	string	{res:itgen_fnx_pricelist}	☐	
PrintTemplate	string	{res:itgen_fnx_printtemplate}	☐	
Project_2	string	{res:itgen_fnx_project_2}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Remarks	string(1024)	{res:itgen_fnx_remarks}	☐	
Reminders	int32	{res:itgen_fnx_reminders}	☐	
RoundOff	double	{res:itgen_fnx_roundoff}	☐	
Row Id	int32	{res:itgen_fnx_row id}	☐	
Sent	boolean	{res:itgen_fnx_sent}	☐	
StockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
TaxReduction	int32	{res:itgen_fnx_taxreduction}	☐	
TaxReductionType	string	{res:itgen_fnx_taxreductiontype}	☐	
TermsOfDelivery	string	{res:itgen_fnx_termsofdelivery}	☐	
TermsOfPayment	string	{res:itgen_fnx_termsofpayment}	☐	
TimeBasisReference	int32	{res:itgen_fnx_timebasisreference}	☐	
Total_2	double	{res:itgen_fnx_total_2}	☐	
Total	double	{res:itgen_fnx_total}	☐	
TotalExcludingVAT	double	{res:itgen_fnx_totalexcludingvat}	☐	
TotalToPay	double	{res:itgen_fnx_totaltopay}	☐	
TotalVAT	double	{res:itgen_fnx_totalvat}	☐	
Unit	string	{res:itgen_fnx_unit}	☐	
VAT	int32	{res:itgen_fnx_vat}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
VATCode	string	{res:itgen_fnx_vatcode}	☐	
VATIncluded	boolean	{res:itgen_fnx_vatincluded}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
VoucherYear	int32	{res:itgen_fnx_voucheryear}	☐	
WarehouseReady	boolean	{res:itgen_fnx_warehouseready}	☐	
WayOfDelivery	string	{res:itgen_fnx_wayofdelivery}	☐	
YourOrderNumber	string(75)	{res:itgen_fnx_yourordernumber}	☐	
YourReference	string(50)	{res:itgen_fnx_yourreference}	☐	
ZipCode	string(1024)	{res:itgen_fnx_zipcode}	☐	

InvoiceLabelsByDocumentNumber

Retrieve a single invoice

Catalog: Fortnox

Schema: Invoices

Label: {res:itgen_fnx_invoicelabelsbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceLabelsByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the invoice
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function InvoiceLabelsByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AccountingMethod	string	{res:itgen_fnx_accountingmethod}	☐	
Address1	string	{res:itgen_fnx_address1}	☐	
Address2	string	{res:itgen_fnx_address2}	☐	
AdministrationFee	double	{res:itgen_fnx_administrationfee}	☐	
AdministrationFeeVAT	double	{res:itgen_fnx_administrationfeevat}	☐	
Balance	double	{res:itgen_fnx_balance}	☐	
BasisTaxReduction	double	{res:itgen_fnx_basistaxreduction}	☐	
Booked	boolean	{res:itgen_fnx_booked}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
c_urlTaxReductionList	string	{res:itgen_fnx_c_urltaxreductionlist}	☐	
Cancelled	boolean	{res:itgen_fnx_cancelled}	☐	
City	string(1024)	{res:itgen_fnx_city}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Comments	string(1024)	{res:itgen_fnx_comments}	☐	
ContractReference	int32	{res:itgen_fnx_contractreference}	☐	
ContributionPercent	double	{res:itgen_fnx_contributionpercent}	☐	
ContributionValue	double	{res:itgen_fnx_contributionvalue}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	
Credit	string	{res:itgen_fnx_credit}	☐	
CreditInvoiceReference	string	{res:itgen_fnx_creditinvoicereference}	☐	
Currency	string	{res:itgen_fnx_currency}	☐	
CurrencyRate	double	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	int32	{res:itgen_fnx_currencyunit}	☐	
CustomerName	string	{res:itgen_fnx_customername}	☐	
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
DeliveryAddress1	string(1024)	{res:itgen_fnx_deliveryaddress1}	☐	
DeliveryAddress2	string(1024)	{res:itgen_fnx_deliveryaddress2}	☐	
DeliveryCity	string(1024)	{res:itgen_fnx_deliverycity}	☐	
DeliveryCountry	string	{res:itgen_fnx_deliverycountry}	☐	
DeliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	
DeliveryName	string(1024)	{res:itgen_fnx_deliveryname}	☐	
DeliveryZipCode	string(1024)	{res:itgen_fnx_deliveryzipcode}	☐	
DocumentNumber	string	{res:itgen_fnx_documentnumber}	☐	
DueDate	datetime	{res:itgen_fnx_duedate}	☐	
EDIInformation_EDIGlobalLocationNumber	string(13)	{res:itgen_fnx_ediinformation_ediglobalocationnumber}	☐	
EDIInformation_EDIGlobalLocationNumberDelivery	string(13)	{res:itgen_fnx_ediinformation_ediglobalocationnumberdelivery}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
EDIInformation_EDIInvoiceExtra1	string	{res:itgen_fnx_ediinformation_ediinvoiceextra1}	☐	
EDIInformation_EDIInvoiceExtra2	string	{res:itgen_fnx_ediinformation_ediinvoiceextra2}	☐	
EDIInformation_EDIOurElectronicReference	string	{res:itgen_fnx_ediinformation_ediourelectronicreference}	☐	
EDIInformation_EDIStatus	string	{res:itgen_fnx_ediinformation_edistatus}	☐	
EDIInformation_EDIYourElectronicReference	string	{res:itgen_fnx_ediinformation_ediyourelectronicreference}	☐	
EmailInformation_EmailAddressBCC	string	{res:itgen_fnx_emailinformation_emailaddressbcc}	☐	
EmailInformation_EmailAddressCC	string	{res:itgen_fnx_emailinformation_emailaddresscc}	☐	
EmailInformation_EmailAddressFrom	string	{res:itgen_fnx_emailinformation_emailaddressfrom}	☐	
EmailInformation_EmailAddressTo	string	{res:itgen_fnx_emailinformation_emailaddressto}	☐	
EmailInformation_EmailBody	string(20000)	{res:itgen_fnx_emailinformation_emailbody}	☐	
EmailInformation_EmailSubject	string(100)	{res:itgen_fnx_emailinformation_emailsubject}	☐	
EUQuarterlyReport	boolean	{res:itgen_fnx_euquarterlyreport}	☐	
ExternalInvoiceReference1	string(80)	{res:itgen_fnx_externalinvoicereference1}	☐	
ExternalInvoiceReference2	string(80)	{res:itgen_fnx_externalinvoicereference2}	☐	
FinalPayDate	datetime	{res:itgen_fnx_finalpaydate}	☐	
Freight	double	{res:itgen_fnx_freight}	☐	
FreightVAT	double	{res:itgen_fnx_freightvat}	☐	
Gross	double	{res:itgen_fnx_gross}	☐	
HouseWork	boolean	{res:itgen_fnx_housework}	☐	
Id	int32	{res:itgen_fnx_id}	☐	
InvoiceDate	datetime	{res:itgen_fnx_invoicedate}	☐	
InvoicePeriodEnd	datetime	{res:itgen_fnx_invoiceperiodend}	☐	
InvoicePeriodReference	string	{res:itgen_fnx_invoiceperiodreference}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
InvoicePeriodStart	datetime	{res:itgen_fnx_invoiceperiodstart}	☐	
InvoiceType	string	{res:itgen_fnx_invoicetype}	☐	
Language	string	{res:itgen_fnx_language}	☐	
LastRemindDate	datetime	{res:itgen_fnx_lastreminddate}	☐	
Net	double	{res:itgen_fnx_net}	☐	
NotCompleted	boolean	{res:itgen_fnx_notcompleted}	☐	
NoxFinans	boolean	{res:itgen_fnx_noxfinans}	☐	
OCR	string	{res:itgen_fnx_ocr}	☐	
OfferReference	string	{res:itgen_fnx_offerreference}	☐	
OrderReference	string	{res:itgen_fnx_orderreference}	☐	
OrganisationNumber	string	{res:itgen_fnx_organisationnumber}	☐	
OurReference	string(50)	{res:itgen_fnx_ourreference}	☐	
OutboundDate	datetime	{res:itgen_fnx_outbounddate}	☐	
PaymentWay	string	{res:itgen_fnx_paymentway}	☐	
Phone1	string(1024)	{res:itgen_fnx_phone1}	☐	
Phone2	string(1024)	{res:itgen_fnx_phone2}	☐	
PriceList	string	{res:itgen_fnx_pricelist}	☐	
PrintTemplate	string	{res:itgen_fnx_printtemplate}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Remarks	string(1024)	{res:itgen_fnx_remarks}	☐	
Reminders	int32	{res:itgen_fnx_reminders}	☐	
RoundOff	double	{res:itgen_fnx_roundoff}	☐	
Sent	boolean	{res:itgen_fnx_sent}	☐	
TaxReduction	int32	{res:itgen_fnx_taxreduction}	☐	
TaxReductionType	string	{res:itgen_fnx_taxreductiontype}	☐	
TermsOfDelivery	string	{res:itgen_fnx_termsofdelivery}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TermsOfPayment	string	{res:itgen_fnx_termsofpayment}	☐	
TimeBasisReference	int32	{res:itgen_fnx_timebasisreference}	☐	
Total	double	{res:itgen_fnx_total}	☐	
TotalToPay	double	{res:itgen_fnx_totaltopay}	☐	
TotalVAT	double	{res:itgen_fnx_totalvat}	☐	
VATIncluded	boolean	{res:itgen_fnx_vatincluded}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
VoucherYear	int32	{res:itgen_fnx_voucheryear}	☐	
WarehouseReady	boolean	{res:itgen_fnx_warehouseready}	☐	
WayOfDelivery	string	{res:itgen_fnx_wayofdelivery}	☐	
YourOrderNumber	string(75)	{res:itgen_fnx_yourordernumber}	☐	
YourReference	string(50)	{res:itgen_fnx_yourreference}	☐	
ZipCode	string(1024)	{res:itgen_fnx_zipcode}	☐	

InvoiceLinesByDocumentNumber

Retrieve a single invoice

Catalog: Fortnox

Schema: Invoices

Label: {res:itgen_fnx_invoicelinesbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceLinesByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the invoice
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function InvoiceLinesByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AccountingMethod	string	{res:itgen_fnx_accountingmethod}	☐	
AccountNumber	int32	{res:itgen_fnx_accountnumber}	☐	
Address1	string	{res:itgen_fnx_address1}	☐	
Address2	string	{res:itgen_fnx_address2}	☐	
AdministrationFee	double	{res:itgen_fnx_administrationfee}	☐	
AdministrationFeeVAT	double	{res:itgen_fnx_administrationfeevat}	☐	
ArticleNumber	string	{res:itgen_fnx_articlenumber}	☐	
Balance	double	{res:itgen_fnx_balance}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
BasisTaxReduction	double	{res:itgen_fnx_basistaxreduction}	☐	
Booked	boolean	{res:itgen_fnx_booked}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
c_urlTaxReductionList	string	{res:itgen_fnx_c_urltaxreductionlist}	☐	
Cancelled	boolean	{res:itgen_fnx_cancelled}	☐	
City	string(1024)	{res:itgen_fnx_city}	☐	
Comments	string(1024)	{res:itgen_fnx_comments}	☐	
ContractReference	int32	{res:itgen_fnx_contractreference}	☐	
ContributionPercent_2	double	{res:itgen_fnx_contributionpercent_2}	☐	
ContributionPercent	string	{res:itgen_fnx_contributionpercent}	☐	
ContributionValue_2	double	{res:itgen_fnx_contributionvalue_2}	☐	
ContributionValue	string	{res:itgen_fnx_contributionvalue}	☐	
CostCenter_2	string	{res:itgen_fnx_costcenter_2}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	
Credit	string	{res:itgen_fnx_credit}	☐	
CreditInvoiceReference	string	{res:itgen_fnx_creditinvoicereference}	☐	
Currency	string	{res:itgen_fnx_currency}	☐	
CurrencyRate	double	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	int32	{res:itgen_fnx_currencyunit}	☐	
CustomerName	string	{res:itgen_fnx_customername}	☐	
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
DeliveredQuantity	string	{res:itgen_fnx_deliveredquantity}	☐	
DeliveryAddress1	string(1024)	{res:itgen_fnx_deliveryaddress1}	☐	
DeliveryAddress2	string(1024)	{res:itgen_fnx_deliveryaddress2}	☐	
DeliveryCity	string(1024)	{res:itgen_fnx_deliverycity}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
DeliveryCountry	string	{res:itgen_fnx_deliverycountry}	☐	
DeliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	
DeliveryName	string(1024)	{res:itgen_fnx_deliveryname}	☐	
DeliveryZipCode	string(1024)	{res:itgen_fnx_deliveryzipcode}	☐	
Description	string(255)	{res:itgen_fnx_description}	☐	
Discount	double	{res:itgen_fnx_discount}	☐	
DiscountType	string	{res:itgen_fnx_discounttype}	☐	
DocumentNumber	string	{res:itgen_fnx_documentnumber}	☐	
DueDate	datetime	{res:itgen_fnx_duedate}	☐	
EDIInformation_EDIGlobalLocationNumber	string(13)	{res:itgen_fnx_ediinformation_ediglobalocationnumber}	☐	
EDIInformation_EDIGlobalLocationNumberDelivery	string(13)	{res:itgen_fnx_ediinformation_ediglobalocationnumberdelivery}	☐	
EDIInformation_EDIInvoiceExtra1	string	{res:itgen_fnx_ediinformation_ediinvoiceextra1}	☐	
EDIInformation_EDIInvoiceExtra2	string	{res:itgen_fnx_ediinformation_ediinvoiceextra2}	☐	
EDIInformation_EDIOurElectronicReference	string	{res:itgen_fnx_ediinformation_ediourelectronicreference}	☐	
EDIInformation_EDIStatus	string	{res:itgen_fnx_ediinformation_edistatus}	☐	
EDIInformation_EDIYourElectronicReference	string	{res:itgen_fnx_ediinformation_ediyourelectronicreference}	☐	
EmailInformation_EmailAddressBCC	string	{res:itgen_fnx_emailinformation_emailaddressbcc}	☐	
EmailInformation_EmailAddressCC	string	{res:itgen_fnx_emailinformation_emailaddresscc}	☐	
EmailInformation_EmailAddressFrom	string	{res:itgen_fnx_emailinformation_emailaddressfrom}	☐	
EmailInformation_EmailAddressTo	string	{res:itgen_fnx_emailinformation_emailaddressto}	☐	
EmailInformation_EmailBody	string(20000)	{res:itgen_fnx_emailinformation_emailbody}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
EmailInformation_EmailSubject	string(100)	{res:itgen_fnx_emailinformation_emailsubject}	☐	
EUQuarterlyReport	boolean	{res:itgen_fnx_euquarterlyreport}	☐	
ExternalInvoiceReference1	string(80)	{res:itgen_fnx_externalinvoicereference1}	☐	
ExternalInvoiceReference2	string(80)	{res:itgen_fnx_externalinvoicereference2}	☐	
FinalPayDate	datetime	{res:itgen_fnx_finalpaydate}	☐	
Freight	double	{res:itgen_fnx_freight}	☐	
FreightVAT	double	{res:itgen_fnx_freightvat}	☐	
Gross	double	{res:itgen_fnx_gross}	☐	
HouseWork_2	boolean	{res:itgen_fnx_housework_2}	☐	
HouseWork	boolean	{res:itgen_fnx_housework}	☐	
HouseWorkHoursToReport	int32	{res:itgen_fnx_houseworkhourstoreport}	☐	
HouseWorkType	string	{res:itgen_fnx_houseworktype}	☐	
InvoiceDate	datetime	{res:itgen_fnx_invoicedate}	☐	
InvoicePeriodEnd	datetime	{res:itgen_fnx_invoiceperiodend}	☐	
InvoicePeriodReference	string	{res:itgen_fnx_invoiceperiodreference}	☐	
InvoicePeriodStart	datetime	{res:itgen_fnx_invoiceperiodstart}	☐	
InvoiceType	string	{res:itgen_fnx_invoicetype}	☐	
Language	string	{res:itgen_fnx_language}	☐	
LastRemindDate	datetime	{res:itgen_fnx_lastreminddate}	☐	
Net	double	{res:itgen_fnx_net}	☐	
NotCompleted	boolean	{res:itgen_fnx_notcompleted}	☐	
NoxFinans	boolean	{res:itgen_fnx_noxfinans}	☐	
OCR	string	{res:itgen_fnx_ocr}	☐	
OfferReference	string	{res:itgen_fnx_offerreference}	☐	
OrderReference	string	{res:itgen_fnx_orderreference}	☐	
OrganisationNumber	string	{res:itgen_fnx_organisationnumber}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
OurReference	string(50)	{res:itgen_fnx_ourreference}	☐	
OutboundDate	datetime	{res:itgen_fnx_outbounddate}	☐	
PaymentWay	string	{res:itgen_fnx_paymentway}	☐	
Phone1	string(1024)	{res:itgen_fnx_phone1}	☐	
Phone2	string(1024)	{res:itgen_fnx_phone2}	☐	
Price	double	{res:itgen_fnx_price}	☐	
PriceExcludingVAT	double	{res:itgen_fnx_priceexcludingvat}	☐	
PriceList	string	{res:itgen_fnx_pricelist}	☐	
PrintTemplate	string	{res:itgen_fnx_printtemplate}	☐	
Project_2	string	{res:itgen_fnx_project_2}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Remarks	string(1024)	{res:itgen_fnx_remarks}	☐	
Reminders	int32	{res:itgen_fnx_reminders}	☐	
RoundOff	double	{res:itgen_fnx_roundoff}	☐	
Row Id	int32	{res:itgen_fnx_row id}	☐	
Sent	boolean	{res:itgen_fnx_sent}	☐	
StockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
TaxReduction	int32	{res:itgen_fnx_taxreduction}	☐	
TaxReductionType	string	{res:itgen_fnx_taxreductiontype}	☐	
TermsOfDelivery	string	{res:itgen_fnx_termsofdelivery}	☐	
TermsOfPayment	string	{res:itgen_fnx_termsofpayment}	☐	
TimeBasisReference	int32	{res:itgen_fnx_timebasisreference}	☐	
Total_2	double	{res:itgen_fnx_total_2}	☐	
Total	double	{res:itgen_fnx_total}	☐	
TotalExcludingVAT	double	{res:itgen_fnx_totalexcludingvat}	☐	
TotalToPay	double	{res:itgen_fnx_totaltopay}	☐	
TotalVAT	double	{res:itgen_fnx_totalvat}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Unit	string	{res:itgen_fnx_unit}	☐	
VAT	int32	{res:itgen_fnx_vat}	☐	
VATCode	string	{res:itgen_fnx_vatcode}	☐	
VATIncluded	boolean	{res:itgen_fnx_vatincluded}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
VoucherYear	int32	{res:itgen_fnx_voucheryear}	☐	
WarehouseReady	boolean	{res:itgen_fnx_warehouseready}	☐	
WayOfDelivery	string	{res:itgen_fnx_wayofdelivery}	☐	
YourOrderNumber	string(75)	{res:itgen_fnx_yourordernumber}	☐	
YourReference	string(50)	{res:itgen_fnx_yourreference}	☐	
ZipCode	string(1024)	{res:itgen_fnx_zipcode}	☐	

InvoicePreviewByDocumentNumber

Preview an invoiceThe difference between this and the print-endpoint is that property Sent is not set to TRUE.

Catalog: Fortnox

Schema: Invoices

Label: {res:itgen_fnx_invoicepreviewbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicePreviewByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the invoice
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function InvoicePreviewByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
TEXT	byte[]	{res:itgen_fnx_text}	☐	

InvoicePrintByDocumentNumber

Print an invoice

Catalog: Fortnox

Schema: Invoices

Label: {res:itgen_fnx_invoiceprintbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicePrintByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the invoice
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function InvoicePrintByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
TEXT	byte[]	{res:itgen_fnx_text}	☐	

Invoices

Retrieve a list of invoices

Catalog: Fortnox

Schema: Invoices

Primary Keys: DocumentNumber

Label: {res:itgen_fnx_invoices_plural}

Retrieve: true

Insert: true

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Update: true

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Invoices. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountnumberfrom	string	☐		
accountnumberto	string	☐		
articledescription	string	☐		
articlenumber	string	☐		
costcenter	string	☐		
credit	string	☐		
currency	string	☐		
customername	string	☐		
customernumber	string	☐		
documentnumber	string	☐		
externalinvoicereference1	string	☐		
externalinvoicereference2	string	☐		
filter	string	☐		possibility to filter invoices (Values: cancelled, fullypaid, unpaid, unpaidoverdue, unbooked)
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	string	☐		
fromfinalpaydate	string	☐		
invoicetype	string	☐		
label	string	☐		
lastmodified	string	☐		
notcompleted	string	☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
ocr	string	☐		
ourreference	string	☐		
project	string	☐		
sent	string	☐		
todate	string	☐		
tofinalpaydate	string	☐		
yourordernumber	string	☐		
yourreference	string	☐		

Columns of Table Function

The columns of the table function Invoices are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
Balance	double	{res:itgen_fnx_balance}	☐	
Booked	boolean	{res:itgen_fnx_booked}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Cancelled	boolean	{res:itgen_fnx_cancelled}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Currency	string	{res:itgen_fnx_currency}	☐	
CurrencyRate	double	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	double	{res:itgen_fnx_currencyunit}	☐	
CustomerName	string	{res:itgen_fnx_customername}	☐	
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
DocumentNumber	string	{res:itgen_fnx_documentnumber}	☐	
DueDate	datetime	{res:itgen_fnx_duedate}	☐	
ExternalInvoiceReference1	string	{res:itgen_fnx_externalinvoicereference1}	☐	
ExternalInvoiceReference2	string	{res:itgen_fnx_externalinvoicereference2}	☐	
FinalPayDate	datetime	{res:itgen_fnx_finalpaydate}	☐	
InvoiceDate	datetime	{res:itgen_fnx_invoicedate}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
InvoiceType	string	{res:itgen_fnx_invoicetype}	☐	
NoxFinans	boolean	{res:itgen_fnx_noxfinans}	☐	
OCR	string	{res:itgen_fnx_ocr}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Sent	boolean	{res:itgen_fnx_sent}	☐	
TermsOfPayment	string	{res:itgen_fnx_termsofpayment}	☐	
Total	double	{res:itgen_fnx_total}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
VoucherYear	int32	{res:itgen_fnx_voucheryear}	☐	
WayOfDelivery	string	{res:itgen_fnx_wayofdelivery}	☐	

3.1.34 Schema: Labels

3.1.34.1 Tables

Labels

Retrieve a list of labels

Catalog: Fortnox

Schema: Labels

Primary Keys: Id

Label: {res:itgen_fnx_labels_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Labels. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Labels are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
Description	string(25)	{res:itgen_fnx_description}	☐	
Id	int32	{res:itgen_fnx_id}	☐	

3.1.35 Schema: LockedPeriod

3.1.35.1 Tables

LockedPeriods

Retrieve the locked periodIf no date is returned, no period is locked.

Catalog: Fortnox

Schema: LockedPeriod

Label: {res:itgen_fnx_lockedperiods_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function LockedPeriods. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function LockedPeriods are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
LockedPeriod_EndDate	datetime	{res:itgen_fnx_locked period_enddate}	☐	

3.1.36 Schema: Me

3.1.36.1 Tables

Me

Retrieve user informationUse this endpoint to retrieve user information related to the used access token

Catalog: Fortnox

Schema: Me

Label: {res:itgen_fnx_me_plural}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Me. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Me are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Me_Email	string	{res:itgen_fnx_me_email}	☐	
Me_Id	string	{res:itgen_fnx_me_id}	☐	
Me_Locale	string	{res:itgen_fnx_me_locale}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Me_Name	string	{res:itgen_fnx_me_name}	☐	
Me_SysAdmin	boolean	{res:itgen_fnx_me_sysadmin}	☐	

3.1.37 Schema: ModesOfPayments

3.1.37.1 Tables

PaymentMethodByCode

Retrieve a single mode of payment

Catalog: Fortnox

Schema: ModesOfPayments

Label: {res:itgen_fnx_paymentmethodbycode_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PaymentMethodByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
Code	string	☑		identifies the mode of payment
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function PaymentMethodByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ModeOfPayment__url	string	{res:itgen_fnx_modeofpayment__url}	☐	
ModeOfPayment_AccountNumber	string(4)	{res:itgen_fnx_modeofpayment_accountnumber}	☐	
ModeOfPayment_Code	string	{res:itgen_fnx_modeofpayment_code}	☐	
ModeOfPayment_Description	string	{res:itgen_fnx_modeofpayment_description}	☐	

PaymentModes

Retrieve a list of modes of paymentsThe modes of payments register can return a list of records or a single record. By specifying a Code in the URL, a single record will be returned. Not specifying a Code will return a list of records.

Catalog: Fortnox

Schema: ModesOfPayments

Primary Keys: Code

Label: {res:itgen_fnx_paymentmodes_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PaymentModes. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PaymentModes are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
AccountNumber	string(4)	{res:itgen_fnx_accountnumber}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Code	string	{res:itgen_fnx_code}	☐	
Description	string	{res:itgen_fnx_description}	☐	

3.1.38 Schema: Native

3.1.38.1 Tables

NATIVEPLATFORMSCALARREQUESTS: Fortnox Native Platform Scalar Requests

{res:itgen_native_platform_scalar_requests_desc}

Catalog: Fortnox

Schema: Native

Alias: `npt`

Label: Native Platform Scalar Requests

Documentation:

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The NativePlatformScalarRequests table provides direct access to the native API protocol over an established connection to the Fortnox API server. It will contain a new row for every row inserted with a native API request in PAYLOAD_TEXT with the results of unaltered forwarding of the payload to the Fortnox API server.

Retrieve: true

Insert: true

Update: false

Delete: false

View Columns

The columns of the view NATIVEPLATFORMSCALARREQUESTS are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
BLOB_PREFERRED	boolean	BLOB Preferred	☑	Indicator w hether a BLOB result is preferred over text.
BOL_RESPONSE_CACHE_MAX_AGE_SEC	int32	Response Cache Maximum Age (sec)	☐	Maximum age in seconds of Bridge Online response cache entries to be used.
CONTENT_TYPE	string(240)	Content Type	☐	
DATE_ENDED	datetime	End Date	☑	
DATE_STARTED	datetime	Start Date	☑	
DRY_RUN	boolean	Run w ithout Actions	☑	
DURATION_MS	int64	Duration (ms)	☑	
ERROR_MESSAGE_CODE	string(30)	Error Message Code	☐	
ERROR_MESSAGE_TEXT	string(32000)	Error Message Text	☐	
FAIL_ON_ERROR	boolean	Fail on Error	☑	Whether to raise an exception w hen processing the native request triggered an error from the provider.
HTTP_DISK_CACHE_MAX_AGE_SEC	int32	HTTP Disk Cache Maximum Age (sec)	☐	Maximum age in seconds of HTTP disk cache entries to be used.
HTTP_DISK_CACHE_SAVE	boolean	Save HTTP Disk Cache	☐	Whether results can be stored in HTTP disk cache.
HTTP_DISK_CACHE_USE	boolean	Use HTTP Disk Cache	☐	Whether results can be fetched from HTTP disk cache.
HTTP_MEMORY_CACHE_MAX_AGE_SEC	int32	HTTP Memory Cache Maximum Age (sec)	☐	Maximum age in seconds of HTTP memory cache entries to be used.
HTTP_MEMORY_CACHE_SAVE	boolean	Save HTTP Memory Cache	☐	Whether results can be stored in HTTP memory cache.
HTTP_MEMORY_CACHE_USE	boolean	Use HTTP Memory Cache	☐	Whether results can be fetched from HTTP memory cache.
HTTP_METHOD	string(30)	HTTP Method	☐	
HTTP_STATUS_CODE	int32	HTTP Status Code	☐	
ORIG_SYSTEM_GROUP	string(4000)	Original System Group	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ORIG_SYSTEM_REFERENCE	string(4000)	Original System Reference	☐	
PAYLOAD_TEXT	string	Payload	☐	
RESULT_BLOB	byte[]	Result BLOB	☐	
RESULT_DATE_TIME_UTC	datetime	Result Date Time	☐	
RESULT_NUMBER	decimal	Result Number	☐	
RESULT_TEXT	string	Result Text	☐	
SUCCESSFUL	boolean	Succesful	☑	
TIMEOUT_SEC	int32	Timeout (sec)	☐	Timeout in seconds.
TRANSACTION_ID	int64	Transaction ID	☑	Incrementing ID of the transaction.
URL	string(4000)	URL	☐	

3.1.39 Schema: Offers

3.1.39.1 Tables

OfferByDocumentNumber

Retrieve a single offer

Catalog: Fortnox

Schema: Offers

Label: {res:itgen_fnx_offerbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function OfferByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the offer
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function OfferByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Offer__url	string	{res:itgen_fnx_offer__url}	☐	
Offer__urlTaxReductionList	string	{res:itgen_fnx_offer__urltaxreductionlist}	☐	
Offer_Address1	string(1024)	{res:itgen_fnx_offer_address1}	☐	
Offer_Address2	string(1024)	{res:itgen_fnx_offer_address2}	☐	
Offer_AdministrationFee	double	{res:itgen_fnx_offer_administrationfee}	☐	
Offer_AdministrationFeeVAT	double	{res:itgen_fnx_offer_administrationfeevat}	☐	
Offer_BasisTaxReduction	double	{res:itgen_fnx_offer_basistaxreduction}	☐	
Offer_Cancelled	boolean	{res:itgen_fnx_offer_cancelled}	☐	
Offer_City	string(1024)	{res:itgen_fnx_offer_city}	☐	
Offer_Comments	string(1024)	{res:itgen_fnx_offer_comments}	☐	
Offer_ContributionPercent	double	{res:itgen_fnx_offer_contributionpercent}	☐	
Offer_ContributionValue	double	{res:itgen_fnx_offer_contributionvalue}	☐	
Offer_CopyRemarks	boolean	{res:itgen_fnx_offer_copyremarks}	☐	
Offer_CostCenter	string	{res:itgen_fnx_offer_costcenter}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Offer_Country	string	{res:itgen_fnx_offer_country}	☐	
Offer_Currency	string	{res:itgen_fnx_offer_currency}	☐	
Offer_CurrencyRate	double	{res:itgen_fnx_offer_currencyrate}	☐	
Offer_CurrencyUnit	int32	{res:itgen_fnx_offer_currencyunit}	☐	
Offer_CustomerName	string	{res:itgen_fnx_offer_customername}	☐	
Offer_CustomerNumber	string	{res:itgen_fnx_offer_customernumber}	☐	
Offer_DeliveryAddress1	string(1024)	{res:itgen_fnx_offer_deliveryaddress1}	☐	
Offer_DeliveryAddress2	string(1024)	{res:itgen_fnx_offer_deliveryaddress2}	☐	
Offer_DeliveryCity	string(1024)	{res:itgen_fnx_offer_deliverycity}	☐	
Offer_DeliveryCountry	string	{res:itgen_fnx_offer_deliverycountry}	☐	
Offer_DeliveryDate	datetime	{res:itgen_fnx_offer_deliverydate}	☐	
Offer_DeliveryName	string(1024)	{res:itgen_fnx_offer_deliveryname}	☐	
Offer_DeliveryZipCode	string	{res:itgen_fnx_offer_deliveryzipcode}	☐	
Offer_DocumentNumber	string	{res:itgen_fnx_offer_documentnumber}	☐	
Offer_EmailInformation_EmailAddressBCC	string	{res:itgen_fnx_offer_emailinformation_emailaddressbcc}	☐	
Offer_EmailInformation_EmailAddressCC	string	{res:itgen_fnx_offer_emailinformation_emailaddresscc}	☐	
Offer_EmailInformation_EmailAddressFrom	string	{res:itgen_fnx_offer_emailinformation_emailaddressfrom}	☐	
Offer_EmailInformation_EmailAddressTo	string	{res:itgen_fnx_offer_emailinformation_emailaddressto}	☐	
Offer_EmailInformation_EmailBody	string(20000)	{res:itgen_fnx_offer_emailinformation_emailbody}	☐	
Offer_EmailInformation_EmailSubject	string(100)	{res:itgen_fnx_offer_emailinformation_emailsubject}	☐	
Offer_ExpireDate	datetime	{res:itgen_fnx_offer_expiredate}	☐	
Offer_Freight	double	{res:itgen_fnx_offer_freight}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Offer_FreightVAT	double	{res:itgen_fnx_offer_freightvat}	☐	
Offer_Gross	double	{res:itgen_fnx_offer_gross}	☐	
Offer_HouseWork	boolean	{res:itgen_fnx_offer_housework}	☐	
Offer_InvoiceReference	string	{res:itgen_fnx_offer_invoicereference}	☐	
Offer_Language	string	{res:itgen_fnx_offer_language}	☐	
Offer_Net	double	{res:itgen_fnx_offer_net}	☐	
Offer_NotCompleted	boolean	{res:itgen_fnx_offer_notcompleted}	☐	
Offer_OfferDate	datetime	{res:itgen_fnx_offer_offerdate}	☐	
Offer_OrderReference	string	{res:itgen_fnx_offer_orderreference}	☐	
Offer_OrganisationNumber	string	{res:itgen_fnx_offer_organisationnumber}	☐	
Offer_OurReference	string	{res:itgen_fnx_offer_ourreference}	☐	
Offer_Phone1	string	{res:itgen_fnx_offer_phone1}	☐	
Offer_Phone2	string	{res:itgen_fnx_offer_phone2}	☐	
Offer_PriceList	string	{res:itgen_fnx_offer_pricelist}	☐	
Offer_PrintTemplate	string	{res:itgen_fnx_offer_printtemplate}	☐	
Offer_Project	string	{res:itgen_fnx_offer_project}	☐	
Offer_Remarks	string	{res:itgen_fnx_offer_remarks}	☐	
Offer_RoundOff	double	{res:itgen_fnx_offer_roundoff}	☐	
Offer_Sent	boolean	{res:itgen_fnx_offer_sent}	☐	
Offer_TaxReduction	int32	{res:itgen_fnx_offer_taxreduction}	☐	
Offer_TaxReductionType	string	{res:itgen_fnx_offer_taxreductiontype}	☐	
Offer_TermsOfDelivery	string	{res:itgen_fnx_offer_termsofdelivery}	☐	
Offer_TermsOfPayment	string	{res:itgen_fnx_offer_termsofpayment}	☐	
Offer_Total	double	{res:itgen_fnx_offer_total}	☐	
Offer_TotalToPay	double	{res:itgen_fnx_offer_totaltopay}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Offer_TotalVAT	double	{res:itgen_fnx_offer_totalvat}	☐	
Offer_VATIncluded	boolean	{res:itgen_fnx_offer_vatincluded}	☐	
Offer_WayOfDelivery	string	{res:itgen_fnx_offer_wayofdelivery}	☐	
Offer_YourReference	string	{res:itgen_fnx_offer_yourreference}	☐	
Offer_YourReferenceNumber	string	{res:itgen_fnx_offer_yourreferencenumber}	☐	
Offer_ZipCode	string	{res:itgen_fnx_offer_zipcode}	☐	

OfferLabelsByDocumentNumber

Retrieve a single offer

Catalog: Fortnox

Schema: Offers

Label: {res:itgen_fnx_offerlabelsbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function OfferLabelsByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the offer
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function OfferLabelsByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Address1	string(1024)	{res:itgen_fnx_address1}	☐	
Address2	string(1024)	{res:itgen_fnx_address2}	☐	
AdministrationFee	double	{res:itgen_fnx_administrationfee}	☐	
AdministrationFeeVAT	double	{res:itgen_fnx_administrationfeevat}	☐	
BasisTaxReduction	double	{res:itgen_fnx_basistaxreduction}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
c_urlTaxReductionList	string	{res:itgen_fnx_c_urltaxreductionlist}	☐	
Cancelled	boolean	{res:itgen_fnx_cancelled}	☐	
City	string(1024)	{res:itgen_fnx_city}	☐	
Comments	string(1024)	{res:itgen_fnx_comments}	☐	
ContributionPercent	double	{res:itgen_fnx_contributionpercent}	☐	
ContributionValue	double	{res:itgen_fnx_contributionvalue}	☐	
CopyRemarks	boolean	{res:itgen_fnx_copyremarks}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	
Currency	string	{res:itgen_fnx_currency}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CurrencyRate	double	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	int32	{res:itgen_fnx_currencyunit}	☐	
CustomerName	string	{res:itgen_fnx_customername}	☐	
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
DeliveryAddress1	string(1024)	{res:itgen_fnx_deliveryaddress1}	☐	
DeliveryAddress2	string(1024)	{res:itgen_fnx_deliveryaddress2}	☐	
DeliveryCity	string(1024)	{res:itgen_fnx_deliverycity}	☐	
DeliveryCountry	string	{res:itgen_fnx_deliverycountry}	☐	
DeliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	
DeliveryName	string(1024)	{res:itgen_fnx_deliveryname}	☐	
DeliveryZipCode	string	{res:itgen_fnx_deliveryzipcode}	☐	
DocumentNumber	string	{res:itgen_fnx_documentnumber}	☐	
EmailInformation_EmailAddressBCC	string	{res:itgen_fnx_emailinformation_emailaddressbcc}	☐	
EmailInformation_EmailAddressCC	string	{res:itgen_fnx_emailinformation_emailaddresscc}	☐	
EmailInformation_EmailAddressFrom	string	{res:itgen_fnx_emailinformation_emailaddressfrom}	☐	
EmailInformation_EmailAddressTo	string	{res:itgen_fnx_emailinformation_emailaddressto}	☐	
EmailInformation_EmailBody	string(20000)	{res:itgen_fnx_emailinformation_emailbody}	☐	
EmailInformation_EmailSubject	string(100)	{res:itgen_fnx_emailinformation_emailsubject}	☐	
ExpireDate	datetime	{res:itgen_fnx_expiredate}	☐	
Freight	double	{res:itgen_fnx_freight}	☐	
FreightVAT	double	{res:itgen_fnx_freightvat}	☐	
Gross	double	{res:itgen_fnx_gross}	☐	
HouseWork	boolean	{res:itgen_fnx_housework}	☐	
Id	int32	{res:itgen_fnx_id}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
InvoiceReference	string	{res:itgen_fnx_invoicereference}	☐	
Language	string	{res:itgen_fnx_language}	☐	
Net	double	{res:itgen_fnx_net}	☐	
NotCompleted	boolean	{res:itgen_fnx_notcompleted}	☐	
OfferDate	datetime	{res:itgen_fnx_offerdate}	☐	
OrderReference	string	{res:itgen_fnx_orderreference}	☐	
OrganisationNumber	string	{res:itgen_fnx_organisationnumber}	☐	
OurReference	string	{res:itgen_fnx_ourreference}	☐	
Phone1	string	{res:itgen_fnx_phone1}	☐	
Phone2	string	{res:itgen_fnx_phone2}	☐	
PriceList	string	{res:itgen_fnx_pricelist}	☐	
PrintTemplate	string	{res:itgen_fnx_printtemplate}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Remarks	string	{res:itgen_fnx_remarks}	☐	
RoundOff	double	{res:itgen_fnx_roundoff}	☐	
Sent	boolean	{res:itgen_fnx_sent}	☐	
TaxReduction	int32	{res:itgen_fnx_taxreduction}	☐	
TaxReductionType	string	{res:itgen_fnx_taxreductiontype}	☐	
TermsOfDelivery	string	{res:itgen_fnx_termsofdelivery}	☐	
TermsOfPayment	string	{res:itgen_fnx_termsofpayment}	☐	
Total	double	{res:itgen_fnx_total}	☐	
TotalToPay	double	{res:itgen_fnx_totaltopay}	☐	
TotalVAT	double	{res:itgen_fnx_totalvat}	☐	
VATIncluded	boolean	{res:itgen_fnx_vatincluded}	☐	
WayOfDelivery	string	{res:itgen_fnx_wayofdelivery}	☐	
YourReference	string	{res:itgen_fnx_yourreference}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
YourReferenceNumber	string	{res:itgen_fnx_yourreferencenumber}	☐	
ZipCode	string	{res:itgen_fnx_zipcode}	☐	

OfferLinesByDocumentNumber

Retrieve a single offer

Catalog: Fortnox

Schema: Offers

Label: {res:itgen_fnx_offerlinesbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function OfferLinesByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the offer
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function OfferLinesByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AccountNumber	int32	{res:itgen_fnx_accountnumber}	☐	
Address1	string(1024)	{res:itgen_fnx_address1}	☐	
Address2	string(1024)	{res:itgen_fnx_address2}	☐	
AdministrationFee	double	{res:itgen_fnx_administrationfee}	☐	
AdministrationFeeVAT	double	{res:itgen_fnx_administrationfeevat}	☐	
ArticleNumber	string	{res:itgen_fnx_articlenumber}	☐	
BasisTaxReduction	double	{res:itgen_fnx_basistaxreduction}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
c_urlTaxReductionList	string	{res:itgen_fnx_c_urltaxreductionlist}	☐	
Cancelled	boolean	{res:itgen_fnx_cancelled}	☐	
City	string(1024)	{res:itgen_fnx_city}	☐	
Comments	string(1024)	{res:itgen_fnx_comments}	☐	
ContributionPercent_2	double	{res:itgen_fnx_contributionpercent_2}	☐	
ContributionPercent	string	{res:itgen_fnx_contributionpercent}	☐	
ContributionValue_2	double	{res:itgen_fnx_contributionvalue_2}	☐	
ContributionValue	string	{res:itgen_fnx_contributionvalue}	☐	
CopyRemarks	boolean	{res:itgen_fnx_copyremarks}	☐	
CostCenter_2	string	{res:itgen_fnx_costcenter_2}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Currency	string	{res:itgen_fnx_currency}	☐	
CurrencyRate	double	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	int32	{res:itgen_fnx_currencyunit}	☐	
CustomerName	string	{res:itgen_fnx_customername}	☐	
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
DeliveryAddress1	string(1024)	{res:itgen_fnx_deliveryaddress1}	☐	
DeliveryAddress2	string(1024)	{res:itgen_fnx_deliveryaddress2}	☐	
DeliveryCity	string(1024)	{res:itgen_fnx_deliverycity}	☐	
DeliveryCountry	string	{res:itgen_fnx_deliverycountry}	☐	
DeliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	
DeliveryName	string(1024)	{res:itgen_fnx_deliveryname}	☐	
DeliveryZipCode	string	{res:itgen_fnx_deliveryzipcode}	☐	
Description	string	{res:itgen_fnx_description}	☐	
Discount	double	{res:itgen_fnx_discount}	☐	
DiscountType	string	{res:itgen_fnx_discounttype}	☐	
DocumentNumber	string	{res:itgen_fnx_documentnumber}	☐	
EmailInformation_EmailAddressBCC	string	{res:itgen_fnx_emailinformation_emailaddressbcc}	☐	
EmailInformation_EmailAddressCC	string	{res:itgen_fnx_emailinformation_emailaddresscc}	☐	
EmailInformation_EmailAddressFrom	string	{res:itgen_fnx_emailinformation_emailaddressfrom}	☐	
EmailInformation_EmailAddressTo	string	{res:itgen_fnx_emailinformation_emailaddressto}	☐	
EmailInformation_EmailBody	string(20000)	{res:itgen_fnx_emailinformation_emailbody}	☐	
EmailInformation_EmailSubject	string(100)	{res:itgen_fnx_emailinformation_emailsubject}	☐	
ExpireDate	datetime	{res:itgen_fnx_expiredate}	☐	
Freight	double	{res:itgen_fnx_freight}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
FreightVAT	double	{res:itgen_fnx_freightvat}	☐	
Gross	double	{res:itgen_fnx_gross}	☐	
HouseWork_2	boolean	{res:itgen_fnx_housework_2}	☐	
HouseWork	boolean	{res:itgen_fnx_housework}	☐	
HouseWorkHoursToReport	double	{res:itgen_fnx_houseworkhourstoreport}	☐	
HouseWorkType	string	{res:itgen_fnx_houseworktype}	☐	
InvoiceReference	string	{res:itgen_fnx_invoicereference}	☐	
Language	string	{res:itgen_fnx_language}	☐	
Net	double	{res:itgen_fnx_net}	☐	
NotCompleted	boolean	{res:itgen_fnx_notcompleted}	☐	
OfferDate	datetime	{res:itgen_fnx_offerdate}	☐	
OrderReference	string	{res:itgen_fnx_orderreference}	☐	
OrganisationNumber	string	{res:itgen_fnx_organisationnumber}	☐	
OurReference	string	{res:itgen_fnx_ourreference}	☐	
Phone1	string	{res:itgen_fnx_phone1}	☐	
Phone2	string	{res:itgen_fnx_phone2}	☐	
Price	double	{res:itgen_fnx_price}	☐	
PriceList	string	{res:itgen_fnx_pricelist}	☐	
PrintTemplate	string	{res:itgen_fnx_printtemplate}	☐	
Project_2	string	{res:itgen_fnx_project_2}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Quantity	string	{res:itgen_fnx_quantity}	☐	
Remarks	string	{res:itgen_fnx_remarks}	☐	
RoundOff	double	{res:itgen_fnx_roundoff}	☐	
Row Id	int32	{res:itgen_fnx_row id}	☐	
Sent	boolean	{res:itgen_fnx_sent}	☐	
TaxReduction	int32	{res:itgen_fnx_taxreduction}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TaxReductionType	string	{res:itgen_fnx_taxreductiontype}	☐	
TermsOfDelivery	string	{res:itgen_fnx_termsofdelivery}	☐	
TermsOfPayment	string	{res:itgen_fnx_termsofpayment}	☐	
Total_2	double	{res:itgen_fnx_total_2}	☐	
Total	double	{res:itgen_fnx_total}	☐	
TotalToPay	double	{res:itgen_fnx_totaltopay}	☐	
TotalVAT	double	{res:itgen_fnx_totalvat}	☐	
Unit	string	{res:itgen_fnx_unit}	☐	
VAT	int32	{res:itgen_fnx_vat}	☐	
VATCode	string	{res:itgen_fnx_vatcode}	☐	
VATIncluded	boolean	{res:itgen_fnx_vatincluded}	☐	
WayOfDelivery	string	{res:itgen_fnx_wayofdelivery}	☐	
YourReference	string	{res:itgen_fnx_yourreference}	☐	
YourReferenceNumber	string	{res:itgen_fnx_yourreferencenumber}	☐	
ZipCode	string	{res:itgen_fnx_zipcode}	☐	

Offers

Retrieve a list of offers

Catalog: Fortnox

Schema: Offers

Primary Keys: DocumentNumber

Label: {res:itgen_fnx_offers_plural}

Retrieve: true

Insert: true

Update: true

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Offers. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
costcenter	string	☐		filter by cost center
customername	string	☐		filter by customer name
customernumber	string	☐		filter by customer number
documentnumber	string	☐		filter by document number
filter	string	☐		possibility to filter offers (Values: cancelled, expired, completed, notcompleted, ordercreated, ordernotcreated)
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	string	☐		filter by from date
label	string	☐		filter by label
lastmodified	string	☐		filter by last modified
notcompleted	boolean	☐		filter by not completed
ourreference	string	☐		filter by our reference
project	string	☐		filter by project
sent	boolean	☐		filter by sent
todate	string	☐		filter by to date
yourreference	string	☐		filter by your reference

Columns of Table Function

The columns of the table function Offers are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
Cancelled	boolean	{res:itgen_fnx_cancelled}	☐	
Currency	string	{res:itgen_fnx_currency}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CustomerName	string	{res:itgen_fnx_customername}	☐	
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
DocumentNumber	string	{res:itgen_fnx_documentnumber}	☐	
OfferDate	datetime	{res:itgen_fnx_offerdate}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Sent	boolean	{res:itgen_fnx_sent}	☐	
Total	double	{res:itgen_fnx_total}	☐	

3.1.40 Schema: Orders

3.1.40.1 Tables

OrderByDocumentNumber

Retrieve a single order

Catalog: Fortnox

Schema: Orders

Label: {res:itgen_fnx_orderbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function OrderByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the order
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function OrderByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Order__url	string	{res:itgen_fnx_order_ _url}	☐	
Order__urlTaxReductionList	string	{res:itgen_fnx_order_ _urltaxreductionlist}	☐	
Order_Address1	string	{res:itgen_fnx_order_ address1}	☐	
Order_Address2	string	{res:itgen_fnx_order_ address2}	☐	
Order_AdministrationFee	double	{res:itgen_fnx_order_ administrationfee}	☐	
Order_AdministrationFeeVAT	double	{res:itgen_fnx_order_ administrationfeevat}	☐	
Order_BasisTaxReduction	double	{res:itgen_fnx_order_ basistaxreduction}	☐	
Order_Cancelled	boolean	{res:itgen_fnx_order_ cancelled}	☐	
Order_City	string	{res:itgen_fnx_order_ city}	☐	
Order_Comments	string	{res:itgen_fnx_order_ comments}	☐	
Order_ContributionPercent	double	{res:itgen_fnx_order_ contributionpercent}	☐	
Order_ContributionValue	double	{res:itgen_fnx_order_ contributionvalue}	☐	
Order_CopyRemarks	boolean	{res:itgen_fnx_order_ copyremarks}	☐	
Order_CostCenter	string	{res:itgen_fnx_order_ costcenter}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Order_Country	string	{res:itgen_fnx_order_country}	☐	
Order_Currency	string(3)	{res:itgen_fnx_order_currency}	☐	
Order_CurrencyRate	double	{res:itgen_fnx_order_currencyrate}	☐	
Order_CurrencyUnit	double	{res:itgen_fnx_order_currencyunit}	☐	
Order_CustomerName	string	{res:itgen_fnx_order_customername}	☐	
Order_CustomerNumber	string	{res:itgen_fnx_order_customernumber}	☐	
Order_DeliveryAddress1	string	{res:itgen_fnx_order_deliveryaddress1}	☐	
Order_DeliveryAddress2	string	{res:itgen_fnx_order_deliveryaddress2}	☐	
Order_DeliveryCity	string	{res:itgen_fnx_order_deliverycity}	☐	
Order_DeliveryCountry	string	{res:itgen_fnx_order_deliverycountry}	☐	
Order_DeliveryDate	datetime	{res:itgen_fnx_order_deliverydate}	☐	
Order_DeliveryName	string	{res:itgen_fnx_order_deliveryname}	☐	
Order_DeliveryState	string	{res:itgen_fnx_order_deliverystate}	☐	
Order_DeliveryZipCode	string	{res:itgen_fnx_order_deliveryzipcode}	☐	
Order_DocumentNumber	string	{res:itgen_fnx_order_documentnumber}	☐	
Order_EmailInformation_EmailAddressBCC	string	{res:itgen_fnx_order_emailinformation_emailaddressbcc}	☐	
Order_EmailInformation_EmailAddressCC	string	{res:itgen_fnx_order_emailinformation_emailaddresscc}	☐	
Order_EmailInformation_EmailAddressFrom	string	{res:itgen_fnx_order_emailinformation_emailaddressfrom}	☐	
Order_EmailInformation_EmailAddressTo	string	{res:itgen_fnx_order_emailinformation_emailaddressto}	☐	
Order_EmailInformation_EmailBody	string	{res:itgen_fnx_order_emailinformation_emailbody}	☐	
Order_EmailInformation_EmailSubject	string	{res:itgen_fnx_order_emailinformation_emailsubject}	☐	
Order_ExternalInvoiceReference1	string	{res:itgen_fnx_order_externalinvoicereference1}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Order_ExternalInvoiceReference2	string	{res:itgen_fnx_order_externalinvoicereference2}	☐	
Order_Freight	double	{res:itgen_fnx_order_freight}	☐	
Order_FreightVAT	double	{res:itgen_fnx_order_freightvat}	☐	
Order_Gross	double	{res:itgen_fnx_order_gross}	☐	
Order_HouseWork	boolean	{res:itgen_fnx_order_housework}	☐	
Order_InvoiceReference	string	{res:itgen_fnx_order_invoicereference}	☐	
Order_Language	string	{res:itgen_fnx_order_language}	☐	
Order_Net	double	{res:itgen_fnx_order_net}	☐	
Order_NotCompleted	boolean	{res:itgen_fnx_order_notcompleted}	☐	
Order_OfferReference	string	{res:itgen_fnx_order_offerreference}	☐	
Order_OrderDate	datetime	{res:itgen_fnx_order_orderdate}	☐	
Order_OrderType	string	{res:itgen_fnx_order_ordertype}	☐	
Order_OrganisationNumber	string	{res:itgen_fnx_order_organisationnumber}	☐	
Order_OurReference	string(50)	{res:itgen_fnx_order_ourreference}	☐	
Order_OutboundDate	datetime	{res:itgen_fnx_order_outbounddate}	☐	
Order_Phone1	string	{res:itgen_fnx_order_phone1}	☐	
Order_Phone2	string	{res:itgen_fnx_order_phone2}	☐	
Order_PriceList	string	{res:itgen_fnx_order_pricelist}	☐	
Order_PrintTemplate	string	{res:itgen_fnx_order_printtemplate}	☐	
Order_Project	string	{res:itgen_fnx_order_project}	☐	
Order_Remarks	string(1024)	{res:itgen_fnx_order_remarks}	☐	
Order_RoundOff	double	{res:itgen_fnx_order_roundoff}	☐	
Order_Sent	boolean	{res:itgen_fnx_order_sent}	☐	
Order_StockPointCode	string	{res:itgen_fnx_order_stockpointcode}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Order_StockPointId	string	{res:itgen_fnx_order_stockpointid}	☐	
Order_TaxReduction	double	{res:itgen_fnx_order_taxreduction}	☐	
Order_TaxReductionType	string	{res:itgen_fnx_order_taxreductiontype}	☐	
Order_TermsOfDelivery	string	{res:itgen_fnx_order_termsofdelivery}	☐	
Order_TermsOfPayment	string	{res:itgen_fnx_order_termsofpayment}	☐	
Order_TimeBasisReference	int32	{res:itgen_fnx_order_timebasisreference}	☐	
Order_Total	double	{res:itgen_fnx_order_total}	☐	
Order_TotalToPay	double	{res:itgen_fnx_order_totaltopay}	☐	
Order_TotalVAT	double	{res:itgen_fnx_order_totalvat}	☐	
Order_VATIncluded	boolean	{res:itgen_fnx_order_vatincluded}	☐	
Order_WarehouseReady	boolean	{res:itgen_fnx_order_warehouseready}	☐	
Order_WayOfDelivery	string	{res:itgen_fnx_order_wayofdelivery}	☐	
Order_YourOrderNumber	string(75)	{res:itgen_fnx_order_yourordernumber}	☐	
Order_YourReference	string(50)	{res:itgen_fnx_order_yourreference}	☐	
Order_ZipCode	string	{res:itgen_fnx_order_zipcode}	☐	

OrderLabelsByDocumentNumber

Retrieve a single order

Catalog: Fortnox

Schema: Orders

Label: {res:itgen_fnx_orderlabelsbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function OrderLabelsByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the order
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function OrderLabelsByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Address1	string	{res:itgen_fnx_address1}	☐	
Address2	string	{res:itgen_fnx_address2}	☐	
AdministrationFee	double	{res:itgen_fnx_administrationfee}	☐	
AdministrationFeeVAT	double	{res:itgen_fnx_administrationfeevat}	☐	
BasisTaxReduction	double	{res:itgen_fnx_basistaxreduction}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
c_urlTaxReductionList	string	{res:itgen_fnx_c_urltaxreductionlist}	☐	
Cancelled	boolean	{res:itgen_fnx_cancelled}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
City	string	{res:itgen_fnx_city}	☐	
Comments	string	{res:itgen_fnx_comments}	☐	
ContributionPercent	double	{res:itgen_fnx_contributionpercent}	☐	
ContributionValue	double	{res:itgen_fnx_contributionvalue}	☐	
CopyRemarks	boolean	{res:itgen_fnx_copyremarks}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	
Currency	string(3)	{res:itgen_fnx_currency}	☐	
CurrencyRate	double	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	double	{res:itgen_fnx_currencyunit}	☐	
CustomerName	string	{res:itgen_fnx_customername}	☐	
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
DeliveryAddress1	string	{res:itgen_fnx_deliveryaddress1}	☐	
DeliveryAddress2	string	{res:itgen_fnx_deliveryaddress2}	☐	
DeliveryCity	string	{res:itgen_fnx_deliverycity}	☐	
DeliveryCountry	string	{res:itgen_fnx_deliverycountry}	☐	
DeliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	
DeliveryName	string	{res:itgen_fnx_deliveryname}	☐	
DeliveryState	string	{res:itgen_fnx_deliverystate}	☐	
DeliveryZipCode	string	{res:itgen_fnx_deliveryzipcode}	☐	
DocumentNumber	string	{res:itgen_fnx_documentnumber}	☐	
EmailInformation_EmailAddressBCC	string	{res:itgen_fnx_emailinformation_emailaddressbcc}	☐	
EmailInformation_EmailAddressCC	string	{res:itgen_fnx_emailinformation_emailaddresscc}	☐	
EmailInformation_EmailAddressFrom	string	{res:itgen_fnx_emailinformation_emailaddressfrom}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
EmailInformation_EmailAddressTo	string	{res:itgen_fnx_emailinformation_emailaddresto}	☐	
EmailInformation_EmailBody	string	{res:itgen_fnx_emailinformation_emailbody}	☐	
EmailInformation_EmailSubject	string	{res:itgen_fnx_emailinformation_emailsubject}	☐	
ExternalInvoiceReference1	string	{res:itgen_fnx_externalinvoicereference1}	☐	
ExternalInvoiceReference2	string	{res:itgen_fnx_externalinvoicereference2}	☐	
Freight	double	{res:itgen_fnx_freight}	☐	
FreightVAT	double	{res:itgen_fnx_freightvat}	☐	
Gross	double	{res:itgen_fnx_gross}	☐	
HouseWork	boolean	{res:itgen_fnx_housework}	☐	
Id	int32	{res:itgen_fnx_id}	☐	
InvoiceReference	string	{res:itgen_fnx_invoicereference}	☐	
Language	string	{res:itgen_fnx_language}	☐	
Net	double	{res:itgen_fnx_net}	☐	
NotCompleted	boolean	{res:itgen_fnx_notcompleted}	☐	
OfferReference	string	{res:itgen_fnx_offerreference}	☐	
OrderDate	datetime	{res:itgen_fnx_orderdate}	☐	
OrderType	string	{res:itgen_fnx_ordertype}	☐	
OrganisationNumber	string	{res:itgen_fnx_organisationnumber}	☐	
OurReference	string(50)	{res:itgen_fnx_ourreference}	☐	
OutboundDate	datetime	{res:itgen_fnx_outbounddate}	☐	
Phone1	string	{res:itgen_fnx_phone1}	☐	
Phone2	string	{res:itgen_fnx_phone2}	☐	
PriceList	string	{res:itgen_fnx_pricelist}	☐	
PrintTemplate	string	{res:itgen_fnx_printtemplate}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Remarks	string(1024)	{res:itgen_fnx_remarks}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
RoundOff	double	{res:itgen_fnx_roundoff}	☐	
Sent	boolean	{res:itgen_fnx_sent}	☐	
StockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
StockPointId	string	{res:itgen_fnx_stockpointid}	☐	
TaxReduction	double	{res:itgen_fnx_taxreduction}	☐	
TaxReductionType	string	{res:itgen_fnx_taxreductiontype}	☐	
TermsOfDelivery	string	{res:itgen_fnx_termsofdelivery}	☐	
TermsOfPayment	string	{res:itgen_fnx_termsofpayment}	☐	
TimeBasisReference	int32	{res:itgen_fnx_timebasisreference}	☐	
Total	double	{res:itgen_fnx_total}	☐	
TotalToPay	double	{res:itgen_fnx_totaltopay}	☐	
TotalVAT	double	{res:itgen_fnx_totalvat}	☐	
VATIncluded	boolean	{res:itgen_fnx_vatincluded}	☐	
WarehouseReady	boolean	{res:itgen_fnx_warehouseready}	☐	
WayOfDelivery	string	{res:itgen_fnx_wayofdelivery}	☐	
YourOrderNumber	string(75)	{res:itgen_fnx_yourordernumber}	☐	
YourReference	string(50)	{res:itgen_fnx_yourreference}	☐	
ZipCode	string	{res:itgen_fnx_zipcode}	☐	

OrderLinesByDocumentNumber

Retrieve a single order

Catalog: Fortnox

Schema: Orders

Label: {res:itgen_fnx_orderlinesbydocumentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function OrderLinesByDocumentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
DocumentNumber	string	☑		identifies the order
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function OrderLinesByDocumentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AccountNumber	int32	{res:itgen_fnx_accountnumber}	☐	
Address1	string	{res:itgen_fnx_address1}	☐	
Address2	string	{res:itgen_fnx_address2}	☐	
AdministrationFee	double	{res:itgen_fnx_administrationfee}	☐	
AdministrationFeeVAT	double	{res:itgen_fnx_administrationfeevat}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ArticleNumber	string	{res:itgen_fnx_articlenumber}	☐	
BasisTaxReduction	double	{res:itgen_fnx_basistaxreduction}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
c_urlTaxReductionList	string	{res:itgen_fnx_c_urltaxreductionlist}	☐	
Cancelled	boolean	{res:itgen_fnx_cancelled}	☐	
City	string	{res:itgen_fnx_city}	☐	
Comments	string	{res:itgen_fnx_comments}	☐	
ContributionPercent_2	double	{res:itgen_fnx_contributionpercent_2}	☐	
ContributionPercent	string	{res:itgen_fnx_contributionpercent}	☐	
ContributionValue_2	double	{res:itgen_fnx_contributionvalue_2}	☐	
ContributionValue	string	{res:itgen_fnx_contributionvalue}	☐	
CopyRemarks	boolean	{res:itgen_fnx_copyremarks}	☐	
CostCenter_2	string	{res:itgen_fnx_costcenter_2}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Country	string	{res:itgen_fnx_country}	☐	
Currency	string(3)	{res:itgen_fnx_currency}	☐	
CurrencyRate	double	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	double	{res:itgen_fnx_currencyunit}	☐	
CustomerName	string	{res:itgen_fnx_customername}	☐	
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
DeliveredQuantity	string	{res:itgen_fnx_deliveredquantity}	☐	
DeliveryAddress1	string	{res:itgen_fnx_deliveryaddress1}	☐	
DeliveryAddress2	string	{res:itgen_fnx_deliveryaddress2}	☐	
DeliveryCity	string	{res:itgen_fnx_deliverycity}	☐	
DeliveryCountry	string	{res:itgen_fnx_deliverycountry}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
DeliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	
DeliveryName	string	{res:itgen_fnx_deliveryname}	☐	
DeliveryState	string	{res:itgen_fnx_deliverystate}	☐	
DeliveryZipCode	string	{res:itgen_fnx_deliveryzipcode}	☐	
Description	string	{res:itgen_fnx_description}	☐	
Discount	double	{res:itgen_fnx_discount}	☐	
DiscountType	string	{res:itgen_fnx_discounttype}	☐	
DocumentNumber	string	{res:itgen_fnx_documentnumber}	☐	
EmailInformation_EmailAddressBCC	string	{res:itgen_fnx_emailinformation_emailaddressbcc}	☐	
EmailInformation_EmailAddressCC	string	{res:itgen_fnx_emailinformation_emailaddresscc}	☐	
EmailInformation_EmailAddressFrom	string	{res:itgen_fnx_emailinformation_emailaddressfrom}	☐	
EmailInformation_EmailAddressTo	string	{res:itgen_fnx_emailinformation_emailaddressto}	☐	
EmailInformation_EmailBody	string	{res:itgen_fnx_emailinformation_emailbody}	☐	
EmailInformation_EmailSubject	string	{res:itgen_fnx_emailinformation_emailsubject}	☐	
ExternalInvoiceReference1	string	{res:itgen_fnx_externalinvoicereference1}	☐	
ExternalInvoiceReference2	string	{res:itgen_fnx_externalinvoicereference2}	☐	
Freight	double	{res:itgen_fnx_freight}	☐	
FreightVAT	double	{res:itgen_fnx_freightvat}	☐	
Gross	double	{res:itgen_fnx_gross}	☐	
HouseWork_2	boolean	{res:itgen_fnx_housework_2}	☐	
HouseWork	boolean	{res:itgen_fnx_housework}	☐	
HouseWorkHoursToReport	int32	{res:itgen_fnx_houseworkhourstoreport}	☐	
HouseWorkType	string	{res:itgen_fnx_houseworktype}	☐	
InvoiceReference	string	{res:itgen_fnx_invoicereference}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Language	string	{res:itgen_fnx_language}	☐	
Net	double	{res:itgen_fnx_net}	☐	
NotCompleted	boolean	{res:itgen_fnx_notcompleted}	☐	
OfferReference	string	{res:itgen_fnx_offerreference}	☐	
OrderDate	datetime	{res:itgen_fnx_orderdate}	☐	
OrderedQuantity	string	{res:itgen_fnx_orderedquantity}	☐	
OrderType	string	{res:itgen_fnx_ordertype}	☐	
OrganisationNumber	string	{res:itgen_fnx_organisationnumber}	☐	
OurReference	string(50)	{res:itgen_fnx_ourreference}	☐	
OutboundDate	datetime	{res:itgen_fnx_outbounddate}	☐	
Phone1	string	{res:itgen_fnx_phone1}	☐	
Phone2	string	{res:itgen_fnx_phone2}	☐	
Price	double	{res:itgen_fnx_price}	☐	
PriceList	string	{res:itgen_fnx_pricelist}	☐	
PrintTemplate	string	{res:itgen_fnx_printtemplate}	☐	
Project_2	string	{res:itgen_fnx_project_2}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Remarks	string(1024)	{res:itgen_fnx_remarks}	☐	
ReservedQuantity	string	{res:itgen_fnx_reservedquantity}	☐	
RoundOff	double	{res:itgen_fnx_roundoff}	☐	
Row Id	int32	{res:itgen_fnx_row id}	☐	
Sent	boolean	{res:itgen_fnx_sent}	☐	
StockPointCode_2	string	{res:itgen_fnx_stockpointcode_2}	☐	
StockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
StockPointId_2	string	{res:itgen_fnx_stockpointid_2}	☐	
StockPointId	string	{res:itgen_fnx_stockpointid}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TaxReduction	double	{res:itgen_fnx_taxreduction}	☐	
TaxReductionType	string	{res:itgen_fnx_taxreductiontype}	☐	
TermsOfDelivery	string	{res:itgen_fnx_termsofdelivery}	☐	
TermsOfPayment	string	{res:itgen_fnx_termsofpayment}	☐	
TimeBasisReference	int32	{res:itgen_fnx_timebasisreference}	☐	
Total_2	double	{res:itgen_fnx_total_2}	☐	
Total	double	{res:itgen_fnx_total}	☐	
TotalToPay	double	{res:itgen_fnx_totaltopay}	☐	
TotalVAT	double	{res:itgen_fnx_totalvat}	☐	
Unit	string	{res:itgen_fnx_unit}	☐	
VAT	double	{res:itgen_fnx_vat}	☐	
VATCode	string	{res:itgen_fnx_vatcode}	☐	
VATIncluded	boolean	{res:itgen_fnx_vatincluded}	☐	
WarehouseReady	boolean	{res:itgen_fnx_warehouseready}	☐	
WayOfDelivery	string	{res:itgen_fnx_wayofdelivery}	☐	
YourOrderNumber	string(75)	{res:itgen_fnx_yourordernumber}	☐	
YourReference	string(50)	{res:itgen_fnx_yourreference}	☐	
ZipCode	string	{res:itgen_fnx_zipcode}	☐	

Orders

Retrieve a list of orders

Catalog: Fortnox

Schema: Orders

Primary Keys: DocumentNumber

Label: {res:itgen_fnx_orders_plural}

Retrieve: true

Insert: true

Update: true

Delete: false

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Orders. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
costcenter	string	☐		filter by cost center
customername	string	☐		filter by customer name
customernumber	string	☐		filter by customer number
documentnumber	string	☐		filter by document number
externalinvoicereference1	string	☐		filter by external invoice reference 1
externalinvoicereference2	string	☐		filter by external invoice reference 2
filter	string	☐		possibility to filter orders (Values: cancelled, expired, invoicecreated, invoicenotcreated)
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	string	☐		filter by from date
label	string	☐		filter by label
lastmodified	string	☐		filter by lastmodified
notcompleted	boolean	☐		filter by not completed
ordertype	string	☐		filter by order type
ourreference	string	☐		filter by ourreference
project	string	☐		filter by project
sent	boolean	☐		filter by sent
todate	string	☐		filter by to date
yourreference	string	☐		filter by your reference

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the table function Orders are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
Cancelled	boolean	{res:itgen_fnx_cancelled}	☐	
Currency	string(3)	{res:itgen_fnx_currency}	☐	
CustomerName	string	{res:itgen_fnx_customername}	☐	
CustomerNumber	string	{res:itgen_fnx_customernumber}	☐	
DeliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	
DocumentNumber	string	{res:itgen_fnx_documentnumber}	☐	
ExternalInvoiceReference1	string	{res:itgen_fnx_externalinvoicereference1}	☐	
ExternalInvoiceReference2	string	{res:itgen_fnx_externalinvoicereference2}	☐	
OrderDate	datetime	{res:itgen_fnx_orderdate}	☐	
OrderType	string	{res:itgen_fnx_ordertype}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Sent	boolean	{res:itgen_fnx_sent}	☐	
Total	double	{res:itgen_fnx_total}	☐	

3.1.41 Schema: PredefinedAccounts

3.1.41.1 Tables

PredefinedAccountByName

Retrieve information for a specific account type

Catalog: Fortnox

Schema: PredefinedAccounts

Label: {res:itgen_fnx_predefinedaccountbyname_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PredefinedAccountByName. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
name	string	☑		identifies the predefined account
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PredefinedAccountByName are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
PreDefinedAccount__url	string	{res:itgen_fnx_predefinedaccount__url}	☐	
PreDefinedAccount_Account	int32	{res:itgen_fnx_predefinedaccount_account}	☐	
PreDefinedAccount_Name	string	{res:itgen_fnx_predefinedaccount_name}	☐	

PredefinedAccounts

Retrieve a list of all predefined accounts

Catalog: Fortnox

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: PredefinedAccounts

Primary Keys: name

Label: {res:itgen_fnx_predefinedaccounts_plural}

Retrieve: true

Insert: false

Update: true

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PredefinedAccounts. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the table function PredefinedAccounts are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during update.

Name	Data Type	Label	Required	Documentation
Account	int32	{res:itgen_fnx_account}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Name	string	{res:itgen_fnx_name}	☐	

3.1.42 Schema: PredefinedVoucherSeries

3.1.42.1 Tables

PredefinedVoucherSeries

Retrieve a list of predefined voucher series

Catalog: Fortnox

Schema: PredefinedVoucherSeries

Primary Keys: Name

Label: {res:itgen_fnx_predefinedvoucherseries_plural}

Retrieve: true

Insert: false

Update: true

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PredefinedVoucherSeries. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PredefinedVoucherSeries are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
Name	string	{res:itgen_fnx_name}	☐	
VoucherSeries	string(1)	{res:itgen_fnx_voucherseries}	☐	

PredefinedVoucherSeriesByName

Retrieve a specific predefined voucher series

Catalog: Fortnox

Schema: PredefinedVoucherSeries

Label: {res:itgen_fnx_predefinedvoucherseriesbyname_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PredefinedVoucherSeriesByName. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
Name	string	☑		identifies the predefined voucher series
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PredefinedVoucherSeriesByName are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
PreDefinedVoucherSeries__url	string	{res:itgen_fnx_predefinedvoucherseries__url}	☐	
PreDefinedVoucherSeries_Name	string	{res:itgen_fnx_predefinedvoucherseries_name}	☐	
PreDefinedVoucherSeries_VoucherSeries	string(1)	{res:itgen_fnx_predefinedvoucherseries_voucherseries}	☐	

3.1.43 Schema: PriceLists

3.1.43.1 Tables

PriceListByCode

Retrieve a single price list

Catalog: Fortnox

Schema: PriceLists

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: {res:itgen_fnx_pricelistbycode_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PriceListByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
Code	string	☑		identifies the price list
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PriceListByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
PriceList__url	string	{res:itgen_fnx_pricelist__url}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
PriceList_Code	string(100)	{res:itgen_fnx_pricelist_code}	☐	
PriceList_Comments	string	{res:itgen_fnx_pricelist_comments}	☐	
PriceList_Description	string(50)	{res:itgen_fnx_pricelist_description}	☐	
PriceList_PreSelected	boolean	{res:itgen_fnx_pricelist_preselected}	☐	

PriceLists

Retrieve a list of price listsThe price lists register can return a list of records or a single record. By specifying a Code in the URL, a single record will be returned. Not specifying a Code will return a list of records.

Catalog: Fortnox

Schema: PriceLists

Primary Keys: Code

Label: {res:itgen_fnx_pricelists_plural}

Retrieve: true

Insert: true

Update: true

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PriceLists. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PriceLists are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
Code	string(100)	{res:itgen_fnx_code}	☐	
Comments	string	{res:itgen_fnx_comments}	☐	
Description	string(50)	{res:itgen_fnx_description}	☐	
PreSelected	boolean	{res:itgen_fnx_preselected}	☐	

3.1.44 Schema: Prices

3.1.44.1 Tables

ArticleFirstPriceByPriceListAndArticleNumber

Retrieve the first price for the specified article

Catalog: Fortnox

Schema: Prices

Label: {res:itgen_fnx_articlefirstpricebypricelistandarticlenumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ArticleFirstPriceByPriceListAndArticleNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
ArticleNumber	string	☑		identifies the article
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
PriceList	string	☑		identifies the price list
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ArticleFirstPriceByPriceListAndArticleNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Price__url	string	{res:itgen_fnx_price__url}	☐	
Price_ArticleNumber	string	{res:itgen_fnx_price_articlenumber}	☐	
Price_Date	datetime	{res:itgen_fnx_price_date}	☐	
Price_FromQuantity	double	{res:itgen_fnx_price_fromquantity}	☐	
Price_Percent	double	{res:itgen_fnx_price_percent}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Price_Price	double	{res:itgen_fnx_price_price}	☐	
Price_PriceList	string	{res:itgen_fnx_price_pricelist}	☐	

ArticlePriceByPriceListAndArticleNumberAndQuantity

Retrieve a price for a specified article

Catalog: Fortnox

Schema: Prices

Label: {res:itgen_fnx_articlepricebypricelistandarticlenumberandquantity_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ArticlePriceByPriceListAndArticleNumberAndQuantity. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
ArticleNumber	string	☑		identifies the article
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
FromQuantity	double	☑		identifies from quantity
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
PriceList	string	☑		identifies the price list
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ArticlePriceByPriceListAndArticleNumberAndQuantity are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Price__url	string	{res:itgen_fnx_price__url}	☐	
Price_ArticleNumber	string	{res:itgen_fnx_price_articlenumber}	☐	
Price_Date	datetime	{res:itgen_fnx_price_date}	☐	
Price_FromQuantity	double	{res:itgen_fnx_price_fromquantity}	☐	
Price_Percent	double	{res:itgen_fnx_price_percent}	☐	
Price_Price	double	{res:itgen_fnx_price_price}	☐	
Price_PriceList	string	{res:itgen_fnx_price_pricelist}	☐	

Prices

Retrieve a list of prices

Catalog: Fortnox

Schema: Prices

Label: {res:itgen_fnx_prices_plural}

Retrieve: true

Insert: true

Update: false

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Prices. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Prices are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
Price__url	string	{res:itgen_fnx_price__url}	☐	
Price_ArticleNumber	string	{res:itgen_fnx_price_articlenumber}	☐	
Price_Date	datetime	{res:itgen_fnx_price_date}	☐	
Price_FromQuantity	double	{res:itgen_fnx_price_fromquantity}	☐	
Price_Percent	double	{res:itgen_fnx_price_percent}	☐	
Price_Price	double	{res:itgen_fnx_price_price}	☐	
Price_PriceList	string	{res:itgen_fnx_price_pricelist}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Prices2

Retrieve a list of articles with all their prices in the specified price listThe list contains a slimmer version of the prices. To get a full entity, use the GET with a price list, article number and from quantity.

Catalog: Fortnox

Schema: Prices

Label: {res:itgen_fnx_prices_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Prices2. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
ArticleNumber	string	☑		identifies the article number of the prices
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
PriceList	string	☑		identifies the price list of the prices
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

The columns of the table function Prices2 are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ArticleNumber	string	{res:itgen_fnx_articlenumber}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
FromQuantity	double	{res:itgen_fnx_fromquantity}	☐	
Price	double	{res:itgen_fnx_price}	☐	
PriceList	string	{res:itgen_fnx_pricelist}	☐	

3.1.45 Schema: PrintTemplates

3.1.45.1 Tables

PrintTemplates

Retrieve a list of print templates

Catalog: Fortnox

Schema: PrintTemplates

Label: {res:itgen_fnx_printtemplates_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PrintTemplates. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PrintTemplates are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Name	string(25)	{res:itgen_fnx_name}	☐	
Template	string	{res:itgen_fnx_template}	☐	

3.1.46 Schema: Projects

3.1.46.1 Tables

ProjectByProjectNumber

Retrieve a single project

Catalog: Fortnox

Schema: Projects

Label: {res:itgen_fnx_projectbyprojectnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProjectByProjectNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
ProjectNumber	int32	☑		identifies the project
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ProjectByProjectNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Project__url	string	{res:itgen_fnx_project__url}	☐	
Project_Comments	string(512)	{res:itgen_fnx_project_comments}	☐	
Project_ContactPerson	string(50)	{res:itgen_fnx_project_contactperson}	☐	
Project_Description	string(50)	{res:itgen_fnx_project_description}	☐	
Project_EndDate	datetime	{res:itgen_fnx_project_enddate}	☐	
Project_ProjectLeader	string(50)	{res:itgen_fnx_project_projectleader}	☐	
Project_ProjectNumber	string(20)	{res:itgen_fnx_project_projectnumber}	☐	
Project_StartDate	datetime	{res:itgen_fnx_project_startdate}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Project_Status	string	{res:itgen_fnx_project_status}	☐	

Projects

Retrieve a list of projectsThe project register can return a list of records or a single record. By specifying a ProjectNumber in the URL, a single record will be returned. If no ProjectNumber is provided, a list of records will be returned.

Catalog: Fortnox

Schema: Projects

Primary Keys: ProjectNumber

Label: {res:itgen_fnx_projects_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Projects. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers,

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function Projects are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
Description	string(50)	{res:itgen_fnx_description}	☐	
EndDate	datetime	{res:itgen_fnx_enddate}	☐	
ProjectLeader	string(50)	{res:itgen_fnx_projectleader}	☐	
ProjectNumber	string(20)	{res:itgen_fnx_projectnumber}	☐	
StartDate	datetime	{res:itgen_fnx_startdate}	☐	
Status	string	{res:itgen_fnx_status}	☐	

3.1.47 Schema: SalaryTransactions

3.1.47.1 Tables

SalaryTransactionByRow

Retrieve a single salary transaction

Catalog: Fortnox

Schema: SalaryTransactions

Label: {res:itgen_fnx_salarytransactionbyrow_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SalaryTransactionByRow. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
SalaryRow	int32	☑		identifies the salary transaction
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SalaryTransactionByRow are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
SalaryTransaction_Amount	string	{res:itgen_fnx_salarytransaction_amount}	☐	
SalaryTransaction_CostCenter	string	{res:itgen_fnx_salarytransaction_costcenter}	☐	
SalaryTransaction_Date	datetime	{res:itgen_fnx_salarytransaction_date}	☐	
SalaryTransaction_EmployeeId	string	{res:itgen_fnx_salarytransaction_employeeid}	☐	
SalaryTransaction_Expense	string(6)	{res:itgen_fnx_salarytransaction_expense}	☐	
SalaryTransaction_Number	string	{res:itgen_fnx_salarytransaction_number}	☐	
SalaryTransaction_Project	string	{res:itgen_fnx_salarytransaction_project}	☐	
SalaryTransaction_SalaryCode	string	{res:itgen_fnx_salarytransaction_salarycode}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SalaryTransaction_SalaryRow	int32	{res:itgen_fnx_salarytransaction_salaryrow}	☐	
SalaryTransaction_TextRow	string(40)	{res:itgen_fnx_salarytransaction_textrow}	☐	
SalaryTransaction_Total	string	{res:itgen_fnx_salarytransaction_total}	☐	
SalaryTransaction_VAT	string	{res:itgen_fnx_salarytransaction_vat}	☐	

SalaryTransactions

List all salary transactions for all employeesSupports query-string parameters employeeid and date for filtering the result.

Catalog: Fortnox

Schema: SalaryTransactions

Primary Keys: SalaryRow

Label: {res:itgen_fnx_salarytransactions_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SalaryTransactions. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
date	datetime	☐		filter on date
employeeId	string	☐		filter on employeeId
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SalaryTransactions are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
Amount	string	{res:itgen_fnx_amount}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Date	datetime	{res:itgen_fnx_date}	☐	
EmployeeId	string	{res:itgen_fnx_employeeid}	☐	
Expense	string(6)	{res:itgen_fnx_expense}	☐	
Number	string	{res:itgen_fnx_number}	☐	
Project	string	{res:itgen_fnx_project}	☐	
SalaryCode	string	{res:itgen_fnx_salarycode}	☐	
SalaryRow	int32	{res:itgen_fnx_salaryrow}	☐	
TextRow	string(40)	{res:itgen_fnx_textrow}	☐	
Total	string	{res:itgen_fnx_total}	☐	
VAT	string	{res:itgen_fnx_vat}	☐	

3.1.48 Schema: ScheduleTimes

3.1.48.1 Tables

ScheduleTimeByEmployeeIdAndDate

Retrieve a specific schedule time

Catalog: Fortnox

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: ScheduleTimes

Label: {res:itgen_fnx_scheduletimebyemployeeidanddate_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ScheduleTimeByEmployeeIdAndDate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
Date	datetime	☑		identifies the date
EmployeeId	string	☑		identifies the employee
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ScheduleTimeByEmployeeIdAndDate are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ScheduleTime_Date	datetime	{res:itgen_fnx_scheduletime_date}	☐	
ScheduleTime_EmployeeId	string	{res:itgen_fnx_scheduletime_employeeid}	☐	
ScheduleTime_Hours	string	{res:itgen_fnx_scheduletime_hours}	☐	
ScheduleTime_IWH1	string	{res:itgen_fnx_scheduletime_iwh1}	☐	
ScheduleTime_IWH2	string	{res:itgen_fnx_scheduletime_iwh2}	☐	
ScheduleTime_IWH3	string	{res:itgen_fnx_scheduletime_iwh3}	☐	
ScheduleTime_IWH4	string	{res:itgen_fnx_scheduletime_iwh4}	☐	
ScheduleTime_IWH5	string	{res:itgen_fnx_scheduletime_iwh5}	☐	
ScheduleTime_ScheduleId	string	{res:itgen_fnx_scheduletime_scheduleid}	☐	

3.1.49 Schema: Sie

3.1.49.1 Tables

SieFileByType

Retrieve a SIE fileRetrieves a SIE file as streamed content

Catalog: Fortnox

Schema: Sie

Label: {res:itgen_fnx_siefilebytype_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SieFileByType. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialYear	int32	☐		financialYear
financialyear1	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
Type	string	☑		type

Columns of Table Function

The columns of the table function SieFileByType are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string	{res:itgen_fnx_result}	☐	Outcome of operation as single plain text column.

3.1.50 Schema: SupplierInvoiceAccruals

3.1.50.1 Tables

SupplierInvoiceAccrualByInvoiceNumber

Retrieve a single supplier invoice accrual

Catalog: Fortnox

Schema: SupplierInvoiceAccruals

Label: {res:itgen_fnx_supplierinvoiceaccrualbyinvoicenumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierInvoiceAccrualByInvoiceNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
SupplierInvoiceNumber	int32	☑		identifies the supplier invoice accrual
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SupplierInvoiceAccrualByInvoiceNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
SupplierInvoiceAccrual__url	string	{res:itgen_fnx_supplierinvoiceaccrual__url}	☐	
SupplierInvoiceAccrual_AccrualAccount	int32	{res:itgen_fnx_supplierinvoiceaccrual_accrualaccount}	☐	
SupplierInvoiceAccrual_CostAccount	int32	{res:itgen_fnx_supplierinvoiceaccrual_costaccount}	☐	
SupplierInvoiceAccrual_Description	string	{res:itgen_fnx_supplierinvoiceaccrual_description}	☐	
SupplierInvoiceAccrual_EndDate	datetime	{res:itgen_fnx_supplierinvoiceaccrual_endda	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		te}		
SupplierInvoiceAccrual_Period	string	{res:itgen_fnx_supplierinvoiceaccrual_period}	☐	
SupplierInvoiceAccrual_StartDate	datetime	{res:itgen_fnx_supplierinvoiceaccrual_startdate}	☐	
SupplierInvoiceAccrual_SupplierInvoiceNumber	int32	{res:itgen_fnx_supplierinvoiceaccrual_supplierinvoicenumber}	☐	
SupplierInvoiceAccrual_Times	int32	{res:itgen_fnx_supplierinvoiceaccrual_times}	☐	
SupplierInvoiceAccrual_Total	double	{res:itgen_fnx_supplierinvoiceaccrual_total}	☐	
SupplierInvoiceAccrual_VATIncluded	boolean	{res:itgen_fnx_supplierinvoiceaccrual_vatincluded}	☐	

SupplierInvoiceAccrualLinesByInvoiceNumber

Retrieve a single supplier invoice accrual

Catalog: Fortnox

Schema: SupplierInvoiceAccruals

Label: {res:itgen_fnx_supplierinvoiceaccruallinesbyinvoicenumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierInvoiceAccrualLinesByInvoiceNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
SupplierInvoiceNumber	int32	☑		identifies the supplier invoice accrual
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SupplierInvoiceAccrualLinesByInvoiceNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Account	int32	{res:itgen_fnx_account}	☐	
AccrualAccount	int32	{res:itgen_fnx_accrualaccount}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
CostAccount	int32	{res:itgen_fnx_costaccount}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Credit	double	{res:itgen_fnx_credit}	☐	
Debit	double	{res:itgen_fnx_debit}	☐	
Description	string	{res:itgen_fnx_description}	☐	
EndDate	datetime	{res:itgen_fnx_enddate}	☐	
Period	string	{res:itgen_fnx_period}	☐	
Project	string	{res:itgen_fnx_project}	☐	
StartDate	datetime	{res:itgen_fnx_startdate}	☐	
SupplierInvoiceNumber	int32	{res:itgen_fnx_supplierinvoicenumber}	☐	
Times	int32	{res:itgen_fnx_times}	☐	
Total	double	{res:itgen_fnx_total}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TransactionInformation	string	{res:itgen_fnx_transactioninformation}	☐	
VATIncluded	boolean	{res:itgen_fnx_vatincluded}	☐	

SupplierInvoiceAccruals

Retrieve a list of supplier invoice accrualsThe supplier invoice accruals register can return a list of records or a single record. By specifying a SupplierInvoiceNumber in the URL, a single record will be returned. Not specifying a SupplierInvoiceNumber will return a list of records.

Catalog: Fortnox

Schema: SupplierInvoiceAccruals

Primary Keys: SupplierInvoiceNumber

Label: {res:itgen_fnx_supplierinvoiceaccruals_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierInvoiceAccruals. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SupplierInvoiceAccruals are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
Description	string	{res:itgen_fnx_description}	☐	
Period	string	{res:itgen_fnx_period}	☐	
SupplierInvoiceNumber	int32	{res:itgen_fnx_supplierinvoicenumber}	☐	

3.1.51 Schema: SupplierInvoiceExternalUrlConnections

3.1.51.1 Tables

SupplierInvoiceExternalUrlConnectionById

Retrieve a single supplier invoice external URL connection

Catalog: Fortnox

Schema: SupplierInvoiceExternalUrlConnections

Label: {res:itgen_fnx_supplierinvoiceexternalurlconnectionbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierInvoiceExternalUrlConnectionById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
Id	int32	☑		id
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SupplierInvoiceExternalUrlConnectionById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
SupplierInvoiceExternalURLConnection_ExternalURLConnection	string	{res:itgen_fnx_supplierinvoiceexternalurlconnection_externalurlconnection}	☐	
SupplierInvoiceExternalURLConnection_Id	int32	{res:itgen_fnx_supplierinvoiceexternalurlconnection_id}	☐	
SupplierInvoiceExternalURLConnection_SupplierInvoiceNumber	int32	{res:itgen_fnx_supplierinvoiceexternalurlconnection_supplierinvoicenumber}	☐	
SupplierInvoiceExternalURLConnection_Url	string	{res:itgen_fnx_supplierinvoiceexternalurlconnection_url}	☐	

3.1.52 Schema: SupplierInvoiceFileConnections

3.1.52.1 Tables

SupplierInvoiceFileConnectionByFileId

Retrieve a single supplier invoice file connection

Catalog: Fortnox

Schema: SupplierInvoiceFileConnections

Label: {res:itgen_fnx_supplierinvoicefileconnectionbyfileid_plural}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierInvoiceFileConnectionByFileId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
FileId	string	☑		identifies the file connection
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SupplierInvoiceFileConnectionByFileId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
SupplierInvoiceFileConnection__url	string	{res:itgen_fnx_supplierinvoicefileconnection__url}	☐	
SupplierInvoiceFileConnection_FileId	string	{res:itgen_fnx_supplierinvoicefileconnection	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		_fileid}		
SupplierInvoiceFileConnection_Name	string	{res:itgen_fnx_supplierinvoicefileconnection_name}	☐	
SupplierInvoiceFileConnection_SupplierInvoiceNumber	string	{res:itgen_fnx_supplierinvoicefileconnection_supplierinvoicenumber}	☐	
SupplierInvoiceFileConnection_SupplierName	string	{res:itgen_fnx_supplierinvoicefileconnection_suppliername}	☐	

SupplierInvoiceFileConnections

Retrieve a list of supplier invoice file connectionsThe supplier invoice file connections register can return a list of records or a single record. By specifying a FileId in the URL, a single record will be returned. Not specifying a FileId will return a list of records.

Catalog: Fortnox

Schema: SupplierInvoiceFileConnections

Primary Keys: FileId

Label: {res:itgen_fnx_supplierinvoicefileconnections_plural}

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierInvoiceFileConnections. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SupplierInvoiceFileConnections are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
FileId	string	{res:itgen_fnx_fileid}	☐	
Name	string	{res:itgen_fnx_name}	☐	
SupplierInvoiceNumber	string	{res:itgen_fnx_supplierinvoicenumber}	☐	
SupplierName	string	{res:itgen_fnx_suppliername}	☐	

3.1.53 Schema: SupplierInvoicePayments

3.1.53.1 Tables

SupplierInvoicePaymentByInvoicePaymentNumber

Retrieve a single supplier invoice payment

Catalog: Fortnox

Schema: SupplierInvoicePayments

Label: {res:itgen_fnx_supplierinvoicepaymentbyinvoicepaymentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierInvoicePaymentByInvoicePaymentNumber. A value must be provided at all times for

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
Number	int32	☑		identifies the supplier invoice payment
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SupplierInvoicePaymentByInvoicePaymentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
SupplierInvoicePayment__url	string	{res:itgen_fnx_supplierinvoicepayment__url}	☐	
SupplierInvoicePayment_Amount	double	{res:itgen_fnx_supplierinvoicepayment_amount}	☐	
SupplierInvoicePayment_AmountCurrency	double	{res:itgen_fnx_supplierinvoicepayment_amountcurrency}	☐	
SupplierInvoicePayment_Booked	boolean	{res:itgen_fnx_supplierinvoicepayment_booked}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SupplierInvoicePayment_Currency	string(3)	{res:itgen_fnx_supplierinvoicepayment_currency}	☐	
SupplierInvoicePayment_CurrencyRate	double	{res:itgen_fnx_supplierinvoicepayment_currencyrate}	☐	
SupplierInvoicePayment_CurrencyUnit	double	{res:itgen_fnx_supplierinvoicepayment_currencyunit}	☐	
SupplierInvoicePayment_Information	string	{res:itgen_fnx_supplierinvoicepayment_information}	☐	
SupplierInvoicePayment_InvoiceDueDate	datetime	{res:itgen_fnx_supplierinvoicepayment_invoiceduedate}	☐	
SupplierInvoicePayment_InvoiceNumber	string	{res:itgen_fnx_supplierinvoicepayment_invoicenumber}	☐	
SupplierInvoicePayment_InvoiceOCR	string	{res:itgen_fnx_supplierinvoicepayment_invoiceocr}	☐	
SupplierInvoicePayment_InvoiceSupplierName	string	{res:itgen_fnx_supplierinvoicepayment_invoicesuppliername}	☐	
SupplierInvoicePayment_InvoiceSupplierNumber	string	{res:itgen_fnx_supplierinvoicepayment_invoicesuppliernumber}	☐	
SupplierInvoicePayment_InvoiceTotal	string	{res:itgen_fnx_supplierinvoicepayment_invoicetotal}	☐	
SupplierInvoicePayment_ModeOfPayment	string	{res:itgen_fnx_supplierinvoicepayment_modeofpayment}	☐	
SupplierInvoicePayment_Number	int32	{res:itgen_fnx_supplierinvoicepayment_number}	☐	
SupplierInvoicePayment_PaymentDate	datetime	{res:itgen_fnx_supplierinvoicepayment_paymentdate}	☐	
SupplierInvoicePayment_Source	string	{res:itgen_fnx_supplierinvoicepayment_source}	☐	
SupplierInvoicePayment_VoucherNumber	int32	{res:itgen_fnx_supplierinvoicepayment_vouchernumber}	☐	
SupplierInvoicePayment_VoucherSeries	string	{res:itgen_fnx_supplierinvoicepayment_voucherseries}	☐	
SupplierInvoicePayment_VoucherYear	int32	{res:itgen_fnx_supplierinvoicepayment_voucheryear}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

SupplierInvoicePayments

Retrieve a list of supplier invoice payments

Catalog: Fortnox

Schema: SupplierInvoicePayments

Primary Keys: Number

Label: {res:itgen_fnx_supplierinvoicepayments_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierInvoicePayments. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SupplierInvoicePayments are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Amount	double	{res:itgen_fnx_amount}	☐	
Booked	boolean	{res:itgen_fnx_booked}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Currency	string(3)	{res:itgen_fnx_currency}	☐	
CurrencyRate	double	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	double	{res:itgen_fnx_currencyunit}	☐	
InvoiceNumber	string	{res:itgen_fnx_invoicenumber}	☐	
Number	int32	{res:itgen_fnx_number}	☐	
PaymentDate	datetime	{res:itgen_fnx_paymentdate}	☐	
Source	string	{res:itgen_fnx_source}	☐	
WriteOffExist	boolean	{res:itgen_fnx_writeoffexist}	☐	

SupplierInvoicePaymentWriteOffsByInvoicePaymentNumber

Retrieve a single supplier invoice payment

Catalog: Fortnox

Schema: SupplierInvoicePayments

Label: {res:itgen_fnx_supplierinvoicepaymentwriteoffsbyinvoicepaymentnumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierInvoicePaymentWriteOffsByInvoicePaymentNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
Number	int32	☑		identifies the supplier invoice payment
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SupplierInvoicePaymentWriteOffsByInvoicePaymentNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AccountNumber	int32	{res:itgen_fnx_accountnumber}	☐	
Amount_2	double	{res:itgen_fnx_amount_2}	☐	
Amount	double	{res:itgen_fnx_amount}	☐	
AmountCurrency	double	{res:itgen_fnx_amountcurrency}	☐	
Booked	boolean	{res:itgen_fnx_booked}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Currency_2	string(3)	{res:itgen_fnx_currency_2}	☐	
Currency	string(3)	{res:itgen_fnx_currency}	☐	
CurrencyRate	double	{res:itgen_fnx_currencyrate}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CurrencyUnit	double	{res:itgen_fnx_currencyunit}	☐	
Description	string	{res:itgen_fnx_description}	☐	
Information	string	{res:itgen_fnx_information}	☐	
InvoiceDueDate	datetime	{res:itgen_fnx_invoiceduedate}	☐	
InvoiceNumber	string	{res:itgen_fnx_invoicenumber}	☐	
InvoiceOCR	string	{res:itgen_fnx_invoiceocr}	☐	
InvoiceSupplierName	string	{res:itgen_fnx_invoicesuppliername}	☐	
InvoiceSupplierNumber	string	{res:itgen_fnx_invoicesuppliernumber}	☐	
InvoiceTotal	string	{res:itgen_fnx_invoicetotal}	☐	
ModeOfPayment	string	{res:itgen_fnx_modeofpayment}	☐	
Number	int32	{res:itgen_fnx_number}	☐	
PaymentDate	datetime	{res:itgen_fnx_paymentdate}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Source	string	{res:itgen_fnx_source}	☐	
TransactionInformation	string	{res:itgen_fnx_transactioninformation}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
VoucherYear	int32	{res:itgen_fnx_voucheryear}	☐	

3.1.54 Schema: SupplierInvoices

3.1.54.1 Tables

SupplierInvoiceByGivenNumber

Retrieve a single supplier invoice

Catalog: Fortnox

Schema: SupplierInvoices

Label: {res:itgen_fnx_supplierinvoicebygivennumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierInvoiceByGivenNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
GivenNumber	int32	☑		identifies the invoice
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SupplierInvoiceByGivenNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
SupplierInvoice__url	string	{res:itgen_fnx_supplierinvoice__url}	☐	
SupplierInvoice_AccountingMethod	string	{res:itgen_fnx_supplierinvoice_accountingmethod}	☐	
SupplierInvoice_AdministrationFee	string	{res:itgen_fnx_supplierinvoice_administrationfee}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SupplierInvoice_Balance	string	{res:itgen_fnx_supplierinvoice_balance}	☐	
SupplierInvoice_Booked	boolean	{res:itgen_fnx_supplierinvoice_booked}	☐	
SupplierInvoice_Cancelled	boolean	{res:itgen_fnx_supplierinvoice_cancelled}	☐	
SupplierInvoice_Comments	string(1000)	{res:itgen_fnx_supplierinvoice_comments}	☐	
SupplierInvoice_CostCenter	string	{res:itgen_fnx_supplierinvoice_costcenter}	☐	
SupplierInvoice_Credit	boolean	{res:itgen_fnx_supplierinvoice_credit}	☐	
SupplierInvoice_CreditReference	int32	{res:itgen_fnx_supplierinvoice_creditreference}	☐	
SupplierInvoice_Currency	string	{res:itgen_fnx_supplierinvoice_currency}	☐	
SupplierInvoice_CurrencyRate	string	{res:itgen_fnx_supplierinvoice_currencyrate}	☐	
SupplierInvoice_CurrencyUnit	double	{res:itgen_fnx_supplierinvoice_currencyunit}	☐	
SupplierInvoice_DisablePaymentFile	boolean	{res:itgen_fnx_supplierinvoice_disablepaymentfile}	☐	
SupplierInvoice_DueDate	datetime	{res:itgen_fnx_supplierinvoice_duedate}	☐	
SupplierInvoice_ExternalInvoiceNumber	string	{res:itgen_fnx_supplierinvoice_externalinvoicenumber}	☐	
SupplierInvoice_ExternalInvoiceSeries	string	{res:itgen_fnx_supplierinvoice_externalinvoiceseries}	☐	
SupplierInvoice_FinalPayDate	datetime	{res:itgen_fnx_supplierinvoice_finalpaydate}	☐	
SupplierInvoice_Freight	string	{res:itgen_fnx_supplierinvoice_freight}	☐	
SupplierInvoice_GivenNumber	string	{res:itgen_fnx_supplierinvoice_givennumber}	☐	
SupplierInvoice_InvoiceDate	datetime	{res:itgen_fnx_supplierinvoice_invoicedate}	☐	
SupplierInvoice_InvoiceNumber	string(64)	{res:itgen_fnx_supplierinvoice_invoicenumber}	☐	
SupplierInvoice_OCR	string	{res:itgen_fnx_supplierinvoice_ocr}	☐	
SupplierInvoice_OurReference	string(50)	{res:itgen_fnx_supplierinvoice_ourreference}	☐	
SupplierInvoice_PaymentPending	boolean	{res:itgen_fnx_supplierinvoice_paymentpending}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SupplierInvoice_Project	string	{res:itgen_fnx_supplierinvoice_project}	☐	
SupplierInvoice_RoundOffValue	string	{res:itgen_fnx_supplierinvoice_roundoffvalue}	☐	
SupplierInvoice_SalesType	string	{res:itgen_fnx_supplierinvoice_salestype}	☐	
SupplierInvoice_SupplierName	string	{res:itgen_fnx_supplierinvoice_suppliername}	☐	
SupplierInvoice_SupplierNumber	string	{res:itgen_fnx_supplierinvoice_suppliernumber}	☐	
SupplierInvoice_Total	string	{res:itgen_fnx_supplierinvoice_total}	☐	
SupplierInvoice_VAT	string	{res:itgen_fnx_supplierinvoice_vat}	☐	
SupplierInvoice_VATType	string	{res:itgen_fnx_supplierinvoice_vattype}	☐	
SupplierInvoice_VoucherNumber	int32	{res:itgen_fnx_supplierinvoice_vouchernumber}	☐	
SupplierInvoice_VoucherSeries	string	{res:itgen_fnx_supplierinvoice_voucherseries}	☐	
SupplierInvoice_VoucherYear	int32	{res:itgen_fnx_supplierinvoice_voucheryear}	☐	
SupplierInvoice_YourReference	string(50)	{res:itgen_fnx_supplierinvoice_yourreference}	☐	

SupplierInvoiceLinesByGivenNumber

Retrieve a single supplier invoice

Catalog: Fortnox

Schema: SupplierInvoices

Label: {res:itgen_fnx_supplierinvoicelinesbygivennumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierInvoiceLinesByGivenNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
GivenNumber	int32	☑		identifies the invoice
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SupplierInvoiceLinesByGivenNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Account	int32	{res:itgen_fnx_account}	☐	
AccountDescription	string	{res:itgen_fnx_accountdescription}	☐	
AccountingMethod	string	{res:itgen_fnx_accountingmethod}	☐	
AdministrationFee	string	{res:itgen_fnx_administrationfee}	☐	
ArticleNumber	string	{res:itgen_fnx_articlenumber}	☐	
Balance	string	{res:itgen_fnx_balance}	☐	
Booked	boolean	{res:itgen_fnx_booked}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Cancelled	boolean	{res:itgen_fnx_cancelled}	☐	
Code	string	{res:itgen_fnx_code}	☐	
Comments	string(1000)	{res:itgen_fnx_comments}	☐	
CostCenter_2	string	{res:itgen_fnx_costcenter_2}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Credit_2	boolean	{res:itgen_fnx_credit_2}	☐	
Credit	double	{res:itgen_fnx_credit}	☐	
CreditCurrency	double	{res:itgen_fnx_creditcurrency}	☐	
CreditReference	int32	{res:itgen_fnx_creditreference}	☐	
Currency	string	{res:itgen_fnx_currency}	☐	
CurrencyRate	string	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	double	{res:itgen_fnx_currencyunit}	☐	
Debit	double	{res:itgen_fnx_debit}	☐	
DebitCurrency	double	{res:itgen_fnx_debitcurrency}	☐	
DisablePaymentFile	boolean	{res:itgen_fnx_disablepaymentfile}	☐	
DueDate	datetime	{res:itgen_fnx_duedate}	☐	
ExternalInvoiceNumber	string	{res:itgen_fnx_externalinvoicenumber}	☐	
ExternalInvoiceSeries	string	{res:itgen_fnx_externalinvoiceseries}	☐	
FinalPayDate	datetime	{res:itgen_fnx_finalpaydate}	☐	
Freight	string	{res:itgen_fnx_freight}	☐	
GivenNumber	string	{res:itgen_fnx_givennumber}	☐	
InvoiceDate	datetime	{res:itgen_fnx_invoicedate}	☐	
InvoiceNumber	string(64)	{res:itgen_fnx_invoicenumber}	☐	
ItemDescription	string	{res:itgen_fnx_itemdescription}	☐	
OCR	string	{res:itgen_fnx_ocr}	☐	
OurReference	string(50)	{res:itgen_fnx_ourreference}	☐	
PaymentPending	boolean	{res:itgen_fnx_paymentpending}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Price	double	{res:itgen_fnx_price}	☐	
Project_2	string	{res:itgen_fnx_project_2}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Quantity	int32	{res:itgen_fnx_quantity}	☐	
RoundOffValue	string	{res:itgen_fnx_roundoffvalue}	☐	
SalesType	string	{res:itgen_fnx_salestype}	☐	
StockLocationCode	string	{res:itgen_fnx_stocklocationcode}	☐	
StockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
SupplierName	string	{res:itgen_fnx_suppliername}	☐	
SupplierNumber	string	{res:itgen_fnx_suppliernumber}	☐	
Total_2	string	{res:itgen_fnx_total_2}	☐	
Total	double	{res:itgen_fnx_total}	☐	
TransactionInformation	string(100)	{res:itgen_fnx_transactioninformation}	☐	
Unit	string	{res:itgen_fnx_unit}	☐	
VAT	string	{res:itgen_fnx_vat}	☐	
VATType	string	{res:itgen_fnx_vattype}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
VoucherYear	int32	{res:itgen_fnx_voucheryear}	☐	
YourReference	string(50)	{res:itgen_fnx_yourreference}	☐	

SupplierInvoices

Retrieve a list of supplier invoices

Catalog: Fortnox

Schema: SupplierInvoices

Primary Keys: GivenNumber

Label: {res:itgen_fnx_supplierinvoices_plural}

Retrieve: true

Insert: true

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Update: true

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierInvoices. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
filter	string	☐		possibility to filter supplier invoices (Values: cancelled, fullypaid, unpaid, unpaidoverdue, unbooked, pendingpayment, authorizepending)
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SupplierInvoices are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
AuthorizerName	string	{res:itgen_fnx_authorizername}	☐	
Balance	string	{res:itgen_fnx_balance}	☐	
Booked	boolean	{res:itgen_fnx_booked}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Cancel	boolean	{res:itgen_fnx_cancel}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Credit	boolean	{res:itgen_fnx_credit}	☐	
Currency	string	{res:itgen_fnx_currency}	☐	
CurrencyRate	string	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	double	{res:itgen_fnx_currencyunit}	☐	
DueDate	datetime	{res:itgen_fnx_duedate}	☐	
ExternalInvoiceNumber	string	{res:itgen_fnx_externalinvoicenumber}	☐	
ExternalInvoiceSeries	string	{res:itgen_fnx_externalinvoiceseries}	☐	
FinalPayDate	datetime	{res:itgen_fnx_finalpaydate}	☐	
GivenNumber	string	{res:itgen_fnx_givennumber}	☐	
InvoiceDate	datetime	{res:itgen_fnx_invoicedate}	☐	
InvoiceNumber	string	{res:itgen_fnx_invoicenumber}	☐	
Project	string	{res:itgen_fnx_project}	☐	
SupplierName	string	{res:itgen_fnx_suppliername}	☐	
SupplierNumber	string	{res:itgen_fnx_suppliernumber}	☐	
Total	string	{res:itgen_fnx_total}	☐	

SupplierInvoiceVouchers

Retrieve a list of supplier invoices

Catalog: Fortnox

Schema: SupplierInvoices

Label: {res:itgen_fnx_supplierinvoicevouchers_plural}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierInvoiceVouchers. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
filter	string	☐		possibility to filter supplier invoices (Values: cancelled, fullypaid, unpaid, unpaidoverdue, unbooked, pendingpayment, authorizepending)
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SupplierInvoiceVouchers are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
AuthorizerName	string	{res:itgen_fnx_authorizername}	☐	
Balance	string	{res:itgen_fnx_balance}	☐	
Booked	boolean	{res:itgen_fnx_booked}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Cancel	boolean	{res:itgen_fnx_cancel}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Credit	boolean	{res:itgen_fnx_credit}	☐	
Currency	string	{res:itgen_fnx_currency}	☐	
CurrencyRate	string	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	double	{res:itgen_fnx_currencyunit}	☐	
DueDate	datetime	{res:itgen_fnx_duedate}	☐	
ExternalInvoiceNumber	string	{res:itgen_fnx_externalinvoicenumber}	☐	
ExternalInvoiceSeries	string	{res:itgen_fnx_externalinvoiceseries}	☐	
FinalPayDate	datetime	{res:itgen_fnx_finalpaydate}	☐	
GivenNumber	string	{res:itgen_fnx_givennumber}	☐	
InvoiceDate	datetime	{res:itgen_fnx_invoicedate}	☐	
InvoiceNumber	string	{res:itgen_fnx_invoicenumber}	☐	
Number	int32	{res:itgen_fnx_number}	☐	
Project	string	{res:itgen_fnx_project}	☐	
ReferenceType	string	{res:itgen_fnx_referencetype}	☐	
Series	string	{res:itgen_fnx_series}	☐	
SupplierName	string	{res:itgen_fnx_suppliername}	☐	
SupplierNumber	string	{res:itgen_fnx_suppliernumber}	☐	
Total	string	{res:itgen_fnx_total}	☐	
Year	int32	{res:itgen_fnx_year}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

SupplierInvoiceVouchersByGivenNumber

Retrieve a single supplier invoice

Catalog: Fortnox

Schema: SupplierInvoices

Label: {res:itgen_fnx_supplierinvoicevouchersbygivennumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierInvoiceVouchersByGivenNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
GivenNumber	int32	☑		identifies the invoice
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the table function SupplierInvoiceVouchersByGivenNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AccountingMethod	string	{res:itgen_fnx_accountingmethod}	☐	
AdministrationFee	string	{res:itgen_fnx_administrationfee}	☐	
Balance	string	{res:itgen_fnx_balance}	☐	
Booked	boolean	{res:itgen_fnx_booked}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Cancelled	boolean	{res:itgen_fnx_cancelled}	☐	
Comments	string(1000)	{res:itgen_fnx_comments}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Credit	boolean	{res:itgen_fnx_credit}	☐	
CreditReference	int32	{res:itgen_fnx_creditreference}	☐	
Currency	string	{res:itgen_fnx_currency}	☐	
CurrencyRate	string	{res:itgen_fnx_currencyrate}	☐	
CurrencyUnit	double	{res:itgen_fnx_currencyunit}	☐	
DisablePaymentFile	boolean	{res:itgen_fnx_disablepaymentfile}	☐	
DueDate	datetime	{res:itgen_fnx_duedate}	☐	
ExternalInvoiceNumber	string	{res:itgen_fnx_externalinvoicenumber}	☐	
ExternalInvoiceSeries	string	{res:itgen_fnx_externalinvoiceseries}	☐	
FinalPayDate	datetime	{res:itgen_fnx_finalpaydate}	☐	
Freight	string	{res:itgen_fnx_freight}	☐	
GivenNumber	string	{res:itgen_fnx_givennumber}	☐	
InvoiceDate	datetime	{res:itgen_fnx_invoicedate}	☐	
InvoiceNumber	string(64)	{res:itgen_fnx_invoicenumber}	☐	
Number	int32	{res:itgen_fnx_number}	☐	
OCR	string	{res:itgen_fnx_ocr}	☐	
OurReference	string(50)	{res:itgen_fnx_ourreference}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
PaymentPending	boolean	{res:itgen_fnx_paymentpending}	☐	
Project	string	{res:itgen_fnx_project}	☐	
ReferenceType	string	{res:itgen_fnx_referencetype}	☐	
RoundOffValue	string	{res:itgen_fnx_roundoffvalue}	☐	
SalesType	string	{res:itgen_fnx_salestype}	☐	
Series	string	{res:itgen_fnx_series}	☐	
SupplierName	string	{res:itgen_fnx_suppliername}	☐	
SupplierNumber	string	{res:itgen_fnx_suppliernumber}	☐	
Total	string	{res:itgen_fnx_total}	☐	
VAT	string	{res:itgen_fnx_vat}	☐	
VATType	string	{res:itgen_fnx_vattype}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
VoucherYear	int32	{res:itgen_fnx_voucheryear}	☐	
Year	int32	{res:itgen_fnx_year}	☐	
YourReference	string(50)	{res:itgen_fnx_yourreference}	☐	

3.1.55 Schema: Suppliers

3.1.55.1 Tables

SupplierByNumber

Retrieve a single supplier

Catalog: Fortnox

Schema: Suppliers

Label: {res:itgen_fnx_supplierbynumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SupplierByNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
SupplierNumber	string	☑		identifies the supplier
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function SupplierByNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Supplier__url	string	{res:itgen_fnx_supplier__url}	☐	
Supplier_Active	boolean	{res:itgen_fnx_supplier_active}	☐	
Supplier_Address1	string	{res:itgen_fnx_supplier_address1}	☐	
Supplier_Address2	string	{res:itgen_fnx_supplier_address2}	☐	
Supplier_Bank	string	{res:itgen_fnx_supplier_bank}	☐	
Supplier_BankAccountNumber	string	{res:itgen_fnx_supplier_bankaccountnumber}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Supplier_BG	string	{res:itgen_fnx_supplier_bg}	☐	
Supplier_BIC	string	{res:itgen_fnx_supplier_bic}	☐	
Supplier_BranchCode	string	{res:itgen_fnx_supplier_branchcode}	☐	
Supplier_City	string(1024)	{res:itgen_fnx_supplier_city}	☐	
Supplier_ClearingNumber	string	{res:itgen_fnx_supplier_clearingnumber}	☐	
Supplier_Comments	string(1024)	{res:itgen_fnx_supplier_comments}	☐	
Supplier_CostCenter	string	{res:itgen_fnx_supplier_costcenter}	☐	
Supplier_Country	string	{res:itgen_fnx_supplier_country}	☐	
Supplier_CountryCode	string(2)	{res:itgen_fnx_supplier_countrycode}	☐	
Supplier_Currency	string(3)	{res:itgen_fnx_supplier_currency}	☐	
Supplier_DisablePaymentFile	boolean	{res:itgen_fnx_supplier_disablepaymentfile}	☐	
Supplier_Email	string	{res:itgen_fnx_supplier_email}	☐	
Supplier_Fax	string	{res:itgen_fnx_supplier_fax}	☐	
Supplier_IBAN	string	{res:itgen_fnx_supplier_iban}	☐	
Supplier_Name	string(1024)	{res:itgen_fnx_supplier_name}	☐	
Supplier_OrganisationNumber	string	{res:itgen_fnx_supplier_organisationnumber}	☐	
Supplier_OurCustomerNumber	string	{res:itgen_fnx_supplier_ourcustomernumber}	☐	
Supplier_OurReference	string	{res:itgen_fnx_supplier_ourreference}	☐	
Supplier_PG	string	{res:itgen_fnx_supplier_pg}	☐	
Supplier_Phone1	string(1024)	{res:itgen_fnx_supplier_phone1}	☐	
Supplier_Phone2	string(1024)	{res:itgen_fnx_supplier_phone2}	☐	
Supplier_PreDefinedAccount	string(4)	{res:itgen_fnx_supplier_predefinedaccount}	☐	
Supplier_Project	string	{res:itgen_fnx_supplier_project}	☐	
Supplier_SupplierNumber	string	{res:itgen_fnx_supplier_suppliernumber}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Supplier_TermsOfPayment	string	{res:itgen_fnx_supplier_termsofpayment}	☐	
Supplier_VATNumber	string	{res:itgen_fnx_supplier_vatnumber}	☐	
Supplier_VATType	string	{res:itgen_fnx_supplier_vattype}	☐	
Supplier_VisitingAddress	string	{res:itgen_fnx_supplier_visitingaddress}	☐	
Supplier_VisitingCity	string	{res:itgen_fnx_supplier_visitingcity}	☐	
Supplier_VisitingCountry	string	{res:itgen_fnx_supplier_visitingcountry}	☐	
Supplier_VisitingCountryCode	string	{res:itgen_fnx_supplier_visitingcountrycode}	☐	
Supplier_VisitingZipCode	string	{res:itgen_fnx_supplier_visitingzipcode}	☐	
Supplier_WorkPlace	string	{res:itgen_fnx_supplier_workplace}	☐	
Supplier_WWW	string	{res:itgen_fnx_supplier_www}	☐	
Supplier_YourReference	string	{res:itgen_fnx_supplier_yourreference}	☐	
Supplier_ZipCode	string	{res:itgen_fnx_supplier_zipcode}	☐	

Suppliers

Retrieve a list of suppliersThe supplier register can return a list of records or a single record. By specifying a SupplierNumber in the URL, a single record will be returned. Not specifying a SupplierNumber will return a list of records.

Catalog: Fortnox

Schema: Suppliers

Primary Keys: SupplierNumber

Label: {res:itgen_fnx_suppliers_plural}

Retrieve: true

Insert: true

Update: true

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Suppliers. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Suppliers are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
Active	boolean	{res:itgen_fnx_active}	☐	
Address1	string	{res:itgen_fnx_address1}	☐	
Address2	string	{res:itgen_fnx_address2}	☐	
BankAccountNumber	string	{res:itgen_fnx_bankaccountnumber}	☐	
BG	string	{res:itgen_fnx_bg}	☐	
BIC	string	{res:itgen_fnx_bic}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
City	string(1024)	{res:itgen_fnx_city}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CountryCode	string(2)	{res:itgen_fnx_countrycode}	☐	
Currency	string(3)	{res:itgen_fnx_currency}	☐	
DisablePaymentFile	boolean	{res:itgen_fnx_disablepaymentfile}	☐	
Email	string	{res:itgen_fnx_email}	☐	
IBAN	string	{res:itgen_fnx_iban}	☐	
Name	string(1024)	{res:itgen_fnx_name}	☐	
OrganisationNumber	string	{res:itgen_fnx_organisationnumber}	☐	
PG	string	{res:itgen_fnx_pg}	☐	
Phone	string	{res:itgen_fnx_phone}	☐	
PreDefinedAccount	string(4)	{res:itgen_fnx_predefinedaccount}	☐	
Project	string	{res:itgen_fnx_project}	☐	
SupplierNumber	string	{res:itgen_fnx_suppliernumber}	☐	
TermsOfPayment	string	{res:itgen_fnx_termsofpayment}	☐	
ZipCode	string	{res:itgen_fnx_zipcode}	☐	

3.1.56 Schema: TaxReductions

3.1.56.1 Tables

TaxReductionAmountsByTaxReductionId

Retrieve a single tax reduction

Catalog: Fortnox

Schema: TaxReductions

Label: {res:itgen_fnx_taxreductionamountsbytaxreductionid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TaxReductionAmountsByTaxReductionId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
Id	int32	☑		identifies the tax reduction
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function TaxReductionAmountsByTaxReductionId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ApprovedAmount	double	{res:itgen_fnx_approvedamount}	☐	
AskedAmount_2	double	{res:itgen_fnx_askedamount_2}	☐	
AskedAmount	double	{res:itgen_fnx_askedamount}	☐	
BilledAmount	double	{res:itgen_fnx_billedamount}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
CustomerName	string(2147483647)	{res:itgen_fnx_customername}	☐	
Id	int32	{res:itgen_fnx_id}	☐	
PropertyDesignation	string	{res:itgen_fnx_propertydesignation}	☐	
ReferenceDocumentType	string	{res:itgen_fnx_referencedocumenttype}	☐	
ReferenceNumber	string	{res:itgen_fnx_referencenumber}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
RequestSent	boolean	{res:itgen_fnx_requestsent}	☐	
ResidenceAssociationOrganisationNumber	string	{res:itgen_fnx_residenceassociationorganisationnumber}	☐	
SocialSecurityNumber	string(13)	{res:itgen_fnx_socialsecuritynumber}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
VoucherYear	int32	{res:itgen_fnx_voucheryear}	☐	
WorkType	string	{res:itgen_fnx_worktype}	☐	

TaxReductionById

Retrieve a single tax reduction

Catalog: Fortnox

Schema: TaxReductions

Label: {res:itgen_fnx_taxreductionbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TaxReductionById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
Id	int32	☑		identifies the tax reduction
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function TaxReductionById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
TaxReduction__url	string	{res:itgen_fnx_taxreduction__url}	☐	
TaxReduction_ApprovedAmount	double	{res:itgen_fnx_taxreduction_approvedamount}	☐	
TaxReduction_AskedAmount	double	{res:itgen_fnx_taxreduction_askedamount}	☐	
TaxReduction_BilledAmount	double	{res:itgen_fnx_taxreduction_billedamount}	☐	
TaxReduction_CustomerName	string(2147483647)	{res:itgen_fnx_taxreduction_customername}	☐	
TaxReduction_Id	int32	{res:itgen_fnx_taxreduction_id}	☐	
TaxReduction_PropertyDesignation	string	{res:itgen_fnx_taxreduction_propertydesignation}	☐	
TaxReduction_ReferenceDocumentType	string	{res:itgen_fnx_taxreduction_referencedocumenttype}	☐	
TaxReduction_ReferenceNumber	string	{res:itgen_fnx_taxreduction_referencenumber}	☐	
TaxReduction_RequestSent	boolean	{res:itgen_fnx_taxreduction_requestsent}	☐	
TaxReduction_ResidenceAssociationOrganisationNumber	string	{res:itgen_fnx_taxreduction_residenceassociationorganisationnumber}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TaxReduction_SocialSecurityNumber	string(13)	{res:itgen_fnx_taxreduction_socialsecuritynumber}	☐	
TaxReduction_VoucherNumber	int32	{res:itgen_fnx_taxreduction_vouchernumber}	☐	
TaxReduction_VoucherSeries	string	{res:itgen_fnx_taxreduction_voucherseries}	☐	
TaxReduction_VoucherYear	int32	{res:itgen_fnx_taxreduction_voucheryear}	☐	

TaxReductions

Retrieve a list of tax reductions

Catalog: Fortnox

Schema: TaxReductions

Primary Keys: Id

Label: {res:itgen_fnx_taxreductions_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TaxReductions. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
filter	string	☐		possibility to filter tax reductions (Values: invoices, orders, offers)
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function TaxReductions are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
ApprovedAmount	double	{res:itgen_fnx_approvedamount}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
CustomerName	string(2147483647)	{res:itgen_fnx_customername}	☐	
Id	int32	{res:itgen_fnx_id}	☐	
ReferenceDocumentType	string	{res:itgen_fnx_referencedocumenttype}	☐	
ReferenceNumber	int32	{res:itgen_fnx_referencenumber}	☐	
SocialSecurityNumber	string(13)	{res:itgen_fnx_socialsecuritynumber}	☐	

3.1.57 Schema: TermsOfDeliveries

3.1.57.1 Tables

DeliveryTermByCode

Retrieve a single terms of delivery

Catalog: Fortnox

Schema: TermsOfDeliveries

Label: {res:itgen_fnx_deliverytermbycode_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeliveryTermByCode. A value must be provided at all times for required parameters, but

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
Code	string	☑		identifies the terms of delivery
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function DeliveryTermByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
TermsOfDelivery__url	string	{res:itgen_fnx_termsofdelivery__url}	☐	
TermsOfDelivery_Code	string(20)	{res:itgen_fnx_termsofdelivery_code}	☐	
TermsOfDelivery_Description	string(200)	{res:itgen_fnx_termsofdelivery_description}	☐	
TermsOfDelivery_DescriptionEnglish	string(100)	{res:itgen_fnx_termsofdelivery_descriptionenglish}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

DeliveryTerms

Retrieve a list of terms of deliveriesThe terms of deliveries register can return a list of records or a single record. By specifying a Code in the URL, a single record will be returned. Not specifying a Code will return a list of records.

Catalog: Fortnox

Schema: TermsOfDeliveries

Primary Keys: Code

Label: {res:itgen_fnx_deliveryterms_plural}

Retrieve: true

Insert: true

Update: true

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeliveryTerms. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

The columns of the table function DeliveryTerms are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
Code	string(20)	{res:itgen_fnx_code}	☐	
Description	string(200)	{res:itgen_fnx_description}	☐	
DescriptionEnglish	string(100)	{res:itgen_fnx_descriptionenglish}	☐	

3.1.58 Schema: TermsOfPayments

3.1.58.1 Tables

PaymentTermByCode

Retrieve a single terms of payment

Catalog: Fortnox

Schema: TermsOfPayments

Label: {res:itgen_fnx_paymenttermbycode_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PaymentTermByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
Code	string	☑		identifies the terms of payment
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PaymentTermByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
TermsOfPayment__url	string	{res:itgen_fnx_termsofpayment__url}	☐	
TermsOfPayment_Code	string(25)	{res:itgen_fnx_termsofpayment_code}	☐	
TermsOfPayment_Description	string	{res:itgen_fnx_termsofpayment_description}	☐	

PaymentTerms

Retrieve a list of all terms of payments

Catalog: Fortnox

Schema: TermsOfPayments

Primary Keys: Code

Label: {res:itgen_fnx_paymentterms_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PaymentTerms. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PaymentTerms are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
Code	string(25)	{res:itgen_fnx_code}	☐	
Description	string	{res:itgen_fnx_description}	☐	

3.1.59 Schema: TimeReporting_Articles

3.1.59.1 Tables

TimeReportingArticleRegistrations

Get full article registrations that match filterResponse property descriptions: id - The unique id of a basic common combination of article registrations. (The basic common combination means "user/purchase date/customer/project/cost center", which leads to a dialog with several article registrations.)   purchaseDate - The date on which the article is purchased or registered for charging.   ownerId - The user ID who creates the basic common combination.   version - The version of the basic common combination (article dialog) being updated, which is used for handling the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

concurrency issue.   registrationType - It is always "ARTICLE" for article list endpoint.   Sub-Class - ArticleRegistration: id - The unique id of an article registration.   registrationId - The id of the basic common combination.   orderIndex - the order index for the article registration in regard of the common combination.   ownerId - The user ID who owns the article registration.   totalQuantity - The quantity of the article.   unitPrice - The unit price connected to the article registration, which might be locked on an invoice/order basis or for non-invoiceable.   unitCost - The unit cost connected to the article registration, which might be locked on an invoice/order basis.   invoiceBasisId - The ID of invoice/order basis which is used for creating an invoice/order.   nonInvoiceable - If the article registration would be ignored for charging or not.   note - The note on the article registration.   documentId - The document ID which includes the article registration and is created in Invoicing application.   documentType - The document type which could be "invoice" or "order".

Catalog: Fortnox

Schema: TimeReporting_Articles

Label: {res:itgen_fnx_timereportingarticleregistrations_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TimeReportingArticleRegistrations. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromDate	datetime	☐		The start date of the search span, the max of which should be 1 year to the end date ("toDate").Example: 2022-11-01
fromdate1	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
includeNonInvoiceablePrice	boolean	☐		If the price of the non-invoiceable article registration is included, or not.
includeRegistrationsWithoutProject	boolean	☐		If the article registration without project is included, or not.
inInvoiceBasis	boolean	☐		If the article registration is locked on an invoice basis, or not.
internalArticles	boolean	☐		If the article registration is internal, which is registered on an internal customer, or not.
invoiced	boolean	☐		If a document is created with the article registration, or not.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
nonInvoiceable	boolean	☐		If the article registration has been moved to non-invoiceable, or not.
toDate	datetime	☐		The end date of the search span, the max of which should be 1 year back to the start date ("fromDate").Example: 2022-11-30
todate1	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function TimeReportingArticleRegistrations are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
costCenter_active	string	{res:itgen_fnx_costcenter_active}	☐	
costCenter_id	string(40)	{res:itgen_fnx_costcenter_id}	☐	
costCenter_name	string(255)	{res:itgen_fnx_costcenter_name}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
createdBy	string	{res:itgen_fnx_createdby}	☐	
createdTime_2	datetime	{res:itgen_fnx_createdtime_2}	☐	
createdTime	datetime	{res:itgen_fnx_createdtime}	☐	
customer_accountManager	string	{res:itgen_fnx_customer_accountmanager}	☐	
customer_active	boolean	{res:itgen_fnx_customer_active}	☐	
customer_defaultProjectId	string	{res:itgen_fnx_customer_defaultprojectid}	☐	
customer_deleted	boolean	{res:itgen_fnx_customer_deleted}	☐	
customer_id	string(40)	{res:itgen_fnx_customer_id}	☐	
customer_invoiceCity	string	{res:itgen_fnx_customer_invoicecity}	☐	
customer_invoiceZipCode	string	{res:itgen_fnx_customer_invoicezipcode}	☐	
customer_isBusiness	int32	{res:itgen_fnx_customer_isbusiness}	☐	
customer_name	string(255)	{res:itgen_fnx_customer_name}	☐	
customer_number	string(1024)	{res:itgen_fnx_customer_number}	☐	
customer_priceList	string	{res:itgen_fnx_customer_pricelist}	☐	
customer_propertyDesignation	string	{res:itgen_fnx_customer_propertydesignation}	☐	
documentId	int64	{res:itgen_fnx_documentid}	☐	
documentType	string	{res:itgen_fnx_documenttype}	☐	
id_2	guid	{res:itgen_fnx_id_2}	☐	
id	guid	{res:itgen_fnx_id}	☐	
invoiceBasisId	int64	{res:itgen_fnx_invoicebasisid}	☐	
invoiceText	string(1000)	{res:itgen_fnx_invoicetext}	☐	
nonInvoiceable	boolean	{res:itgen_fnx_noninvoiceable}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
orderIndex	int32	{res:itgen_fnx_orderindex}	☐	
ownerId_2	string	{res:itgen_fnx_ownerid_2}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ownerId	string	{res:itgen_fnx_ownerid}	☐	
project_description	string(50)	{res:itgen_fnx_project_description}	☐	
project_endDate	datetime	{res:itgen_fnx_project_enddate}	☐	
project_id	string(40)	{res:itgen_fnx_project_id}	☐	
project_invoicedAmount	double	{res:itgen_fnx_project_invoicedamount}	☐	
project_orderAmount	double	{res:itgen_fnx_project_orderamount}	☐	
project_orderTime	double	{res:itgen_fnx_project_ordertime}	☐	
project_projectLeader	string	{res:itgen_fnx_project_projectleader}	☐	
project_startDate	datetime	{res:itgen_fnx_project_startdate}	☐	
project_status	int32	{res:itgen_fnx_project_status}	☐	
project_totalAmountInInvoices	double	{res:itgen_fnx_project_totalamountininvoices}	☐	
purchaseDate	datetime	{res:itgen_fnx_purchasedate}	☐	
registeredArticle_createdBy	string	{res:itgen_fnx_registeredarticle_createdby}	☐	
registeredArticle_createdTime	datetime	{res:itgen_fnx_registeredarticle_createdtime}	☐	
registeredArticle_documentId	int64	{res:itgen_fnx_registeredarticle_documentid}	☐	
registeredArticle_documentType	string	{res:itgen_fnx_registeredarticle_documenttype}	☐	
registeredArticle_id	guid	{res:itgen_fnx_registeredarticle_id}	☐	
registeredArticle_invoiceBasisId	int64	{res:itgen_fnx_registeredarticle_invoicebasisid}	☐	
registeredArticle_invoiceText	string(1000)	{res:itgen_fnx_registeredarticle_invoicetext}	☐	
registeredArticle_nonInvoiceable	boolean	{res:itgen_fnx_registeredarticle_noninvoiceable}	☐	
registeredArticle_note	string(1000)	{res:itgen_fnx_registeredarticle_note}	☐	
registeredArticle_orderIndex	int32	{res:itgen_fnx_registeredarticle_orderindex}	☐	
registeredArticle_ownerId	string	{res:itgen_fnx_registeredarticle_ownerid}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
registeredArticle_timeLocked	boolean	{res:itgen_fnx_registeredarticle_timelocked}	☐	
registeredArticle_totalQuantity	decimal	{res:itgen_fnx_registeredarticle_totalquantity}	☐	
registeredArticle_unitCost	decimal	{res:itgen_fnx_registeredarticle_unitcost}	☐	
registeredArticle_unitPrice	decimal	{res:itgen_fnx_registeredarticle_unitprice}	☐	
registrationType	string	{res:itgen_fnx_registrationtype}	☐	
timeLocked	boolean	{res:itgen_fnx_timelocked}	☐	
totalQuantity	decimal	{res:itgen_fnx_totalquantity}	☐	
unitCost	decimal	{res:itgen_fnx_unitcost}	☐	
unitPrice	decimal	{res:itgen_fnx_unitprice}	☐	
version	int64	{res:itgen_fnx_version}	☐	

TimeReportingArticleRegistrationSalesPrices

Get full article registrations that match filterResponse property descriptions: id - The unique id of a basic common combination of article registrations. (The basic common combination means "user/purchase date/customer/project/cost center", which leads to a dialog with several article registrations.)   purchaseDate - The date on which the article is purchased or registered for charging.   ownerId - The user ID who creates the basic common combination.   version - The version of the basic common combination (article dialog) being updated, which is used for handling the concurrency issue.   registrationType - It is always "ARTICLE" for article list endpoint.   Sub-Class - ArticleRegistration: id - The unique id of an article registration.   registrationId - The id of the basic common combination.   orderIndex - the order index for the article registration in regard of the common combination.   ownerId - The user ID who owns the article registration.   totalQuantity - The quantity of the article.   unitPrice - The unit price connected to the article registration, which might be locked on an invoice/order basis or for non-invoiceable.   unitCost - The unit cost connected to the article registration, which might be locked on an invoice/order basis.   invoiceBasisId - The ID of invoice/order basis which is used for creating an invoice/order.   nonInvoiceable - If the article registration would be ignored for charging or not.   note - The note on the article

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

registration.   documentId - The document ID which includes the article registration and is created in Invoicing application.   documentType - The document type which could be "invoice" or "order".

Catalog: Fortnox

Schema: TimeReporting_Articles

Label: {res:itgen_fnx_timereportingarticleregistrationsalesprices_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TimeReportingArticleRegistrationSalesPrices. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromDate	datetime	☐		The start date of the search span, the max of which should be 1 year to the end date ("toDate").Example: 2022-11-01
fromdate1	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
includeNonInvoiceablePrice	boolean	☐		If the price of the non-invoiceable article registration is included, or not.
includeRegistrationsWithoutProject	boolean	☐		If the article registration without project is included, or not.
inInvoiceBasis	boolean	☐		If the article registration is locked on an invoice basis, or not.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
internalArticles	boolean	☐		If the article registration is internal, which is registered on an internal customer, or not.
invoiced	boolean	☐		If a document is created with the article registration, or not.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
nonInvoiceable	boolean	☐		If the article registration has been moved to non-invoiceable, or not.
toDate	datetime	☐		The end date of the search span, the max of which should be 1 year back to the start date ("fromDate").Example: 2022-11-30
todate1	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function TimeReportingArticleRegistrationSalesPrices are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
costCenter_active	string	{res:itgen_fnx_costcenter_active}	☐	
costCenter_id	string(40)	{res:itgen_fnx_costcenter_id}	☐	
costCenter_name	string(255)	{res:itgen_fnx_costcenter_name}	☐	
createdBy	string	{res:itgen_fnx_createdby}	☐	
createdTime_2	datetime	{res:itgen_fnx_createdtime_2}	☐	
createdTime	datetime	{res:itgen_fnx_createdtime}	☐	
customer_accountManager	string	{res:itgen_fnx_customer_accountmanager}	☐	
customer_active	boolean	{res:itgen_fnx_customer_active}	☐	
customer_defaultProjectId	string	{res:itgen_fnx_customer_defaultprojectid}	☐	
customer_deleted	boolean	{res:itgen_fnx_customer_deleted}	☐	
customer_id	string(40)	{res:itgen_fnx_customer_id}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
customer_invoiceCity	string	{res:itgen_fnx_customer_invoicecity}	☐	
customer_invoiceZipCode	string	{res:itgen_fnx_customer_invoicezipcode}	☐	
customer_isBusiness	int32	{res:itgen_fnx_customer_isbusiness}	☐	
customer_name	string(255)	{res:itgen_fnx_customer_name}	☐	
customer_number	string(1024)	{res:itgen_fnx_customer_number}	☐	
customer_priceList	string	{res:itgen_fnx_customer_pricelist}	☐	
customer_propertyDesignation	string	{res:itgen_fnx_customer_propertydesignation}	☐	
documentId	int64	{res:itgen_fnx_documentid}	☐	
documentType	string	{res:itgen_fnx_documenttype}	☐	
id_2	guid	{res:itgen_fnx_id_2}	☐	
id	guid	{res:itgen_fnx_id}	☐	
invoiceBasisId	int64	{res:itgen_fnx_invoicebasisid}	☐	
invoiceText	string(1000)	{res:itgen_fnx_invoicetext}	☐	
nonInvoiceable	boolean	{res:itgen_fnx_noninvoiceable}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
orderIndex	int32	{res:itgen_fnx_orderindex}	☐	
ownerId_2	string	{res:itgen_fnx_ownerid_2}	☐	
ownerId	string	{res:itgen_fnx_ownerid}	☐	
project_description	string(50)	{res:itgen_fnx_project_description}	☐	
project_endDate	datetime	{res:itgen_fnx_project_enddate}	☐	
project_id	string(40)	{res:itgen_fnx_project_id}	☐	
project_invoicedAmount	double	{res:itgen_fnx_project_invoicedamount}	☐	
project_orderAmount	double	{res:itgen_fnx_project_orderamount}	☐	
project_orderTime	double	{res:itgen_fnx_project_ordertime}	☐	
project_projectLeader	string	{res:itgen_fnx_project_projectleader}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
project_startDate	datetime	{res:itgen_fnx_project_startdate}	☐	
project_status	int32	{res:itgen_fnx_project_status}	☐	
project_totalAmountInInvoices	double	{res:itgen_fnx_project_totalamountininvoices}	☐	
purchaseDate	datetime	{res:itgen_fnx_purchasedate}	☐	
registeredArticle_createdBy	string	{res:itgen_fnx_registeredarticle_createdby}	☐	
registeredArticle_createdTime	datetime	{res:itgen_fnx_registeredarticle_createdtime}	☐	
registeredArticle_documentId	int64	{res:itgen_fnx_registeredarticle_documentid}	☐	
registeredArticle_documentType	string	{res:itgen_fnx_registeredarticle_documenttype}	☐	
registeredArticle_id	guid	{res:itgen_fnx_registeredarticle_id}	☐	
registeredArticle_invoiceBasisId	int64	{res:itgen_fnx_registeredarticle_invoicebasisid}	☐	
registeredArticle_invoiceText	string(1000)	{res:itgen_fnx_registeredarticle_invoicetext}	☐	
registeredArticle_nonInvoiceable	boolean	{res:itgen_fnx_registeredarticle_noninvoiceable}	☐	
registeredArticle_note	string(1000)	{res:itgen_fnx_registeredarticle_note}	☐	
registeredArticle_orderIndex	int32	{res:itgen_fnx_registeredarticle_orderindex}	☐	
registeredArticle_ow nerId	string	{res:itgen_fnx_registeredarticle_ow nerid}	☐	
registeredArticle_timeLocked	boolean	{res:itgen_fnx_registeredarticle_timelocked}	☐	
registeredArticle_totalQuantity	decimal	{res:itgen_fnx_registeredarticle_totalquantity}	☐	
registeredArticle_unitCost	decimal	{res:itgen_fnx_registeredarticle_unitcost}	☐	
registeredArticle_unitPrice	decimal	{res:itgen_fnx_registeredarticle_unitprice}	☐	
registrationType	string	{res:itgen_fnx_registrationtype}	☐	
timeLocked	boolean	{res:itgen_fnx_timelocked}	☐	
totalQuantity	decimal	{res:itgen_fnx_totalquantity}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
unitCost	decimal	{res:itgen_fnx_unitcost}	☐	
unitPrice	decimal	{res:itgen_fnx_unitprice}	☐	
version	int64	{res:itgen_fnx_version}	☐	

TimeReportingArticles

Get full article registrations that match filterResponse property descriptions: id - The unique id of a basic common combination of article registrations. (The basic common combination means "user/purchase date/customer/project/cost center", which leads to a dialog with several article registrations.)   purchaseDate - The date on which the article is purchased or registered for charging.   ownerId - The user ID who creates the basic common combination.   version - The version of the basic common combination (article dialog) being updated, which is used for handling the concurrency issue.   registrationType - It is always "ARTICLE" for article list endpoint.   Sub-Class - ArticleRegistration: id - The unique id of an article registration.   registrationId - The id of the basic common combination.   orderIndex - the order index for the article registration in regard of the common combination.   ownerId - The user ID who owns the article registration.   totalQuantity - The quantity of the article.   unitPrice - The unit price connected to the article registration, which might be locked on an invoice/order basis or for non-invoiceable.   unitCost - The unit cost connected to the article registration, which might be locked on an invoice/order basis.   invoiceBasisId - The ID of invoice/order basis which is used for creating an invoice/order.   nonInvoiceable - If the article registration would be ignored for charging or not.   note - The note on the article registration.   documentId - The document ID which includes the article registration and is created in Invoicing application.   documentType - The document type which could be "invoice" or "order".

Catalog: Fortnox

Schema: TimeReporting_Articles

Label: {res:itgen_fnx_timereportingarticles_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function TimeReportingArticles. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromDate	datetime	☐		The start date of the search span, the max of which should be 1 year to the end date ("toDate").Example: 2022-11-01
fromdate1	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
includeNonInvoiceablePrice	boolean	☐		If the price of the non-invoiceable article registration is included, or not.
includeRegistrationsWithoutProject	boolean	☐		If the article registration without project is included, or not.
inInvoiceBasis	boolean	☐		If the article registration is locked on an invoice basis, or not.
internalArticles	boolean	☐		If the article registration is internal, which is registered on an internal customer, or not.
invoiced	boolean	☐		If a document is created with the article registration, or not.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
nonInvoiceable	boolean	☐		If the article registration has been moved to non-invoiceable, or not.
toDate	datetime	☐		The end date of the search span, the max of which should be 1 year back to the start date ("fromDate").Example: 2022-11-30

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
todate1	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function TimeReportingArticles are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
costCenter_active	string	{res:itgen_fnx_costcenter_active}	☐	
costCenter_id	string(40)	{res:itgen_fnx_costcenter_id}	☐	
costCenter_name	string(255)	{res:itgen_fnx_costcenter_name}	☐	
createdTime	datetime	{res:itgen_fnx_createdtime}	☐	
customer_accountManager	string	{res:itgen_fnx_customer_accountmanager}	☐	
customer_active	boolean	{res:itgen_fnx_customer_active}	☐	
customer_defaultProjectId	string	{res:itgen_fnx_customer_defaultprojectid}	☐	
customer_deleted	boolean	{res:itgen_fnx_customer_deleted}	☐	
customer_id	string(40)	{res:itgen_fnx_customer_id}	☐	
customer_invoiceCity	string	{res:itgen_fnx_customer_invoicecity}	☐	
customer_invoiceZipCode	string	{res:itgen_fnx_customer_invoicezipcode}	☐	
customer_isBusiness	int32	{res:itgen_fnx_customer_isbusiness}	☐	
customer_name	string(255)	{res:itgen_fnx_customer_name}	☐	
customer_number	string(1024)	{res:itgen_fnx_customer_number}	☐	
customer_priceList	string	{res:itgen_fnx_customer_pricelist}	☐	
customer_propertyDesignation	string	{res:itgen_fnx_customer_propertydesignation}	☐	
id	guid	{res:itgen_fnx_id}	☐	
ownerId	string	{res:itgen_fnx_ownerid}	☐	
project_description	string(50)	{res:itgen_fnx_project_description}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
project_endDate	datetime	{res:itgen_fnx_project_enddate}	☐	
project_id	string(40)	{res:itgen_fnx_project_id}	☐	
project_invoicedAmount	double	{res:itgen_fnx_project_invoicedamount}	☐	
project_orderAmount	double	{res:itgen_fnx_project_orderamount}	☐	
project_orderTime	double	{res:itgen_fnx_project_ordertime}	☐	
project_projectLeader	string	{res:itgen_fnx_project_projectleader}	☐	
project_startDate	datetime	{res:itgen_fnx_project_startdate}	☐	
project_status	int32	{res:itgen_fnx_project_status}	☐	
project_totalAmountInInvoices	double	{res:itgen_fnx_project_totalamountininvoices}	☐	
purchaseDate	datetime	{res:itgen_fnx_purchasedate}	☐	
registeredArticle_createdBy	string	{res:itgen_fnx_registeredarticle_createdby}	☐	
registeredArticle_createdTime	datetime	{res:itgen_fnx_registeredarticle_createdtime}	☐	
registeredArticle_documentId	int64	{res:itgen_fnx_registeredarticle_documentid}	☐	
registeredArticle_documentType	string	{res:itgen_fnx_registeredarticle_documenttype}	☐	
registeredArticle_id	guid	{res:itgen_fnx_registeredarticle_id}	☐	
registeredArticle_invoiceBasisId	int64	{res:itgen_fnx_registeredarticle_invoicebasisid}	☐	
registeredArticle_invoiceText	string(1000)	{res:itgen_fnx_registeredarticle_invoicetext}	☐	
registeredArticle_nonInvoiceable	boolean	{res:itgen_fnx_registeredarticle_noninvoiceable}	☐	
registeredArticle_note	string(1000)	{res:itgen_fnx_registeredarticle_note}	☐	
registeredArticle_orderIndex	int32	{res:itgen_fnx_registeredarticle_orderindex}	☐	
registeredArticle_ownerId	string	{res:itgen_fnx_registeredarticle_ownerid}	☐	
registeredArticle_timeLocked	boolean	{res:itgen_fnx_registeredarticle_timelocked}	☐	
registeredArticle_totalQuantity	decimal	{res:itgen_fnx_registeredarticle_totalquantity	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		}		
registeredArticle_unitCost	decimal	{res:itgen_fnx_registeredarticle_unitcost}	☐	
registeredArticle_unitPrice	decimal	{res:itgen_fnx_registeredarticle_unitprice}	☐	
registrationType	string	{res:itgen_fnx_registrationtype}	☐	
version	int64	{res:itgen_fnx_version}	☐	

TimeReportingRegisteredArticleSalesPrices

Get full article registrations that match filterResponse property descriptions: id - The unique id of a basic common combination of article registrations. (The basic common combination means "user/purchase date/customer/project/cost center", which leads to a dialog with several article registrations.)   purchaseDate - The date on which the article is purchased or registered for charging.   ownerId - The user ID who creates the basic common combination.   version - The version of the basic common combination (article dialog) being updated, which is used for handling the concurrency issue.   registrationType - It is always "ARTICLE" for article list endpoint.   Sub-Class - ArticleRegistration: id - The unique id of an article registration.   registrationId - The id of the basic common combination.   orderIndex - the order index for the article registration in regard of the common combination.   ownerId - The user ID who owns the article registration.   totalQuantity - The quantity of the article.   unitPrice - The unit price connected to the article registration, which might be locked on an invoice/order basis or for non-invoiceable.   unitCost - The unit cost connected to the article registration, which might be locked on an invoice/order basis.   invoiceBasisId - The ID of invoice/order basis which is used for creating an invoice/order.   nonInvoiceable - If the article registration would be ignored for charging or not.   note - The note on the article registration.   documentId - The document ID which includes the article registration and is created in Invoicing application.   documentType - The document type which could be "invoice" or "order".

Catalog: Fortnox

Schema: TimeReporting_Articles

Label: {res:itgen_fnx_timereportingregisteredarticlesalesprices_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TimeReportingRegisteredArticleSalesPrices. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromDate	datetime	☐		The start date of the search span, the max of which should be 1 year to the end date ("toDate").Example: 2022-11-01
fromdate1	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
includeNonInvoiceablePrice	boolean	☐		If the price of the non-invoiceable article registration is included, or not.
includeRegistrationsWithoutProject	boolean	☐		If the article registration without project is included, or not.
inInvoiceBasis	boolean	☐		If the article registration is locked on an invoice basis, or not.
internalArticles	boolean	☐		If the article registration is internal, which is registered on an internal customer, or not.
invoiced	boolean	☐		If a document is created with the article registration, or not.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
nonInvoiceable	boolean	☐		If the article registration has been moved to non-invoiceable, or not.
toDate	datetime	☐		The end date of the search span, the max of which should be 1 year back to the start date

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				("fromDate").Example: 2022-11-30
todate1	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function TimeReportingRegisteredArticleSalesPrices are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
costCenter_active	string	{res:itgen_fnx_costcenter_active}	☐	
costCenter_id	string(40)	{res:itgen_fnx_costcenter_id}	☐	
costCenter_name	string(255)	{res:itgen_fnx_costcenter_name}	☐	
createdBy	string	{res:itgen_fnx_createdby}	☐	
createdTime_2	datetime	{res:itgen_fnx_createdtime_2}	☐	
createdTime	datetime	{res:itgen_fnx_createdtime}	☐	
customer_accountManager	string	{res:itgen_fnx_customer_accountmanager}	☐	
customer_active	boolean	{res:itgen_fnx_customer_active}	☐	
customer_defaultProjectId	string	{res:itgen_fnx_customer_defaultprojectid}	☐	
customer_deleted	boolean	{res:itgen_fnx_customer_deleted}	☐	
customer_id	string(40)	{res:itgen_fnx_customer_id}	☐	
customer_invoiceCity	string	{res:itgen_fnx_customer_invoicecity}	☐	
customer_invoiceZipCode	string	{res:itgen_fnx_customer_invoicezipcode}	☐	
customer_isBusiness	int32	{res:itgen_fnx_customer_isbusiness}	☐	
customer_name	string(255)	{res:itgen_fnx_customer_name}	☐	
customer_number	string(1024)	{res:itgen_fnx_customer_number}	☐	
customer_priceList	string	{res:itgen_fnx_customer_pricelist}	☐	
customer_propertyDesignation	string	{res:itgen_fnx_customer_propertydesignatio	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		n}		
documentId	int64	{res:itgen_fnx_documentid}	☐	
documentType	string	{res:itgen_fnx_documenttype}	☐	
id_2	guid	{res:itgen_fnx_id_2}	☐	
id	guid	{res:itgen_fnx_id}	☐	
invoiceBasisId	int64	{res:itgen_fnx_invoicebasisid}	☐	
invoiceText	string(1000)	{res:itgen_fnx_invoicetext}	☐	
nonInvoiceable	boolean	{res:itgen_fnx_noninvoiceable}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
orderIndex	int32	{res:itgen_fnx_orderindex}	☐	
ownerId_2	string	{res:itgen_fnx_ownerid_2}	☐	
ownerId	string	{res:itgen_fnx_ownerid}	☐	
project_description	string(50)	{res:itgen_fnx_project_description}	☐	
project_endDate	datetime	{res:itgen_fnx_project_enddate}	☐	
project_id	string(40)	{res:itgen_fnx_project_id}	☐	
project_invoicedAmount	double	{res:itgen_fnx_project_invoicedamount}	☐	
project_orderAmount	double	{res:itgen_fnx_project_orderamount}	☐	
project_orderTime	double	{res:itgen_fnx_project_ordertime}	☐	
project_projectLeader	string	{res:itgen_fnx_project_projectleader}	☐	
project_startDate	datetime	{res:itgen_fnx_project_startdate}	☐	
project_status	int32	{res:itgen_fnx_project_status}	☐	
project_totalAmountInInvoices	double	{res:itgen_fnx_project_totalamountininvoices}	☐	
purchaseDate	datetime	{res:itgen_fnx_purchasedate}	☐	
registrationType	string	{res:itgen_fnx_registrationtype}	☐	
timeLocked	boolean	{res:itgen_fnx_timelocked}	☐	
totalQuantity	decimal	{res:itgen_fnx_totalquantity}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
unitCost	decimal	{res:itgen_fnx_unitcost}	☐	
unitPrice	decimal	{res:itgen_fnx_unitprice}	☐	
version	int64	{res:itgen_fnx_version}	☐	

3.1.60 Schema: TimeReporting_Registrations

3.1.60.1 Tables

TimeReportingRegistrations

Get time/absence registrations that match filterResponse property descriptions: id - The unique id of the registration.   userId - The user ID who owns the registration.   workedDate - The date for which the registration is created.   workedHours - The time spent, or the time of absence.   startTime - The start of clock time.   stopTime - The end of clock time.   invoiceText - The text to be included in the invoice/order basis which would be used to create an invoice/order.   note - The note on the registration.   chargeHours - The time to be invoiced, or 0 for the absence, or locked for non-invoiceable.   childId - The child ID related to the absence registration of parental leave (FPE), which comes from Payroll application.   nonInvoiceable - If the registration would be ignored for charging or not.   invoiceBasisId - The ID of invoice/order basis which is used for creating an invoice/order.   documentId - The document ID which includes the registration and is created in Invoicing application.   documentType - The document type which could be "invoice" or "order".   unitCost - The unit cost from the registration owner who takes the work.   unitPrice - The unit price for the service on the registration, which comes in priority from "invoice/order basis", "price group" or "service".

Catalog: Fortnox

Schema: TimeReporting_Registrations

Label: {res:itgen_fnx_timereportingregistrations_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TimeReportingRegistrations. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromDate	datetime	☐		The start date of the search span, the max of which should be 1 year to the end date ("toDate").Example: 2022-11-01
fromdate1	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
includeNonInvoiceableChargeHours	boolean	☐		If the price of the non-invoiceable time/absence registration is included, or not.
includeRegistrationsWithoutProject	boolean	☐		If the time/absence registration without project is included, or not.
inInvoiceBasis	boolean	☐		If the time/absence registration is locked on an invoice basis, or not.
internalTime	boolean	☐		If the time/absence registration is internal, which is registered on an internal customer, or not.
invoiced	boolean	☐		If a document is created with the time/absence registration, or not.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
nonInvoiceable	boolean	☐		If the time/absence registration has been moved to non-invoiceable, or not.
toDate	datetime	☐		The end date of the search span, the max of which should be 1 year back to the start date ("fromDate").Example: 2022-11-30
todate1	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

The columns of the table function TimeReportingRegistrations are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
chargeHours	float	{res:itgen_fnx_chargehours}	☐	
childId	guid	{res:itgen_fnx_childid}	☐	
costCenter_active	string	{res:itgen_fnx_costcenter_active}	☐	
costCenter_id	string(40)	{res:itgen_fnx_costcenter_id}	☐	
costCenter_name	string(255)	{res:itgen_fnx_costcenter_name}	☐	
createdBy	string	{res:itgen_fnx_createdby}	☐	
createdTime	datetime	{res:itgen_fnx_createdtime}	☐	
customer_accountManager	string	{res:itgen_fnx_customer_accountmanager}	☐	
customer_active	boolean	{res:itgen_fnx_customer_active}	☐	
customer_defaultProjectId	string	{res:itgen_fnx_customer_defaultprojectid}	☐	
customer_deleted	boolean	{res:itgen_fnx_customer_deleted}	☐	
customer_id	string(40)	{res:itgen_fnx_customer_id}	☐	
customer_invoiceCity	string	{res:itgen_fnx_customer_invoicecity}	☐	
customer_invoiceZipCode	string	{res:itgen_fnx_customer_invoicezipcode}	☐	
customer_isBusiness	int32	{res:itgen_fnx_customer_isbusiness}	☐	
customer_name	string(255)	{res:itgen_fnx_customer_name}	☐	
customer_number	string(1024)	{res:itgen_fnx_customer_number}	☐	
customer_priceList	string	{res:itgen_fnx_customer_pricelist}	☐	
customer_propertyDesignation	string	{res:itgen_fnx_customer_propertydesignation}	☐	
documentId	int64	{res:itgen_fnx_documentid}	☐	
documentType	string	{res:itgen_fnx_documenttype}	☐	
id	guid	{res:itgen_fnx_id}	☐	
invoiceBasisId	int64	{res:itgen_fnx_invoicebasisid}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
invoiceText	string(1000)	{res:itgen_fnx_invoicetext}	☐	
nonInvoiceable	boolean	{res:itgen_fnx_noninvoiceable}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
project_description	string(50)	{res:itgen_fnx_project_description}	☐	
project_endDate	datetime	{res:itgen_fnx_project_enddate}	☐	
project_id	string(40)	{res:itgen_fnx_project_id}	☐	
project_invoicedAmount	double	{res:itgen_fnx_project_invoicedamount}	☐	
project_orderAmount	double	{res:itgen_fnx_project_orderamount}	☐	
project_orderTime	double	{res:itgen_fnx_project_ordertime}	☐	
project_projectLeader	string	{res:itgen_fnx_project_projectleader}	☐	
project_startDate	datetime	{res:itgen_fnx_project_startdate}	☐	
project_status	int32	{res:itgen_fnx_project_status}	☐	
project_totalAmountInInvoices	double	{res:itgen_fnx_project_totalamountininvoices}	☐	
registrationCode_active	boolean	{res:itgen_fnx_registrationcode_active}	☐	
registrationCode_code	string(3)	{res:itgen_fnx_registrationcode_code}	☐	
registrationCode_costMultiplier	float	{res:itgen_fnx_registrationcode_costmultiplier}	☐	
registrationCode_id	string	{res:itgen_fnx_registrationcode_id}	☐	
registrationCode_name	string	{res:itgen_fnx_registrationcode_name}	☐	
registrationCode_priceMultiplier	float	{res:itgen_fnx_registrationcode_pricemultiplier}	☐	
registrationCode_type	string	{res:itgen_fnx_registrationcode_type}	☐	
service_active	boolean	{res:itgen_fnx_service_active}	☐	
service_description	string(255)	{res:itgen_fnx_service_description}	☐	
service_id	string(40)	{res:itgen_fnx_service_id}	☐	
service_isStock	boolean	{res:itgen_fnx_service_isstock}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
service_purchasePrice	decimal	{res:itgen_fnx_service_purchaseprice}	☐	
service_type	string	{res:itgen_fnx_service_type}	☐	
service_unit	string	{res:itgen_fnx_service_unit}	☐	
startTime	datetime	{res:itgen_fnx_starttime}	☐	
stopTime	datetime	{res:itgen_fnx_stoptime}	☐	
unitCost	decimal	{res:itgen_fnx_unitcost}	☐	
unitPrice	decimal	{res:itgen_fnx_unitprice}	☐	
updatedBy	string	{res:itgen_fnx_updatedby}	☐	
userId	string	{res:itgen_fnx_userid}	☐	
w orkedDate	datetime	{res:itgen_fnx_w orkeddate}	☐	
w orkedHours	float	{res:itgen_fnx_w orkedhours}	☐	

TimeReportingRegistrationServiceSalesPrices

Get time/absence registrations that match filterResponse property descriptions: id - The unique id of the registration.   userId - The user ID who owns the registration.   workedDate - The date for which the registration is created.   workedHours - The time spent, or the time of absence.   startTime - The start of clock time.   stopTime - The end of clock time.   invoiceText - The text to be included in the invoice/order basis which would be used to create an invoice/order.   note - The note on the registration.   chargeHours - The time to be invoiced, or 0 for the absence, or locked for non-invoiceable.   childId - The child ID related to the absence registration of parental leave (FPE), which comes from Payroll application.   nonInvoiceable - If the registration would be ignored for charging or not.   invoiceBasisId - The ID of invoice/order basis which is used for creating an invoice/order.   documentId - The document ID which includes the registration and is created in Invoicing application.   documentType - The document type which could be "invoice" or "order".   unitCost - The unit cost from the registration owner who takes the work.   unitPrice - The unit price for the service on the registration, which comes in priority from "invoice/order basis", "price group" or "service".

Catalog: Fortnox

Schema: TimeReporting_Registrations

Label: {res:itgen_fnx_timereportingregistrationservicesalesprices_plural}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TimeReportingRegistrationServiceSalesPrices. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromDate	datetime	☐		The start date of the search span, the max of which should be 1 year to the end date ("toDate").Example: 2022-11-01
fromdate1	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
includeNonInvoiceableChargeHours	boolean	☐		If the price of the non-invoiceable time/absence registration is included, or not.
includeRegistrationsWithoutProject	boolean	☐		If the time/absence registration without project is included, or not.
inInvoiceBasis	boolean	☐		If the time/absence registration is locked on an invoice basis, or not.
internalTime	boolean	☐		If the time/absence registration is internal, which is registered on an internal customer, or not.
invoiced	boolean	☐		If a document is created with the time/absence registration, or not.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
nonInvoiceable	boolean	☐		If the time/absence registration has been moved to non-invoiceable, or not.
toDate	datetime	☐		The end date of the search span, the max of w hich should be 1 year back to the start date ("fromDate").Example: 2022-11-30
todate1	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function TimeReportingRegistrationServiceSalesPrices are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
active	boolean	{res:itgen_fnx_active}	☐	
chargeHours	float	{res:itgen_fnx_chargehours}	☐	
childId	guid	{res:itgen_fnx_childid}	☐	
costCenter_active	string	{res:itgen_fnx_costcenter_active}	☐	
costCenter_id	string(40)	{res:itgen_fnx_costcenter_id}	☐	
costCenter_name	string(255)	{res:itgen_fnx_costcenter_name}	☐	
createdBy	string	{res:itgen_fnx_createdby}	☐	
createdTime	datetime	{res:itgen_fnx_createdtime}	☐	
customer_accountManager	string	{res:itgen_fnx_customer_accountmanager}	☐	
customer_active	boolean	{res:itgen_fnx_customer_active}	☐	
customer_defaultProjectId	string	{res:itgen_fnx_customer_defaultprojectid}	☐	
customer_deleted	boolean	{res:itgen_fnx_customer_deleted}	☐	
customer_id	string(40)	{res:itgen_fnx_customer_id}	☐	
customer_invoiceCity	string	{res:itgen_fnx_customer_invoicecity}	☐	
customer_invoiceZipCode	string	{res:itgen_fnx_customer_invoicezipcode}	☐	
customer_isBusiness	int32	{res:itgen_fnx_customer_isbusiness}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
customer_name	string(255)	{res:itgen_fnx_customer_name}	☐	
customer_number	string(1024)	{res:itgen_fnx_customer_number}	☐	
customer_priceList	string	{res:itgen_fnx_customer_pricelist}	☐	
customer_propertyDesignation	string	{res:itgen_fnx_customer_propertydesignation}	☐	
description	string(255)	{res:itgen_fnx_description}	☐	
documentId	int64	{res:itgen_fnx_documentid}	☐	
documentType	string	{res:itgen_fnx_documenttype}	☐	
id_2	guid	{res:itgen_fnx_id_2}	☐	
id	string(40)	{res:itgen_fnx_id}	☐	
invoiceBasisId	int64	{res:itgen_fnx_invoicebasisid}	☐	
invoiceText	string(1000)	{res:itgen_fnx_invoicetext}	☐	
isStock	boolean	{res:itgen_fnx_isstock}	☐	
nonInvoiceable	boolean	{res:itgen_fnx_noninvoiceable}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
project_description	string(50)	{res:itgen_fnx_project_description}	☐	
project_endDate	datetime	{res:itgen_fnx_project_enddate}	☐	
project_id	string(40)	{res:itgen_fnx_project_id}	☐	
project_invoicedAmount	double	{res:itgen_fnx_project_invoicedamount}	☐	
project_orderAmount	double	{res:itgen_fnx_project_orderamount}	☐	
project_orderTime	double	{res:itgen_fnx_project_ordertime}	☐	
project_projectLeader	string	{res:itgen_fnx_project_projectleader}	☐	
project_startDate	datetime	{res:itgen_fnx_project_startdate}	☐	
project_status	int32	{res:itgen_fnx_project_status}	☐	
project_totalAmountInInvoices	double	{res:itgen_fnx_project_totalamountininvoices}	☐	
purchasePrice	decimal	{res:itgen_fnx_purchaseprice}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
registrationCode_active	boolean	{res:itgen_fnx_registrationcode_active}	☐	
registrationCode_code	string(3)	{res:itgen_fnx_registrationcode_code}	☐	
registrationCode_costMultiplier	float	{res:itgen_fnx_registrationcode_costmultiplier}	☐	
registrationCode_id	string	{res:itgen_fnx_registrationcode_id}	☐	
registrationCode_name	string	{res:itgen_fnx_registrationcode_name}	☐	
registrationCode_priceMultiplier	float	{res:itgen_fnx_registrationcode_pricemultiplier}	☐	
registrationCode_type	string	{res:itgen_fnx_registrationcode_type}	☐	
startTime	datetime	{res:itgen_fnx_starttime}	☐	
stopTime	datetime	{res:itgen_fnx_stoptime}	☐	
type	string	{res:itgen_fnx_type}	☐	
unit	string	{res:itgen_fnx_unit}	☐	
unitCost	decimal	{res:itgen_fnx_unitcost}	☐	
unitPrice	decimal	{res:itgen_fnx_unitprice}	☐	
updatedBy	string	{res:itgen_fnx_updatedby}	☐	
userId	string	{res:itgen_fnx_userid}	☐	
workedDate	datetime	{res:itgen_fnx_workeddate}	☐	
workedHours	float	{res:itgen_fnx_workedhours}	☐	

3.1.61 Schema: TrustedEmailSenders

3.1.61.1 Tables

RejectedSenders

Retrieve a list of all trusted and rejected senders

Catalog: Fortnox

Schema: TrustedEmailSenders

Label: {res:itgen_fnx_rejectedsenders_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function RejectedSenders. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function RejectedSenders are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Email	string	{res:itgen_fnx_email}	☐	
Id	int32	{res:itgen_fnx_id}	☐	

Senders

Retrieve a list of all trusted and rejected senders

Catalog: Fortnox

Schema: TrustedEmailSenders

Label: {res:itgen_fnx_senders_plural}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Senders. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Senders are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DUMMY	string(1)	{res:itgen_fnx_dummy}	☑	Default column added since the specification specifies that no data is returned.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

TrustedSenders

Retrieve a list of all trusted and rejected senders

Catalog: Fortnox

Schema: TrustedEmailSenders

Label: {res:itgen_fnx_trustedsenders_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TrustedSenders. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function TrustedSenders are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Email	string	{res:itgen_fnx_email}	☐	
Id	int32	{res:itgen_fnx_id}	☐	

3.1.62 Schema: Units

3.1.62.1 Tables

UnitByCode

Retrieve a single unit

Catalog: Fortnox

Schema: Units

Label: {res:itgen_fnx_unitbycode_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UnitByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
Code	string	☑		identifies the unit
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function UnitByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Unit__url	string	{res:itgen_fnx_unit__url}	☐	
Unit_Code	string(20)	{res:itgen_fnx_unit_code}	☐	
Unit_CodeEnglish	string(100)	{res:itgen_fnx_unit_codeenglish}	☐	
Unit_Description	string(100)	{res:itgen_fnx_unit_description}	☐	

Units

Retrieve a list of unitsThe units register can return a list of records or a single record. By specifying a Code in the URL, a single record will be returned. Not specifying a Code will return a list of records.

Catalog: Fortnox

Schema: Units

Primary Keys: Code

Label: {res:itgen_fnx_units_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Units. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Units are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
Code	string(20)	{res:itgen_fnx_code}	☐	
CodeEnglish	string(100)	{res:itgen_fnx_codeenglish}	☐	
Description	string(100)	{res:itgen_fnx_description}	☐	

3.1.63 Schema: VacationDebtBasis

3.1.63.1 Tables

VacationDebtBasisByYearAndMonth

Retrieve a specific vacation debt basis for a posted voucher

Catalog: Fortnox

Schema: VacationDebtBasis

Label: {res:itgen_fnx_vacationdebtbasisbyyearandmonth_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function VacationDebtBasisByYearAndMonth. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
Month	int32	☑		
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
Year	int32	☑		

Columns of Table Function

The columns of the table function VacationDebtBasisByYearAndMonth are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
VacationDebtBasis_LastDay	datetime	{res:itgen_fnx_vacationdebtbasis_lastday}	☐	
VacationDebtBasis_Month	int32	{res:itgen_fnx_vacationdebtbasis_month}	☐	
VacationDebtBasis_VoucherNumber	int32	{res:itgen_fnx_vacationdebtbasis_vouchernumber}	☐	
VacationDebtBasis_VoucherSeries	string	{res:itgen_fnx_vacationdebtbasis_vouchers	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		eries}		
VacationDebtBasis_VoucherYear	int32	{res:itgen_fnx_vacationdebtbasis_voucheryear}	☐	
VacationDebtBasis_Year	int32	{res:itgen_fnx_vacationdebtbasis_year}	☐	

VacationDebtBasisEmployeesByYearAndMonth

Retrieve a specific vacation debt basis for a posted voucher

Catalog: Fortnox

Schema: VacationDebtBasis

Label: {res:itgen_fnx_vacationdebtbasisemployeesbyyearandmonth_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function VacationDebtBasisEmployeesByYearAndMonth. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
Month	int32	☑		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
Year	int32	☑		

Columns of Table Function

The columns of the table function VacationDebtBasisEmployeesByYearAndMonth are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DaysEarned	double	{res:itgen_fnx_daysearned}	☐	
DaysSaved	double	{res:itgen_fnx_dayssaved}	☐	
DaysUnused	double	{res:itgen_fnx_daysunused}	☐	
DebtAdvance	double	{res:itgen_fnx_debtadvance}	☐	
DebtEarned	double	{res:itgen_fnx_debtearned}	☐	
DebtSaved	double	{res:itgen_fnx_debtsaved}	☐	
DebtUnused	double	{res:itgen_fnx_debtunused}	☐	
EmployeeId	string(15)	{res:itgen_fnx_employeeid}	☐	
EmployeeName	string	{res:itgen_fnx_employeename}	☐	
LastDay	datetime	{res:itgen_fnx_lastday}	☐	
Month	int32	{res:itgen_fnx_month}	☐	
TotalDebtEmployee	double	{res:itgen_fnx_totaldebtemployee}	☐	
TotalDebtEmployerContribution	double	{res:itgen_fnx_totaldebtemployercontribution}	☐	
VariableEarned	double	{res:itgen_fnx_variableearned}	☐	
VariableUnused	double	{res:itgen_fnx_variableunused}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
VoucherYear	int32	{res:itgen_fnx_voucheryear}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
WageEarned	double	{res:itgen_fnx_wageearned}	☐	
WageSaved	double	{res:itgen_fnx_wagesaved}	☐	
WageUnused	double	{res:itgen_fnx_wageunused}	☐	
Year	int32	{res:itgen_fnx_year}	☐	

3.1.64 Schema: Views

3.1.64.1 Views

AccountActiveBalances

Catalog: Fortnox

Schema: Views

Label: {res:itgen_fortnox_active_account_balances}

This is a read-only view. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

View Columns

The columns of the view AccountActiveBalances are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTACTIVE	boolean	{res:itgen_fnx_active}	☐	
ACCOUNTDESCRIPTION	string(200)	{res:itgen_fnx_description}	☐	
ACCOUNTNUMBER	int32	{res:itgen_fnx_number}	☐	
ACCOUNTYEAR	int32	{res:itgen_fnx_year}	☐	
BALANCEBROUGHTFORWARD	double	{res:itgen_fnx_balancebroughtforward}	☐	
BALANCECARRIEDFORWARD	double	{res:itgen_fnx_account_balancecarriedforward}	☐	

VoucherLines: Fortnox Voucher Lines

Catalog: Fortnox

Schema: Views

Label: Voucher Lines

This is a read-only view. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

View Columns

The columns of the view VoucherLines are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Account	int32	{res:itgen_fnx_account}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Credit	double	{res:itgen_fnx_credit}	☐	
Debit	double	{res:itgen_fnx_debit}	☐	
Description	string	{res:itgen_fnx_description}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Quantity	double	{res:itgen_fnx_quantity}	☐	
Removed	boolean	{res:itgen_fnx_removed}	☐	
TransactionInformation	string(100)	{res:itgen_fnx_transactioninformation}	☐	
VOUCHERAPPROVALSTATE	int32	{res:itgen_fnx_approvalstate}	☐	
VOUCHERDESCRIPTION	string	{res:itgen_fnx_description}	☐	
VOUCHERNUMBER	int32	{res:itgen_fnx_vouchernumber}	☐	
VOUCHERREFERENCENUMBER	string	{res:itgen_fnx_referencenumber}	☐	
VOUCHERSERIES	string	{res:itgen_fnx_voucherseries}	☐	
VOUCHERTRANSACTIONDATE	datetime	{res:itgen_fnx_transactiondate}	☐	
YEARACCOUNTINGMETHOD	string	{res:itgen_fnx_accountingmethod}	☐	
YEAREND	datetime	{res:itgen_fnx_todate}	☐	
YEARID	int32	{res:itgen_fnx_id}	☐	
YEARSTART	datetime	{res:itgen_fnx_fromdate}	☐	

Vouchers

Catalog: Fortnox

Schema: Views

Label: {res:itgen_fortnox_vouchers}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only view. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

View Columns

The columns of the view Vouchers are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
VOUCHERAPPROVALSTATE	int32	{res:itgen_fnx_approvalstate}	☐	
VOUCHERDESCRIPTION	string	{res:itgen_fnx_description}	☐	
VOUCHERNUMBER	int32	{res:itgen_fnx_vouchernumber}	☐	
VOUCHERREFERENCENUMBER	string	{res:itgen_fnx_referencenumber}	☐	
VOUCHERSERIES	string	{res:itgen_fnx_voucherseries}	☐	
VOUCHERTRANSACTIONDATE	datetime	{res:itgen_fnx_transactiondate}	☐	
YEARACCOUNTINGMETHOD	string	{res:itgen_fnx_accountingmethod}	☐	
YEAREND	datetime	{res:itgen_fnx_todate}	☐	
YEARID	int32	{res:itgen_fnx_id}	☐	
YEARSTART	datetime	{res:itgen_fnx_fromdate}	☐	

3.1.65 Schema: VoucherFileConnections

3.1.65.1 Tables

VoucherFileConnectionByFileId

Retrieve a single voucher file connection

Catalog: Fortnox

Schema: VoucherFileConnections

Label: {res:itgen_fnx_voucherfileconnectionbyfileid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function VoucherFileConnectionByFileId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
FileId	string	☑		identifies the voucher file connection
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function VoucherFileConnectionByFileId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
VoucherFileConnection__url	string	{res:itgen_fnx_voucherfileconnection__url}	☐	
VoucherFileConnection_FileId	string	{res:itgen_fnx_voucherfileconnection_fileid}	☐	
VoucherFileConnection_VoucherDescription	string	{res:itgen_fnx_voucherfileconnection_voucherdescription}	☐	
VoucherFileConnection_VoucherNumber	string	{res:itgen_fnx_voucherfileconnection_vouchernumber}	☐	
VoucherFileConnection_VoucherSeries	string	{res:itgen_fnx_voucherfileconnection_voucherseries}	☐	
VoucherFileConnection_VoucherYear	int32	{res:itgen_fnx_voucherfileconnection_voucheryear}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

VoucherFileConnections

Retrieve a list of voucher file connectionsThe voucher file connections register can return a list of records or a single record. By specifying a FileId in the URL, a single record will be returned. Not specifying a FileId will return a list of records.

Catalog: Fortnox

Schema: VoucherFileConnections

Primary Keys: FileId

Label: {res:itgen_fnx_voucherfileconnections_plural}

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function VoucherFileConnections. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

The columns of the table function VoucherFileConnections are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
FileId	string	{res:itgen_fnx_fileid}	☐	
VoucherDescription	string	{res:itgen_fnx_voucherdescription}	☐	
VoucherNumber	string	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
VoucherYear	int32	{res:itgen_fnx_voucheryear}	☐	

3.1.66 Schema: Vouchers

3.1.66.1 Tables

VoucherLinesByVoucherSeriesAndVoucherNumber

Retrieve a specific voucher

Catalog: Fortnox

Schema: Vouchers

Label: {res:itgen_fnx_voucherlinesbyvoucherseriesandvouchernumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function VoucherLinesByVoucherSeriesAndVoucherNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyear	int32	☐		filter on financial year
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
VoucherNumber	int32	☑		identifies the voucher number
VoucherSeries	string	☑		identifies the voucher series

Columns of Table Function

The columns of the table function VoucherLinesByVoucherSeriesAndVoucherNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Account	int32	{res:itgen_fnx_account}	☐	
ApprovalState	int32	{res:itgen_fnx_approvalstate}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Comments	string(1000)	{res:itgen_fnx_comments}	☐	
CostCenter_2	string	{res:itgen_fnx_costcenter_2}	☐	
CostCenter	string	{res:itgen_fnx_costcenter}	☐	
Credit	double	{res:itgen_fnx_credit}	☐	
Debit	double	{res:itgen_fnx_debit}	☐	
Description_2	string(200)	{res:itgen_fnx_description_2}	☐	
Description	string	{res:itgen_fnx_description}	☐	
Project_2	string	{res:itgen_fnx_project_2}	☐	
Project	string	{res:itgen_fnx_project}	☐	
Quantity	double	{res:itgen_fnx_quantity}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ReferenceNumber	string	{res:itgen_fnx_referencenumber}	☐	
ReferenceType	string	{res:itgen_fnx_referencetype}	☐	
Removed	boolean	{res:itgen_fnx_removed}	☐	
TransactionDate	datetime	{res:itgen_fnx_transactiondate}	☐	
TransactionInformation	string(100)	{res:itgen_fnx_transactioninformation}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
Year	int32	{res:itgen_fnx_year}	☐	

Vouchers

Retrieve all vouchersNote that vouchers have two keys, one for voucher series and one for voucher number. The financial year is also specified for each voucher, this is due to the same voucher series and number is used each year.To get a unique voucher you need the voucher series, the voucher number and the financial year. These properties will always be returned where ever vouchers is used.

Catalog: Fortnox

Schema: Vouchers

Label: {res:itgen_fnx_vouchers_plural}

Retrieve: true

Insert: true

Update: false

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Vouchers. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyear	int32	☐		filter on financial year
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Vouchers are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
ApprovalState	int32	{res:itgen_fnx_approvalstate}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Comments	string	{res:itgen_fnx_comments}	☐	
Description	string	{res:itgen_fnx_description}	☐	
ReferenceNumber	string	{res:itgen_fnx_referencenumber}	☐	
ReferenceType	string	{res:itgen_fnx_referencetype}	☐	
TransactionDate	datetime	{res:itgen_fnx_transactiondate}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
Year	int32	{res:itgen_fnx_year}	☐	

VouchersByVoucherSeries

Retrieve a list of vouchers for a specific series

Catalog: Fortnox

Schema: Vouchers

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: {res:itgen_fnx_vouchersbyvoucherseries_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function VouchersByVoucherSeries. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int32	☐		filter on financial year
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
VoucherSeries	string	☑		identifies the voucher series

Columns of Table Function

The columns of the table function VouchersByVoucherSeries are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ApprovalState	int32	{res:itgen_fnx_approvalstate}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
Comments	string	{res:itgen_fnx_comments}	☐	
Description	string	{res:itgen_fnx_description}	☐	
ReferenceNumber	string	{res:itgen_fnx_referencenumber}	☐	
ReferenceType	string	{res:itgen_fnx_referencetype}	☐	
TransactionDate	datetime	{res:itgen_fnx_transactiondate}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
Year	int32	{res:itgen_fnx_year}	☐	

VouchersByVoucherSeriesAndVoucherNumber

Retrieve a specific voucher

Catalog: Fortnox

Schema: Vouchers

Label: {res:itgen_fnx_vouchersbyvoucherseriesandvouchernumber_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function VouchersByVoucherSeriesAndVoucherNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int32	☐		filter on financial year
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
VoucherNumber	int32	☑		identifies the voucher number
VoucherSeries	string	☑		identifies the voucher series

Columns of Table Function

The columns of the table function VouchersByVoucherSeriesAndVoucherNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Voucher__url	string	{res:itgen_fnx_voucher__url}	☐	
Voucher_ApprovalState	int32	{res:itgen_fnx_voucher_approvalstate}	☐	
Voucher_Comments	string(1000)	{res:itgen_fnx_voucher_comments}	☐	
Voucher_CostCenter	string	{res:itgen_fnx_voucher_costcenter}	☐	
Voucher_Description	string(200)	{res:itgen_fnx_voucher_description}	☐	
Voucher_Project	string	{res:itgen_fnx_voucher_project}	☐	
Voucher_ReferenceNumber	string	{res:itgen_fnx_voucher_referencenumber}	☐	
Voucher_ReferenceType	string	{res:itgen_fnx_voucher_referencetype}	☐	
Voucher_TransactionDate	datetime	{res:itgen_fnx_voucher_transactiondate}	☐	
Voucher_VoucherNumber	int32	{res:itgen_fnx_voucher_vouchernumber}	☐	
Voucher_VoucherSeries	string	{res:itgen_fnx_voucher_voucherseries}	☐	
Voucher_Year	int32	{res:itgen_fnx_voucher_year}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

VouchersForCurrentFinancialYear

Retrieve all vouchers for the current financial year

Catalog: Fortnox

Schema: Vouchers

Label: {res:itgen_fnx_vouchersforcurrentfinancialyear_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function VouchersForCurrentFinancialYear. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function VouchersForCurrentFinancialYear are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ApprovalState	int32	{res:itgen_fnx_approvalstate}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Comments	string	{res:itgen_fnx_comments}	☐	
Description	string	{res:itgen_fnx_description}	☐	
ReferenceNumber	string	{res:itgen_fnx_referencenumber}	☐	
ReferenceType	string	{res:itgen_fnx_referencetype}	☐	
TransactionDate	datetime	{res:itgen_fnx_transactiondate}	☐	
VoucherNumber	int32	{res:itgen_fnx_vouchernumber}	☐	
VoucherSeries	string	{res:itgen_fnx_voucherseries}	☐	
Year	int32	{res:itgen_fnx_year}	☐	

3.1.67 Schema: VoucherSeries

3.1.67.1 Tables

VoucherSeries

Retrieve a list of voucher seriesThe voucher series register can return a list of records or a single record. By specifying a Code in the URL, a single record will be returned. Not specifying a Code will return a list of records.

Catalog: Fortnox

Schema: VoucherSeries

Primary Keys: Code

Label: {res:itgen_fnx_voucherseries_plural}

Retrieve: true

Insert: true

Update: true

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function VoucherSeries. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function VoucherSeries are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
Approver_Id	int32	{res:itgen_fnx_approver_id}	☐	
Approver_Name	string	{res:itgen_fnx_approver_name}	☐	
c_url	string	{res:itgen_fnx_c_url}	☐	
Code	string(10)	{res:itgen_fnx_code}	☐	
Description	string(200)	{res:itgen_fnx_description}	☐	
Manual	boolean	{res:itgen_fnx_manual}	☐	
Year	int32	{res:itgen_fnx_year}	☐	

VoucherSeriesByCode

Retrieve a single voucher series

Catalog: Fortnox

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: VoucherSeries

Label: {res:itgen_fnx_voucherseriesbycode_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function VoucherSeriesByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
Code	string	☑		identifies the voucher series
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

Columns of Table Function

The columns of the table function VoucherSeriesByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
VoucherSeries__url	string	{res:itgen_fnx_vouche	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		rseries__url}		
VoucherSeries_Approver_Id	int32	{res:itgen_fnx_voucherseries_approver_id}	☐	
VoucherSeries_Approver_Name	string	{res:itgen_fnx_voucherseries_approver_name}	☐	
VoucherSeries_Code	string(10)	{res:itgen_fnx_voucherseries_code}	☐	
VoucherSeries_Description	string(200)	{res:itgen_fnx_voucherseries_description}	☐	
VoucherSeries_Manual	boolean	{res:itgen_fnx_voucherseries_manual}	☐	
VoucherSeries_NextVoucherNumber	int32	{res:itgen_fnx_voucherseries_nextvouchernumber}	☐	
VoucherSeries_Year	int32	{res:itgen_fnx_voucherseries_year}	☐	

3.1.68 Schema: Warehouse_CustomDocumentType

3.1.68.1 Tables

CustomDocumentTypeByType

Get custom document type

Catalog: Fortnox

Schema: Warehouse_CustomDocumentType

Label: {res:itgen_fnx_customdocumenttypebytype_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomDocumentTypeByType. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
type	string	☑		the name of the reference type

Columns of Table Function

The columns of the table function CustomDocumentTypeByType are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
category	string	{res:itgen_fnx_category}	☐	
referenceType	string(25)	{res:itgen_fnx_referencetype}	☐	

CustomDocumentTypes

List custom document types

Catalog: Fortnox

Schema: Warehouse_CustomDocumentType

Label: {res:itgen_fnx_customdocumenttypes_plural}

Retrieve: true

Insert: true

Update: false

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomDocumentTypes. A value must be provided at all times for required parameters, but

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function CustomDocumentTypes are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
category	string	{res:itgen_fnx_category}	☐	
referenceType	string(25)	{res:itgen_fnx_referencetype}	☐	

3.1.69 Schema: Warehouse_CustomInboundDocument

3.1.69.1 Tables

CustomInboundDocumentRows

Get custom inbound document

Catalog: Fortnox

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: Warehouse_CustomInboundDocument

Label: {res:itgen_fnx_custominbounddocumentrows_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomInboundDocumentRows. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	string	☑		Document id.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
type	string	☑		Document type.

Columns of Table Function

The columns of the table function CustomInboundDocumentRows are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
batch	string(50)	{res:itgen_fnx_batch}	☐	
costCenterCode	string(25)	{res:itgen_fnx_costcentercode}	☐	
currency_currency	string(3)	{res:itgen_fnx_currency_currency}	☐	
currency_rate	decimal	{res:itgen_fnx_currency_rate}	☐	
currency_unit	int32	{res:itgen_fnx_currency_unit}	☐	
date	datetime	{res:itgen_fnx_date}	☐	
directCost	decimal	{res:itgen_fnx_directcost}	☐	
freightCost	decimal	{res:itgen_fnx_freightcost}	☐	
id	string(25)	{res:itgen_fnx_id}	☐	
itemDescription	string(200)	{res:itgen_fnx_itemdescription}	☐	
itemId	string(50)	{res:itgen_fnx_itemid}	☐	
itemUnit	string(50)	{res:itgen_fnx_itemunit}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
otherCost	decimal	{res:itgen_fnx_othercost}	☐	
projectId	string(25)	{res:itgen_fnx_projectid}	☐	
quantity	decimal	{res:itgen_fnx_quantity}	☐	
row Id	int64	{res:itgen_fnx_row id}	☐	
stockLocationCode	string	{res:itgen_fnx_stocklocationcode}	☐	
stockLocationId	guid	{res:itgen_fnx_stocklocationid}	☐	
stockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
type	string(25)	{res:itgen_fnx_type}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	
w arehouseReady	boolean	{res:itgen_fnx_w arehouseready}	☐	

CustomInboundDocuments

Get custom inbound document

Catalog: Fortnox

Schema: Warehouse_CustomInboundDocument

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: {res:itgen_fnx_custominbounddocuments_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomInboundDocuments. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	string	☑		Document id.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
type	string	☑		Document type.

Columns of Table Function

The columns of the table function CustomInboundDocuments are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
currency_currency	string(3)	{res:itgen_fnx_currency_currency}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
currency_rate	decimal	{res:itgen_fnx_currency_rate}	☐	
currency_unit	int32	{res:itgen_fnx_currency_unit}	☐	
date	datetime	{res:itgen_fnx_date}	☐	
id	string(25)	{res:itgen_fnx_id}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
type	string(25)	{res:itgen_fnx_type}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	
warehouseReady	boolean	{res:itgen_fnx_warehouseready}	☐	

3.1.70 Schema: Warehouse_CustomOutboundDocument

3.1.70.1 Tables

CustomOutboundDocumentAverageCostsByTypeAndId

Get custom outbound document

Catalog: Fortnox

Schema: Warehouse_CustomOutboundDocument

Label: {res:itgen_fnx_customoutbounddocumentaveragecostsbytypeandid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomOutboundDocumentAverageCostsByTypeAndId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
id	string	☑		document id
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
type	string	☑		document type

Columns of Table Function

The columns of the table function CustomOutboundDocumentAverageCostsByTypeAndId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
averageCostInSEK	decimal	{res:itgen_fnx_averagecostinsek}	☐	
date	datetime	{res:itgen_fnx_date}	☐	
deliveryState	string	{res:itgen_fnx_deliverystate}	☐	
forcedDelivery	boolean	{res:itgen_fnx_forceddelivery}	☐	
id	string(25)	{res:itgen_fnx_id}	☐	
itemId	string	{res:itgen_fnx_itemid}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
referenceType	string(25)	{res:itgen_fnx_referencetype}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	
w arehouseReady	boolean	{res:itgen_fnx_w arehouseready}	☐	

CustomOutboundDocumentByTypeAndId

Get custom outbound document

Catalog: Fortnox

Schema: Warehouse_CustomOutboundDocument

Label: {res:itgen_fnx_customoutbounddocumentbytypeandid_plural}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomOutboundDocumentByTypeAndId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	string	☑		document id
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
type	string	☑		document type

Columns of Table Function

The columns of the table function CustomOutboundDocumentByTypeAndId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
date	datetime	{res:itgen_fnx_date}	☐	
deliveryState	string	{res:itgen_fnx_deliverystate}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
forcedDelivery	boolean	{res:itgen_fnx_forced delivery}	☐	
id	string(25)	{res:itgen_fnx_id}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
referenceType	string(25)	{res:itgen_fnx_referencetype}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	
warehouseReady	boolean	{res:itgen_fnx_warehouseready}	☐	

CustomOutboundDocumentLinesByTypeAndId

Get custom outbound document

Catalog: Fortnox

Schema: Warehouse_CustomOutboundDocument

Label: {res:itgen_fnx_customoutbounddocumentlinesbytypeandid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomOutboundDocumentLinesByTypeAndId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
id	string	☑		document id
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
type	string	☑		document type

Columns of Table Function

The columns of the table function CustomOutboundDocumentLinesByTypeAndId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
costCenterCode	string(25)	{res:itgen_fnx_costcentercode}	☐	
date	datetime	{res:itgen_fnx_date}	☐	
deliveredQuantity	decimal	{res:itgen_fnx_deliveredquantity}	☐	
deliveryState	string	{res:itgen_fnx_deliverystate}	☐	
forcedDelivery	boolean	{res:itgen_fnx_forceddelivery}	☐	
forcedQuantity	decimal	{res:itgen_fnx_forcedquantity}	☐	
id	string(25)	{res:itgen_fnx_id}	☐	
itemDescription	string(200)	{res:itgen_fnx_itemdescription}	☐	
itemId	string(50)	{res:itgen_fnx_itemid}	☐	
itemUnit	string(50)	{res:itgen_fnx_itemunit}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
projectId	string(25)	{res:itgen_fnx_projectid}	☐	
quantity	decimal	{res:itgen_fnx_quantity}	☐	
referenceType	string(25)	{res:itgen_fnx_referencetype}	☐	
reservedQuantity	decimal	{res:itgen_fnx_reservedquantity}	☐	
row Id	int64	{res:itgen_fnx_row id}	☐	
stockLocationId	guid	{res:itgen_fnx_stocklocationid}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
voided	boolean	{res:itgen_fnx_voided}	☐	
warehouseReady	boolean	{res:itgen_fnx_warehouseready}	☐	

3.1.71 Schema: Warehouse_IncomingGoods

3.1.71.1 Tables

IncomingGoodDocumentById

Get Incoming Goods document

Catalog: Fortnox

Schema: Warehouse_IncomingGoods

Label: {res:itgen_fnx_incominggooddocumentbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function IncomingGoodDocumentById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Incoming goods document id.
ignoreSupplierInvoiceId	int64	☐		This Supplier Invoice id will be excluded when calculating the takenQuantity.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function IncomingGoodDocumentById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
completed	boolean	{res:itgen_fnx_completed}	☐	
costCenterCode	string	{res:itgen_fnx_costcentercode}	☐	
date	datetime	{res:itgen_fnx_date}	☐	
deliveryNoteId	string(50)	{res:itgen_fnx_deliverynoteid}	☐	
hasDeliveryNote	boolean	{res:itgen_fnx_hasdeliverynote}	☐	
id	int64	{res:itgen_fnx_id}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
projectId	string	{res:itgen_fnx_projectid}	☐	
released	boolean	{res:itgen_fnx_released}	☐	
stockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
stockPointName	string	{res:itgen_fnx_stockpointname}	☐	
supplierName	string(1024)	{res:itgen_fnx_suppliername}	☐	
supplierNumber	string(1024)	{res:itgen_fnx_suppliernumber}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	

IncomingGoodDocumentLinesById

Get Incoming Goods document

Catalog: Fortnox

Schema: Warehouse_IncomingGoods

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: {res:itgen_fnx_incominggooddocumentlinesbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function IncomingGoodDocumentLinesById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Incoming goods document id.
ignoreSupplierInvoiceId	int64	☐		This Supplier Invoice id will be excluded when calculating the takenQuantity.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function IncomingGoodDocumentLinesById are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
backOrderQuantity	decimal	{res:itgen_fnx_backorderquantity}	☐	
batch	string(50)	{res:itgen_fnx_batch}	☐	
completed	boolean	{res:itgen_fnx_completed}	☐	
costCenterCode_2	string	{res:itgen_fnx_costcentercode_2}	☐	
costCenterCode	string	{res:itgen_fnx_costcentercode}	☐	
date	datetime	{res:itgen_fnx_date}	☐	
deliveryNoteId	string(50)	{res:itgen_fnx_deliverynoteid}	☐	
directCost	decimal	{res:itgen_fnx_directcost}	☐	
hasDeliveryNote	boolean	{res:itgen_fnx_hasdeliverynote}	☐	
id_2	int64	{res:itgen_fnx_id_2}	☐	
id	guid	{res:itgen_fnx_id}	☐	
invoicedQuantity	decimal	{res:itgen_fnx_invoicedquantity}	☐	
isStockItem	boolean	{res:itgen_fnx_isstockitem}	☐	
itemDescription	string	{res:itgen_fnx_itemdescription}	☐	
itemId	string(50)	{res:itgen_fnx_itemid}	☐	
itemUnit	string	{res:itgen_fnx_itemunit}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
orderedQuantity	decimal	{res:itgen_fnx_orderedquantity}	☐	
projectId_2	string	{res:itgen_fnx_projectid_2}	☐	
projectId	string	{res:itgen_fnx_projectid}	☐	
purchaseOrderId	int64	{res:itgen_fnx_purchaseorderid}	☐	
purchaseOrderRow Id	guid	{res:itgen_fnx_purchaseorderrow id}	☐	
receivedQuantity	decimal	{res:itgen_fnx_receivedquantity}	☐	
released	boolean	{res:itgen_fnx_released}	☐	
remainingOrderedQuantity	decimal	{res:itgen_fnx_remainingorderedquantity}	☐	
row Order	int32	{res:itgen_fnx_row order}	☐	
stockLocationCode	string	{res:itgen_fnx_stocklocationcode}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
stockLocationId	guid	{res:itgen_fnx_stocklocationid}	☐	
stockLocationName	string	{res:itgen_fnx_stocklocationname}	☐	
stockPointCode_2	string	{res:itgen_fnx_stockpointcode_2}	☐	
stockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
stockPointId_2	guid	{res:itgen_fnx_stockpointid_2}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
stockPointName_2	string	{res:itgen_fnx_stockpointname_2}	☐	
stockPointName	string	{res:itgen_fnx_stockpointname}	☐	
supplierName	string(1024)	{res:itgen_fnx_suppliername}	☐	
supplierNumber	string(1024)	{res:itgen_fnx_suppliernumber}	☐	
takenQuantity	decimal	{res:itgen_fnx_takenquantity}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	

IncomingGoodDocuments

List Incoming Goods DocumentsList incoming goods documents matching the given parameters.Sortable fields:id,has_delivery_note,delivery_note_id,supplier_number,date

Catalog: Fortnox

Schema: Warehouse_IncomingGoods

Primary Keys: id

Label: {res:itgen_fnx_incominggooddocuments_plural}

Retrieve: true

Insert: true

Update: true

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function IncomingGoodDocuments. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
completed	boolean	☐		`true` to include only completed documents.`false` to include only non-completed documents.
deliveryNote	string	☐		Include only documents where `deliveryNote`-field contains the given text (case-insensitive).
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
itemId	string	☐		Include only documents with the given `itemId`.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
note	string	☐		Include only documents where `note`-field contains the given text (case-insensitive).
q	string	☐		Include only documents where `id` or `deliveryNote`-field contains the given text (case-insensitive).
released	boolean	☐		`true` to include only released documents.`false` to include only non-released documents.
supplierNumber	string	☐		Include only documents with the given `supplierNumber`.
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
voided	boolean	☐		`true` to include only voided documents.`false` to include only non-voided documents.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

The columns of the table function IncomingGoodDocuments are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
completed	boolean	{res:itgen_fnx_completed}	☐	
date	datetime	{res:itgen_fnx_date}	☐	
deliveryNoteId	string	{res:itgen_fnx_deliverynoteid}	☐	
hasDeliveryNote	boolean	{res:itgen_fnx_hasdeliverynote}	☐	
id	int64	{res:itgen_fnx_id}	☐	
note	string	{res:itgen_fnx_note}	☐	
released	boolean	{res:itgen_fnx_released}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
supplierName	string	{res:itgen_fnx_suppliername}	☐	
supplierNumber	string	{res:itgen_fnx_suppliernumber}	☐	
unmatchedValue	decimal	{res:itgen_fnx_unmatchedvalue}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	

3.1.72 Schema: Warehouse_ManualDocument

3.1.72.1 Tables

ManualDocuments

List manual documents

Catalog: Fortnox

Schema: Warehouse_ManualDocument

Label: {res:itgen_fnx_manualdocuments_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ManualDocuments. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
itemId	string	☐		Include only documents containing the given item.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
state	string	☐		Include only documents with given state. (Values: all, unreleased, released, voided)
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
type	string	☐		Include only documents with given type. (Values: all, inbound, outbound, stocktransfer)

Columns of Table Function

The columns of the table function ManualDocuments are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
date	datetime	{res:itgen_fnx_date}	☐	
deliveryId	int64	{res:itgen_fnx_deliveryid}	☐	
entityId	string	{res:itgen_fnx_entityid}	☐	
note	string	{res:itgen_fnx_note}	☐	
released	boolean	{res:itgen_fnx_released}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
type	string	{res:itgen_fnx_type}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	

3.1.73 Schema: Warehouse_ManualInboundDocument

3.1.73.1 Tables

ManualInboundDocumentRows

Get manual inbound document

Catalog: Fortnox

Schema: Warehouse_ManualInboundDocument

Label: {res:itgen_fnx_manualinbounddocumentrows_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ManualInboundDocumentRows. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
id	int64	☑		Manual Inbound document id.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ManualInboundDocumentRows are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
batch	string(50)	{res:itgen_fnx_batch}	☐	
costCenterCode	string	{res:itgen_fnx_costcentercode}	☐	
currency	string(3)	{res:itgen_fnx_currency}	☐	
currencyRate	decimal	{res:itgen_fnx_currencyrate}	☐	
currencyUnit	int32	{res:itgen_fnx_currencyunit}	☐	
date	datetime	{res:itgen_fnx_date}	☐	
directCost	decimal	{res:itgen_fnx_directcost}	☐	
freightCost	decimal	{res:itgen_fnx_freightcost}	☐	
id	int64	{res:itgen_fnx_id}	☐	
itemDescription	string	{res:itgen_fnx_itemdescription}	☐	
itemId	string(50)	{res:itgen_fnx_itemid}	☐	
itemUnit	string	{res:itgen_fnx_itemunit}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
otherCost	decimal	{res:itgen_fnx_othercost}	☐	
projectId	string	{res:itgen_fnx_projectid}	☐	
quantity	decimal	{res:itgen_fnx_quantity}	☐	
released	boolean	{res:itgen_fnx_released}	☐	
row Id	int64	{res:itgen_fnx_row id}	☐	
stockLocationCode	string	{res:itgen_fnx_stocklocationcode}	☐	
stockLocationId	guid	{res:itgen_fnx_stocklocationid}	☐	
stockLocationName	string	{res:itgen_fnx_stocklocationname}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
stockPointCode_2	string	{res:itgen_fnx_stockpointcode_2}	☐	
stockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
stockPointId_2	guid	{res:itgen_fnx_stockpointid_2}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
stockPointName_2	string	{res:itgen_fnx_stockpointname_2}	☐	
stockPointName	string	{res:itgen_fnx_stockpointname}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	

ManualInboundDocuments

Get manual inbound document

Catalog: Fortnox

Schema: Warehouse_ManualInboundDocument

Label: {res:itgen_fnx_manualinbounddocuments_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ManualInboundDocuments. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Manual Inbound document id.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ManualInboundDocuments are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
currency	string(3)	{res:itgen_fnx_currency}	☐	
currencyRate	decimal	{res:itgen_fnx_currencyrate}	☐	
currencyUnit	int32	{res:itgen_fnx_currencyunit}	☐	
date	datetime	{res:itgen_fnx_date}	☐	
id	int64	{res:itgen_fnx_id}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
released	boolean	{res:itgen_fnx_released}	☐	
stockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
stockPointName	string	{res:itgen_fnx_stockpointname}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	

3.1.74 Schema: Warehouse_ManualOutboundDocument

3.1.74.1 Tables

ManualOutboundDocumentRows

Get manual outbound document

Catalog: Fortnox

Schema: Warehouse_ManualOutboundDocument

Label: {res:itgen_fnx_manualoutbounddocumentrows_plural}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ManualOutboundDocumentRows. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Manual Outbound document id.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ManualOutboundDocumentRows are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
costCenterCode	string	{res:itgen_fnx_costcentercode}	☐	
date	datetime	{res:itgen_fnx_date}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
deliveredQuantity	decimal	{res:itgen_fnx_deliveredquantity}	☐	
forcedQuantity	decimal	{res:itgen_fnx_forcedquantity}	☐	
id	int64	{res:itgen_fnx_id}	☐	
itemDescription	string	{res:itgen_fnx_itemdescription}	☐	
itemId	string(50)	{res:itgen_fnx_itemid}	☐	
itemUnit	string	{res:itgen_fnx_itemunit}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
projectId	string	{res:itgen_fnx_projectid}	☐	
quantity	decimal	{res:itgen_fnx_quantity}	☐	
released	boolean	{res:itgen_fnx_released}	☐	
stockLocationCode	string	{res:itgen_fnx_stocklocationcode}	☐	
stockLocationId	guid	{res:itgen_fnx_stocklocationid}	☐	
stockLocationName	string	{res:itgen_fnx_stocklocationname}	☐	
stockPointCode_2	string	{res:itgen_fnx_stockpointcode_2}	☐	
stockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
stockPointId_2	guid	{res:itgen_fnx_stockpointid_2}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
stockPointName_2	string	{res:itgen_fnx_stockpointname_2}	☐	
stockPointName	string	{res:itgen_fnx_stockpointname}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	

ManualOutboundDocuments

Get manual outbound document

Catalog: Fortnox

Schema: Warehouse_ManualOutboundDocument

Label: {res:itgen_fnx_manualoutbounddocuments_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ManualOutboundDocuments. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Manual Outbound document id.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ManualOutboundDocuments are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
date	datetime	{res:itgen_fnx_date}	☐	
id	int64	{res:itgen_fnx_id}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
released	boolean	{res:itgen_fnx_released}	☐	
stockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
stockPointName	string	{res:itgen_fnx_stockpointname}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	

3.1.75 Schema: Warehouse_ProductionOrder

3.1.75.1 Tables

ProductionOrderDocumentById

Get Production Order document

Catalog: Fortnox

Schema: Warehouse_ProductionOrder

Label: {res:itgen_fnx_productionorderdocumentbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductionOrderDocumentById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Production Order document id.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ProductionOrderDocumentById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
batch	string	{res:itgen_fnx_batch}	☐	
costCenterCode	string	{res:itgen_fnx_costcentercode}	☐	
documentState	string	{res:itgen_fnx_documentstate}	☐	
id	int64	{res:itgen_fnx_id}	☐	
inboundStockLocationId	guid	{res:itgen_fnx_inboundstocklocationid}	☐	
inboundStockPointId	guid	{res:itgen_fnx_inboundstockpointid}	☐	
itemDescription	string	{res:itgen_fnx_itemdescription}	☐	
itemId	string	{res:itgen_fnx_itemid}	☐	
itemUnit	string	{res:itgen_fnx_itemunit}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
outboundStockPointId	guid	{res:itgen_fnx_outboundstockpointid}	☐	
productionDate	datetime	{res:itgen_fnx_productiondate}	☐	
productionState	string	{res:itgen_fnx_productionstate}	☐	
projectId	string	{res:itgen_fnx_projectid}	☐	
quantity	decimal	{res:itgen_fnx_quantity}	☐	
startDate	datetime	{res:itgen_fnx_startdate}	☐	

ProductionOrderDocumentPackageItemsById

Get Production Order document

Catalog: Fortnox

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: Warehouse_ProductionOrder

Label: {res:itgen_fnx_productionorderdocumentpackageitemsbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductionOrderDocumentPackageItemsById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Production Order document id.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ProductionOrderDocumentPackageItemsById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
batch	string	{res:itgen_fnx_batch}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
costCenterCode	string	{res:itgen_fnx_costcentercode}	☐	
documentState	string	{res:itgen_fnx_documentstate}	☐	
id	int64	{res:itgen_fnx_id}	☐	
inboundStockLocationId	guid	{res:itgen_fnx_inboundstocklocationid}	☐	
inboundStockPointId	guid	{res:itgen_fnx_inboundstockpointid}	☐	
itemDescription_2	string	{res:itgen_fnx_itemdescription_2}	☐	
itemDescription	string	{res:itgen_fnx_itemdescription}	☐	
itemId_2	string	{res:itgen_fnx_itemid_2}	☐	
itemId	string	{res:itgen_fnx_itemid}	☐	
itemUnit_2	string	{res:itgen_fnx_itemunit_2}	☐	
itemUnit	string	{res:itgen_fnx_itemunit}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
outboundStockPointId	guid	{res:itgen_fnx_outboundstockpointid}	☐	
productionDate	datetime	{res:itgen_fnx_productiondate}	☐	
productionState	string	{res:itgen_fnx_productionstate}	☐	
projectId	string	{res:itgen_fnx_projectid}	☐	
quantity	decimal	{res:itgen_fnx_quantity}	☐	
quantityRequired	decimal	{res:itgen_fnx_quantityrequired}	☐	
quantityReserved	decimal	{res:itgen_fnx_quantityreserved}	☐	
startDate	datetime	{res:itgen_fnx_startdate}	☐	
totalQuantityRequired	decimal	{res:itgen_fnx_totalquantityrequired}	☐	

ProductionOrderPackageItems

List production orders

Catalog: Fortnox

Schema: Warehouse_ProductionOrder

Label: {res:itgen_fnx_productionorderpackageitems_plural}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductionOrderPackageItems. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
itemId	string	☐		Include only production orders with the given production item.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
state	string	☐		Include only production orders with the given state.Allowed states: all, incomplete, delayed, completed, voided.(Default is incomplete) (Values: all, incomplete, delayed, completed, voided)
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ProductionOrderPackageItems are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
batch	string	{res:itgen_fnx_batch}	☐	
costCenterCode	string	{res:itgen_fnx_costcentercode}	☐	
documentState	string	{res:itgen_fnx_documentstate}	☐	
id	int64	{res:itgen_fnx_id}	☐	
inboundStockLocationId	guid	{res:itgen_fnx_inboundstocklocationid}	☐	
inboundStockPointId	guid	{res:itgen_fnx_inboundstockpointid}	☐	
itemDescription_2	string	{res:itgen_fnx_itemdescription_2}	☐	
itemDescription	string	{res:itgen_fnx_itemdescription}	☐	
itemId_2	string	{res:itgen_fnx_itemid_2}	☐	
itemId	string	{res:itgen_fnx_itemid}	☐	
itemUnit_2	string	{res:itgen_fnx_itemunit_2}	☐	
itemUnit	string	{res:itgen_fnx_itemunit}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
outboundStockPointId	guid	{res:itgen_fnx_outboundstockpointid}	☐	
productionDate	datetime	{res:itgen_fnx_productiondate}	☐	
productionState	string	{res:itgen_fnx_productionstate}	☐	
projectId	string	{res:itgen_fnx_projectid}	☐	
quantity	decimal	{res:itgen_fnx_quantity}	☐	
quantityRequired	decimal	{res:itgen_fnx_quantityrequired}	☐	
quantityReserved	decimal	{res:itgen_fnx_quantityreserved}	☐	
startDate	datetime	{res:itgen_fnx_startdate}	☐	
totalQuantityRequired	decimal	{res:itgen_fnx_totalquantityrequired}	☐	

ProductionOrders

List production orders

Catalog: Fortnox

Schema: Warehouse_ProductionOrder

Primary Keys: id

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: {res:itgen_fnx_productionorders_plural}

Retrieve: true

Insert: true

Update: true

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductionOrders. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
itemId	string	☐		Include only production orders with the given production item.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
state	string	☐		Include only production orders with the given state.Allowed states: all, incomplete, delayed, completed, voided.(Default is incomplete) (Values: all, incomplete, delayed, completed, voided)
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

The columns of the table function ProductionOrders are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
batch	string	{res:itgen_fnx_batch}	☐	
costCenterCode	string	{res:itgen_fnx_costcentercode}	☐	
documentState	string	{res:itgen_fnx_documentstate}	☐	
id	int64	{res:itgen_fnx_id}	☐	
inboundStockLocationId	guid	{res:itgen_fnx_inboundstocklocationid}	☐	
inboundStockPointId	guid	{res:itgen_fnx_inboundstockpointid}	☐	
itemDescription	string	{res:itgen_fnx_itemdescription}	☐	
itemId	string	{res:itgen_fnx_itemid}	☐	
itemUnit	string	{res:itgen_fnx_itemunit}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
outboundStockPointId	guid	{res:itgen_fnx_outboundstockpointid}	☐	
productionDate	datetime	{res:itgen_fnx_productiondate}	☐	
productionState	string	{res:itgen_fnx_productionstate}	☐	
projectId	string	{res:itgen_fnx_projectid}	☐	
quantity	decimal	{res:itgen_fnx_quantity}	☐	
startDate	datetime	{res:itgen_fnx_startdate}	☐	

RequiredProductionPartsByItemId

Get the package items (Bill Of Materials, BOMs) for a production articleIf no parameters are supplied, the totalQuantityRequired for producing 1 unit is returned.Query parameter quantity can optionally be supplied, which willcalculate totalQuantityRequired.If query parameter id is supplied, it will get the quantity from thatProduction Order (the quantity query parameter is ignored if idis included).

Catalog: Fortnox

Schema: Warehouse_ProductionOrder

Label: {res:itgen_fnx_requiredproductionpartsbyitemid_plural}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function RequiredProductionPartsByItemId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☐		the id of the production order (optional)
itemId	string	☑		Production Article id
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
quantity	string	☐		the quantity of the production order (assumed 1 if left empty)
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function RequiredProductionPartsByItemId are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
itemDescription	string	{res:itgen_fnx_itemdescription}	☐	
itemId	string	{res:itgen_fnx_itemid}	☐	
itemUnit	string	{res:itgen_fnx_itemunit}	☐	
quantityRequired	decimal	{res:itgen_fnx_quantityrequired}	☐	
quantityReserved	decimal	{res:itgen_fnx_quantityreserved}	☐	
totalQuantityRequired	decimal	{res:itgen_fnx_totalquantityrequired}	☐	

3.1.76 Schema: Warehouse_PurchaseOrder

3.1.76.1 Tables

MatchedDocumentsByPurchaseOrderId

List matched documentsGet a list of DocumentReference of linked/connected purchase orders to incoming goods and/or invoice document.

Catalog: Fortnox

Schema: Warehouse_PurchaseOrder

Label: {res:itgen_fnx_matcheddocumentsbypurchaseorderid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function MatchedDocumentsByPurchaseOrderId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Purchase order id.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function MatchedDocumentsByPurchaseOrderId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
id	string	{res:itgen_fnx_id}	☐	
type	string	{res:itgen_fnx_type}	☐	

ProductionOrderRowNotes

List Purchase OrdersList purchase orders matching the given parameters.Sortable fields:id,supplier_number,order_date,internal_reference,response_state,delivery_date

Catalog: Fortnox

Schema: Warehouse_PurchaseOrder

Label: {res:itgen_fnx_productionorderrownotes_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductionOrderRowNotes. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
internalReference	string	☐		Include only documents where `internalReference' contains the given text (case-insensitive).
itemId	string	☐		Include only documents with the given `itemId`.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
note	string	☐		Include only documents where `note`-field contains the given text (case-insensitive).
purchaseType	string	☐		Include only documents with the given `purchaseType` (Values: WAREHOUSE, DROPSHIP)
q	string	☐		Include only documents where `id` or `internalReference`-field contains the given text (case-insensitive).
state	string	☐		Include only documents with the given `purchaseOrderState`. (Values: NOT_SENT, SENT, SENT_NOT_REJECTED, DELAYED, RECEIVED, VOIDED, CURRENT, ALL)
supplierNumber	string	☐		Include only documents with the given `supplierNumber`.
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function ProductionOrderRowNotes are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
backOrderQuantity	decimal	{res:itgen_fnx_backorderquantity}	☐	
confirmationEmail	string(100)	{res:itgen_fnx_confirmationemail}	☐	
costCenterCode_2	string(25)	{res:itgen_fnx_costcentercode_2}	☐	
costCenterCode	string(25)	{res:itgen_fnx_costcentercode}	☐	
currencyCode_2	string(3)	{res:itgen_fnx_currencycode_2}	☐	
currencyCode	string(3)	{res:itgen_fnx_currencycode}	☐	
currencyRate	decimal	{res:itgen_fnx_currencyrate}	☐	
currencyUnit	int32	{res:itgen_fnx_currencyunit}	☐	
customerId	string(25)	{res:itgen_fnx_customerid}	☐	
customerName	string(1024)	{res:itgen_fnx_customername}	☐	
customerNumber	string(1024)	{res:itgen_fnx_customernumber}	☐	
deliveryAddress	string(1024)	{res:itgen_fnx_deliveryaddress}	☐	
deliveryAddress2	string(1024)	{res:itgen_fnx_deliveryaddress2}	☐	
deliveryCity	string(50)	{res:itgen_fnx_deliverycity}	☐	
deliveryCountryCode	string(3)	{res:itgen_fnx_deliverycountrycode}	☐	
deliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	
deliveryName	string(50)	{res:itgen_fnx_deliveryname}	☐	
deliveryZipCode	string(10)	{res:itgen_fnx_deliveryzipcode}	☐	
dropship	boolean	{res:itgen_fnx_dropship}	☐	
id_2	int64	{res:itgen_fnx_id_2}	☐	
id	guid	{res:itgen_fnx_id}	☐	
internalReference	string(50)	{res:itgen_fnx_internalreference}	☐	
isStockItem	boolean	{res:itgen_fnx_isstockitem}	☐	
itemDescription	string	{res:itgen_fnx_itemdescription}	☐	
itemId	string(50)	{res:itgen_fnx_itemid}	☐	
itemUnit	string	{res:itgen_fnx_itemunit}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
languageCode	string(3)	{res:itgen_fnx_languagecode}	☐	
manuallyCompleted	boolean	{res:itgen_fnx_manuallycompleted}	☐	
messageToSupplier	string(1000)	{res:itgen_fnx_messagetosupplier}	☐	
note_2	string(1000)	{res:itgen_fnx_note_2}	☐	
note	string	{res:itgen_fnx_note}	☐	
orderDate	datetime	{res:itgen_fnx_orderdate}	☐	
orderedQuantity	decimal	{res:itgen_fnx_orderedquantity}	☐	
orderValue	decimal	{res:itgen_fnx_ordervalue}	☐	
orderValueInSEK	decimal	{res:itgen_fnx_ordervalueinsek}	☐	
ourReference	string(50)	{res:itgen_fnx_ourreference}	☐	
outboundDocumentReference_id	string	{res:itgen_fnx_outbounddocumentreference_id}	☐	
outboundDocumentReference_type	string	{res:itgen_fnx_outbounddocumentreference_type}	☐	
paymentTermsCode	string(20)	{res:itgen_fnx_paymenttermscode}	☐	
price	decimal	{res:itgen_fnx_price}	☐	
projectId_2	string(25)	{res:itgen_fnx_projectid_2}	☐	
projectId	string(25)	{res:itgen_fnx_projectid}	☐	
purchaseOrderId_2	int64	{res:itgen_fnx_purchaseorderid_2}	☐	
purchaseOrderId	int64	{res:itgen_fnx_purchaseorderid}	☐	
purchaseOrderRow Id	guid	{res:itgen_fnx_purchaseorderrow id}	☐	
purchaseOrderState	string	{res:itgen_fnx_purchaseorderstate}	☐	
purchaseType	string	{res:itgen_fnx_purchasetype}	☐	
receivedQuantity	decimal	{res:itgen_fnx_receivedquantity}	☐	
remainingOrderedQuantity	decimal	{res:itgen_fnx_remainingorderedquantity}	☐	
responseState	string	{res:itgen_fnx_responsestate}	☐	
row Num_2	int64	{res:itgen_fnx_row num_2}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
row Num	int64	{res:itgen_fnx_row num}	☐	
stockLocationCode	string	{res:itgen_fnx_stocklocationcode}	☐	
stockLocationId	guid	{res:itgen_fnx_stocklocationid}	☐	
stockLocationName	string	{res:itgen_fnx_stocklocationname}	☐	
stockPointCode_2	string	{res:itgen_fnx_stockpointcode_2}	☐	
stockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
stockPointId_2	guid	{res:itgen_fnx_stockpointid_2}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
stockPointName	string	{res:itgen_fnx_stockpointname}	☐	
supplier	string	{res:itgen_fnx_supplier}	☐	
supplierAddress	string(1024)	{res:itgen_fnx_supplieraddress}	☐	
supplierAddress2	string(1024)	{res:itgen_fnx_supplieraddress2}	☐	
supplierCity	string(1024)	{res:itgen_fnx_suppliercity}	☐	
supplierCountryCode	string(3)	{res:itgen_fnx_suppliercountrycode}	☐	
supplierEmail	string(100)	{res:itgen_fnx_supplieremail}	☐	
supplierName	string(1024)	{res:itgen_fnx_suppliername}	☐	
supplierNumber	string(1024)	{res:itgen_fnx_suppliernumber}	☐	
supplierPostCode	string(1024)	{res:itgen_fnx_supplierpostcode}	☐	
totalReceivedQuantity	decimal	{res:itgen_fnx_totalreceivedquantity}	☐	
translatedResponseState	string	{res:itgen_fnx_translatedresponsestate}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	
yourReference	string(50)	{res:itgen_fnx_yourreference}	☐	

PurchaseOrderAttachedNotesByPurchaseOrderId

Get notesGet notes for a purchase order.

Catalog: Fortnox

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: Warehouse_PurchaseOrder

Label: {res:itgen_fnx_purchaseorderattachednotesbypurchaseorderid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PurchaseOrderAttachedNotesByPurchaseOrderId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Purchase order id.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PurchaseOrderAttachedNotesByPurchaseOrderId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
note	string	{res:itgen_fnx_note}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
purchaseOrderId	int64	{res:itgen_fnx_purchaseorderid}	☐	
purchaseOrderRowId	guid	{res:itgen_fnx_purchaseorderrowid}	☐	
rowNum	int64	{res:itgen_fnx_rownum}	☐	

PurchaseOrderById

Get Purchase Order

Catalog: Fortnox

Schema: Warehouse_PurchaseOrder

Label: {res:itgen_fnx_purchaseorderbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PurchaseOrderById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Purchase order id.
ignoreIncomingGoodsId	int64	☐		used for calculating the remaining ordered quantity. null

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				will take the received quantity from all incoming goods
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PurchaseOrderById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
confirmationEmail	string(100)	{res:itgen_fnx_confirmationemail}	☐	
costCenterCode	string(25)	{res:itgen_fnx_costcentercode}	☐	
currencyCode	string(3)	{res:itgen_fnx_currencycode}	☐	
currencyRate	decimal	{res:itgen_fnx_currencyrate}	☐	
currencyUnit	int32	{res:itgen_fnx_currencyunit}	☐	
customerId	string(25)	{res:itgen_fnx_customerid}	☐	
customerName	string(1024)	{res:itgen_fnx_customername}	☐	
customerNumber	string(1024)	{res:itgen_fnx_customernumber}	☐	
deliveryAddress	string(1024)	{res:itgen_fnx_deliveryaddress}	☐	
deliveryAddress2	string(1024)	{res:itgen_fnx_deliveryaddress2}	☐	
deliveryCity	string(50)	{res:itgen_fnx_deliverycity}	☐	
deliveryCountryCode	string(3)	{res:itgen_fnx_deliverycountrycode}	☐	
deliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	
deliveryName	string(50)	{res:itgen_fnx_deliveryname}	☐	
deliveryZipCode	string(10)	{res:itgen_fnx_deliveryzipcode}	☐	
dropship	boolean	{res:itgen_fnx_dropship}	☐	
id	int64	{res:itgen_fnx_id}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
internalReference	string(50)	{res:itgen_fnx_internalreference}	☐	
languageCode	string(3)	{res:itgen_fnx_languagecode}	☐	
manuallyCompleted	boolean	{res:itgen_fnx_manuallycompleted}	☐	
messageToSupplier	string(1000)	{res:itgen_fnx_messagetosupplier}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
orderDate	datetime	{res:itgen_fnx_orderdate}	☐	
orderValue	decimal	{res:itgen_fnx_ordervalue}	☐	
orderValueInSEK	decimal	{res:itgen_fnx_ordervalueinsek}	☐	
ourReference	string(50)	{res:itgen_fnx_ourreference}	☐	
outboundDocumentReference_id	string	{res:itgen_fnx_outbounddocumentreference_id}	☐	
outboundDocumentReference_type	string	{res:itgen_fnx_outbounddocumentreference_type}	☐	
paymentTermsCode	string(20)	{res:itgen_fnx_paymenttermscode}	☐	
projectId	string(25)	{res:itgen_fnx_projectid}	☐	
purchaseOrderState	string	{res:itgen_fnx_purchaseorderstate}	☐	
purchaseType	string	{res:itgen_fnx_purchasetype}	☐	
responseState	string	{res:itgen_fnx_responsestate}	☐	
stockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
supplier	string	{res:itgen_fnx_supplier}	☐	
supplierAddress	string(1024)	{res:itgen_fnx_supplieraddress}	☐	
supplierAddress2	string(1024)	{res:itgen_fnx_supplieraddress2}	☐	
supplierCity	string(1024)	{res:itgen_fnx_suppliercity}	☐	
supplierCountryCode	string(3)	{res:itgen_fnx_suppliercountrycode}	☐	
supplierEmail	string(100)	{res:itgen_fnx_supplieremail}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
supplierName	string(1024)	{res:itgen_fnx_suppliername}	☐	
supplierNumber	string(1024)	{res:itgen_fnx_suppliernumber}	☐	
supplierPostCode	string(1024)	{res:itgen_fnx_supplierpostcode}	☐	
totalReceivedQuantity	decimal	{res:itgen_fnx_totalreceivedquantity}	☐	
translatedResponseState	string	{res:itgen_fnx_translatedresponsestate}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	
yourReference	string(50)	{res:itgen_fnx_yourreference}	☐	

PurchaseOrderCsvReports

Get CSV list of Purchase Orders

Catalog: Fortnox

Schema: Warehouse_PurchaseOrder

Label: {res:itgen_fnx_purchaseordercsvreports_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PurchaseOrderCsvReports. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
internalReference	string	☐		Include only documents where `internalReference' contains the given text (case-insensitive).
itemId	string	☐		Include only documents with the given `itemId`.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
note	string	☐		Include only documents where `note`-field contains the given text (case-insensitive).
purchaseType	string	☐		Include only documents with the given `purchaseType` (Values: WAREHOUSE, DROPSHIP)
q	string	☐		Include only documents where `id` or `internalReference`-field contains the given text (case-insensitive).
showPurchaseTypeColumn	boolean	☐		True to include the purchase type column, default is false.
state	string	☐		Include only documents with the given `purchaseOrderState`. (Values: NOT_SENT, SENT, SENT_NOT_REJECTED, DELAYED, RECEIVED, VOIDED, CURRENT, ALL)
supplierNumber	string	☐		Include only documents with the given `supplierNumber`.
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PurchaseOrderCsvReports are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
TEXT	string	{res:itgen_fnx_text}	☐	

PurchaseOrderLinesById

Get Purchase Order

Catalog: Fortnox

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: Warehouse_PurchaseOrder

Label: {res:itgen_fnx_purchaseorderlinesbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PurchaseOrderLinesById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Purchase order id.
ignoreIncomingGoodsId	int64	☐		used for calculating the remaining ordered quantity. null will take the received quantity from all incoming goods
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PurchaseOrderLinesById are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
backOrderQuantity	decimal	{res:itgen_fnx_backorderquantity}	☐	
confirmationEmail	string(100)	{res:itgen_fnx_confirmationemail}	☐	
costCenterCode_2	string(25)	{res:itgen_fnx_costcentercode_2}	☐	
costCenterCode	string(25)	{res:itgen_fnx_costcentercode}	☐	
currencyCode_2	string(3)	{res:itgen_fnx_currencycode_2}	☐	
currencyCode	string(3)	{res:itgen_fnx_currencycode}	☐	
currencyRate	decimal	{res:itgen_fnx_currencyrate}	☐	
currencyUnit	int32	{res:itgen_fnx_currencyunit}	☐	
customerId	string(25)	{res:itgen_fnx_customerid}	☐	
customerName	string(1024)	{res:itgen_fnx_customername}	☐	
customerNumber	string(1024)	{res:itgen_fnx_customernumber}	☐	
deliveryAddress	string(1024)	{res:itgen_fnx_deliveryaddress}	☐	
deliveryAddress2	string(1024)	{res:itgen_fnx_deliveryaddress2}	☐	
deliveryCity	string(50)	{res:itgen_fnx_deliverycity}	☐	
deliveryCountryCode	string(3)	{res:itgen_fnx_deliverycountrycode}	☐	
deliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	
deliveryName	string(50)	{res:itgen_fnx_deliveryname}	☐	
deliveryZipCode	string(10)	{res:itgen_fnx_deliveryzipcode}	☐	
dropship	boolean	{res:itgen_fnx_dropship}	☐	
id_2	int64	{res:itgen_fnx_id_2}	☐	
id	guid	{res:itgen_fnx_id}	☐	
internalReference	string(50)	{res:itgen_fnx_internalreference}	☐	
isStockItem	boolean	{res:itgen_fnx_isstockitem}	☐	
itemDescription	string	{res:itgen_fnx_itemdescription}	☐	
itemId	string(50)	{res:itgen_fnx_itemid}	☐	
itemUnit	string	{res:itgen_fnx_itemunit}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
languageCode	string(3)	{res:itgen_fnx_languagecode}	☐	
manuallyCompleted	boolean	{res:itgen_fnx_manuallycompleted}	☐	
messageToSupplier	string(1000)	{res:itgen_fnx_messagetosupplier}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
orderDate	datetime	{res:itgen_fnx_orderdate}	☐	
orderedQuantity	decimal	{res:itgen_fnx_orderedquantity}	☐	
orderValue	decimal	{res:itgen_fnx_ordervalue}	☐	
orderValueInSEK	decimal	{res:itgen_fnx_ordervalueinsek}	☐	
ourReference	string(50)	{res:itgen_fnx_ourreference}	☐	
outboundDocumentReference_id	string	{res:itgen_fnx_outbounddocumentreference_id}	☐	
outboundDocumentReference_type	string	{res:itgen_fnx_outbounddocumentreference_type}	☐	
paymentTermsCode	string(20)	{res:itgen_fnx_paymenttermscode}	☐	
price	decimal	{res:itgen_fnx_price}	☐	
projectId_2	string(25)	{res:itgen_fnx_projectid_2}	☐	
projectId	string(25)	{res:itgen_fnx_projectid}	☐	
purchaseOrderId	int64	{res:itgen_fnx_purchaseorderid}	☐	
purchaseOrderState	string	{res:itgen_fnx_purchaseorderstate}	☐	
purchaseType	string	{res:itgen_fnx_purchasetype}	☐	
receivedQuantity	decimal	{res:itgen_fnx_receivedquantity}	☐	
remainingOrderedQuantity	decimal	{res:itgen_fnx_remainingorderedquantity}	☐	
responseState	string	{res:itgen_fnx_responsestate}	☐	
row Num	int64	{res:itgen_fnx_row num}	☐	
stockLocationCode	string	{res:itgen_fnx_stocklocationcode}	☐	
stockLocationId	guid	{res:itgen_fnx_stocklocationid}	☐	
stockLocationName	string	{res:itgen_fnx_stocklocationname}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
stockPointCode_2	string	{res:itgen_fnx_stockpointcode_2}	☐	
stockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
stockPointId_2	guid	{res:itgen_fnx_stockpointid_2}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
stockPointName	string	{res:itgen_fnx_stockpointname}	☐	
supplier	string	{res:itgen_fnx_supplier}	☐	
supplierAddress	string(1024)	{res:itgen_fnx_supplieraddress}	☐	
supplierAddress2	string(1024)	{res:itgen_fnx_supplieraddress2}	☐	
supplierCity	string(1024)	{res:itgen_fnx_suppliercity}	☐	
supplierCountryCode	string(3)	{res:itgen_fnx_suppliercountrycode}	☐	
supplierEmail	string(100)	{res:itgen_fnx_supplieremail}	☐	
supplierName	string(1024)	{res:itgen_fnx_suppliername}	☐	
supplierNumber	string(1024)	{res:itgen_fnx_suppliernumber}	☐	
supplierPostCode	string(1024)	{res:itgen_fnx_supplierpostcode}	☐	
totalReceivedQuantity	decimal	{res:itgen_fnx_totalreceivedquantity}	☐	
translatedResponseState	string	{res:itgen_fnx_translatedresponsestate}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	
yourReference	string(50)	{res:itgen_fnx_yourreference}	☐	

PurchaseOrderRowNotesById

Get Purchase Order

Catalog: Fortnox

Schema: Warehouse_PurchaseOrder

Label: {res:itgen_fnx_purchaseorderrownotesbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function PurchaseOrderRowNotesById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Purchase order id.
ignoreIncomingGoodsId	int64	☐		used for calculating the remaining ordered quantity. null will take the received quantity from all incoming goods
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PurchaseOrderRowNotesById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
backOrderQuantity	decimal	{res:itgen_fnx_backorderquantity}	☐	
confirmationEmail	string(100)	{res:itgen_fnx_confirmationemail}	☐	
costCenterCode_2	string(25)	{res:itgen_fnx_costcentercode_2}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
costCenterCode	string(25)	{res:itgen_fnx_costcentercode}	☐	
currencyCode_2	string(3)	{res:itgen_fnx_currencycode_2}	☐	
currencyCode	string(3)	{res:itgen_fnx_currencycode}	☐	
currencyRate	decimal	{res:itgen_fnx_currencyrate}	☐	
currencyUnit	int32	{res:itgen_fnx_currencyunit}	☐	
customerId	string(25)	{res:itgen_fnx_customerid}	☐	
customerName	string(1024)	{res:itgen_fnx_customername}	☐	
customerNumber	string(1024)	{res:itgen_fnx_customernumber}	☐	
deliveryAddress	string(1024)	{res:itgen_fnx_deliveryaddress}	☐	
deliveryAddress2	string(1024)	{res:itgen_fnx_deliveryaddress2}	☐	
deliveryCity	string(50)	{res:itgen_fnx_deliverycity}	☐	
deliveryCountryCode	string(3)	{res:itgen_fnx_deliverycountrycode}	☐	
deliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	
deliveryName	string(50)	{res:itgen_fnx_deliveryname}	☐	
deliveryZipCode	string(10)	{res:itgen_fnx_deliveryzipcode}	☐	
dropship	boolean	{res:itgen_fnx_dropship}	☐	
id_2	int64	{res:itgen_fnx_id_2}	☐	
id	guid	{res:itgen_fnx_id}	☐	
internalReference	string(50)	{res:itgen_fnx_internalreference}	☐	
isStockItem	boolean	{res:itgen_fnx_isstockitem}	☐	
itemDescription	string	{res:itgen_fnx_itemdescription}	☐	
itemId	string(50)	{res:itgen_fnx_itemid}	☐	
itemUnit	string	{res:itgen_fnx_itemunit}	☐	
languageCode	string(3)	{res:itgen_fnx_languagecode}	☐	
manuallyCompleted	boolean	{res:itgen_fnx_manuallycompleted}	☐	
messageToSupplier	string(1000)	{res:itgen_fnx_messagetosupplier}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
note_2	string(1000)	{res:itgen_fnx_note_2}	☐	
note	string	{res:itgen_fnx_note}	☐	
orderDate	datetime	{res:itgen_fnx_orderdate}	☐	
orderedQuantity	decimal	{res:itgen_fnx_orderedquantity}	☐	
orderValue	decimal	{res:itgen_fnx_ordervalue}	☐	
orderValueInSEK	decimal	{res:itgen_fnx_ordervalueinsek}	☐	
ourReference	string(50)	{res:itgen_fnx_ourreference}	☐	
outboundDocumentReference_id	string	{res:itgen_fnx_outbounddocumentreference_id}	☐	
outboundDocumentReference_type	string	{res:itgen_fnx_outbounddocumentreference_type}	☐	
paymentTermsCode	string(20)	{res:itgen_fnx_paymenttermscode}	☐	
price	decimal	{res:itgen_fnx_price}	☐	
projectId_2	string(25)	{res:itgen_fnx_projectid_2}	☐	
projectId	string(25)	{res:itgen_fnx_projectid}	☐	
purchaseOrderId_2	int64	{res:itgen_fnx_purchaseorderid_2}	☐	
purchaseOrderId	int64	{res:itgen_fnx_purchaseorderid}	☐	
purchaseOrderRowId	guid	{res:itgen_fnx_purchaseorderrowid}	☐	
purchaseOrderState	string	{res:itgen_fnx_purchaseorderstate}	☐	
purchaseType	string	{res:itgen_fnx_purchasetype}	☐	
receivedQuantity	decimal	{res:itgen_fnx_receivedquantity}	☐	
remainingOrderedQuantity	decimal	{res:itgen_fnx_remainingorderedquantity}	☐	
responseState	string	{res:itgen_fnx_responsestate}	☐	
rowNum_2	int64	{res:itgen_fnx_rownum_2}	☐	
rowNum	int64	{res:itgen_fnx_rownum}	☐	
stockLocationCode	string	{res:itgen_fnx_stocklocationcode}	☐	
stockLocationId	guid	{res:itgen_fnx_stocklocationid}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
stockLocationName	string	{res:itgen_fnx_stocklocationname}	☐	
stockPointCode_2	string	{res:itgen_fnx_stockpointcode_2}	☐	
stockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
stockPointId_2	guid	{res:itgen_fnx_stockpointid_2}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
stockPointName	string	{res:itgen_fnx_stockpointname}	☐	
supplier	string	{res:itgen_fnx_supplier}	☐	
supplierAddress	string(1024)	{res:itgen_fnx_supplieraddress}	☐	
supplierAddress2	string(1024)	{res:itgen_fnx_supplieraddress2}	☐	
supplierCity	string(1024)	{res:itgen_fnx_suppliercity}	☐	
supplierCountryCode	string(3)	{res:itgen_fnx_suppliercountrycode}	☐	
supplierEmail	string(100)	{res:itgen_fnx_supplieremail}	☐	
supplierName	string(1024)	{res:itgen_fnx_suppliername}	☐	
supplierNumber	string(1024)	{res:itgen_fnx_suppliernumber}	☐	
supplierPostCode	string(1024)	{res:itgen_fnx_supplierpostcode}	☐	
totalReceivedQuantity	decimal	{res:itgen_fnx_totalreceivedquantity}	☐	
translatedResponseState	string	{res:itgen_fnx_translatedresponsestate}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	
yourReference	string(50)	{res:itgen_fnx_yourreference}	☐	

PurchaseOrderRows

List Purchase OrdersList purchase orders matching the given parameters.Sortable fields:id,supplier_number,order_date,internal_reference,response_state,delivery_date

Catalog: Fortnox

Schema: Warehouse_PurchaseOrder

Label: {res:itgen_fnx_purchaseorderrows_plural}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PurchaseOrderRows. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
internalReference	string	☐		Include only documents where `internalReference' contains the given text (case-insensitive).
itemId	string	☐		Include only documents with the given `itemId`.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
note	string	☐		Include only documents where `note`-field contains the given text (case-insensitive).
purchaseType	string	☐		Include only documents with the given `purchaseType` (Values: WAREHOUSE, DROPSHIP)
q	string	☐		Include only documents where `id` or `internalReference`-field contains the given text (case-insensitive).
state	string	☐		Include only documents with the given `purchaseOrderState`. (Values: NOT_SENT, SENT, SENT_NOT_REJECTED,

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				DELAYED, RECEIVED, VOIDED, CURRENT, ALL)
supplierNumber	string	☐		Include only documents with the given `supplierNumber`.
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PurchaseOrderRows are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
backOrderQuantity	decimal	{res:itgen_fnx_backorderquantity}	☐	
confirmationEmail	string(100)	{res:itgen_fnx_confirmationemail}	☐	
costCenterCode_2	string(25)	{res:itgen_fnx_costcentercode_2}	☐	
costCenterCode	string(25)	{res:itgen_fnx_costcentercode}	☐	
currencyCode_2	string(3)	{res:itgen_fnx_currencycode_2}	☐	
currencyCode	string(3)	{res:itgen_fnx_currencycode}	☐	
currencyRate	decimal	{res:itgen_fnx_currencyrate}	☐	
currencyUnit	int32	{res:itgen_fnx_currencyunit}	☐	
customerId	string(25)	{res:itgen_fnx_customerid}	☐	
customerName	string(1024)	{res:itgen_fnx_customername}	☐	
customerNumber	string(1024)	{res:itgen_fnx_customernumber}	☐	
deliveryAddress	string(1024)	{res:itgen_fnx_deliveryaddress}	☐	
deliveryAddress2	string(1024)	{res:itgen_fnx_deliveryaddress2}	☐	
deliveryCity	string(50)	{res:itgen_fnx_deliverycity}	☐	
deliveryCountryCode	string(3)	{res:itgen_fnx_deliverycountrycode}	☐	
deliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	
deliveryName	string(50)	{res:itgen_fnx_deliveryname}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
deliveryZipCode	string(10)	{res:itgen_fnx_deliveryzipcode}	☐	
dropship	boolean	{res:itgen_fnx_dropship}	☐	
id_2	int64	{res:itgen_fnx_id_2}	☐	
id	guid	{res:itgen_fnx_id}	☐	
internalReference	string(50)	{res:itgen_fnx_internalreference}	☐	
isStockItem	boolean	{res:itgen_fnx_isstockitem}	☐	
itemDescription	string	{res:itgen_fnx_itemdescription}	☐	
itemId	string(50)	{res:itgen_fnx_itemid}	☐	
itemUnit	string	{res:itgen_fnx_itemunit}	☐	
languageCode	string(3)	{res:itgen_fnx_languagecode}	☐	
manuallyCompleted	boolean	{res:itgen_fnx_manuallycompleted}	☐	
messageToSupplier	string(1000)	{res:itgen_fnx_messagetosupplier}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
orderDate	datetime	{res:itgen_fnx_orderdate}	☐	
orderedQuantity	decimal	{res:itgen_fnx_orderedquantity}	☐	
orderValue	decimal	{res:itgen_fnx_ordervalue}	☐	
orderValueInSEK	decimal	{res:itgen_fnx_ordervalueinsek}	☐	
ourReference	string(50)	{res:itgen_fnx_ourreference}	☐	
outboundDocumentReference_id	string	{res:itgen_fnx_outbounddocumentreference_id}	☐	
outboundDocumentReference_type	string	{res:itgen_fnx_outbounddocumentreference_type}	☐	
paymentTermsCode	string(20)	{res:itgen_fnx_paymenttermscode}	☐	
price	decimal	{res:itgen_fnx_price}	☐	
projectId_2	string(25)	{res:itgen_fnx_projectid_2}	☐	
projectId	string(25)	{res:itgen_fnx_projectid}	☐	
purchaseOrderId	int64	{res:itgen_fnx_purchaseorderid}	☐	
purchaseOrderState	string	{res:itgen_fnx_purchaseorderstate}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
purchaseType	string	{res:itgen_fnx_purchasetype}	☐	
receivedQuantity	decimal	{res:itgen_fnx_receivedquantity}	☐	
remainingOrderedQuantity	decimal	{res:itgen_fnx_remainingorderedquantity}	☐	
responseState	string	{res:itgen_fnx_responsestate}	☐	
row Num	int64	{res:itgen_fnx_row num}	☐	
stockLocationCode	string	{res:itgen_fnx_stocklocationcode}	☐	
stockLocationId	guid	{res:itgen_fnx_stocklocationid}	☐	
stockLocationName	string	{res:itgen_fnx_stocklocationname}	☐	
stockPointCode_2	string	{res:itgen_fnx_stockpointcode_2}	☐	
stockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
stockPointId_2	guid	{res:itgen_fnx_stockpointid_2}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
stockPointName	string	{res:itgen_fnx_stockpointname}	☐	
supplier	string	{res:itgen_fnx_supplier}	☐	
supplierAddress	string(1024)	{res:itgen_fnx_supplieraddress}	☐	
supplierAddress2	string(1024)	{res:itgen_fnx_supplieraddress2}	☐	
supplierCity	string(1024)	{res:itgen_fnx_suppliercity}	☐	
supplierCountryCode	string(3)	{res:itgen_fnx_suppliercountrycode}	☐	
supplierEmail	string(100)	{res:itgen_fnx_supplieremail}	☐	
supplierName	string(1024)	{res:itgen_fnx_suppliername}	☐	
supplierNumber	string(1024)	{res:itgen_fnx_suppliernumber}	☐	
supplierPostCode	string(1024)	{res:itgen_fnx_supplierpostcode}	☐	
totalReceivedQuantity	decimal	{res:itgen_fnx_totalreceivedquantity}	☐	
translatedResponseState	string	{res:itgen_fnx_translatedresponsestate}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
yourReference	string(50)	{res:itgen_fnx_yourreference}	☐	

PurchaseOrders

List Purchase OrdersList purchase orders matching the given parameters.Sortable fields:id,supplier_number,order_date,internal_reference,response_state,delivery_date

Catalog: Fortnox

Schema: Warehouse_PurchaseOrder

Primary Keys: id

Label: {res:itgen_fnx_purchaseorders_plural}

Retrieve: true

Insert: true

Update: true

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PurchaseOrders. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
internalReference	string	☐		Include only documents where `internalReference' contains the given text (case-insensitive).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
itemId	string	☐		Include only documents with the given `itemId`.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
note	string	☐		Include only documents where `note`-field contains the given text (case-insensitive).
purchaseType	string	☐		Include only documents with the given `purchaseType` (Values: WAREHOUSE, DROPSHIP)
q	string	☐		Include only documents where `id` or `internalReference`-field contains the given text (case-insensitive).
state	string	☐		Include only documents with the given `purchaseOrderState`. (Values: NOT_SENT, SENT, SENT_NOT_REJECTED, DELAYED, RECEIVED, VOIDED, CURRENT, ALL)
supplierNumber	string	☐		Include only documents with the given `supplierNumber`.
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function PurchaseOrders are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
confirmationEmail	string(100)	{res:itgen_fnx_confirmationemail}	☐	
costCenterCode	string(25)	{res:itgen_fnx_costcentercode}	☐	
currencyCode	string(3)	{res:itgen_fnx_currencycode}	☐	
currencyRate	decimal	{res:itgen_fnx_currencyrate}	☐	
currencyUnit	int32	{res:itgen_fnx_currencyunit}	☐	
customerId	string(25)	{res:itgen_fnx_customerid}	☐	
customerName	string(1024)	{res:itgen_fnx_customername}	☐	
customerNumber	string(1024)	{res:itgen_fnx_customernumber}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
deliveryAddress	string(1024)	{res:itgen_fnx_deliveryaddress}	☐	
deliveryAddress2	string(1024)	{res:itgen_fnx_deliveryaddress2}	☐	
deliveryCity	string(50)	{res:itgen_fnx_deliverycity}	☐	
deliveryCountryCode	string(3)	{res:itgen_fnx_deliverycountrycode}	☐	
deliveryDate	datetime	{res:itgen_fnx_deliverydate}	☐	
deliveryName	string(50)	{res:itgen_fnx_deliveryname}	☐	
deliveryZipCode	string(10)	{res:itgen_fnx_deliveryzipcode}	☐	
dropship	boolean	{res:itgen_fnx_dropship}	☐	
id	int64	{res:itgen_fnx_id}	☐	
internalReference	string(50)	{res:itgen_fnx_internalreference}	☐	
languageCode	string(3)	{res:itgen_fnx_languagecode}	☐	
manuallyCompleted	boolean	{res:itgen_fnx_manuallycompleted}	☐	
messageToSupplier	string(1000)	{res:itgen_fnx_messagetosupplier}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
orderDate	datetime	{res:itgen_fnx_orderdate}	☐	
orderValue	decimal	{res:itgen_fnx_ordervalue}	☐	
orderValueInSEK	decimal	{res:itgen_fnx_ordervalueinsek}	☐	
ourReference	string(50)	{res:itgen_fnx_ourreference}	☐	
outboundDocumentReference_id	string	{res:itgen_fnx_outbounddocumentreference_id}	☐	
outboundDocumentReference_type	string	{res:itgen_fnx_outbounddocumentreference_type}	☐	
paymentTermsCode	string(20)	{res:itgen_fnx_paymenttermscode}	☐	
projectId	string(25)	{res:itgen_fnx_projectid}	☐	
purchaseOrderState	string	{res:itgen_fnx_purchaseorderstate}	☐	
purchaseType	string	{res:itgen_fnx_purchasetype}	☐	
responseState	string	{res:itgen_fnx_responsestate}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
stockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
supplier	string	{res:itgen_fnx_supplier}	☐	
supplierAddress	string(1024)	{res:itgen_fnx_supplieraddress}	☐	
supplierAddress2	string(1024)	{res:itgen_fnx_supplieraddress2}	☐	
supplierCity	string(1024)	{res:itgen_fnx_suppliercity}	☐	
supplierCountryCode	string(3)	{res:itgen_fnx_suppliercountrycode}	☐	
supplierEmail	string(100)	{res:itgen_fnx_supplieremail}	☐	
supplierName	string(1024)	{res:itgen_fnx_suppliername}	☐	
supplierNumber	string(1024)	{res:itgen_fnx_suppliernumber}	☐	
supplierPostCode	string(1024)	{res:itgen_fnx_supplierpostcode}	☐	
totalReceivedQuantity	decimal	{res:itgen_fnx_totalreceivedquantity}	☐	
translatedResponseState	string	{res:itgen_fnx_translatedresponsestate}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	
yourReference	string(50)	{res:itgen_fnx_yourreference}	☐	

3.1.77 Schema: Warehouse_StockPoint

3.1.77.1 Tables

MultiStockLocations

Get stock pointsGet stock points by IDs.Use query param `state` to filter on ACTIVE, INACTIVE or ALL (default is to include ALL stock points).Stock locations are NOT included in the response.

Catalog: Fortnox

Schema: Warehouse_StockPoint

Label: {res:itgen_fnx_multistocklocations_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function MultiStockLocations. A value must be provided at all times for required parameters, but

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
state	string	☐		filter on <code>StockPointState</code>, default is to include ALL stock points. (Values: ALL, ACTIVE, INACTIVE)
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function MultiStockLocations are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
active	boolean	{res:itgen_fnx_active}	☐	
code_2	string(10)	{res:itgen_fnx_code_2}	☐	
code	string(20)	{res:itgen_fnx_code}	☐	
deliveryAddress	string(50)	{res:itgen_fnx_deliveryaddress}	☐	
deliveryAddress2	string(50)	{res:itgen_fnx_deliveryaddress2}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
deliveryCity	string(50)	{res:itgen_fnx_deliverycity}	☐	
deliveryCountryCode	string(3)	{res:itgen_fnx_deliverycountrycode}	☐	
deliveryName	string(50)	{res:itgen_fnx_deliveryname}	☐	
deliveryPhone	string(50)	{res:itgen_fnx_deliveryphone}	☐	
deliveryZipCode	string(10)	{res:itgen_fnx_deliveryzipcode}	☐	
id_2	guid	{res:itgen_fnx_id_2}	☐	
id	guid	{res:itgen_fnx_id}	☐	
name_2	string(25)	{res:itgen_fnx_name_2}	☐	
name	string(25)	{res:itgen_fnx_name}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
usingCompanyAddress	boolean	{res:itgen_fnx_usingcompanyaddress}	☐	

MultiStockpoints

Get stock pointsGet stock points by IDs.Use query param `state` to filter on ACTIVE, INACTIVE or ALL (default is to include ALL stock points).Stock locations are NOT included in the response.

Catalog: Fortnox

Schema: Warehouse_StockPoint

Label: {res:itgen_fnx_multistockpoints_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function MultiStockpoints. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
state	string	☐		filter on <code>StockPointState</code>, default is to include ALL stock points. (Values: ALL, ACTIVE, INACTIVE)
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function MultiStockpoints are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
active	boolean	{res:itgen_fnx_active}	☐	
code	string(10)	{res:itgen_fnx_code}	☐	
deliveryAddress	string(50)	{res:itgen_fnx_deliveryaddress}	☐	
deliveryAddress2	string(50)	{res:itgen_fnx_deliveryaddress2}	☐	
deliveryCity	string(50)	{res:itgen_fnx_deliverycity}	☐	
deliveryCountryCode	string(3)	{res:itgen_fnx_deliverycountrycode}	☐	
deliveryName	string(50)	{res:itgen_fnx_deliveryname}	☐	
deliveryPhone	string(50)	{res:itgen_fnx_deliveryphone}	☐	
deliveryZipCode	string(10)	{res:itgen_fnx_deliveryzipcode}	☐	
id	guid	{res:itgen_fnx_id}	☐	
name	string(25)	{res:itgen_fnx_name}	☐	
usingCompanyAddress	boolean	{res:itgen_fnx_usingcompanyaddress}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

StockLocations2ByStockPointId

Get stock locationsList stock locations for a specific stock point.Optionally include a query parameter `q` to filter on stock location code or name.

Catalog: Fortnox

Schema: Warehouse_StockPoint

Label: {res:itgen_fnx_stocklocations2bystockpointid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function StockLocations2ByStockPointId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	string	☑		stock point id or code
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
q	string	☐		filters on stock location code or name.
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

The columns of the table function StockLocations2ByStockPointId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
code	string(20)	{res:itgen_fnx_code}	☐	
id	guid	{res:itgen_fnx_id}	☐	
name	string(25)	{res:itgen_fnx_name}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	

StockLocationsByStockPointId

Get stock pointGet stock point by id or code.

Catalog: Fortnox

Schema: Warehouse_StockPoint

Label: {res:itgen_fnx_stocklocationsbystockpointid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function StockLocationsByStockPointId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers,

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				vouchers and supplierinvoices (does not work on every table).
id	string	☑		stock point code, or stock point id
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function StockLocationsByStockPointId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
active	boolean	{res:itgen_fnx_active}	☐	
code_2	string(10)	{res:itgen_fnx_code_2}	☐	
code	string(20)	{res:itgen_fnx_code}	☐	
deliveryAddress	string(50)	{res:itgen_fnx_deliveryaddress}	☐	
deliveryAddress2	string(50)	{res:itgen_fnx_deliveryaddress2}	☐	
deliveryCity	string(50)	{res:itgen_fnx_deliverycity}	☐	
deliveryCountryCode	string(3)	{res:itgen_fnx_deliverycountrycode}	☐	
deliveryName	string(50)	{res:itgen_fnx_deliveryname}	☐	
deliveryPhone	string(50)	{res:itgen_fnx_deliveryphone}	☐	
deliveryZipCode	string(10)	{res:itgen_fnx_deliveryzipcode}	☐	
id_2	guid	{res:itgen_fnx_id_2}	☐	
id	guid	{res:itgen_fnx_id}	☐	
name_2	string(25)	{res:itgen_fnx_name_2}	☐	
name	string(25)	{res:itgen_fnx_name}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
usingCompanyAddress	boolean	{res:itgen_fnx_usingcompanyaddress}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

StockpointById

Get stock pointGet stock point by id or code.

Catalog: Fortnox

Schema: Warehouse_StockPoint

Label: {res:itgen_fnx_stockpointbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function StockpointById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	string	☑		stock point code, or stock point id
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the table function StockpointById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
active	boolean	{res:itgen_fnx_active}	☐	
code	string(10)	{res:itgen_fnx_code}	☐	
deliveryAddress	string(50)	{res:itgen_fnx_deliveryaddress}	☐	
deliveryAddress2	string(50)	{res:itgen_fnx_deliveryaddress2}	☐	
deliveryCity	string(50)	{res:itgen_fnx_deliverycity}	☐	
deliveryCountryCode	string(3)	{res:itgen_fnx_deliverycountrycode}	☐	
deliveryName	string(50)	{res:itgen_fnx_deliveryname}	☐	
deliveryPhone	string(50)	{res:itgen_fnx_deliveryphone}	☐	
deliveryZipCode	string(10)	{res:itgen_fnx_deliveryzipcode}	☐	
id	guid	{res:itgen_fnx_id}	☐	
name	string(25)	{res:itgen_fnx_name}	☐	
usingCompanyAddress	boolean	{res:itgen_fnx_usingcompanyaddress}	☐	

StockpointLocations

List stock pointsList stock points, optionally include a query parameter `q` to filter on stock point code or name.Use query param `state` to filter on ACTIVE, INACTIVE or ALL (default is to include only ACTIVE stock points).Stock locations are NOT included in the response.

Catalog: Fortnox

Schema: Warehouse_StockPoint

Label: {res:itgen_fnx_stockpointlocations_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function StockpointLocations. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
q	string	☐		filters on stock point code or name.
state	string	☐		filter on stock point state (Values: ALL, ACTIVE, INACTIVE)
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function StockpointLocations are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
active	boolean	{res:itgen_fnx_active}	☐	
code_2	string(10)	{res:itgen_fnx_code_2}	☐	
code	string(20)	{res:itgen_fnx_code}	☐	
deliveryAddress	string(50)	{res:itgen_fnx_deliveryaddress}	☐	
deliveryAddress2	string(50)	{res:itgen_fnx_deliveryaddress2}	☐	
deliveryCity	string(50)	{res:itgen_fnx_deliverycity}	☐	
deliveryCountryCode	string(3)	{res:itgen_fnx_deliverycountrycode}	☐	
deliveryName	string(50)	{res:itgen_fnx_deliveryname}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
deliveryPhone	string(50)	{res:itgen_fnx_deliveryphone}	☐	
deliveryZipCode	string(10)	{res:itgen_fnx_deliveryzipcode}	☐	
id_2	guid	{res:itgen_fnx_id_2}	☐	
id	guid	{res:itgen_fnx_id}	☐	
name_2	string(25)	{res:itgen_fnx_name_2}	☐	
name	string(25)	{res:itgen_fnx_name}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
usingCompanyAddress	boolean	{res:itgen_fnx_usingcompanyaddress}	☐	

Stockpoints

List stock pointsList stock points, optionally include a query parameter `q` to filter on stock point code or name.Use query param `state` to filter on ACTIVE, INACTIVE or ALL (default is to include only ACTIVE stock points).Stock locations are NOT included in the response.

Catalog: Fortnox

Schema: Warehouse_StockPoint

Primary Keys: id

Label: {res:itgen_fnx_stockpoints_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Stockpoints. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
q	string	☐		filters on stock point code or name.
state	string	☐		filter on stock point state (Values: ALL, ACTIVE, INACTIVE)
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function Stockpoints are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
active	boolean	{res:itgen_fnx_active}	☐	
code	string(10)	{res:itgen_fnx_code}	☐	
deliveryAddress	string(50)	{res:itgen_fnx_deliveryaddress}	☐	
deliveryAddress2	string(50)	{res:itgen_fnx_deliveryaddress2}	☐	
deliveryCity	string(50)	{res:itgen_fnx_deliverycity}	☐	
deliveryCountryCode	string(3)	{res:itgen_fnx_deliverycountrycode}	☐	
deliveryName	string(50)	{res:itgen_fnx_deliveryname}	☐	
deliveryPhone	string(50)	{res:itgen_fnx_deliveryphone}	☐	
deliveryZipCode	string(10)	{res:itgen_fnx_deliveryzipcode}	☐	
id	guid	{res:itgen_fnx_id}	☐	
name	string(25)	{res:itgen_fnx_name}	☐	
usingCompanyAddress	boolean	{res:itgen_fnx_usingcompanyaddress}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

3.1.78 Schema: Warehouse_StockStatus

3.1.78.1 Tables

StockBalances

Get stock balanceGet stock balance for each stockpoint.Returns a list of itemId, stockPointCode,availableStock, inStock.(The difference between availableStock and inStockis the reserved amount.)

Catalog: Fortnox

Schema: Warehouse_StockStatus

Label: {res:itgen_fnx_stockbalances_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function StockBalances. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

The columns of the table function StockBalances are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
availableStock	decimal	{res:itgen_fnx_availablestock}	☐	
inStock	decimal	{res:itgen_fnx_instock}	☐	
itemId	string	{res:itgen_fnx_itemid}	☐	
stockPointCode	string	{res:itgen_fnx_stockpointcode}	☐	

3.1.79 Schema: Warehouse_StockTaking

3.1.79.1 Tables

StockTakingCandidateRows

Get candidate rowsA candidate row is a combination of itemId, stockPointId and stockLocationIdthat can be added to the Stock Taking document.Rows already added to the Stock Taking are excluded from this list.

Catalog: Fortnox

Schema: Warehouse_StockTaking

Label: {res:itgen_fnx_stocktakingcandidaterows_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function StockTakingCandidateRows. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
excludeZeroBalanceItems	boolean	☐		
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Stock Taking document id.
includeNonInboundItems	boolean	☐		Include items that do not exist on inbound deliveries.
itemDescriptionSearch	string	☐		
itemIdSearch	string	☐		
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
transactionDate	datetime	☐		

Columns of Table Function

The columns of the table function StockTakingCandidateRows are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
countedBy	string(100)	{res:itgen_fnx_countedby}	☐	
currentRow No	int32	{res:itgen_fnx_currentrow no}	☐	
hasPostReleaseStockChanges	boolean	{res:itgen_fnx_hasposttreleasestockchanges}	☐	
id	string	{res:itgen_fnx_id}	☐	
itemId	string(50)	{res:itgen_fnx_itemid}	☐	
stockLocationId	guid	{res:itgen_fnx_stocklocationid}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
stockTakenQuantity	decimal	{res:itgen_fnx_stocktakenquantity}	☐	
stockTakingId	int64	{res:itgen_fnx_stocktakingid}	☐	
stockTakingRow Id	string	{res:itgen_fnx_stocktakingrow id}	☐	
totalQuantityInStock	decimal	{res:itgen_fnx_totalquantityinstock}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

StockTakingLinesByDocumentId

Get Stock Taking document

Catalog: Fortnox

Schema: Warehouse_StockTaking

Label: {res:itgen_fnx_stocktakinglinesbydocumentid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function StockTakingLinesByDocumentId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Stock Taking document id.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the table function StockTakingLinesByDocumentId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
costCenterCode	string	{res:itgen_fnx_costcentercode}	☐	
countedBy	string(100)	{res:itgen_fnx_countedby}	☐	
currentRow No	int32	{res:itgen_fnx_currentrow no}	☐	
date	datetime	{res:itgen_fnx_date}	☐	
hasPostReleaseStockChanges	boolean	{res:itgen_fnx_haspostreleasestockchanges}	☐	
id_2	int64	{res:itgen_fnx_id_2}	☐	
id	string	{res:itgen_fnx_id}	☐	
itemId	string(50)	{res:itgen_fnx_itemid}	☐	
name	string(50)	{res:itgen_fnx_name}	☐	
projectId	string	{res:itgen_fnx_projectid}	☐	
responsible	string(50)	{res:itgen_fnx_responsible}	☐	
sortingId	int32	{res:itgen_fnx_sortingid}	☐	
sortParams_primarySort	string	{res:itgen_fnx_sortparams_primarysort}	☐	
sortParams_primarySortOrder	string	{res:itgen_fnx_sortparams_primarysortorder}	☐	
sortParams_secondarySort	string	{res:itgen_fnx_sortparams_secondarysort}	☐	
sortParams_secondarySortOrder	string	{res:itgen_fnx_sortparams_secondarysortorder}	☐	
state	string	{res:itgen_fnx_state}	☐	
stockLocationId	guid	{res:itgen_fnx_stocklocationid}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
stockTakenQuantity	decimal	{res:itgen_fnx_stocktakenquantity}	☐	
stockTakingId	int64	{res:itgen_fnx_stocktakingid}	☐	
stockTakingRow Id	string	{res:itgen_fnx_stocktakingrow id}	☐	
totalQuantityInStock	decimal	{res:itgen_fnx_totalquantityinstock}	☐	
usingStockPoints	boolean	{res:itgen_fnx_usingstockpoints}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

StockTakingRows

Get Stock Taking Rows

Catalog: Fortnox

Schema: Warehouse_StockTaking

Label: {res:itgen_fnx_stocktakingrows_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function StockTakingRows. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
excludeZeroBalanceItems	boolean	☐		
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Stock Taking document id.
itemDescriptionSearch	string	☐		
itemIdSearch	string	☐		
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
secondaryorder	string	☐		Secondary sorting order
secondarysortby	string	☐		Secondary sorting column
startingItemId	string	☐		the itemId that should be on top of the rows list (used to jump to specific row), can be empty

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
startingRow No	int32	☐		the row number to start the search from, used with startingItemId to jump to specific rows, can be empty
stateFilter	string	☐		Values: all, notStockTaken, stockTakenNoDeviation, stockTakenWithDeviation.
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
transactionDate	datetime	☐		

Columns of Table Function

The columns of the table function StockTakingRows are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
countedBy	string(100)	{res:itgen_fnx_countedby}	☐	
currentRow No	int32	{res:itgen_fnx_currentrowno}	☐	
hasPostReleaseStockChanges	boolean	{res:itgen_fnx_hasposttreleasestockchanges}	☐	
id	string	{res:itgen_fnx_id}	☐	
itemId	string(50)	{res:itgen_fnx_itemid}	☐	
stockLocationId	guid	{res:itgen_fnx_stocklocationid}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
stockTakenQuantity	decimal	{res:itgen_fnx_stocktakenquantity}	☐	
stockTakingId	int64	{res:itgen_fnx_stocktakingid}	☐	
stockTakingRow Id	string	{res:itgen_fnx_stocktakingrow id}	☐	
totalQuantityInStock	decimal	{res:itgen_fnx_totalquantityinstock}	☐	

StockTakingRows2

List stock takingsSortable fields:id,name,date,responsible,state

Catalog: Fortnox

Schema: Warehouse_StockTaking

Label: {res:itgen_fnx_stocktakingrows2_plural}

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function StockTakingRows2. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
itemId	string	☐		Include only stock takings with the given item.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
state	string	☐		Include only stock takings with the given state. (Values: all, planning, started, completed, voided)
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function StockTakingRows2 are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
costCenterCode	string	{res:itgen_fnx_costcentercode}	☐	
countedBy	string(100)	{res:itgen_fnx_countedby}	☐	
currentRow No	int32	{res:itgen_fnx_currentrow no}	☐	
date	datetime	{res:itgen_fnx_date}	☐	
hasPostReleaseStockChanges	boolean	{res:itgen_fnx_haspostreleasestockchanges}	☐	
id_2	int64	{res:itgen_fnx_id_2}	☐	
id	string	{res:itgen_fnx_id}	☐	
itemId	string(50)	{res:itgen_fnx_itemid}	☐	
name	string(50)	{res:itgen_fnx_name}	☐	
projectId	string	{res:itgen_fnx_projectid}	☐	
responsible	string(50)	{res:itgen_fnx_responsible}	☐	
sortingId	int32	{res:itgen_fnx_sortingid}	☐	
sortParams_primarySort	string	{res:itgen_fnx_sortparams_primarysort}	☐	
sortParams_primarySortOrder	string	{res:itgen_fnx_sortparams_primarysortorder}	☐	
sortParams_secondarySort	string	{res:itgen_fnx_sortparams_secondarysort}	☐	
sortParams_secondarySortOrder	string	{res:itgen_fnx_sortparams_secondarysortorder}	☐	
state	string	{res:itgen_fnx_state}	☐	
stockLocationId	guid	{res:itgen_fnx_stocklocationid}	☐	
stockPointId	guid	{res:itgen_fnx_stockpointid}	☐	
stockTakenQuantity	decimal	{res:itgen_fnx_stocktakenquantity}	☐	
stockTakingId	int64	{res:itgen_fnx_stocktakingid}	☐	
stockTakingRow Id	string	{res:itgen_fnx_stocktakingrow id}	☐	
totalQuantityInStock	decimal	{res:itgen_fnx_totalquantityinstock}	☐	
usingStockPoints	boolean	{res:itgen_fnx_usingstockpoints}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

StockTakings

List stock takingsSortable fields:id,name,date,responsible,state

Catalog: Fortnox

Schema: Warehouse_StockTaking

Primary Keys: id

Label: {res:itgen_fnx_stocktakings_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function StockTakings. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
itemId	string	☐		Include only stock takings w ith the given item.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
state	string	☐		Include only stock takings w ith the given state. (Values: all, planning, started, completed, voided)
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

The columns of the table function StockTakings are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
costCenterCode	string	{res:itgen_fnx_costcentercode}	☐	
date	datetime	{res:itgen_fnx_date}	☐	
id	int64	{res:itgen_fnx_id}	☐	
name	string(50)	{res:itgen_fnx_name}	☐	
projectId	string	{res:itgen_fnx_projectid}	☐	
responsible	string(50)	{res:itgen_fnx_responsible}	☐	
sortingId	int32	{res:itgen_fnx_sortingid}	☐	
sortParams_primarySort	string	{res:itgen_fnx_sortparams_primarysort}	☐	
sortParams_primarySortOrder	string	{res:itgen_fnx_sortparams_primarysortorder}	☐	
sortParams_secondarySort	string	{res:itgen_fnx_sortparams_secondarysort}	☐	
sortParams_secondarySortOrder	string	{res:itgen_fnx_sortparams_secondarysortorder}	☐	
state	string	{res:itgen_fnx_state}	☐	
usingStockPoints	boolean	{res:itgen_fnx_usingstockpoints}	☐	

StockTakingsByDocumentId

Get Stock Taking document

Catalog: Fortnox

Schema: Warehouse_StockTaking

Label: {res:itgen_fnx_stocktakingsbydocumentid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function StockTakingsByDocumentId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Stock Taking document id.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function StockTakingsByDocumentId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
costCenterCode	string	{res:itgen_fnx_costcentercode}	☐	
date	datetime	{res:itgen_fnx_date}	☐	
id	int64	{res:itgen_fnx_id}	☐	
name	string(50)	{res:itgen_fnx_name}	☐	
projectId	string	{res:itgen_fnx_projectid}	☐	
responsible	string(50)	{res:itgen_fnx_responsible}	☐	
sortingId	int32	{res:itgen_fnx_sortingid}	☐	
sortParams_primarySort	string	{res:itgen_fnx_sortparams_primarysort}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
sortParams_primarySortOrder	string	{res:itgen_fnx_sortparams_primarysortorder}	☐	
sortParams_secondarySort	string	{res:itgen_fnx_sortparams_secondarysort}	☐	
sortParams_secondarySortOrder	string	{res:itgen_fnx_sortparams_secondarysortorder}	☐	
state	string	{res:itgen_fnx_state}	☐	
usingStockPoints	boolean	{res:itgen_fnx_usingstockpoints}	☐	

3.1.80 Schema: Warehouse_StockTransfer

3.1.80.1 Tables

StockTransferDocumentLinesById

Get stock transfer document

Catalog: Fortnox

Schema: Warehouse_StockTransfer

Label: {res:itgen_fnx_stocktransferdocumentlinesbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function StockTransferDocumentLinesById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Stock Transfer document id.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function StockTransferDocumentLinesById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
fromStockLocationCode	string	{res:itgen_fnx_fromstocklocationcode}	☐	
fromStockLocationId	guid	{res:itgen_fnx_fromstocklocationid}	☐	
fromStockLocationName	string	{res:itgen_fnx_fromstocklocationname}	☐	
fromStockPointCode	string	{res:itgen_fnx_fromstockpointcode}	☐	
fromStockPointId	guid	{res:itgen_fnx_fromstockpointid}	☐	
fromStockPointName	string	{res:itgen_fnx_fromstockpointname}	☐	
id	int64	{res:itgen_fnx_id}	☐	
itemDescription	string	{res:itgen_fnx_itemdescription}	☐	
itemId	string(50)	{res:itgen_fnx_itemid}	☐	
itemUnit	string	{res:itgen_fnx_itemunit}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
quantity	decimal	{res:itgen_fnx_quantity}	☐	
released	boolean	{res:itgen_fnx_released}	☐	
requestedQuantity	decimal	{res:itgen_fnx_requestedquantity}	☐	
row Num	int64	{res:itgen_fnx_row num}	☐	
toStockLocationCode	string	{res:itgen_fnx_tostocklocationcode}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
toStockLocationId	guid	{res:itgen_fnx_tostocklocationid}	☐	
toStockLocationName	string	{res:itgen_fnx_tostocklocationname}	☐	
toStockPointCode	string	{res:itgen_fnx_tostockpointcode}	☐	
toStockPointId	guid	{res:itgen_fnx_tostockpointid}	☐	
toStockPointName	string	{res:itgen_fnx_tostockpointname}	☐	
transferDate	datetime	{res:itgen_fnx_transferdate}	☐	
version	int64	{res:itgen_fnx_version}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	

StockTransferDocumentsById

Get stock transfer document

Catalog: Fortnox

Schema: Warehouse_StockTransfer

Label: {res:itgen_fnx_stocktransferdocumentsbyid_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function StockTransferDocumentsById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
id	int64	☑		Stock Transfer document id.
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function StockTransferDocumentsById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
id	int64	{res:itgen_fnx_id}	☐	
note	string(1000)	{res:itgen_fnx_note}	☐	
released	boolean	{res:itgen_fnx_released}	☐	
transferDate	datetime	{res:itgen_fnx_transferdate}	☐	
version	int64	{res:itgen_fnx_version}	☐	
voided	boolean	{res:itgen_fnx_voided}	☐	

3.1.81 Schema: Warehouse_Tenant

3.1.81.1 Tables

HasWarehouseFeatures

Get Warehouse activation statusCheck if current tenant has activated Fortnox Warehouse.

Catalog: Fortnox

Schema: Warehouse_Tenant

Label: {res:itgen_fnx_haswarehousefeatures_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HasWarehouseFeatures. A value must be provided at all times for required parameters, but

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function HasWarehouseFeatures are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
activated	boolean	{res:itgen_fnx_activated}	☐	
tenantId	int64	{res:itgen_fnx_tenantid}	☐	

3.1.82 Schema: WayOfDeliveries

3.1.82.1 Tables

DeliveryMethodByCode

Retrieve a single way of delivery

Catalog: Fortnox

Schema: WayOfDeliveries

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: {res:itgen_fnx_deliverymethodbycode_plural}

This is a read-only table function. The Fortnox API may not support changing the data or the Invantive UniversalSQL driver for Fortnox does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Fortnox API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeliveryMethodByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
Code	string	☑		identifies the way of delivery
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function DeliveryMethodByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
WayOfDelivery__url	string	{res:itgen_fnx_wayofdelivery__url}	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
WayOfDelivery_Code	string(50)	{res:itgen_fnx_wayofdelivery_code}	☐	
WayOfDelivery_Description	string(100)	{res:itgen_fnx_wayofdelivery_description}	☐	
WayOfDelivery_DescriptionEnglish	string(100)	{res:itgen_fnx_wayofdelivery_descriptionenglish}	☐	

DeliveryMethods

Retrieve a list of way of deliveriesThe way of delivery register can return a list of records or a single record. By specifying a Code in the URL, a single record will be returned. Not specifying a Code will return a list of records.

Catalog: Fortnox

Schema: WayOfDeliveries

Primary Keys: Code

Label: {res:itgen_fnx_deliverymethods_plural}

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeliveryMethods. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

Columns of Table Function

The columns of the table function DeliveryMethods are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
c_url	string	{res:itgen_fnx_c_url}	☐	
Code	string(50)	{res:itgen_fnx_code}	☐	
Description	string(100)	{res:itgen_fnx_description}	☐	
DescriptionEnglish	string(100)	{res:itgen_fnx_descriptionenglish}	☐	

4 Package: dcr_metadata

4.1 Procedures

4.1.1 dcr_metadata.get_partitions: Fortnox Data container metadata package

Get all partitions.

Documentation:

List all partitions.

5 Package: fnx_fileattachments_attachment

5.1 Procedures

5.1.1 fnx_fileattachments_attachment.attach: Fortnox Package for fileattachments_Attachment operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fileattachments_attachment.attach. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
p_financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

5.1.2 fnx_fileattachments_attachment.detach: Fortnox Package for fileattachments_Attachment operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fileattachments_attachment.detach. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_attachmentId	guid	☑		id of the attachment to be detached
p_financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
p_financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				not work on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

5.1.3 fnx_fileattachments_attachment.updateattachment: Fortnox Package for fileattachments_Attachment operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fileattachments_attachment.updateattachment. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_attachmentId	guid	☑		id of the attachment to be updated
entityId	int64	☐		
entityType	string	☐		Values: OF, O, F, C, LGR_IO, LGR_IG.
fileId	string	☐		
id	guid	☐		
includeOnSend	boolean	☐		
p_financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

5.1.4 fnx_fileattachments_attachment.validateincludedonsend: Fortnox Package for fileattachments_Attachment operations that could not be mapped to a table.

6 Package: fnx_fortnox_archive

6.1 Procedures

6.1.1 fnx_fortnox_archive.removebyid: Fortnox Package for fortnox_Archive operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_archive.removebyid. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	string	☑		identifies file/folder to remove

6.1.2 fnx_fortnox_archive.removebypath: Fortnox Package for fortnox_Archive operations that could not be mapped to a table.

Please note that removing a folder will also resulting in removal of all the contents within!

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_archive.removebypath. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_path	string	☐		identifies file/folder to remove
p_financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
p_financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

7 Package: fnx_fortnox_assets

7.1 Procedures

7.1.1 fnx_fortnox_assets.changeob: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.

The updated asset will be returned if everything succeeded, if there was any problems an error will be returned.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_assets.changeob. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_GivenNumber	string	☑		Asset number
Amount	int32	☐		
Comment	string	☐		

7.1.2 fnx_fortnox_assets.delete: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.

By specifying a {GivenNumber} in the URL a single "Not active" asset or asset with a type "Not depreciable" can be deleted. By specifying a {GivenNumber} in the URL a single "Active" or "Fully depreciated" assets can be voided and in this case in request body voiddate should be provided, otherwise it will use todays date.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_assets.delete. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_GivenNumber	string	☑		Asset number
Date	string	☐		
p_financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

7.1.3 fnx_fortnox_assets.depreciate: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.

The created vouchers list will be returned if everything succeeded, if there was any problems an error will be returned.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_assets.depreciate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
AssetIds	string	☐		
DepreciateUntil	string	☐		

7.1.4 fnx_fortnox_assets.scrap: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.

The updated asset will be returned if everything succeeded, if there was any problems an error will be returned.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_assets.scrap. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_GivenNumber	string	☑		Asset number
Comment	string	☐		
Date	string	☐		
Percentage	int32	☐		

7.1.5 fnx_fortnox_assets.sell: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.

Partial sell or full sell of an asset.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_assets.sell. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_GivenNumber	string	☑		Asset number
Comment	string	☐		
Date	string	☐		
Percentage	int32	☐		
Price	int32	☐		

7.1.6 fnx_fortnox_assets.update: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.

The updated asset will be returned if everything succeeded, if there were any problems an error will be returned.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_assets.update. A value must be provided at all times for required parameters,

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_GivenNumber	string	☑		Asset number
Description	string	☐		
Notes	string	☐		
p_financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

7.1.7 fnx_fortnox_assets.writedown: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.

The updated asset will be returned if everything succeeded, if there was any problems an error will be returned.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_assets.writedown. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_GivenNumber	string	☑		Asset number
Amount	int32	☐		
Comment	string	☐		
Date	string	☐		

7.1.8 fnx_fortnox_assets.writeup: Fortnox Package for fortnox_Assets operations that could not be mapped to a table.

The updated asset will be returned if everything succeeded, if there was any problems an error will be returned.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_assets.writeup. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_GivenNumber	string	☑		Asset number
Amount	int32	☐		
Comment	string	☐		
Date	string	☐		

8 Package: fnx_fortnox_contracts

8.1 Procedures

8.1.1 fnx_fortnox_contracts.createinvoice: Fortnox Package for fortnox_Contracts operations that could not be mapped to a table.

Parameters of Procedure

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_contracts.createinvoice. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the contract

8.1.2 fnx_fortnox_contracts.finish: Fortnox Package for fortnox_Contracts operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_contracts.finish. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the contract

8.1.3 fnx_fortnox_contracts.increaseinvoicecount: Fortnox Package for fortnox_Contracts operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_contracts.increaseinvoicecount. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the contract

9 Package: fnx_fortnox_contracttemplates

9.1 Procedures

9.1.1 fnx_fortnox_contracttemplates.update: Fortnox Package for fortnox_ContractTemplates operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_contracttemplates.update. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_TemplateNumber	int32	☑		identifies the contract template
c_url	string	☐		
AdministrationFee	double	☐		
Continuous	boolean	☐		
ContractLength	int32	☐		
Freight	double	☐		
InvoiceInterval	int32	☐		
OurReference	string	☐		
PrintTemplate	string	☐		
Remarks	string	☐		
TemplateName	string	☐		
TemplateNumber	int32	☐		
TermsOfDelivery	string	☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
TermsOfPayment	string	☐		
WayOfDelivery	string	☐		
p_financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

10 Package: fnx_fortnox_customerreferences

10.1 Procedures

10.1.1 fnx_fortnox_customerreferences.remove: Fortnox Package for fortnox_CustomerReferences operations that could not be mapped to a table.

Deletes the customer reference row permanently. If everything succeeded the response will be of the type 204, No content and the response body will be empty. If there was any problems an error will be returned. You need to supply the unique customer reference row id of the customer reference row that you want to delete.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_customerreferences.remove. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_CustomerReferenceRowId	string	☑		identifies the customer reference row

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

10.1.2 fnx_fortnox_customerreferences.update: Fortnox Package for fortnox_CustomerReferences operations that could not be mapped to a table.

The updated customer reference row will be returned if everything succeeded, if there was any problems an error will be returned. You need to supply the unique customer reference row id of the customer reference row that you want to update. Only the properties provided in the request body will be updated, properties not provided will be left unchanged. CustomerNumber cannot be changed by this request.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_customerreferences.update. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_CustomerReferenceRowId	string	☑		identifies the customer reference row
CustomerNumber	string	☐		
Id	int32	☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
Reference	string	☐		
p_financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

11 Package: fnx_fortnox_financeinvoices

11.1 Procedures

11.1.1 fnx_fortnox_financeinvoices.create: Fortnox Package for fortnox_FinanceInvoices operations that could not be mapped to a table.

<p>When sending an invoice with Fortnox Finans you will get the invoice status returned if everything succeeded,if there were any problems, an error will be returned.<p>Please note that it can take 1 min to several hours before you will get back status, OCR number and link toPDF document, meanwhile the invoice will have status UNKNOWN or NOT_AUTHORIZED.<p>Fortnox Finans is currently only accepting invoices in SEK<p>Parameters in the body:
InvoiceNumber: the invoice number for the invoice which should be sent with Fortnox Finans
SendMethod: how to send the invoice; EMAIL, LETTER, EINVOICE or NONE
Service: which service to use; LEDGERBASE or REMINDER
<p>

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_financeinvoices.create. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
InvoiceNumber	string	☐		
SendMethod	string	☐		
Service	string	☐		

11.1.2 fnx_fortnox_financeinvoices.pause: Fortnox Package for fortnox_FinanceInvoices operations that could not be mapped to a table.

<p>Pauses an invoice for up to 60 days. Pause means that Fortnox Finans reminder process will stop for the invoice. All invoices which have the status OPEN can be paused.<p>Note: this action is not available for invoices sent by the old Noxbox platform<p>Parameters in the body:
PausedUntilDate: the invoice will be paused to and including this date.
<p>

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_financeinvoices.pause. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_Number	string	☑		The Fortnox invoice number
PausedUntilDate	datetime	☐		

11.1.3 fnx_fortnox_financeinvoices.reportpayment: Fortnox Package for fortnox_FinanceInvoices operations that could not be mapped to a table.

<p>If a customer has paid some or all of the capital on an invoice directly to the client, this can be reportedfor bookkeeping purposes and reported to Fortnox Finans to actually deduct the paid amount from the invoice.<p>Note: this action is not available for invoices sent by the old Noxbox platform<p>Parameters in the body:
ClientTakesFees: a boolean indicating if the client should take the customer fees or not.
BookkeepPaymentInFortnox: a boolean indicating if the payment should be bookkept in Fortnox or not. Usually the payment should be bookkept.
ReportToFinance: a boolean indicating if the payment should be reported to Fortnox Finans or not. Usually the payment should be reported.
PaymentAmount: a decimal field with the amount to report.
PaymentMethodCode: a string with the method code (e.g. BG, PG or other). Could be omitted if BookkeepPaymentInFortnox is false.
PaymentMethodAccount: an integer with the account number to

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

bookkeep the payment on (e.g. 1920 or other). Could be omitted if BookkeepPaymentInFortnox is false.
<p>

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_financeinvoices.reportpayment. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_Number	string	☑		The Fortnox invoice number
BookkeepPaymentInFortnox	boolean	☑		
ClientTakesFees	boolean	☑		
PaymentAmount	float	☑		
PaymentMethodAccount	string	☑		
PaymentMethodCode	string	☑		
ReportToFinance	boolean	☑		

11.1.4 fnx_fortnox_financeinvoices.stop: Fortnox Package for fortnox_FinanceInvoices operations that could not be mapped to a table.

<p>Removes the invoice from Fortnox Finans process. The invoice can still be handled manually, but no further automatic process will be applied<p>Note: this action is not available for invoices sent by the old Noxbox platform

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_financeinvoices.stop. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_Number	string	☑		The Fortnox invoice number

11.1.5 fnx_fortnox_financeinvoices.takefees: Fortnox Package for fortnox_FinanceInvoices operations that could not be mapped to a table.

<p>If fees have been added to an invoice, e.g. reminder fees, the client can choose to pay those fees instead of letting the customer pay.<p>Note: this action is not available for invoices sent by the old Noxbox platform

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_financeinvoices.takefees. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_Number	string	☑		The Fortnox invoice number

11.1.6 fnx_fortnox_financeinvoices.unpause: Fortnox Package for fortnox_FinanceInvoices operations that could not be mapped to a table.

<p>Unpauses a paused invoice. If the invoice is manually paused, then this action will remove the pause status immediately. Invoices which are paused by the system cannot be unpaused.<p>Note: this action is not available for invoices sent by the old Noxbox platform

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_financeinvoices.unpause. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_Number	string	☑		The Fortnox invoice number

12 Package: fnx_fortnox_inbox

12.1 Procedures

12.1.1 fnx_fortnox_inbox.remove: Fortnox Package for fortnox_Inbox operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_inbox.remove. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_Id	string	☑		identifies the file to remove
p_financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

13 Package: fnx_fortnox_invoicepayments

13.1 Procedures

13.1.1 fnx_fortnox_invoicepayments.bookkeep: Fortnox Package for fortnox_InvoicePayments operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_invoicepayments.bookkeep. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_Number	string	☑		identifies the invoice payment
c_url	string	☐		
Amount	double	☐		
AmountCurrency	double	☐		
Booked	boolean	☐		
Currency	string	☐		
CurrencyRate	double	☐		
CurrencyUnit	double	☐		
ExternalInvoiceReference1	string	☐		
ExternalInvoiceReference2	string	☐		
InvoiceCustomerName	string	☐		
InvoiceCustomerNumber	string	☐		
InvoiceDueDate	datetime	☐		
InvoiceNumber	int32	☐		
InvoiceOCR	string	☐		
InvoiceTotal	string	☐		
ModeOfPayment	string	☐		
ModeOfPaymentAccount	int32	☐		
Number	string	☐		
PaymentDate	datetime	☐		
Source	string	☐		
VoucherNumber	int32	☐		
VoucherSeries	string	☐		
VoucherYear	int32	☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

14 Package: fnx_fortnox_invoices

14.1 Procedures

14.1.1 fnx_fortnox_invoices.bookkeep: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_invoices.bookkeep. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the invoice

14.1.2 fnx_fortnox_invoices.cancel: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_invoices.cancel. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the invoice

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

14.1.3 fnx_fortnox_invoices.credit: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.

The created credit invoice will be referenced in the property CreditInvoiceReference.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_invoices.credit. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the invoice

14.1.4 fnx_fortnox_invoices.einvoice: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_invoices.einvoice. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the invoice
p_financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

14.1.5 fnx_fortnox_invoices.eprint: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_invoices.eprint. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the invoice
p_financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers,

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				vouchers and supplierinvoices (does not w ork on every table).

14.1.6 fnx_fortnox_invoices.externalprint: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.

Use this endpoint to set invoice as sent, without generating an invoice.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_invoices.externalprint. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the invoice

14.1.7 fnx_fortnox_invoices.printreminder: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_invoices.printreminder. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the invoice

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

14.1.8 fnx_fortnox_invoices.warehouseready: Fortnox Package for fortnox_Invoices operations that could not be mapped to a table.

Used for marking a document as ready in the warehouse module. DeliveryState needs to be set to "delivery".

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_invoices.warehouseready. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the invoice

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

15 Package: fnx_fortnox_offers

15.1 Procedures

15.1.1 fnx_fortnox_offers.cancel: Fortnox Package for fortnox_Offers operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_offers.cancel. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the offer

15.1.2 fnx_fortnox_offers.createinvoice: Fortnox Package for fortnox_Offers operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_offers.createinvoice. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the offer

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

15.1.3 fnx_fortnox_offers.createorder: Fortnox Package for fortnox_Offers operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_offers.createorder. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the offer

15.1.4 fnx_fortnox_offers.email: Fortnox Package for fortnox_Offers operations that could not be mapped to a table.

You can use the properties in the EmailInformation to customize the e-mail message on each offer.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_offers.email. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the offer
p_financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
p_financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

15.1.5 fnx_fortnox_offers.externalprint: Fortnox Package for fortnox_Offers operations that could not be mapped to a table.

Use this endpoint to set offer as sent, without generating an offer.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_offers.externalprint. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the offer

15.1.6 fnx_fortnox_offers.preview: Fortnox Package for fortnox_Offers operations that could not be mapped to a table.

The difference between this and the print-endpoint is that property Sent is not set to TRUE.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_offers.preview. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the offer
p_financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

15.1.7 fnx_fortnox_offers.print: Fortnox Package for fortnox_Offers operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_offers.print. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the offer

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

16 Package: fnx_fortnox_orders

16.1 Procedures

16.1.1 fnx_fortnox_orders.cancel: Fortnox Package for fortnox_Orders operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_orders.cancel. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the order

16.1.2 fnx_fortnox_orders.createinvoice: Fortnox Package for fortnox_Orders operations that could not be mapped to a table.

Parameters of Procedure

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_orders.createinvoice. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the order

16.1.3 fnx_fortnox_orders.email: Fortnox Package for fortnox_Orders operations that could not be mapped to a table.

You can use the properties in the EmailInformation to customize the e-mail message on each order.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_orders.email. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the order
p_financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
p_financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers,

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

16.1.4 fnx_fortnox_orders.externalprint: Fortnox Package for fortnox_Orders operations that could not be mapped to a table.

Use this endpoint to set order as sent, without generating an order.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_orders.externalprint. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the order

16.1.5 fnx_fortnox_orders.preview: Fortnox Package for fortnox_Orders operations that could not be mapped to a table.

The difference between this and the print-endpoint is that property Sent is not set to TRUE.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_orders.preview. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the offer
p_financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

16.1.6 fnx_fortnox_orders.print: Fortnox Package for fortnox_Orders operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_orders.print. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_DocumentNumber	string	☑		identifies the order
p_financialyear	int64	☐		Selects what financial year that should be used (does not work

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

17 Package: fnx_fortnox_prices

17.1 Procedures

17.1.1 fnx_fortnox_prices.remove: Fortnox Package for fortnox_Prices operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_prices.remove. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_PriceList	string	☑		identifies the price list
p_ArticleNumber	string	☑		identifies the article number
p_FromQuantity	double	☑		identifies the from quantity
p_financialyear	int64	☐		Selects what financial year that should be used (does not work on every table).
p_financialyeardate	datetime	☐		Selects by date, what financial year that should be used (does not work on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

17.1.2 fnx_fortnox_prices.update: Fortnox Package for fortnox_Prices operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_prices.update. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_PriceList	string	☑		identifies the price list
p_ArticleNumber	string	☑		identifies the article number
c_url	string	☐		
ArticleNumber	string	☐		
Date	datetime	☐		
FromQuantity	double	☐		
Percent	double	☐		
Price	double	☐		
PriceList	string	☐		

17.1.3 fnx_fortnox_prices.update_2: Fortnox Package for fortnox_Prices operations that could not be mapped to a table.

Parameters of Procedure

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_prices.update_2. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_PriceList	string	☑		identifies the price list
p_ArticleNumber	string	☑		identifies the article number
p_FromQuantity	double	☑		identifies the from quantity
c_url	string	☐		
ArticleNumber	string	☐		
Date	datetime	☐		
FromQuantity	double	☐		
Percent	double	☐		
Price	double	☐		
PriceList	string	☐		
p_financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
p_financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

18 Package: fnx_fortnox_scheduletimes

18.1 Procedures

18.1.1 fnx_fortnox_scheduletimes.reset: Fortnox Package for fortnox_ScheduleTimes operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_scheduletimes.reset. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_EmployeeId	string	☑		identifies the employee
p_Date	datetime	☑		identifies the date

18.1.2 fnx_fortnox_scheduletimes.update: Fortnox Package for fortnox_ScheduleTimes operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_scheduletimes.update. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_EmployeeId	string	☑		identifies the employee
p_Date	datetime	☑		identifies the date
Date	datetime	☐		
EmployeeId	string	☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
Hours	string	☐		
IWH1	string	☐		
IWH2	string	☐		
IWH3	string	☐		
IWH4	string	☐		
IWH5	string	☐		
ScheduleId	string	☐		

19 Package: fnx_fortnox_supplierinvoiceexternalurlconnections

19.1 Procedures

19.1.1 fnx_fortnox_supplierinvoiceexternalurlconnections.create: Fortnox Package for fortnox_SupplierInvoiceExternalUrlConnections operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_supplierinvoiceexternalurlconnections.create. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
ExternalURLConnection	string	☐		
SupplierInvoiceNumber	int32	☐		

19.1.2 fnx_fortnox_supplierinvoiceexternalurlconnections.delete: Fortnox Package for fortnox_SupplierInvoiceExternalUrlConnections operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_supplierinvoiceexternalurlconnections.delete. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_Id	int32	☑		id

19.1.3 fnx_fortnox_supplierinvoiceexternalurlconnections.update: Fortnox Package for fortnox_SupplierInvoiceExternalUrlConnections operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_supplierinvoiceexternalurlconnections.update. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_Id	int32	☑		id
ExternalURLConnection	string	☐		
SupplierInvoiceNumber	int32	☐		

20 Package: fnx_fortnox_supplierinvoicepayments

20.1 Procedures

20.1.1 fnx_fortnox_supplierinvoicepayments.bookkeep: Fortnox Package for fortnox_SupplierInvoicePayments operations that could not be mapped to a table.

Parameters of Procedure

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_supplierinvoicepayments.bookkeep. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_Number	int32	☑		identifies the supplier invoice payment

21 Package: fnx_fortnox_supplierinvoices

21.1 Procedures

21.1.1 fnx_fortnox_supplierinvoices.approvalbookkeep: Fortnox Package for fortnox_SupplierInvoices operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_supplierinvoices.approvalbookkeep. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_GivenNumber	int32	☑		identifies the invoice

21.1.2 fnx_fortnox_supplierinvoices.approvalpayment: Fortnox Package for fortnox_SupplierInvoices operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_supplierinvoices.approvalpayment. A value must be provided at all times for

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_GivenNumber	int32	☑		identifies the invoice

21.1.3 fnx_fortnox_supplierinvoices.bookkeep: Fortnox Package for fortnox_SupplierInvoices operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_supplierinvoices.bookkeep. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_GivenNumber	int32	☑		identifies the invoice

21.1.4 fnx_fortnox_supplierinvoices.cancel: Fortnox Package for fortnox_SupplierInvoices operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_supplierinvoices.cancel. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_GivenNumber	int32	☑		identifies the invoice

21.1.5 fnx_fortnox_supplierinvoices.credit: Fortnox Package for fortnox_SupplierInvoices operations that could not be mapped to a table.

The created credit invoice will be referenced in the property CreditReference.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_supplierinvoices.credit. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_GivenNumber	int32	☑		identifies the invoice

22 Package: fnx_fortnox_trustedemailsenders

22.1 Procedures

22.1.1 fnx_fortnox_trustedemailsenders.create: Fortnox Package for fortnox_TrustedEmailSenders operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_trustedemailsenders.create. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
Email	string	☐		
Id	int32	☐		

22.1.2 fnx_fortnox_trustedemailsenders.remove: Fortnox Package for fortnox_TrustedEmailSenders operations that could not be mapped to a table.

Provide an id matching an email to delete.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_fortnox_trustedemailsenders.remove. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_Id	int32	☑		identifies the trusted email sender to delete

23 Package: fnx_warehouse_custominbounddocument

23.1 Procedures

23.1.1 fnx_warehouse_custominbounddocument.release: Fortnox Package for warehouse_CustomInboundDocument operations that could not be mapped to a table.

The document will be locked and bookkept.The inbound deliveries will affect available stock.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_custominbounddocument.release. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_type	string	☑		document type
p_id	string	☑		document id

23.1.2 fnx_warehouse_custominbounddocument.save: Fortnox Package for warehouse_CustomInboundDocument operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_custominbounddocument.save. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation

p_type	string	☑		min 1 character, max 25 characters, may contain letters A-Z, digits 0-9, underscore (_), and dash (-), type is case-insensitive<blockquote><pre>Type is a custom name/reference of the document that will be used to reference the document type
				* If type is not known, it will be registered as belonging to the INBOUND category.
				* If type is an existing custom document type of category OUTBOUND an error is thrown.
				* If type is invalid an error is thrown.
				</pre></blockquote>

p_id	string	☑		min 1 character, max 25 characters, may only contain digits 0-9
currency	string	☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
rate	decimal	☐		
unit	int32	☐		
date	datetime	☑		
id	string	☐		
note	string	☐		
type	string	☐		
voided	boolean	☐		
w arehouseReady	boolean	☐		

23.1.3 fnx_warehouse_custominbounddocument.voiddocument: Fortnox Package for warehouse_CustomInboundDocument operations that could not be mapped to a table.

Voiding a document will undo the possible stock changes that the document had made,note that the document and the transactions created are not deleted. Some limitations apply, see below.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_custominbounddocument.voiddocument. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_type	string	☑		document type
p_id	string	☑		document id
p_force	boolean	☐		true if the document should be voided even if the document has connected outbounds, defaults to false.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

24 Package: fnx_warehouse_customoutbounddocument

24.1 Procedures

24.1.1 fnx_warehouse_customoutbounddocument.release: Fortnox Package for warehouse_CustomOutboundDocument operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_customoutbounddocument.release. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_type	string	☑		document type
p_id	string	☑		document id

24.1.2 fnx_warehouse_customoutbounddocument.save: Fortnox Package for warehouse_CustomOutboundDocument operations that could not be mapped to a table.

If type is not known, it will be registered as belonging to the OUTBOUND category.
If type is an existing custom document type of category INBOUND an error is thrown.
If type is invalid an error is thrown.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_customoutbounddocument.save. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_type	string	☑		the type of the custom outbound document, min 1 character, max

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				25 characters, may contain letters A-Z, digits 0-9, underscore (_), and dash (-). Always stored as upper case.
p_id	string	☑		the id of the custom outbound document, min 1 character, max 25 characters, may only contain digits 0-9
date	datetime	☑		
deliveryState	string	☑		Values: registration, reservation, delivery.
forcedDelivery	boolean	☐		
id	string	☐		
note	string	☐		
referenceType	string	☐		
voided	boolean	☐		
warehouseReady	boolean	☐		

24.1.3 fnx_warehouse_customoutbounddocument.voiddocument: Fortnox Package for warehouse_CustomOutboundDocument operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_customoutbounddocument.voiddocument. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_type	string	☑		document type
p_id	string	☑		document id
p_force	boolean	☐		true if the document should be voided even if the document has connected outbounds, defaults to false.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

25 Package: fnx_warehouse_incominggoods

25.1 Procedures

25.1.1 fnx_warehouse_incominggoods.completed: Fortnox Package for warehouse_IncomingGoods operations that could not be mapped to a table.

Mark a released Incoming Goods document as Completed.Bookkeeping will be finalized.A Completed Incoming Goods document cannot be matched againstany more Supplier Invoices.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_incominggoods.completed. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Incoming goods document id.

25.1.2 fnx_warehouse_incominggoods.patch: Fortnox Package for warehouse_IncomingGoods operations that could not be mapped to a table.

Perform a partial update of an <code>IncomingGoods</code> document. The partial update will update<code>note</code>, <code>deliveryNoteId</code>, <code>supplierName</code> and <code>hasDeliveryNote</code>It is possible to perform a partial update of a released/completed (TODO: ?) document.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_incominggoods.patch. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Incoming goods document id.
completed	boolean	☐		
costCenterCode	string	☐		
date	datetime	☐		
deliveryNoteId	string	☑		
hasDeliveryNote	boolean	☐		
id	int64	☐		
note	string	☐		
projectId	string	☐		
released	boolean	☐		
stockPointCode	string	☐		
stockPointId	guid	☐		
stockPointName	string	☐		
supplierName	string	☐		
supplierNumber	string	☐		
voided	boolean	☐		

25.1.3 fnx_warehouse_incominggoods.release: Fortnox Package for warehouse_IncomingGoods operations that could not be mapped to a table.

The document will be locked and bookkept.The inbound deliveries will affect available stock.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_incominggoods.release. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Incoming goods document id.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

25.1.4 fnx_warehouse_incominggoods.voiddocument: Fortnox Package for warehouse_IncomingGoods operations that could not be mapped to a table.

Void a document.If an Incoming Goods document has been Completed, or matched againstSupplier Invoice, it cannot be voided.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_incominggoods.voiddocument. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Incoming goods document id.

26 Package: fnx_warehouse_manualinbounddocument

26.1 Procedures

26.1.1 fnx_warehouse_manualinbounddocument.create: Fortnox Package for warehouse_ManualInboundDocument operations that could not be mapped to a table.

<p>The <code>id</code> is set automatically.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_manualinbounddocument.create. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
currency	string	☑		
currencyRate	decimal	☑		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
currencyUnit	int32	☐		
date	datetime	☑		
id	int64	☐		
note	string	☐		
released	boolean	☐		
stockPointCode	string	☐		
stockPointId	guid	☐		
stockPointName	string	☐		
voided	boolean	☐		

26.1.2 fnx_warehouse_manualinbounddocument.release: Fortnox Package for warehouse_ManualInboundDocument operations that could not be mapped to a table.

<p>The document will be locked and bookkept.<p>The following error codes might be thrown:<dl><dt>cannot_release_later_than_current_date</dt><dd>Document date cannot be in the future.</dd><dt>document_is_voided</dt><dd>Document is voided.</dd><dt>period_locked</dt><dd>Document date is within a locked bookkeeping period.</dd></dl>

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_manualinbounddocument.release. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Manual Inbound document id.

26.1.3 fnx_warehouse_manualinbounddocument.update: Fortnox Package for warehouse_ManualInboundDocument operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_manualinbounddocument.update. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Manual Inbound document id.
currency	string	☑		
currencyRate	decimal	☑		
currencyUnit	int32	☐		
date	datetime	☑		
id	int64	☐		
note	string	☐		
released	boolean	☐		
stockPointCode	string	☐		
stockPointId	guid	☐		
stockPointName	string	☐		
voided	boolean	☐		

26.1.4 fnx_warehouse_manualinbounddocument.updatenote: Fortnox Package for warehouse_ManualInboundDocument operations that could not be mapped to a table.

<p>When a Manual Inbound has been released, it is locked.The <code>note</code> field can still be updated using this endpoint.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_manualinbounddocument.updatenote. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Manual Inbound document id.
note	string	☐		

26.1.5 fnx_warehouse_manualinbounddocument.voiddocument: Fortnox Package for warehouse_ManualInboundDocument operations that could not be mapped to a table.

<p>A released manual inbound document might have connected outbounds, and can only be force voided.Note that a force void operation might cause a negative stock.<p>The following error codes might be thrown:<dl><dt>void_linked_outbound</dt><dd>If this document has any outbounds transactions connected to it.</dd></dl>

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_manualinbounddocument.voiddocument. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Manual Inbound document id.
p_force	boolean	☐		true if w e should force void, defaults to false
p_customVoidDate	datetime	☐		date the void operation should be bookkeept on

27 Package: fnx_warehouse_manualoutbounddocument

27.1 Procedures

27.1.1 fnx_warehouse_manualoutbounddocument.create: Fortnox Package for warehouse_ManualOutboundDocument operations that could not be mapped to a table.

<p>The <code>id</code> is set automatically.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_manualoutbounddocument.create. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
date	datetime	☑		
id	int64	☐		
note	string	☐		
released	boolean	☐		
stockPointCode	string	☐		
stockPointId	guid	☐		
stockPointName	string	☐		
voided	boolean	☐		

27.1.2 fnx_warehouse_manualoutbounddocument.release: Fortnox Package for warehouse_ManualOutboundDocument operations that could not be mapped to a table.

<p>The document will be locked and bookkept.<p>The following error codes might be thrown:<dl><dt>cannot_release_later_than_current_date</dt><dd>Document date cannot be in the future.</dd><dt>document_is_voided</dt><dd>Document is voided.</dd><dt>period_locked</dt><dd>Document date is within a locked bookkeeping period.</dd></dl>

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_manualoutbounddocument.release. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Manual Outbound document id.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

27.1.3 fnx_warehouse_manualoutbounddocument.update: Fortnox Package for warehouse_ManualOutboundDocument operations that could not be mapped to a table.

<p>HTTP code 400 cannot_modify_released_documentHTTP code 400 document_is_voided Document is voided.HTTP code 404 not found

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_manualoutbounddocument.update. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Manual Outbound document id.
date	datetime	☑		
id	int64	☐		
note	string	☐		
released	boolean	☐		
stockPointCode	string	☐		
stockPointId	guid	☐		
stockPointName	string	☐		
voided	boolean	☐		

27.1.4 fnx_warehouse_manualoutbounddocument.updatenote: Fortnox Package for warehouse_ManualOutboundDocument operations that could not be mapped to a table.

<p>When a Manual Outbound has been released, it is locked.The <code>note</code> field can still be updated using this endpoint.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_manualoutbounddocument.updatenote. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Manual Outbound document id
note	string	☐		

27.1.5 fnx_warehouse_manualoutbounddocument.voiddocument: Fortnox Package for warehouse_ManualOutboundDocument operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_manualoutbounddocument.voiddocument. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Manual Outbound document id.
p_customVoidDate	datetime	☐		if provided this date will be used as the voided date instead of the document date

28 Package: fnx_warehouse_productionorder

28.1 Procedures

28.1.1 fnx_warehouse_productionorder.release: Fortnox Package for warehouse_ProductionOrder operations that could not be mapped to a table.

<p>The document will be locked and bookkept.<p>The following error codes might be thrown:<dl><dt>PRODUCTION_DATE_IS_EMPTY</dt><dd>Production date cannot be empty.</dd><dt>CANNOT_RELEASE_AFTER_CURRENT_DATE</dt><dd>Document date cannot be in the future.</dd><dt>DOCUMENT_IS_VOIDED</dt><dd> The document has been voided and can no longer be modified.</dd><dt>DOCUMENT_IS_RELEASED</dt><dd>The document has already been released and can no longer be

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

modified.</dd><dt>DOCUMENT_NOT_FULLY_RESERVED</dt><dd>The documents package items (BOMs, Bill Of Materials) has not been fully reserved(reserved quantity is not equal to total required quantity for one or more package item).</dd></dl>

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_productionorder.release. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Production Order document id.

28.1.2 fnx_warehouse_productionorder.updatenote: Fortnox Package for warehouse_ProductionOrder operations that could not be mapped to a table.

<p>When a Production Order has been released it is locked.However, the <code>note</code> field can still be updated using this endpoint.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_productionorder.updatenote. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Production Order document id.
note	string	☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

28.1.3 fnx_warehouse_productionorder.voidproductionorder: Fortnox Package for warehouse_ProductionOrder operations that could not be mapped to a table.

<p>A released production order might have connected outbounds, and can only be force voided.Note that a force void operation might cause a negative stock.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_productionorder.voidproductionorder. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Production Order document id.
p_force	boolean	☐		true to force void a released document, default false

29 Package: fnx_warehouse_purchaseorder

29.1 Procedures

29.1.1 fnx_warehouse_purchaseorder.batchupdateresponsestate: Fortnox Package for warehouse_PurchaseOrder operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_purchaseorder.batchupdateresponsestate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
responseState	string	☐		Values: NOT_SENT, SENT, ACCEPTED_WITH_REQ_DLV_DATE, ACCEPTED_WITH_CHANGED_DLV_DATE, PARTLY_ACCEPTED_WITH_REQ_DLV_DATE, PARTLY_ACCEPTED_WITH_CHANGED_DLV_DATE, REJECTED.

29.1.2 fnx_warehouse_purchaseorder.sendpurchaseorder: Fortnox Package for warehouse_PurchaseOrder operations that could not be mapped to a table.

<p>Sends the purchase order with the specified <code>id</code> to the recipient and sets the purchase order state to SENT

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_purchaseorder.sendpurchaseorder. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Purchase order id.
body	string	☑		
bodyAsHtml	string	☐		
receiver	string	☑		
receiverCopy	string	☐		
receiverSecretCopy	string	☐		
replyTo	string	☑		
senderName	string	☐		
subject	string	☑		

29.1.3 fnx_warehouse_purchaseorder.sendpurchaseorders: Fortnox Package for warehouse_PurchaseOrder operations that could not be mapped to a table.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

29.1.4 fnx_warehouse_purchaseorder.setdropshipmanuallycompleted: Fortnox Package for warehouse_PurchaseOrder operations that could not be mapped to a table.

<p>The dropship order will be treated as fully received.Any remaining quantity will be ignored.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_purchaseorder.setdropshipmanuallycompleted. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Purchase order id.

29.1.5 fnx_warehouse_purchaseorder.setmanuallycompleted: Fortnox Package for warehouse_PurchaseOrder operations that could not be mapped to a table.

<p>The purchase order will be treated as fully received.Any remaining quantity will be ignored.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_purchaseorder.setmanuallycompleted. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Purchase order id.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

29.1.6 fnx_warehouse_purchaseorder.updatepartial: Fortnox Package for warehouse_PurchaseOrder operations that could not be mapped to a table.

<p>Perform a partial update of a purchase order, see <code>PartialPurchaseOrder</code> for possiblefields that are updateable.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_purchaseorder.updatepartial. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Purchase order id.
deliveryDate	datetime	☐		
internalReference	string	☐		
messageToSupplier	string	☐		
note	string	☐		
supplierName	string	☐		

29.1.7 fnx_warehouse_purchaseorder.updateresponsestate: Fortnox Package for warehouse_PurchaseOrder operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_purchaseorder.updateresponsestate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Purchase order id.
responseState	string	☐		Values: NOT_SENT, SENT, ACCEPTED_WITH_REQ_DLV_DATE, ACCEPTED_WITH_CHANGED_DLV_DATE, PARTLY_ACCEPTED_WITH_REQ_DLV_DATE, PARTLY_ACCEPTED_WITH_CHANGED_DLV_DATE, REJECTED.

29.1.8 fnx_warehouse_purchaseorder.voiddocument: Fortnox Package for warehouse_PurchaseOrder operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_purchaseorder.voiddocument. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Purchase order id.

30 Package: fnx_warehouse_stockpoint

30.1 Procedures

30.1.1 fnx_warehouse_stockpoint.appendstocklocations: Fortnox Package for warehouse_StockPoint operations that could not be mapped to a table.

<p>Add new stock locations to specific <code>StockPoint</code>.<p>If you include an already existing stock location code, it will be ignored.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_stockpoint.appendstocklocations. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	guid	☑		stock point id

31 Package: fnx_warehouse_stocktaking

31.1 Procedures

31.1.1 fnx_warehouse_stocktaking.addstocktakingrows: Fortnox Package for warehouse_StockTaking operations that could not be mapped to a table.

<p>Add rows to a stock taking.If you add an already existing row noting happens.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_stocktaking.addstocktakingrows. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Stock Taking document id.
p_financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
p_financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not w ork on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not w ork on every table).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

31.1.2 fnx_warehouse_stocktaking.addstocktakingrowsbyfilter: Fortnox Package for warehouse_StockTaking operations that could not be mapped to a table.

<p>Add all matching candidate rows to a stock taking, as specified by filters.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_stocktaking.addstocktakingrowsbyfilter. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Stock Taking document id.
p_transactionDate	datetime	☐		
p_itemIdSearch	string	☐		
p_itemDescriptionSearch	string	☐		
p_excludeZeroBalanceItems	boolean	☐		
p_excludeNonInboundItems	boolean	☐		

31.1.3 fnx_warehouse_stocktaking.deletestocktakingrow: Fortnox Package for warehouse_StockTaking operations that could not be mapped to a table.

<p>Remove single row by id from the Stock Taking document.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_stocktaking.deletestocktakingrow. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Stock Taking document id.
p_row Id	string	☑		row id

31.1.4 fnx_warehouse_stocktaking.deletestocktakingrowbyfilter: Fortnox Package for warehouse_StockTaking operations that could not be mapped to a table.

<p>Remove all rows matching the filter parameters from the Stock Taking document.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_stocktaking.deletestocktakingrowbyfilter. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Stock Taking document id.
p_transactionDate	datetime	☐		
p_itemIdSearch	string	☐		
p_itemDescriptionSearch	string	☐		
p_excludeZeroBalanceItems	boolean	☐		
p_financialyear	int64	☐		Selects w hat financial year that should be used (does not w ork on every table).
p_financialyeardate	datetime	☐		Selects by date, w hat financial year that should be used (does not w ork on every table).
p_fromdate	datetime	☐		Defines a selection based on a start date. Only available for invoices, orders, offers,

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				vouchers and supplierinvoices (does not work on every table).
p_lastmodified	datetime	☐		Retrieves all records since the provided timestamp (does not work on every table).
p_todate	datetime	☐		Defines a selection based on an end date. Only available for invoices, orders, offers, vouchers and supplierinvoices (does not work on every table).

31.1.5 fnx_warehouse_stocktaking.release: Fortnox Package for warehouse_StockTaking operations that could not be mapped to a table.

<p>The document will be locked and bookkept.The Stock Taking document state will be set to <code>completed</code>.The stock amount will be adjusted according to the stock taken quantity.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_stocktaking.release. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Stock Taking document id.

31.1.6 fnx_warehouse_stocktaking.voidstocktaking: Fortnox Package for warehouse_StockTaking operations that could not be mapped to a table.

<p>Sets the Stock Taking document state to <code>voided</code>.<p>Only documents in state <code>planning</code> and <code>started</code>can be voided. A <code>completed</code> document may not be voided.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_stocktaking.voidstocktaking. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Stock Taking document id.

32 Package: fnx_warehouse_stocktransfer

32.1 Procedures

32.1.1 fnx_warehouse_stocktransfer.create: Fortnox Package for warehouse_StockTransfer operations that could not be mapped to a table.

<p>Outbounds will be reserved in the from-place.Inbounds are created upon release of the stock transfer document.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_stocktransfer.create. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☐		
note	string	☐		
released	boolean	☐		
transferDate	datetime	☐		
version	int64	☐		
voided	boolean	☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

32.1.2 fnx_warehouse_stocktransfer.release: Fortnox Package for warehouse_StockTransfer operations that could not be mapped to a table.

<p>This will deliver all outbounds which are reserved in from-place, andcreate inbounds in the to-place.Nothing happens if you releasr an already released stock transfer document.<p>Returns <code>document_is_voided</code> if document is voided.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_stocktransfer.release. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Stock Transfer document id.

32.1.3 fnx_warehouse_stocktransfer.update: Fortnox Package for warehouse_StockTransfer operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_stocktransfer.update. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Stock Transfer document id.
id	int64	☐		
note	string	☐		
released	boolean	☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
transferDate	datetime	☐		
version	int64	☐		
voided	boolean	☐		

32.1.4 fnx_warehouse_stocktransfer.voidstocktransfer: Fortnox Package for warehouse_StockTransfer operations that could not be mapped to a table.

<p>Voiding a released stock transfer document is not allowed, andwill return <code>cannot_modify_released_document</code>

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure fnx_warehouse_stocktransfer.voidstocktransfer. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
p_id	int64	☑		Stock Transfer document id.
p_force	boolean	☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Index

- @ -

@url_2 31, 32, 151, 153

- A -

AbsenceTransaction__url 17
AbsenceTransaction_CauseCode 17
AbsenceTransaction_CostCenter 17
AbsenceTransaction_Date 17
AbsenceTransaction_EmployeeId 17
AbsenceTransaction_Extent 17
AbsenceTransaction_HolidayEntitling 17
AbsenceTransaction_Hours 17
AbsenceTransaction_id 17
AbsenceTransaction_Project 17
AbsenceTransactionById 17
AbsenceTransactionsByEmployeeId 19
AbsenceTransactionsByEmployeeIdAndDateAndCode 20
Account 67, 140, 170, 257, 282, 298, 359, 364
Account__url 23
Account_Active 23
Account_BalanceBroughtForward 23
Account_BalanceCarriedForward 23
Account_CostCenter 23
Account_CostCenterSettings 23
Account_Description 23
Account_Number 23
Account_Project 23
Account_ProjectSettings 23
Account_QuantitySettings 23
Account_QuantityUnit 23
Account_SRU 23
Account_TransactionInformation 23
Account_TransactionInformationSettings 23
Account_VATCode 23
Account_Year 23
ACCOUNTACTIVE 359
AccountActiveBalances 359
AccountAsset 55
AccountAssetId 55
AccountByNumber 23
AccountCharts 22
AccountChartType 147
AccountDepreciation 55
AccountDepreciationId 55
ACCOUNTDESCRIPTION 298, 359
AccountingMethod 147, 192, 197, 203, 208, 298, 306
ACCOUNTNUMBER 74, 82, 179, 197, 208, 224, 235, 249, 293, 359
accountnumberfrom 216
accountnumberto 216
AccountOpeningQuantitiesByAccountNumber 25
AccountRevaluation 55
AccountRevaluationId 55
Accounts 27
AccountSaleLoss 55
AccountSaleLossId 55
AccountSaleWin 55
AccountSaleWinId 55
AccountValueLoss 55
AccountValueLossId 55
AccountWriteDown 55
AccountWriteDownAck 55
AccountWriteDownAckId 55
AccountWriteDownId 55
ACCOUNTYEAR 359
AccrualAccount 67, 170, 282
AcquisitionDate 47, 50, 52
AcquisitionStart 47
AcquisitionValue 47, 50, 52
activated 464
activationTime 163, 169
active 25, 27, 74, 86, 311, 345, 437, 439, 442, 444, 445, 447
add-odata-mandatory-filters 2
Address1 100, 106, 111, 115, 120, 124, 128, 133, 192, 197, 203, 208, 231, 235, 245, 249, 311
Address2 100, 106, 111, 115, 120, 124, 128, 133, 192, 197, 203, 208, 231, 235, 245, 249, 311
AdministrationFee 74, 82, 192, 197, 203, 208, 231, 235, 245, 249, 298, 306
AdministrationFeeVAT 192, 197, 203, 208, 231, 235, 245, 249
Amount 47, 156, 159, 162, 165, 167, 177, 179, 276, 292, 293
Amount_2 179, 293
AmountCurrency 179, 293
analysis-enforce-row-uniqueness 2
api-access-token 2
api-client-id 2
api-client-secret 2
api-pre-expiry-refresh-sec 2
api-redirect-url 2
api-refresh-token 2
api-scope 2

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

api-token-url 2
api-url 2
appId 157, 158, 159, 161, 162, 163, 165, 166, 167, 169
ApprovalState 364, 366, 367, 371
ApprovedAmount 313, 317
ApprovedDays 106, 111
Approver_Id 372
Approver_Name 372
AppSalesByAppId 157
AppSalesByAppIdAndTenantId 158
AppSalesTenantPurchasesByAppId 159
AppSalesTenantsByAppId 161
AppSalesTenantTerminationsByAppId 162
AppSalesTenantUsersByAppId 163
AppSaleTenantPurchasesByAppIdAndTenantId 165
AppSaleTenantsByAppIdAndTenantId 166
AppSaleTenantTerminationsByAppIdAndTenantId 167
AppSaleTenantUsersByAppIdAndTenantId 169
Archive 28
ArchiveById 30
ArchiveFileId 31, 151
ArchiveFiles 31
ArchiveFolders 32
Article__url 36
Article_Active 36
Article_ArticleNumber 36
Article_Bulky 36
Article_ConstructionAccount 36
Article_CostCalculationMethod 36
Article_DefaultStockLocation 36
Article_DefaultStockPoint 36
Article_Depth 36
Article_Description 36
Article_DirectCost 36
Article_DisposableQuantity 36
Article_EAN 36
Article_EUAccount 36
Article_EUVATAccount 36
Article_Expired 36
Article_ExportAccount 36
Article_FreightCost 36
Article_Height 36
Article_Housework 36
Article_HouseworkType 36
Article_Manufacturer 36
Article_ManufacturerArticleNumber 36
Article_Note 36
Article_OtherCost 36
Article_PurchaseAccount 36
Article_PurchasePrice 36
Article_QuantityInStock 36
Article_ReservedQuantity 36
Article_SalesAccount 36
Article_SalesPrice 36
Article_StockAccount 36
Article_StockChangeAccount 36
Article_StockGoods 36
Article_StockPlace 36
Article_StockValue 36
Article_StockWarning 36
Article_SupplierName 36
Article_SupplierNumber 36
Article_Type 36
Article_Unit 36
Article_VAT 36
Article_WebshopArticle 36
Article_Weight 36
Article_Width 36
ArticleByNumber 36
articledescription 216
ArticleFileConnection__url 34
ArticleFileConnection_ArticleNumber 34
ArticleFileConnection_FileId 34
ArticleFileConnectionById 34
ArticleFileConnections 35
ArticleFirstPriceByPriceListAndArticleNumber 264
ArticleNumber 35, 36, 39, 43, 74, 82, 197, 208, 216, 235, 249, 264, 266, 269, 298
ArticlePriceByPriceListAndArticleNumberAndQuantity 266
Articles 39
ArticleUrlConnection__url 41
ArticleUrlConnection_ArticleNumber 41
ArticleUrlConnection_Id 41
ArticleUrlConnection_URLConnection 41
ArticleUrlConnectionByEmployeeIdAndDateAndCode 57
ArticleUrlConnectionById 41
ArticleUrlConnections 43
AskedAmount 313
AskedAmount_2 313
AssetByGivenNumber 45
AssetFileConnectors 44
AssetHistoryByGivenAssetNumber 47
AssetId 44
Assets 50
Assets__url 45
Assets_AcquisitionDate 45
Assets_AcquisitionStart 45
Assets_AcquisitionValue 45
Assets_Brand 45

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Assets_CostCenter 45
Assets_Department 45
Assets_DepreciatedTo 45
Assets_DepreciateToResidualValue 45
Assets_DepreciationFinal 45
Assets_DepreciationMethod 45
Assets_Description 45
Assets_Group 45
Assets_Id 45
Assets_InsuredNumber 45
Assets_InsuredWith 45
Assets_ManualOb 45
Assets_Notes 45
Assets_Number 45
Assets_Placement 45
Assets_Project 45
Assets_Reference 45
Assets_Room 45
Assets_Status 45
Assets_StatusId 45
Assets_Type 45
Assets_TypeId 45
AssetTypeById 53
AssetTypes 55
AttachmentCountByEntityType 141
AttachmentsByEntityType 143
AttendanceTransaction_CauseCode 59
AttendanceTransaction_CostCenter 59
AttendanceTransaction_Date 59
AttendanceTransaction_EmployeeId 59
AttendanceTransaction_Hours 59
AttendanceTransaction_Project 59
AttendanceTransactionById 59
AttendanceTransactions 60
AuthorizerName 301, 303
AutoNonRecurringTax 106, 111, 115, 120, 124, 128, 133
availableStock 449
averageCostInSEK 381
AverageHourlyWage 106, 111, 115, 120, 124, 128, 133
AverageWeeklyHours 106, 111, 115, 120, 124, 128, 133

- B -

backOrderQuantity 387, 411, 421, 425, 429
Balance 25, 192, 197, 203, 208, 216, 298, 301, 303, 306
BALANCEBROUGHTFORWARD 25, 27, 359
BALANCECARRIEDFORWARD 25, 359
BankAccountNo 106, 111, 115, 120, 124, 128, 133
BankAccountNumber 311
BasisTaxReduction 74, 192, 197, 203, 208, 231, 235, 245, 249
batch 377, 387, 394, 401, 402, 404, 406
BG 311
BIC 311
BilledAmount 313
BLOB Preferred 225
BLOB_PREFERRED 225
BOL_RESPONSE_CACHE_MAX_AGE_SEC 225
Booked 177, 179, 192, 197, 203, 208, 216, 292, 293, 298, 301, 303, 306
Brand 47
bulk-delete-page-size-rows 2
bulk-insert-page-size-bytes 2
bulk-insert-page-size-rows 2
BuyRate 88

- C -

c_url 19, 20, 25, 27, 31, 32, 35, 39, 43, 44, 47, 50, 52, 55, 57, 60, 67, 69, 74, 78, 82, 84, 86, 88, 100, 106, 115, 124, 133, 140, 147, 151, 153, 170, 172, 177, 179, 192, 197, 203, 208, 216, 224, 231, 235, 239, 245, 249, 254, 257, 259, 263, 269, 273, 276, 282, 284, 288, 292, 293, 298, 301, 303, 306, 311, 313, 317, 320, 322, 354, 363, 364, 366, 367, 371, 372, 467
c_urlTaxReductionList 74, 192, 197, 203, 208, 231, 235, 245, 249
Cancel 301, 303
Cancelled 192, 197, 203, 208, 216, 231, 235, 239, 245, 249, 254, 298, 306
category 375, 376
CauseCode 19, 20, 57, 60
chargeHours 341, 345
Child 106, 111
childId 341, 345
City 100, 106, 111, 115, 120, 124, 128, 133, 192, 197, 203, 208, 231, 235, 245, 249, 311
ClearingNo 106, 111, 115, 120, 124, 128, 133
code_2 437, 442, 445
CodeEnglish 354
comment 154
Comments 31, 74, 151, 192, 197, 203, 208, 231, 235, 245, 249, 263, 298, 306, 364, 366, 367, 371
companyEmployeeRange 154
CompanyInformation 62
CompanyInformation_Address 62
CompanyInformation_City 62
CompanyInformation_CompanyName 62
CompanyInformation_CountryCode 62

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

CompanyInformation_DatabaseNumber 62
CompanyInformation_OrganizationNumber 62
CompanyInformation_VisitAddress 62
CompanyInformation_VisitCity 62
CompanyInformation_VisitCountryCode 62
CompanyInformation_VisitZipCode 62
CompanyInformation_ZipCode 62
CompanySettings 63
CompanySettings_Address 63
CompanySettings_BG 63
CompanySettings_BIC 63
CompanySettings_BranchCode 63
CompanySettings_City 63
CompanySettings_ContactFirstName 63
CompanySettings_ContactLastName 63
CompanySettings_Country 63
CompanySettings_CountryCode 63
CompanySettings_DatabaseNumber 63
CompanySettings_Domicile 63
CompanySettings_Email 63
CompanySettings_Fax 63
CompanySettings_IBAN 63
CompanySettings_Name 63
CompanySettings_OrganizationNumber 63
CompanySettings_PG 63
CompanySettings_Phone1 63
CompanySettings_Phone2 63
CompanySettings_TaxEnabled 63
CompanySettings_VATNumber 63
CompanySettings_VisitAddress 63
CompanySettings_VisitCity 63
CompanySettings_VisitCountry 63
CompanySettings_VisitCountryCode 63
CompanySettings_VisitName 63
CompanySettings_VisitZipCode 63
CompanySettings_WWW 63
CompanySettings_ZipCode 63
completed 386, 387, 390
confirmationEmail 411, 417, 421, 425, 429, 434
Content Type 225
CONTENT_TYPE 225
Continuous 74, 78, 82
Contract__url 70
Contract__urlTaxReductionList 70
Contract_Active 70
Contract_AdministrationFee 70
Contract_BasisTaxReduction 70
Contract_Comments 70
Contract_Continuous 70
Contract_ContractDate 70
Contract_ContractLength 70
Contract_ContributionPercent 70
Contract_ContributionValue 70
Contract_CostCenter 70
Contract_Currency 70
Contract_CustomerName 70
Contract_CustomerNumber 70
Contract_DocumentNumber 70
Contract_EmailInformation_EmailAddressBCC 70
Contract_EmailInformation_EmailAddressCC 70
Contract_EmailInformation_EmailAddressFrom 70
Contract_EmailInformation_EmailAddressTo 70
Contract_EmailInformation_EmailBody 70
Contract_EmailInformation_EmailSubject 70
Contract_ExternalInvoiceReference1 70
Contract_ExternalInvoiceReference2 70
Contract_Freight 70
Contract_Gross 70
Contract_HouseWork 70
Contract_InvoiceDiscount 70
Contract_InvoiceInterval 70
Contract_InvoicesRemaining 70
Contract_Language 70
Contract_LastInvoiceDate 70
Contract_Net 70
Contract_OurReference 70
Contract_PeriodEnd 70
Contract_PeriodStart 70
Contract_PriceList 70
Contract_PrintTemplate 70
Contract_Project 70
Contract_Remarks 70
Contract_RoundOff 70
Contract_TaxReduction 70
Contract_TaxReductionType 70
Contract_TemplateName 70
Contract_TemplateNumber 70
Contract_TermsOfDelivery 70
Contract_TermsOfPayment 70
Contract_Total 70
Contract_TotalToPay 70
Contract_TotalVAT 70
Contract_VatIncluded 70
Contract_WayOfDelivery 70
Contract_YourOrderNumber 70
Contract_YourReference 70
ContractAccrual__url 66
ContractAccrual_AccrualAccount 66
ContractAccrual_CostAccount 66
ContractAccrual_Description 66
ContractAccrual_DocumentNumber 66
ContractAccrual_Period 66

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ContractAccrual_Times 66
ContractAccrual_Total 66
ContractAccrual_VATIncluded 66
ContractAccrualByDocumentNumber 66
ContractAccrualLinesByDocumentNumber 67
ContractAccruals 69
ContractByDocumentNumber 70
ContractDate 74
ContractInvoiceLinesByDocumentNumber 74
ContractLength 74, 78, 82, 84
ContractReference 192, 197, 203, 208
Contracts 78
ContractTemplate 84
ContractTemplate__url 80
ContractTemplate_AdministrationFee 80
ContractTemplate_Continuous 80
ContractTemplate_ContractLength 80
ContractTemplate_Freight 80
ContractTemplate_InvoiceInterval 80
ContractTemplate_OurReference 80
ContractTemplate_PrintTemplate 80
ContractTemplate_Remarks 80
ContractTemplate_TemplateName 80
ContractTemplate_TemplateNumber 80
ContractTemplate_TermsOfDelivery 80
ContractTemplate_TermsOfPayment 80
ContractTemplate_WayOfDelivery 80
ContractTemplateByTemplateNumber 80
ContractTemplateInvoiceLinesByTemplateNumber 82
ContractTemplateName 84
ContractTemplates 84
ContributionPercent 74, 192, 197, 203, 208, 231, 235, 245, 249
ContributionPercent_2 74, 197, 208, 235, 249
ContributionValue 74, 192, 197, 203, 208, 231, 235, 245, 249
ContributionValue_2 74, 197, 208, 235, 249
CopyRemarks 231, 235, 245, 249
CostAccount 67, 282
CostCenter 19, 20, 25, 27, 47, 57, 60, 67, 74, 82, 106, 111, 115, 120, 124, 128, 133, 170, 179, 192, 197, 203, 208, 216, 231, 235, 239, 245, 249, 254, 276, 282, 293, 298, 301, 303, 306, 311, 359, 364
CostCenter__url 85
CostCenter_2 74, 197, 208, 235, 249, 298, 364
costCenter_active 85, 323, 328, 333, 337, 341, 345
CostCenter_Code 85
CostCenter_Description 85
costCenter_id 323, 328, 333, 337, 341, 345
costCenter_name 323, 328, 333, 337, 341, 345
CostCenter_Note 85
CostCenterByCode 85
costCenterCode 377, 384, 386, 387, 394, 397, 401, 402, 404, 406, 411, 417, 421, 425, 429, 434, 452, 455, 458, 459
costCenterCode_2 387, 411, 421, 425, 429
CostCenters 86
CostCenterSettings 25, 27
countedBy 450, 452, 454, 455
Country 106, 111, 115, 120, 124, 128, 133, 192, 197, 203, 208, 231, 235, 245, 249
CountryCode 311
created 154
createdBy 323, 328, 337, 341, 345
createdTime 323, 328, 333, 337, 341, 345
createdTime_2 323, 328, 337
Credit 67, 170, 192, 197, 203, 208, 216, 282, 298, 301, 303, 306, 359, 364
Credit_2 298
CreditCurrency 298
CreditInvoiceReference 192, 197, 203, 208
CreditReference 298, 306
Currencies 88
currency 74, 78, 177, 179, 192, 197, 203, 208, 216, 231, 235, 239, 245, 249, 254, 292, 293, 298, 301, 303, 306, 311, 394, 396
Currency__url 89
Currency_2 179, 293
Currency_BuyRate 89
Currency_Code 89
currency_currency 377, 379
Currency_Date 89
Currency_Description 89
Currency_IsAutomatic 89
currency_rate 377, 379
Currency_SellRate 89
currency_unit 89, 377, 379
CurrencyByCode 89
currencyCode 411, 417, 421, 425, 429, 434
currencyCode_2 411, 421, 425, 429
currencyRate 177, 179, 192, 197, 203, 208, 216, 231, 235, 245, 249, 292, 293, 298, 301, 303, 306, 394, 396, 411, 417, 421, 425, 429, 434
currencyUnit 177, 179, 192, 197, 203, 208, 216, 231, 235, 245, 249, 292, 293, 298, 301, 303, 306, 394, 396, 411, 417, 421, 425, 429, 434
currentRowNo 450, 452, 454, 455
CustomDocumentTypeByType 375
CustomDocumentTypes 376
customer 92, 94
Customer__url 95
customer_accountManager 323, 328, 333, 337, 341, 345
customer_active 95, 323, 328, 333, 337, 341, 345

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Customer_Address1 95
Customer_Address2 95
Customer_City 95
Customer_Comments 95
Customer_CostCenter 95
Customer_Country 95
Customer_CountryCode 95
Customer_Currency 95
Customer_CustomerNumber 95
Customer_DefaultDeliveryTypes_Invoice 95
Customer_DefaultDeliveryTypes_Offer 95
Customer_DefaultDeliveryTypes_Order 95
customer_defaultProjectId 323, 328, 333, 337, 341, 345
Customer_DefaultTemplates_CashInvoice 95
Customer_DefaultTemplates_Invoice 95
Customer_DefaultTemplates_Offer 95
Customer_DefaultTemplates_Order 95
customer_deleted 323, 328, 333, 337, 341, 345
Customer_DeliveryAddress1 95
Customer_DeliveryAddress2 95
Customer_DeliveryCity 95
Customer_DeliveryCountry 95
Customer_DeliveryCountryCode 95
Customer_DeliveryFax 95
Customer_DeliveryName 95
Customer_DeliveryPhone1 95
Customer_DeliveryPhone2 95
Customer_DeliveryZipCode 95
Customer_Email 95
Customer_EmailInvoice 95
Customer_EmailInvoiceBCC 95
Customer_EmailInvoiceCC 95
Customer_EmailOffer 95
Customer_EmailOfferBCC 95
Customer_EmailOfferCC 95
Customer_EmailOrder 95
Customer_EmailOrderBCC 95
Customer_EmailOrderCC 95
Customer_ExternalReference 95
Customer_Fax 95
Customer_GLN 95
Customer_GLNDelivery 95
customer_id 323, 328, 333, 337, 341, 345
Customer_InvoiceAdministrationFee 95
customer_invoiceCity 323, 328, 333, 337, 341, 345
Customer_InvoiceDiscount 95
Customer_InvoiceFreight 95
Customer_InvoiceRemark 95
customer_invoiceZipCode 323, 328, 333, 337, 341, 345
customer_isBusiness 323, 328, 333, 337, 341, 345
customer_name 95, 323, 328, 333, 337, 341, 345
customer_number 323, 328, 333, 337, 341, 345
Customer_OrganisationNumber 95
Customer_OurReference 95
Customer_Phone1 95
Customer_Phone2 95
customer_priceList 95, 323, 328, 333, 337, 341, 345
Customer_Project 95
customer_propertyDesignation 323, 328, 333, 337, 341, 345
Customer_SalesAccount 95
Customer_ShowPriceVATIncluded 95
Customer_TermsOfDelivery 95
Customer_TermsOfPayment 95
Customer_Type 95
Customer_VATNumber 95
Customer_VATType 95
Customer_VisitingAddress 95
Customer_VisitingCity 95
Customer_VisitingCountry 95
Customer_VisitingCountryCode 95
Customer_VisitingZipCode 95
Customer_WayOfDelivery 95
Customer_WWW 95
Customer_YourReference 95
Customer_ZipCode 95
CustomerByNumber 95
customerId 411, 417, 421, 425, 429, 434
customerName 74, 78, 192, 197, 203, 208, 216, 231, 235, 239, 245, 249, 254, 313, 317, 411, 417, 421, 425, 429, 434
customerNumber 74, 78, 92, 93, 95, 100, 192, 197, 203, 208, 216, 231, 235, 239, 245, 249, 254, 411, 417, 421, 425, 429, 434
CustomerReferenceById 90
CustomerReferenceLines 92
CustomerReferenceLinesByReferenceId 93
CustomerReferenceRowId 90, 93
CustomerReferences 94
Customers 100
CustomInboundDocumentRows 377
CustomInboundDocuments 379
CustomOutboundDocumentAverageCostsByTypeAndId 381
CustomOutboundDocumentByTypeAndId 382
CustomOutboundDocumentLinesByTypeAndId 384

- D -

Database Driver 1

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

date 19, 20, 47, 57, 60, 88, 147, 156, 276, 277, 377, 379, 381, 382, 384, 386, 387, 390, 392, 394, 396, 397, 399, 452, 455, 458, 459
DATE_ENDED 225
DATE_STARTED 225
DaysEarned 357
DaysSaved 357
DaysUnused 357
Debit 67, 170, 282, 298, 359, 364
DebitCurrency 298
DebtAdvance 357
DebtEarned 357
DebtSaved 357
DebtUnused 357
deliveredQuantity 74, 82, 197, 208, 249, 384, 397
deliveryAddress 411, 417, 421, 425, 429, 434, 437, 439, 442, 444, 445, 447
DeliveryAddress1 192, 197, 203, 208, 231, 235, 245, 249
deliveryAddress2 192, 197, 203, 208, 231, 235, 245, 249, 411, 417, 421, 425, 429, 434, 437, 439, 442, 444, 445, 447
deliveryCity 192, 197, 203, 208, 231, 235, 245, 249, 411, 417, 421, 425, 429, 434, 437, 439, 442, 444, 445, 447
DeliveryCountry 192, 197, 203, 208, 231, 235, 245, 249
deliveryCountryCode 411, 417, 421, 425, 429, 434, 437, 439, 442, 444, 445, 447
deliveryDate 192, 197, 203, 208, 231, 235, 245, 249, 254, 411, 417, 421, 425, 429, 434
deliveryId 392
DeliveryMethodByCode 465
DeliveryMethods 467
deliveryName 192, 197, 203, 208, 231, 235, 245, 249, 411, 417, 421, 425, 429, 434, 437, 439, 442, 444, 445, 447
deliveryNote 390
deliveryNoteId 386, 387, 390
deliveryPhone 437, 439, 442, 444, 445, 447
deliveryState 245, 249, 381, 382, 384
DeliveryTermByCode 318
DeliveryTerms 320
deliveryZipCode 192, 197, 203, 208, 231, 235, 245, 249, 411, 417, 421, 425, 429, 434, 437, 439, 442, 444, 445, 447
Department 47
Deprecations 52
DepreciatedTo 47, 50, 52
DepreciateToResidualValue 47
DepreciationFinal 47, 50, 52
DepreciationMethod 47
Description_2 364
DescriptionEnglish 320, 467
directCost 377, 387, 394
DisablePaymentFile 298, 306, 311
Discount 74, 82, 197, 208, 235, 249
DiscountType 74, 82, 197, 208, 235, 249
DisposableQuantity 39
documentId 323, 328, 337, 341, 345
DocumentNumber 66, 67, 69, 70, 74, 78, 181, 186, 192, 197, 203, 208, 214, 215, 216, 227, 231, 235, 239, 241, 245, 249, 254
documentState 401, 402, 404, 406
documentType 323, 328, 337, 341, 345
download-error-400-bad-request-max-tries 2
download-error-400-bad-request-sleep-initial-ms 2
download-error-400-bad-request-sleep-max-ms 2
download-error-400-bad-request-sleep-multiplicator 2
download-error-403-forbidden-max-tries 2
download-error-403-forbidden-sleep-initial-ms 2
download-error-403-forbidden-sleep-max-ms 2
download-error-403-forbidden-sleep-multiplicator 2
download-error-408-request-timeout-max-tries 2
download-error-408-request-timeout-sleep-initial-ms 2
download-error-408-request-timeout-sleep-max-ms 2
download-error-408-request-timeout-sleep-multiplicator 2
download-error-422-bad-request-max-tries 2
download-error-422-bad-request-sleep-initial-ms 2
download-error-422-bad-request-sleep-max-ms 2
download-error-422-bad-request-sleep-multiplicator 2
download-error-429-too-many-requests-max-tries 2
download-error-429-too-many-requests-sleep-initial-ms 2
download-error-429-too-many-requests-sleep-max-ms 2
download-error-429-too-many-requests-sleep-multiplicator 2
download-error-500-internal-server-error-max-tries 2
download-error-500-internal-server-error-sleep-initial-ms 2
download-error-500-internal-server-error-sleep-max-ms 2
download-error-500-internal-server-error-sleep-multiplicator 2
download-error-502-server-unavailable-max-tries 2
download-error-502-server-unavailable-sleep-initial-ms 2
download-error-502-server-unavailable-sleep-max-ms 2
download-error-502-server-unavailable-sleep-multiplicator 2

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

download-error-503-server-unavailable-max-tries 2
download-error-503-server-unavailable-sleep-initial-ms 2
download-error-503-server-unavailable-sleep-max-ms 2
download-error-503-server-unavailable-sleep-multiplicator 2
download-error-504-gateway-timeout-max-tries 2
download-error-504-gateway-timeout-sleep-initial-ms 2
download-error-504-gateway-timeout-sleep-max-ms 2
download-error-504-gateway-timeout-sleep-multiplicator 2
download-error-590-network-connect-timeout-max-tries 2
download-error-590-network-connect-timeout-sleep-initial-ms 2
download-error-590-network-connect-timeout-sleep-max-ms 2
download-error-590-network-connect-timeout-sleep-multiplicator 2
download-error-599-network-connect-timeout-max-tries 2
download-error-599-network-connect-timeout-sleep-initial-ms 2
download-error-599-network-connect-timeout-sleep-max-ms 2
download-error-599-network-connect-timeout-sleep-multiplicator 2
download-error-argument-exception-max-tries 2
download-error-argument-exception-sleep-initial-ms 2
download-error-argument-exception-sleep-max-ms 2
download-error-argument-exception-sleep-multiplicator 2
download-error-internet-down-max-tries 2
download-error-internet-down-sleep-initial-ms 2
download-error-internet-down-sleep-max-ms 2
download-error-internet-down-sleep-multiplicator 2
download-error-io-exception-max-tries 2
download-error-io-exception-sleep-initial-ms 2
download-error-io-exception-sleep-max-ms 2
download-error-io-exception-sleep-multiplicator 2
download-error-json-exception-max-tries 2
download-error-json-exception-sleep-initial-ms 2
download-error-json-exception-sleep-max-ms 2
download-error-json-exception-sleep-multiplicator 2
download-error-name-resolution-failure-max-tries 2
download-error-name-resolution-failure-sleep-initial-ms 2
download-error-name-resolution-failure-sleep-max-ms 2
download-error-name-resolution-failure-sleep-multiplicator 2
download-error-other-exception-max-tries 2
download-error-other-exception-sleep-initial-ms 2
download-error-other-exception-sleep-max-ms 2
download-error-other-exception-sleep-multiplicator 2
download-error-socket-exception-max-tries 2
download-error-socket-exception-sleep-initial-ms 2
download-error-socket-exception-sleep-max-ms 2
download-error-socket-exception-sleep-multiplicator 2
download-error-web-exception-max-tries 2
download-error-web-exception-sleep-initial-ms 2
download-error-web-exception-sleep-max-ms 2
download-error-web-exception-sleep-multiplicator 2
download-error-web-not-found-max-tries 2
download-error-web-not-found-sleep-initial-ms 2
download-error-web-not-found-sleep-max-ms 2
download-error-web-not-found-sleep-multiplicator 2
download-error-web-not-implemented-max-tries 2
download-error-web-not-implemented-sleep-initial-ms 2
download-error-web-not-implemented-sleep-max-ms 2
download-error-web-not-implemented-sleep-multiplicator 2
download-error-web-timeout-max-tries 2
download-error-web-timeout-sleep-initial-ms 2
download-error-web-timeout-sleep-max-ms 2
download-error-web-timeout-sleep-multiplicator 2
download-error-web-unauthorized-max-tries 2
download-error-web-unauthorized-sleep-initial-ms 2
download-error-web-unauthorized-sleep-max-ms 2
download-error-web-unauthorized-sleep-multiplicator 2
download-error-web-unknown-max-tries 2
download-error-web-unknown-sleep-initial-ms 2
download-error-web-unknown-sleep-max-ms 2
download-error-web-unknown-sleep-multiplicator 2
dropship 411, 417, 421, 425, 429, 434
DRY_RUN 225
DueDate 192, 197, 203, 208, 216, 298, 301, 303, 306
DUMMY 90, 94, 350
duplicate-row-mode 2
Duration (ms) 225
DURATION_MS 225

- E -

EAN 39

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

EDIInformation_EDIGlobalLocationNumber 192, 197, 203, 208
EDIInformation_EDIGlobalLocationNumberDelivery 192, 197, 203, 208
EDIInformation_EDIInvoiceExtra1 192, 197, 203, 208
EDIInformation_EDIInvoiceExtra2 192, 197, 203, 208
EDIInformation_EDIOurElectronicReference 192, 197, 203, 208
EDIInformation_EDIStatus 192, 197, 203, 208
EDIInformation_EDIYourElectronicReference 192, 197, 203, 208
Email 31, 32, 100, 106, 111, 115, 120, 124, 128, 133, 151, 153, 163, 169, 311, 349, 352
EmailInformation_EmailAddressBCC 74, 192, 197, 203, 208, 231, 235, 245, 249
EmailInformation_EmailAddressCC 74, 192, 197, 203, 208, 231, 235, 245, 249
EmailInformation_EmailAddressFrom 74, 192, 197, 203, 208, 231, 235, 245, 249
EmailInformation_EmailAddressTo 74, 192, 197, 203, 208, 231, 235, 245, 249
EmailInformation_EmailBody 74, 192, 197, 203, 208, 231, 235, 245, 249
EmailInformation_EmailSubject 74, 192, 197, 203, 208, 231, 235, 245, 249
EmployedTo 106, 111, 115, 120, 124, 128, 133
Employee_Address1 101
Employee_Address2 101
Employee_AutoNonRecurringTax 101
Employee_AverageHourlyWage 101
Employee_AverageWeeklyHours 101
Employee_BankAccountNo 101
Employee_City 101
Employee_ClearingNo 101
Employee_CostCenter 101
Employee_Country 101
Employee_Email 101
Employee_EmployedTo 101
Employee_EmployeeId 101
Employee_EmploymentDate 101
Employee_EmploymentForm 101
Employee_FirstName 101
Employee_ForaType 101
Employee_FullName 101
Employee_HourlyPay 101
Employee_Inactive 101
Employee_JobTitle 101
Employee_LastName 101
Employee_MonthlySalary 101
Employee_NonRecurringTax 101
Employee_PersonalIdentityNumber 101
Employee_PersonelType 101
Employee_Phone1 101
Employee_Phone2 101
Employee_PostCode 101
Employee_Project 101
Employee_SalaryForm 101
Employee_ScheduleId 101
Employee_TaxAllowance 101
Employee_TaxColumn 101
Employee_TaxTable 101
Employee_VacationDaysPaid 101
Employee_VacationDaysPendingPaid 101
Employee_VacationDaysPendingPrepaid 101
Employee_VacationDaysPendingSaved 101
Employee_VacationDaysPendingSavedYear1 101
Employee_VacationDaysPendingSavedYear2 101
Employee_VacationDaysPendingSavedYear3 101
Employee_VacationDaysPendingSavedYear4 101
Employee_VacationDaysPendingSavedYear5 101
Employee_VacationDaysPendingSavedYear6Plus 101
Employee_VacationDaysPendingUnpaid 101
Employee_VacationDaysPrepaid 101
Employee_VacationDaysRegisteredPaid 101
Employee_VacationDaysRegisteredPrepaid 101
Employee_VacationDaysRegisteredSaved 101
Employee_VacationDaysRegisteredSavedYear1 101
Employee_VacationDaysRegisteredSavedYear2 101
Employee_VacationDaysRegisteredSavedYear3 101
Employee_VacationDaysRegisteredSavedYear4 101
Employee_VacationDaysRegisteredSavedYear5 101
Employee_VacationDaysRegisteredSavedYear6Plus 101
Employee_VacationDaysRegisteredUnpaid 101
Employee_VacationDaysSaved 101
Employee_VacationDaysSavedEmploymentRateYear1 101
Employee_VacationDaysSavedEmploymentRateYear2 101
Employee_VacationDaysSavedEmploymentRateYear3 101
Employee_VacationDaysSavedEmploymentRateYear4 101
Employee_VacationDaysSavedEmploymentRateYear5 101
Employee_VacationDaysSavedEmploymentRateYear6Plus 101
Employee_VacationDaysSavedYear1 101

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Employee_VacationDaysSavedYear2 101
Employee_VacationDaysSavedYear3 101
Employee_VacationDaysSavedYear4 101
Employee_VacationDaysSavedYear5 101
Employee_VacationDaysSavedYear6Plus 101
Employee_VacationDaysUnpaid 101
EmployeeById 101
EmployeeChildren 106
EmployeeChildrenByEmployeeId 111
EmployeeDatedSchedules 115
EmployeeDatedSchedulesByEmployeeId 120
EmployeeDatedWages 124
EmployeeDatedWagesByEmployeeId 128
EmployeeId 19, 20, 57, 60, 101, 106, 111, 115, 120, 124, 128, 133, 276, 277, 357
EmployeeId_2 106, 111, 115, 120, 124, 128
EmployeeName 357
Employees 133
EmploymentDate 106, 111, 115, 120, 124, 128, 133
EmploymentForm 106, 111, 115, 120, 124, 128, 133
End Date 225
EndDate 170, 273, 282
entityId 141, 143, 392
entityType 141, 143
Error Message Code 225
Error Message Text 225
ERROR_MESSAGE_CODE 225
ERROR_MESSAGE_TEXT 225
EUQuarterlyReport 192, 197, 203, 208
EUVatLimitRegulation_IsOverLimit 137
EUVatLimitRegulation_Limit 137
EUVatLimitRegulation_TotalExclVat 137
EUVatLimitRegulation_Year 137
EUVatLimitRegulationByYear 137
EventId 47
excludeZeroBalanceItems 450, 454
Expense 276
Expense_Account 139
Expense_Code 139
Expense_Text 139
ExpenseByCode 139
ExpenseCode 139
Expenses 140
ExpireDate 231, 235
Extent 19, 20
ExternalInvoiceNumber 298, 301, 303, 306
ExternalInvoiceReference1 74, 179, 192, 197, 203, 208, 216, 245, 249, 254
ExternalInvoiceReference2 74, 179, 192, 197, 203, 208, 216, 245, 249, 254
ExternalInvoiceSeries 298, 301, 303, 306
externalSourceId 163, 169

- F -

Fail on Error 225
FAIL_ON_ERROR 225
FileId 28, 30, 31, 32, 34, 35, 44, 143, 286, 288, 361, 363
filter 39, 78, 100, 216, 239, 254, 301, 303, 317
FinalPayDate 192, 197, 203, 208, 216, 298, 301, 303, 306
FinancialInvoiceByNumber 144
financialyear 17, 19, 20, 22, 23, 25, 27, 28, 30, 31, 32, 34, 35, 36, 39, 41, 43, 44, 45, 47, 50, 52, 53, 55, 57, 59, 60, 62, 63, 66, 67, 69, 70, 74, 78, 80, 82, 84, 85, 86, 88, 89, 90, 92, 93, 94, 95, 100, 101, 106, 111, 115, 120, 124, 128, 133, 137, 139, 140, 141, 143, 144, 146, 147, 149, 150, 151, 153, 154, 156, 157, 158, 159, 161, 162, 163, 165, 166, 167, 169, 170, 172, 173, 175, 177, 179, 181, 186, 192, 197, 203, 208, 214, 215, 216, 219, 220, 221, 223, 224, 227, 231, 235, 239, 241, 245, 249, 254, 256, 257, 259, 260, 261, 263, 264, 266, 267, 269, 270, 271, 273, 274, 276, 277, 279, 280, 282, 284, 285, 286, 288, 289, 292, 293, 295, 298, 301, 303, 306, 308, 311, 313, 315, 317, 318, 320, 321, 322, 323, 328, 333, 337, 341, 345, 349, 350, 352, 353, 354, 355, 357, 361, 363, 364, 366, 367, 369, 371, 372, 373, 375, 376, 377, 379, 381, 382, 384, 386, 387, 390, 392, 394, 396, 397, 399, 401, 402, 404, 406, 408, 410, 411, 415, 417, 420, 421, 425, 429, 434, 437, 439, 441, 442, 444, 445, 447, 449, 450, 452, 454, 455, 458, 459, 461, 463, 464, 465, 467
FinancialYear__url 146
FinancialYear_AccountChartType 146
FinancialYear_AccountingMethod 146
FinancialYear_FromDate 146
FinancialYear_Id 146
FinancialYear_ToDate 146
financialyear1 279
FinancialYearById 146

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

financialyeardate 17, 19, 20, 22, 23, 25, 27, 28, 30, 31, 32, 34, 35, 36, 39, 41, 43, 44, 45, 47, 50, 52, 53, 55, 57, 59, 60, 62, 63, 66, 67, 69, 70, 74, 78, 80, 82, 84, 85, 86, 88, 89, 90, 92, 93, 94, 95, 100, 101, 106, 111, 115, 120, 124, 128, 133, 137, 139, 140, 141, 143, 144, 146, 147, 149, 150, 151, 153, 154, 156, 157, 158, 159, 161, 162, 163, 165, 166, 167, 169, 170, 172, 173, 175, 177, 179, 181, 186, 192, 197, 203, 208, 214, 215, 216, 219, 220, 221, 223, 224, 227, 231, 235, 239, 241, 245, 249, 254, 256, 257, 259, 260, 261, 263, 264, 266, 267, 269, 270, 271, 273, 274, 276, 277, 279, 280, 282, 284, 285, 286, 288, 289, 292, 293, 295, 298, 301, 303, 306, 308, 311, 313, 315, 317, 318, 320, 321, 322, 323, 328, 333, 337, 341, 345, 349, 350, 352, 353, 354, 355, 357, 361, 363, 364, 366, 367, 369, 371, 372, 373, 375, 376, 377, 379, 381, 382, 384, 386, 387, 390, 392, 394, 396, 397, 399, 401, 402, 404, 406, 408, 410, 411, 415, 417, 420, 421, 425, 429, 434, 437, 439, 441, 442, 444, 445, 447, 449, 450, 452, 454, 455, 458, 459, 461, 463, 464, 465, 467
FinancialYears 147
FirstDay 115, 120, 124, 128
FirstName 106, 111, 115, 120, 124, 128, 133
fnx 1
Folder__url 28, 149
Folder_Email 28, 149
Folder_Id 28, 149
Folder_Name 28, 149
ForaType 106, 111, 115, 120, 124, 128, 133
force-case-sensitive-identifiers 2
forced-casing-identifiers 2
forcedDelivery 381, 382, 384
forcedQuantity 384, 397
Fortnox 1, 17, 19, 20, 22, 23, 25, 27, 28, 30, 31, 32, 34, 35, 36, 39, 41, 43, 44, 45, 47, 50, 52, 53, 55, 57, 59, 60, 62, 63, 66, 67, 69, 70, 74, 78, 80, 82, 84, 85, 86, 88, 89, 90, 92, 93, 94, 95, 100, 101, 106, 111, 115, 120, 124, 128, 133, 137, 139, 140, 141, 143, 144, 146, 147, 149, 150, 151, 153, 154, 156, 157, 158, 159, 161, 162, 163, 165, 166, 167, 169, 170, 172, 173, 175, 177, 179, 181, 186, 192, 197, 203, 208, 214, 215, 216, 219, 220, 221, 223, 224, 225, 227, 231, 235, 239, 241, 245, 249, 254, 256, 257, 259, 260, 261, 263, 264, 266, 267, 269, 270, 271, 273, 274, 276, 277, 279, 280, 282, 284, 285, 286, 288, 289, 292, 293, 295, 298, 301, 303, 306, 308, 311, 313, 315, 317, 318, 320, 321, 322, 323, 328, 333, 337, 341, 345, 349, 350, 352, 353, 354, 355, 357, 359, 360, 361, 363, 364, 366, 367, 369, 371, 372, 373, 375, 376, 377, 379, 381, 382, 384, 386, 387, 390, 392, 394, 396, 397, 399, 401, 402, 404, 406, 408, 410, 411, 415, 417, 420, 421, 425, 429, 434, 437, 439, 441, 442, 444, 445, 447, 449, 450, 452, 454, 455, 458, 459, 461, 463, 464, 465, 467
Freight 74, 82, 192, 197, 203, 208, 231, 235, 245, 249, 298, 306
freightCost 377, 394
FreightVAT 192, 197, 203, 208, 231, 235, 245, 249
fromdate 17, 19, 20, 22, 23, 25, 27, 28, 30, 31, 32, 34, 35, 36, 39, 41, 43, 44, 45, 47, 50, 52, 53, 55, 57, 59, 60, 62, 63, 66, 67, 69, 70, 74, 78, 80, 82, 84, 85, 86, 88, 89, 90, 92, 93, 94, 95, 100, 101, 106, 111, 115, 120, 124, 128, 133, 137, 139, 140, 141, 143, 144, 146, 147, 149, 150, 151, 153, 154, 156, 157, 158, 159, 161, 162, 163, 165, 166, 167, 169, 170, 172, 173, 175, 177, 179, 181, 186, 192, 197, 203, 208, 214, 215, 216, 219, 220, 221, 223, 224, 227, 231, 235, 239, 241, 245, 249, 254, 256, 257, 259, 260, 261, 263, 264, 266, 267, 269, 270, 271, 273, 274, 276, 277, 279, 280, 282, 284, 285, 286, 288, 289, 292, 293, 295, 298, 301, 303, 306, 308, 311, 313, 315, 317, 318, 320, 321, 322, 323, 328, 333, 337, 341, 345, 349, 350, 352, 353, 354, 355, 357, 361, 363, 364, 366, 367, 369, 371, 372, 373, 375, 376, 377, 379, 381, 382, 384, 386, 387, 390, 392, 394, 396, 397, 399, 401, 402, 404, 406, 408, 410, 411, 415, 417, 420, 421, 425, 429, 434, 437, 439, 441, 442, 444, 445, 447, 449, 450, 452, 454, 455, 458, 459, 461, 463, 464, 465, 467
fromdate1 323, 328, 333, 337, 341, 345
fromfinalpaydate 216
FromQuantity 266, 269
fromStockLocationCode 461
fromStockLocationId 461
fromStockLocationName 461
fromStockPointCode 461
fromStockPointId 461
fromStockPointName 461
FullName 106, 111, 115, 120, 124, 128, 133

- G -

GivenNumber 45, 47, 295, 298, 301, 303, 306
gln 100
gndelivery 100
Gross 74, 192, 197, 203, 208, 231, 235, 245, 249
Group 47

- H -

HasDeliveryNote 386, 387, 390
HasPostReleaseStockChanges 450, 452, 454, 455
HasWarehouseFeatures 464
HolidayEntitling 19, 20
HourlyPay 106, 111, 115, 120, 124, 128, 133
HourlyPay_2 124, 128
Hours 19, 20, 57, 60
HouseWork 39, 74, 192, 197, 203, 208, 231, 235, 245, 249
HouseWork_2 74, 197, 208, 235, 249
HouseWorkHoursToReport 74, 197, 208, 235, 249

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

HouseWorkType 74, 197, 208, 235, 249
HTTP Disk Cache Maximum Age (sec) 225
HTTP Memory Cache Maximum Age (sec) 225
HTTP Method 225
HTTP Status Code 225
HTTP_DISK_CACHE_MAX_AGE_SEC 225
HTTP_DISK_CACHE_SAVE 225
HTTP_DISK_CACHE_USE 225
HTTP_MEMORY_CACHE_MAX_AGE_SEC 225
HTTP_MEMORY_CACHE_SAVE 225
HTTP_MEMORY_CACHE_USE 225
HTTP_METHOD 225
HTTP_STATUS_CODE 225
http-disk-cache-compression-level 2
http-disk-cache-directory 2
http-disk-cache-ignore-write-errors 2
http-disk-cache-max-age-sec 2
http-get-timeout-max-ms 2
http-get-timeout-ms 2
http-memory-cache-compression-level 2
http-memory-cache-max-age-sec 2
http-post-timeout-max-ms 2
http-post-timeout-ms 2

- I -

IBAN 311
id 17, 20, 30, 41, 53, 57, 59, 146, 150, 285, 313, 315, 377, 379, 381, 382, 384, 386, 387, 394, 396, 397, 399, 401, 402, 408, 410, 415, 417, 421, 425, 441, 442, 444, 450, 452, 454, 459, 461, 463
id_2 31, 32, 47, 151, 153, 323, 328, 337, 345, 387, 411, 421, 425, 429, 437, 442, 445, 452, 455
identifier 159, 162, 165, 167
ignore-http-400-errors 2
ignore-http-401-errors 2
ignore-http-402-errors 2
ignore-http-403-errors 2
ignore-http-404-errors 2
ignore-http-422-errors 2
ignore-http-429-errors 2
ignore-http-500-errors 2
ignore-http-502-errors 2
ignore-http-503-errors 2
ignoreIncomingGoodsId 417, 421, 425
ignoreSupplierInvoiceId 386, 387
ignore-unknown-path-type 2
ignore-values-unknown-path 2
Inactive 106, 111, 115, 120, 124, 128, 133
inboundStockLocationId 401, 402, 404, 406
inboundStockPointId 401, 402, 404, 406
Inbox 149
InboxById 150
InboxFiles 151
InboxFolders 153
includeNonInboundItems 450
includeNonInvoiceableChargeHours 341, 345
includeNonInvoiceablePrice 323, 328, 333, 337
includeOnSend 143
includeRegistrationsWithoutProject 323, 328, 333, 337, 341, 345
IncomingGoodDocumentById 386
IncomingGoodDocumentLinesById 387
IncomingGoodDocuments 390
Information 293
IngoingWithdrawnDays 106, 111
inInvoiceBasis 323, 328, 333, 337, 341, 345
inStock 449
InsuredNumber 47
InsuredWith 47
integrationId 154, 156
IntegrationSalesByIntegrationId 156
internalArticles 323, 328, 333, 337
internalReference 411, 417, 420, 421, 425, 429, 434
internalTime 341, 345
InUse 55
invalid-json-on-get-max-tries 2
invalid-json-on-get-sleep-initial-ms 2
invalid-json-on-get-sleep-max-ms 2
invalid-json-on-get-sleep-multiplicator 2
invalid-json-on-post-max-tries 2
invalid-json-on-post-sleep-initial-ms 2
invalid-json-on-post-sleep-max-ms 2
invalid-json-on-post-sleep-multiplicator 2
invantive-sql-compress-sparse-arrays 2
invantive-sql-correct-invalid-date 2
invantive-sql-execution-profile-disk-path 2
invantive-sql-execution-profile-to-disk 2
invantive-sql-forward-filters-to-data-containers 2
invantive-sql-share-byte-arrays 2
invantive-sql-share-strings 2
invantive-sql-shuffle-fetch-results-data-containers 2
invantive-use-cache 2
Invoice__url 181, 186
Invoice__urlTaxReductionList 181, 186
Invoice_AccountingMethod 181, 186
Invoice_Address1 181, 186
Invoice_Address2 181, 186
Invoice_AdministrationFee 181, 186
Invoice_AdministrationFeeVAT 181, 186
Invoice_Balance 181, 186
Invoice_BasisTaxReduction 181, 186

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Invoice_Booked 181, 186
Invoice_Cancelled 181, 186
Invoice_City 181, 186
Invoice_Comments 181, 186
Invoice_ContractReference 181, 186
Invoice_ContributionPercent 181, 186
Invoice_ContributionValue 181, 186
Invoice_CostCenter 181, 186
Invoice_Country 181, 186
Invoice_Credit 181, 186
Invoice_CreditInvoiceReference 181, 186
Invoice_Currency 181, 186
Invoice_CurrencyRate 181, 186
Invoice_CurrencyUnit 181, 186
Invoice_CustomerName 181, 186
Invoice_CustomerNumber 181, 186
Invoice_DeliveryAddress1 181, 186
Invoice_DeliveryAddress2 181, 186
Invoice_DeliveryCity 181, 186
Invoice_DeliveryCountry 181, 186
Invoice_DeliveryDate 181, 186
Invoice_DeliveryName 181, 186
Invoice_DeliveryZipCode 181, 186
Invoice_DocumentNumber 181, 186
Invoice_DueDate 181, 186
Invoice_EDIInformation_EDIGlobalLocationNumber 181, 186
Invoice_EDIInformation_EDIGlobalLocationNumberDelivery 181, 186
Invoice_EDIInformation_EDIInvoiceExtra1 181, 186
Invoice_EDIInformation_EDIInvoiceExtra2 181, 186
Invoice_EDIInformation_EDIOurElectronicReference 181, 186
Invoice_EDIInformation_EDIStatus 181, 186
Invoice_EDIInformation_EDIYourElectronicReference 181, 186
Invoice_EmailInformation_EmailAddressBCC 181, 186
Invoice_EmailInformation_EmailAddressCC 181, 186
Invoice_EmailInformation_EmailAddressFrom 181, 186
Invoice_EmailInformation_EmailAddressTo 181, 186
Invoice_EmailInformation_EmailBody 181, 186
Invoice_EmailInformation_EmailSubject 181, 186
Invoice_EUQuarterlyReport 181, 186
Invoice_ExternalInvoiceReference1 181, 186
Invoice_ExternalInvoiceReference2 181, 186
Invoice_FinalPayDate 181, 186
Invoice_Freight 181, 186
Invoice_FreightVAT 181, 186
Invoice_Gross 181, 186
Invoice_HouseWork 181, 186
Invoice_InvoiceDate 181, 186
Invoice_InvoicePeriodEnd 181, 186
Invoice_InvoicePeriodReference 181, 186
Invoice_InvoicePeriodStart 181, 186
Invoice_InvoiceType 181, 186
Invoice_Language 181, 186
Invoice_LastRemindDate 181, 186
Invoice_Net 181, 186
Invoice_NotCompleted 181, 186
Invoice_NoxFinans 181, 186
Invoice_OCR 181, 186
Invoice_OfferReference 181, 186
Invoice_OrderReference 181, 186
Invoice_OrganisationNumber 181, 186
Invoice_OurReference 181, 186
Invoice_OutboundDate 181, 186
Invoice_PaymentWay 181, 186
Invoice_Phone1 181, 186
Invoice_Phone2 181, 186
Invoice_PriceList 181, 186
Invoice_PrintTemplate 181, 186
Invoice_Project 181, 186
Invoice_Remarks 181, 186
Invoice_Reminders 181, 186
Invoice_RoundOff 181, 186
Invoice_Sent 181, 186
Invoice_TaxReduction 181, 186
Invoice_TaxReductionType 181, 186
Invoice_TermsOfDelivery 181, 186
Invoice_TermsOfPayment 181, 186
Invoice_TimeBasisReference 181, 186
Invoice_Total 181, 186
Invoice_TotalToPay 181, 186
Invoice_TotalVAT 181, 186
Invoice_VATIncluded 181, 186
Invoice_VoucherNumber 181, 186
Invoice_VoucherSeries 181, 186
Invoice_VoucherYear 181, 186
Invoice_WarehouseReady 181, 186
Invoice_WayOfDelivery 181, 186
Invoice_YourOrderNumber 181, 186
Invoice_YourReference 181, 186
Invoice_ZipCode 181, 186
InvoiceAccrual__url 173
InvoiceAccrual_AccrualAccount 173
InvoiceAccrual_Description 173
InvoiceAccrual_EndDate 173
InvoiceAccrual_InvoiceNumber 173
InvoiceAccrual_Period 173
InvoiceAccrual_RevenueAccount 173

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

InvoiceAccrual_StartDate 173
InvoiceAccrual_Times 173
InvoiceAccrual_Total 173
InvoiceAccrual_VATIncluded 173
InvoiceAccrualLinesByInvoiceNumber 170
InvoiceAccruals 172
InvoiceAccrualsByInvoiceNumber 173
invoiceBasisId 323, 328, 337, 341, 345
InvoiceByDocumentNumber 181
InvoiceCustomerName 179
InvoiceCustomerNumber 179
invoiced 323, 328, 333, 337, 341, 345
InvoiceDate 192, 197, 203, 208, 216, 298, 301, 303, 306
InvoiceDiscount 74
invoicedQuantity 387
InvoiceDueDate 179, 293
InvoiceEmailByDocumentNumber 186
InvoiceEmailInvoiceLabelsByDocumentNumber 192
InvoiceEmailInvoiceLinesByDocumentNumber 197
InvoiceInterval 74, 78, 82, 84
InvoiceLabelsByDocumentNumber 203
InvoiceLinesByDocumentNumber 208
InvoiceNumber 170, 172, 173, 177, 179, 292, 293, 298, 301, 303, 306
InvoiceOCR 179, 293
InvoicePayment__url 175
InvoicePayment_Amount 175
InvoicePayment_AmountCurrency 175
InvoicePayment_Booked 175
InvoicePayment_Currency 175
InvoicePayment_CurrencyRate 175
InvoicePayment_CurrencyUnit 175
InvoicePayment_ExternalInvoiceReference1 175
InvoicePayment_ExternalInvoiceReference2 175
InvoicePayment_InvoiceCustomerName 175
InvoicePayment_InvoiceCustomerNumber 175
InvoicePayment_InvoiceDueDate 175
InvoicePayment_InvoiceNumber 175
InvoicePayment_InvoiceOCR 175
InvoicePayment_InvoiceTotal 175
InvoicePayment_ModeOfPayment 175
InvoicePayment_ModeOfPaymentAccount 175
InvoicePayment_Number 175
InvoicePayment_PaymentDate 175
InvoicePayment_Source 175
InvoicePayment_VoucherNumber 175
InvoicePayment_VoucherSeries 175
InvoicePayment_VoucherYear 175
InvoicePaymentByNumber 175
InvoicePayments 177
InvoicePaymentWriteOffsByPaymentNumber 179
InvoicePeriodEnd 192, 197, 203, 208
InvoicePeriodReference 192, 197, 203, 208
InvoicePeriodStart 192, 197, 203, 208
InvoicePreviewByDocumentNumber 214
InvoicePrintByDocumentNumber 215
InvoiceReference 231, 235, 245, 249
Invoices 216
InvoicesRemaining 74, 78
InvoiceSupplierName 293
InvoiceSupplierNumber 293
invoiceText 323, 328, 337, 341, 345
InvoiceTotal 179, 293
InvoiceType 192, 197, 203, 208, 216
IsAutomatic 88
isMyRating 154
isStock 345
isStockItem 387, 411, 421, 425, 429
itemDescription 298, 377, 384, 387, 394, 397, 401, 402, 404, 406, 408, 411, 421, 425, 429, 461
itemDescription_2 402, 404
itemDescriptionSearch 450, 454
itemId 377, 381, 384, 387, 390, 392, 394, 397, 401, 402, 404, 406, 408, 411, 420, 421, 425, 429, 434, 449, 450, 452, 454, 455, 458, 461
itemId_2 402, 404
itemIdSearch 450, 454
itemUnit 377, 384, 387, 394, 397, 401, 402, 404, 406, 408, 411, 421, 425, 429, 461
itemUnit_2 402, 404

- J -

JobTitle 106, 111, 115, 120, 124, 128, 133
join-set-points-per-request 2

- L -

label 216, 239, 254
Labels 219
Language 74, 192, 197, 203, 208, 231, 235, 245, 249
languageCode 411, 417, 421, 425, 429, 434
LastDay 357
LastInvoiceDate 74, 78

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

lastmodified 17, 19, 20, 22, 23, 25, 27, 28, 30, 31, 32, 34, 35, 36, 39, 41, 43, 44, 45, 47, 50, 52, 53, 55, 57, 59, 60, 62, 63, 66, 67, 69, 70, 74, 78, 80, 82, 84, 85, 86, 88, 89, 90, 92, 93, 94, 95, 100, 101, 106, 111, 115, 120, 124, 128, 133, 137, 139, 140, 141, 143, 144, 146, 147, 149, 150, 151, 153, 154, 156, 157, 158, 159, 161, 162, 163, 165, 166, 167, 169, 170, 172, 173, 175, 177, 179, 181, 186, 192, 197, 203, 208, 214, 215, 216, 219, 220, 221, 223, 224, 227, 231, 235, 239, 241, 245, 249, 254, 256, 257, 259, 260, 261, 263, 264, 266, 267, 269, 270, 271, 273, 274, 276, 277, 279, 280, 282, 284, 285, 286, 288, 289, 292, 293, 295, 298, 301, 303, 306, 308, 311, 313, 315, 317, 318, 320, 321, 322, 323, 328, 333, 337, 341, 345, 349, 350, 352, 353, 354, 355, 357, 361, 363, 364, 366, 367, 369, 371, 372, 373, 375, 376, 377, 379, 381, 382, 384, 386, 387, 390, 392, 394, 396, 397, 399, 401, 402, 404, 406, 408, 410, 411, 415, 417, 420, 421, 425, 429, 434, 437, 439, 441, 442, 444, 445, 447, 449, 450, 452, 454, 455, 458, 459, 461, 463, 464, 465, 467
LastName 106, 111, 115, 120, 124, 128, 133
LastRemindDate 192, 197, 203, 208
limit-partition-calls-left 2
LockedPeriod_EndDate 220
LockedPeriods 220
log-native-calls-to-disk-max-events 2
log-native-calls-to-disk-max-seconds 2
log-native-calls-to-disk-on-error 2
log-native-calls-to-disk-on-success 2
log-native-calls-to-trace 2

- M -

Manual 372
ManualDocuments 392
ManualInboundDocumentRows 394
ManualInboundDocuments 396
manuallyCompleted 411, 417, 421, 425, 429, 434
ManualOb 47
ManualOutboundDocumentRows 397
ManualOutboundDocuments 399
manufacturer 39
manufacturerarticlenumber 39
MatchedDocumentsByPurchaseOrderId 410
maximum-discovered-column-count 2
maximum-length-identifiers 2
max-odata-filters 2
max-odata-rewrite-in-count 2
max-url-length-accepted 2
max-url-length-desired 2
Me 221
Me_Email 221
Me_Id 221
Me_Locale 221
Me_Name 221
Me_SysAdmin 221
messageToSupplier 411, 417, 421, 425, 429, 434
metadata-cache-max-age-sec 2
ModeOfPayment 179, 293
ModeOfPayment__url 223
ModeOfPayment_AccountNumber 223
ModeOfPayment_Code 223
ModeOfPayment_Description 223
ModeOfPaymentAccount 179
Month 355, 357
MonthlySalary 106, 111, 115, 120, 124, 128, 133
MonthlySalary_2 124, 128
MultiStockLocations 437
MultiStockpoints 439

- N -

name 22, 31, 32, 44, 100, 151, 153, 163, 169, 256, 257, 259, 260, 270, 288, 311, 437, 439, 441, 442, 444, 445, 447, 452, 455, 458, 459
name_2 31, 32, 151, 153, 437, 442, 445
Native Platform Scalar Requests 225
NATIVEPLATFORMSCALARREQUESTS 225
Net 74, 192, 197, 203, 208, 231, 235, 245, 249
nonInvoiceable 323, 328, 333, 337, 341, 345
NonRecurringTax 106, 111, 115, 120, 124, 128, 133
notcompleted 192, 197, 203, 208, 216, 231, 235, 239, 245, 249, 254
note 86, 323, 328, 337, 341, 345, 377, 379, 381, 382, 384, 386, 387, 390, 392, 394, 396, 397, 399, 401, 402, 404, 406, 411, 415, 417, 420, 421, 425, 429, 434, 461, 463
note_2 411, 425
Notes_2 47
NoxFinans 192, 197, 203, 208, 216
NoxFinansInvoice__url 144
NoxFinansInvoice_BalanceIncludeFees 144
NoxFinansInvoice_BalanceIncludeFeesCurrency 144
NoxFinansInvoice_CurrentCapitalBalance 144
NoxFinansInvoice_CurrentCapitalBalanceCurrency 144
NoxFinansInvoice_InvoiceDocumentURL 144
NoxFinansInvoice_InvoiceNumber 144
NoxFinansInvoice_NextEvent 144
NoxFinansInvoice_NextEventDate 144
NoxFinansInvoice_OCRNumber 144
NoxFinansInvoice_Service 144
NoxFinansInvoice_ServiceName 144
NoxFinansInvoice_Status 144

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

npt 225
Number 23, 25, 27, 47, 50, 52, 55, 144, 175, 177, 179, 276, 289, 292, 293, 303, 306
numberOfAttachments 141

- O -

oauth-unauthorized-max-tries 2
oauth-unauthorized-sleep-initial-ms 2
oauth-unauthorized-sleep-max-ms 2
oauth-unauthorized-sleep-multiplicator 2
OCR 192, 197, 203, 208, 216, 298, 306
Offer__url 227
Offer__urlTaxReductionList 227
Offer_Address1 227
Offer_Address2 227
Offer_AdministrationFee 227
Offer_AdministrationFeeVAT 227
Offer_BasisTaxReduction 227
Offer_Cancelled 227
Offer_City 227
Offer_Comments 227
Offer_ContributionPercent 227
Offer_ContributionValue 227
Offer_CopyRemarks 227
Offer_CostCenter 227
Offer_Country 227
Offer_Currency 227
Offer_CurrencyRate 227
Offer_CurrencyUnit 227
Offer_CustomerName 227
Offer_CustomerNumber 227
Offer_DeliveryAddress1 227
Offer_DeliveryAddress2 227
Offer_DeliveryCity 227
Offer_DeliveryCountry 227
Offer_DeliveryDate 227
Offer_DeliveryName 227
Offer_DeliveryZipCode 227
Offer_DocumentNumber 227
Offer_EmailInformation_EmailAddressBCC 227
Offer_EmailInformation_EmailAddressCC 227
Offer_EmailInformation_EmailAddressFrom 227
Offer_EmailInformation_EmailAddressTo 227
Offer_EmailInformation_EmailBody 227
Offer_EmailInformation_EmailSubject 227
Offer_ExpireDate 227
Offer_Freight 227
Offer_FreightVAT 227
Offer_Gross 227
Offer_HouseWork 227
Offer_InvoiceReference 227
Offer_Language 227
Offer_Net 227
Offer_NotCompleted 227
Offer_OfferDate 227
Offer_OrderReference 227
Offer_OrganisationNumber 227
Offer_OurReference 227
Offer_Phone1 227
Offer_Phone2 227
Offer_PriceList 227
Offer_PrintTemplate 227
Offer_Project 227
Offer_Remarks 227
Offer_RoundOff 227
Offer_Sent 227
Offer_TaxReduction 227
Offer_TaxReductionType 227
Offer_TermsOfDelivery 227
Offer_TermsOfPayment 227
Offer_Total 227
Offer_TotalToPay 227
Offer_TotalVAT 227
Offer_VATIncluded 227
Offer_WayOfDelivery 227
Offer_YourReference 227
Offer_YourReferenceNumber 227
Offer_ZipCode 227
OfferByDocumentNumber 227
OfferDate 231, 235, 239
OfferLabelsByDocumentNumber 231
OfferLinesByDocumentNumber 235
OfferReference 192, 197, 203, 208, 245, 249
Offers 239
onCancellationTime 162, 167
Order__url 241
Order__urlTaxReductionList 241
Order_Address1 241
Order_Address2 241
Order_AdministrationFee 241
Order_AdministrationFeeVAT 241
Order_BasisTaxReduction 241
Order_Cancelled 241
Order_City 241
Order_Comments 241
Order_ContributionPercent 241
Order_ContributionValue 241
Order_CopyRemarks 241
Order_CostCenter 241
Order_Country 241
Order_Currency 241

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Order_CurrencyRate 241
Order_CurrencyUnit 241
Order_CustomerName 241
Order_CustomerNumber 241
Order_DeliveryAddress1 241
Order_DeliveryAddress2 241
Order_DeliveryCity 241
Order_DeliveryCountry 241
Order_DeliveryDate 241
Order_DeliveryName 241
Order_DeliveryState 241
Order_DeliveryZipCode 241
Order_DocumentNumber 241
Order_EmailInformation_EmailAddressBCC 241
Order_EmailInformation_EmailAddressCC 241
Order_EmailInformation_EmailAddressFrom 241
Order_EmailInformation_EmailAddressTo 241
Order_EmailInformation_EmailBody 241
Order_EmailInformation_EmailSubject 241
Order_ExternalInvoiceReference1 241
Order_ExternalInvoiceReference2 241
Order_Freight 241
Order_FreightVAT 241
Order_Gross 241
Order_HouseWork 241
Order_InvoiceReference 241
Order_Language 241
Order_Net 241
Order_NotCompleted 241
Order_OfferReference 241
Order_OrderDate 241
Order_OrderType 241
Order_OrganisationNumber 241
Order_OurReference 241
Order_OutboundDate 241
Order_Phone1 241
Order_Phone2 241
Order_PriceList 241
Order_PrintTemplate 241
Order_Project 241
Order_Remarks 241
Order_RoundOff 241
Order_Sent 241
Order_StockPointCode 241
Order_StockPointId 241
Order_TaxReduction 241
Order_TaxReductionType 241
Order_TermsOfDelivery 241
Order_TermsOfPayment 241
Order_TimeBasisReference 241
Order_Total 241
Order_TotalToPay 241
Order_TotalVAT 241
Order_VATIncluded 241
Order_WarehouseReady 241
Order_WayOfDelivery 241
Order_YourOrderNumber 241
Order_YourReference 241
Order_ZipCode 241
OrderByDocumentNumber 241
orderDate 245, 249, 254, 411, 417, 421, 425, 429, 434
orderedQuantity 249, 387, 411, 421, 425, 429
orderId 156
orderIndex 323, 328, 337
OrderLabelsByDocumentNumber 245
OrderLinesByDocumentNumber 249
OrderReference 192, 197, 203, 208, 231, 235
Orders 254
OrderType 245, 249, 254
orderValue 411, 417, 421, 425, 429, 434
orderValueInSEK 411, 417, 421, 425, 429, 434
OrganisationNumber 100, 192, 197, 203, 208, 231, 235, 245, 249, 311
ORIG_SYSTEM_GROUP 225
ORIG_SYSTEM_REFERENCE 225
Original System Group 225
Original System Reference 225
otherCost 377, 394
ourReference 74, 82, 192, 197, 203, 208, 216, 231, 235, 239, 245, 249, 254, 298, 306, 411, 417, 421, 425, 429, 434
OutboundDate 192, 197, 203, 208, 245, 249
outboundDocumentReference_id 411, 417, 421, 425, 429, 434
outboundDocumentReference_type 411, 417, 421, 425, 429, 434
outboundStockPointId 401, 402, 404, 406
ownerId 323, 328, 333, 337
ownerId_2 323, 328, 337

- P -

partition-slot-based-rate-limit-length-ms 2
partition-slot-based-rate-limit-slots 2
Path 28, 31, 32, 151
Payload 225
PAYLOAD_TEXT 225
PaymentDate 177, 179, 292, 293
PaymentMethodByCode 223
PaymentModes 224
PaymentPending 298, 306
PaymentTermByCode 321

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

PaymentTerms 322
paymentTermsCode 411, 417, 421, 425, 429, 434
PaymentWay 192, 197, 203, 208
Period 67, 69, 170, 172, 282, 284
PeriodEnd 74, 78
PeriodStart 74, 78
PersonalIdentityNumber 106, 111, 115, 120, 124, 128, 133
PersonelType 106, 111, 115, 120, 124, 128, 133
PG 311
Phone 100, 311
Phone1 106, 111, 115, 120, 124, 128, 133, 192, 197, 203, 208, 231, 235, 245, 249
Phone2 106, 111, 115, 120, 124, 128, 133, 192, 197, 203, 208, 231, 235, 245, 249
Placement 47
PostCode 106, 111, 115, 120, 124, 128, 133
PreDefinedAccount 311
PreDefinedAccount__url 256
PreDefinedAccount_Account 256
PreDefinedAccount_Name 256
PredefinedAccountByName 256
PredefinedAccounts 257
PredefinedVoucherSeries 259
PreDefinedVoucherSeries__url 260
PreDefinedVoucherSeries_Name 260
PreDefinedVoucherSeries_VoucherSeries 260
PredefinedVoucherSeriesByName 260
pre-request-delay-ms 2
PreSelected 263
price 74, 82, 197, 208, 235, 249, 269, 298, 411, 421, 425, 429
Price__url 264, 266, 267
Price_ArticleNumber 264, 266, 267
Price_Date 264, 266, 267
Price_FromQuantity 264, 266, 267
Price_Percent 264, 266, 267
Price_Price 264, 266, 267
Price_PriceList 264, 266, 267
PriceExcludingVAT 74, 197, 208
PriceList 74, 192, 197, 203, 208, 231, 235, 245, 249, 264, 266, 269
PriceList__url 261
PriceList_Code 261
PriceList_Comments 261
PriceList_Description 261
PriceList_PreSelected 261
PriceListByCode 261
PriceLists 263
Prices 267
Prices2 269
PrintTemplate 74, 82, 192, 197, 203, 208, 231, 235, 245, 249
PrintTemplates 270
productionDate 401, 402, 404, 406
ProductionOrderDocumentById 401
ProductionOrderDocumentPackageItemsById 402
ProductionOrderPackageItems 404
ProductionOrderRowNotes 411
ProductionOrders 406
productionState 401, 402, 404, 406
Project 19, 20, 25, 27, 47, 57, 60, 67, 74, 82, 106, 111, 115, 120, 124, 128, 133, 170, 179, 192, 197, 203, 208, 216, 231, 235, 239, 245, 249, 254, 276, 282, 293, 298, 301, 303, 306, 311, 359, 364
Project__url 271
Project_2 25, 74, 197, 208, 235, 249, 298, 364
Project_Comments 271
Project_ContactPerson 271
project_description 271, 323, 328, 333, 337, 341, 345
project_endDate 271, 323, 328, 333, 337, 341, 345
project_id 323, 328, 333, 337, 341, 345
project_invoicedAmount 323, 328, 333, 337, 341, 345
project_orderAmount 323, 328, 333, 337, 341, 345
project_orderTime 323, 328, 333, 337, 341, 345
project_projectLeader 271, 323, 328, 333, 337, 341, 345
Project_ProjectNumber 271
project_startDate 271, 323, 328, 333, 337, 341, 345
project_status 271, 323, 328, 333, 337, 341, 345
project_totalAmountInInvoices 323, 328, 333, 337, 341, 345
ProjectByProjectNumber 271
projectId 377, 384, 386, 387, 394, 397, 401, 402, 404, 406, 411, 417, 421, 425, 429, 434, 452, 455, 458, 459
projectId_2 387, 411, 421, 425, 429
ProjectLeader 273
ProjectNumber 271, 273
Projects 273
ProjectSettings 25, 27
PropertyDesignation 313
purchaseDate 323, 328, 333, 337
PurchaseOrderAttachedNotesByPurchaseOrderId 415
PurchaseOrderById 417
PurchaseOrderCsvReports 420
purchaseOrderId 387, 411, 415, 421, 425, 429
purchaseOrderId_2 411, 425
PurchaseOrderLinesById 421
purchaseOrderRowId 387, 411, 415, 425
PurchaseOrderRowNotesById 425

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

PurchaseOrderRows 429
PurchaseOrders 434
purchaseOrderState 411, 417, 421, 425, 429, 434
purchasePrice 39, 345
purchaseTime 159, 165
purchaseType 411, 417, 420, 421, 425, 429, 434

- Q -

q 390, 411, 420, 429, 434, 441, 445, 447
quantity 235, 298, 359, 364, 377, 384, 394, 397, 401, 402, 404, 406, 408, 461
QuantityInStock 39
quantityRequired 402, 404, 408
quantityReserved 402, 404, 408
QuantitySettings 25
QuantityUnit 25

- R -

rating 154
ratingId 154
Ratings 154
receivedQuantity 387, 411, 421, 425, 429
Reference 47, 92, 93
ReferenceDocumentType 313, 317
ReferenceNumber 313, 317, 364, 366, 367, 371
referenceType 303, 306, 364, 366, 367, 371, 375, 376, 381, 382, 384
registeredArticle_createdBy 323, 328, 333
registeredArticle_createdTime 323, 328, 333
registeredArticle_documentId 323, 328, 333
registeredArticle_documentType 323, 328, 333
registeredArticle_id 323, 328, 333
registeredArticle_invoiceBasisId 323, 328, 333
registeredArticle_invoiceText 323, 328, 333
registeredArticle_nonInvoiceable 323, 328, 333
registeredArticle_note 323, 328, 333
registeredArticle_orderIndex 323, 328, 333
registeredArticle_ownerId 323, 328, 333
registeredArticle_timeLocked 323, 328, 333
registeredArticle_totalQuantity 323, 328, 333
registeredArticle_unitCost 323, 328, 333
registeredArticle_unitPrice 323, 328, 333
registrationCode_active 341, 345
registrationCode_code 341, 345
registrationCode_costMultiplier 341, 345
registrationCode_id 341, 345
registrationCode_name 341, 345
registrationCode_priceMultiplier 341, 345
registrationCode_type 341, 345
registrationType 323, 328, 333, 337
RejectedSenders 349
released 386, 387, 390, 392, 394, 396, 397, 399, 461, 463
remainingOrderedQuantity 387, 411, 421, 425, 429
Remarks 74, 82, 192, 197, 203, 208, 231, 235, 245, 249
Reminders 192, 197, 203, 208
Removed 359, 364
requested-page-size 2
requestedQuantity 461
RequestSent 313
requests-parallel-max 2
RequiredProductionPartsByItemId 408
reservedQuantity 39, 249, 384
ResidenceAssociationOrganisationNumber 313
Response Cache Maximum Age (sec) 225
response_comment 154
response_created 154
response_updated 154
responseState 411, 417, 421, 425, 429, 434
responsible 452, 455, 458, 459
RESULT 279
Result BLOB 225
Result Date Time 225
Result Number 225
Result Text 225
RESULT_BLOB 225
RESULT_DATE_TIME_UTC 225
RESULT_NUMBER 225
RESULT_TEXT 225
RevenueAccount 170
Room 47
RoundOff 74, 192, 197, 203, 208, 231, 235, 245, 249
RoundOffValue 298, 306
rowId 74, 197, 208, 235, 249, 377, 384, 394
rowNum 411, 415, 421, 425, 429, 461
rowNum_2 411, 425
rowOrder 387
Run without Actions 225

- S -

SalaryCode 276
SalaryForm 106, 111, 115, 120, 124, 128, 133
SalaryRow 274, 276
SalaryTransaction_Amount 274
SalaryTransaction_CostCenter 274
SalaryTransaction_Date 274
SalaryTransaction_EmployeeId 274

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

SalaryTransaction_Expense 274
SalaryTransaction_Number 274
SalaryTransaction_Project 274
SalaryTransaction_SalaryCode 274
SalaryTransaction_SalaryRow 274
SalaryTransaction_TextRow 274
SalaryTransaction_Total 274
SalaryTransaction_VAT 274
SalaryTransactionByRow 274
SalaryTransactions 276
SalesPrice 39
SalesType 298, 306
Save HTTP Disk Cache 225
Save HTTP Memory Cache 225
ScheduleId 106, 111, 115, 120, 124, 128, 133
ScheduleId_2 115, 120
ScheduleTime_Date 277
ScheduleTime_EmployeeId 277
ScheduleTime_Hours 277
ScheduleTime_IWH1 277
ScheduleTime_IWH2 277
ScheduleTime_IWH3 277
ScheduleTime_IWH4 277
ScheduleTime_IWH5 277
ScheduleTime_ScheduleId 277
ScheduleTimeByEmployeeIdAndDate 277
secondaryorder 454
secondarysortby 454
SellRate 88
Senders 350
Sent 192, 197, 203, 208, 216, 231, 235, 239, 245, 249, 254
Series 303, 306
service_active 341
service_description 341
service_id 341
service_isStock 341
service_purchasePrice 341
service_type 341
service_unit 341
showPurchaseTypeColumn 420
SieFileByType 279
simulate-http-400-errors 2
simulate-http-400-errors-percentage 2
simulate-http-401-errors 2
simulate-http-401-errors-percentage 2
simulate-http-403-errors 2
simulate-http-403-errors-percentage 2
simulate-http-408-errors 2
simulate-http-408-errors-percentage 2
simulate-http-429-errors 2
simulate-http-429-errors-percentage 2
simulate-http-500-errors 2
simulate-http-500-errors-percentage 2
simulate-http-502-errors 2
simulate-http-502-errors-percentage 2
simulate-http-503-errors 2
simulate-http-503-errors-percentage 2
simulate-http-504-errors 2
simulate-http-504-errors-percentage 2
simulate-http-522-errors 2
simulate-http-522-errors-percentage 2
simulate-http-524-errors 2
simulate-http-524-errors-percentage 2
simulate-http-protocol-errors 2
simulate-http-protocol-errors-percentage 2
simulate-http-timeout-errors 2
simulate-http-timeout-errors-percentage 2
Size 31, 151
slot-based-rate-limit-length-ms 2
slot-based-rate-limit-slots 2
SocialSecurityNumber 313, 317
sortingId 452, 455, 458, 459
sortParams_primarySort 452, 455, 458, 459
sortParams_primarySortOrder 452, 455, 458, 459
sortParams_secondarySort 452, 455, 458, 459
sortParams_secondarySortOrder 452, 455, 458, 459
Source 177, 179, 292, 293
SRU 25, 27
standardize-identifiers 2
standardize-identifiers-casing 2
Start Date 225
startDate 170, 273, 282, 401, 402, 404, 406
startingItemId 454
startingRowNo 454
startTime 341, 345
state 392, 404, 406, 411, 420, 429, 434, 437, 439, 445, 447, 452, 455, 458, 459
stateFilter 454
Status 47, 50, 52, 78, 273
StatusId 47, 50, 52
StockBalances 449
stockLocationCode 298, 377, 387, 394, 397, 411, 421, 425, 429
stockLocationId 377, 384, 387, 394, 397, 411, 421, 425, 429, 450, 452, 454, 455
stockLocationName 387, 394, 397, 411, 421, 425, 429
StockLocations2ByStockPointId 441
StockLocationsByStockPointId 442
StockPlace 39
StockpointById 444

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

stockPointCode 197, 208, 245, 249, 298, 377, 386, 387, 394, 396, 397, 399, 411, 417, 421, 425, 429, 434, 449
stockPointCode_2 249, 387, 394, 397, 411, 421, 425, 429
stockPointId 245, 249, 377, 381, 384, 386, 387, 390, 394, 396, 397, 399, 411, 417, 421, 425, 429, 434, 437, 441, 442, 445, 450, 452, 454, 455
stockPointId_2 249, 387, 394, 397, 411, 421, 425, 429
StockpointLocations 445
stockPointName 386, 387, 394, 396, 397, 399, 411, 421, 425, 429
stockPointName_2 387, 394, 397
Stockpoints 447
stockTakenQuantity 450, 452, 454, 455
StockTakingCandidateRows 450
stockTakingId 450, 452, 454, 455
StockTakingLinesByDocumentId 452
stockTakingRowId 450, 452, 454, 455
StockTakingRows 454
StockTakingRows2 455
StockTakings 458
StockTakingsByDocumentId 459
StockTransferDocumentLinesById 461
StockTransferDocumentsById 463
StockValue 39
stopTime 341, 345
Succesful 225
SUCCESSFUL 225
supplier 411, 417, 421, 425, 429, 434
Supplier__url 308
Supplier_Active 308
Supplier_Address1 308
Supplier_Address2 308
Supplier_Bank 308
Supplier_BankAccountNumber 308
Supplier_BG 308
Supplier_BIC 308
Supplier_BranchCode 308
Supplier_City 308
Supplier_ClearingNumber 308
Supplier_Comments 308
Supplier_CostCenter 308
Supplier_Country 308
Supplier_CountryCode 308
Supplier_Currency 308
Supplier_DisablePaymentFile 308
Supplier_Email 308
Supplier_Fax 308
Supplier_IBAN 308
Supplier_Name 308
Supplier_OrganisationNumber 308
Supplier_OurCustomerNumber 308
Supplier_OurReference 308
Supplier_PG 308
Supplier_Phone1 308
Supplier_Phone2 308
Supplier_PreDefinedAccount 308
Supplier_Project 308
Supplier_SupplierNumber 308
Supplier_TermsOfPayment 308
Supplier_VATNumber 308
Supplier_VATType 308
Supplier_VisitingAddress 308
Supplier_VisitingCity 308
Supplier_VisitingCountry 308
Supplier_VisitingCountryCode 308
Supplier_VisitingZipCode 308
Supplier_WorkPlace 308
Supplier_WWW 308
Supplier_YourReference 308
Supplier_ZipCode 308
supplierAddress 411, 417, 421, 425, 429, 434
supplierAddress2 411, 417, 421, 425, 429, 434
SupplierByNumber 308
supplierCity 411, 417, 421, 425, 429, 434
supplierCountryCode 411, 417, 421, 425, 429, 434
supplierEmail 411, 417, 421, 425, 429, 434
SupplierInvoice 47
SupplierInvoice__url 295
SupplierInvoice_AccountingMethod 295
SupplierInvoice_AdministrationFee 295
SupplierInvoice_Balance 295
SupplierInvoice_Booked 295
SupplierInvoice_Cancelled 295
SupplierInvoice_Comments 295
SupplierInvoice_CostCenter 295
SupplierInvoice_Credit 295
SupplierInvoice_CreditReference 295
SupplierInvoice_Currency 295
SupplierInvoice_CurrencyRate 295
SupplierInvoice_CurrencyUnit 295
SupplierInvoice_DisablePaymentFile 295
SupplierInvoice_DueDate 295
SupplierInvoice_ExternalInvoiceNumber 295
SupplierInvoice_ExternalInvoiceSeries 295
SupplierInvoice_FinalPayDate 295
SupplierInvoice_Freight 295
SupplierInvoice_GivenNumber 295
SupplierInvoice_InvoiceDate 295
SupplierInvoice_InvoiceNumber 295
SupplierInvoice_OCR 295

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

SupplierInvoice_OurReference 295
SupplierInvoice_PaymentPending 295
SupplierInvoice_Project 295
SupplierInvoice_RoundOffValue 295
SupplierInvoice_SalesType 295
SupplierInvoice_SupplierName 295
SupplierInvoice_SupplierNumber 295
SupplierInvoice_Total 295
SupplierInvoice_VAT 295
SupplierInvoice_VATType 295
SupplierInvoice_VoucherNumber 295
SupplierInvoice_VoucherSeries 295
SupplierInvoice_VoucherYear 295
SupplierInvoice_YourReference 295
SupplierInvoiceAccrual__url 280
SupplierInvoiceAccrual_AccrualAccount 280
SupplierInvoiceAccrual_CostAccount 280
SupplierInvoiceAccrual_Description 280
SupplierInvoiceAccrual_EndDate 280
SupplierInvoiceAccrual_Period 280
SupplierInvoiceAccrual_StartDate 280
SupplierInvoiceAccrual_SupplierInvoiceNumber 280
SupplierInvoiceAccrual_Times 280
SupplierInvoiceAccrual_Total 280
SupplierInvoiceAccrual_VATIncluded 280
SupplierInvoiceAccrualByInvoiceNumber 280
SupplierInvoiceAccrualLinesByInvoiceNumber 282
SupplierInvoiceAccruals 284
SupplierInvoiceByGivenNumber 295
SupplierInvoiceExternalURLConnection_ExternalURLConnection 285
SupplierInvoiceExternalURLConnection_Id 285
SupplierInvoiceExternalURLConnection_SupplierInvoiceNumber 285
SupplierInvoiceExternalURLConnection_Url 285
SupplierInvoiceExternalUrlConnectionById 285
SupplierInvoiceFileConnection__url 286
SupplierInvoiceFileConnection_FileId 286
SupplierInvoiceFileConnection_Name 286
SupplierInvoiceFileConnection_SupplierInvoiceNumber 286
SupplierInvoiceFileConnection_SupplierName 286
SupplierInvoiceFileConnectionByFileId 286
SupplierInvoiceFileConnections 288
SupplierInvoiceLinesByGivenNumber 298
SupplierInvoiceNumber 280, 282, 284, 288
SupplierInvoicePayment__url 289
SupplierInvoicePayment_Amount 289
SupplierInvoicePayment_AmountCurrency 289
SupplierInvoicePayment_Booked 289
SupplierInvoicePayment_Currency 289
SupplierInvoicePayment_CurrencyRate 289
SupplierInvoicePayment_CurrencyUnit 289
SupplierInvoicePayment_Information 289
SupplierInvoicePayment_InvoiceDueDate 289
SupplierInvoicePayment_InvoiceNumber 289
SupplierInvoicePayment_InvoiceOCR 289
SupplierInvoicePayment_InvoiceSupplierName 289
SupplierInvoicePayment_InvoiceSupplierNumber 289
SupplierInvoicePayment_InvoiceTotal 289
SupplierInvoicePayment_ModeOfPayment 289
SupplierInvoicePayment_Number 289
SupplierInvoicePayment_PaymentDate 289
SupplierInvoicePayment_Source 289
SupplierInvoicePayment_VoucherNumber 289
SupplierInvoicePayment_VoucherSeries 289
SupplierInvoicePayment_VoucherYear 289
SupplierInvoicePaymentByInvoicePaymentNumber 289
SupplierInvoicePayments 292
SupplierInvoicePaymentWriteOffsByInvoicePaymentNumber 293
SupplierInvoices 301
SupplierInvoiceVouchers 303
SupplierInvoiceVouchersByGivenNumber 306
supplierName 288, 298, 301, 303, 306, 386, 387, 390, 411, 417, 421, 425, 429, 434
supplierNumber 39, 298, 301, 303, 306, 308, 311, 386, 387, 390, 411, 417, 420, 421, 425, 429, 434
supplierPostCode 411, 417, 421, 425, 429, 434
Suppliers 311
swagger-specification-allow-fallback 2
swagger-specification-download-tries 2
swagger-specification-file 2
swagger-specification-http-disk-cache-max-age-sec 2
swagger-specification-url 2

- T -

takenQuantity 387
TaxAllowance 106, 111, 115, 120, 124, 128, 133
TaxColumn 106, 111, 115, 120, 124, 128, 133
TaxReduction 74, 192, 197, 203, 208, 231, 235, 245, 249
TaxReduction__url 315
TaxReduction_ApprovedAmount 315
TaxReduction_AskedAmount 315
TaxReduction_BilledAmount 315
TaxReduction_CustomerName 315
TaxReduction_Id 315
TaxReduction_PropertyDesignation 315
TaxReduction_ReferenceDocumentType 315

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

TaxReduction_ReferenceNumber 315
TaxReduction_RequestSent 315
TaxReduction_ResidenceAssociationOrganisationNumber 315
TaxReduction_SocialSecurityNumber 315
TaxReduction_VoucherNumber 315
TaxReduction_VoucherSeries 315
TaxReduction_VoucherYear 315
TaxReductionAmountsByTaxReductionId 313
TaxReductionById 315
TaxReductions 317
TaxReductionType 74, 192, 197, 203, 208, 231, 235, 245, 249
TaxTable 106, 111, 115, 120, 124, 128, 133
Template 270
TemplateName 74, 82
TemplateNumber 74, 78, 80, 82
tenantId 156, 158, 159, 161, 162, 163, 165, 166, 167, 169, 464
TermsOfDelivery 74, 82, 192, 197, 203, 208, 231, 235, 245, 249
TermsOfDelivery__url 318
TermsOfDelivery_Code 318
TermsOfDelivery_Description 318
TermsOfDelivery_DescriptionEnglish 318
TermsOfPayment 74, 82, 192, 197, 203, 208, 216, 231, 235, 245, 249, 311
TermsOfPayment__url 321
TermsOfPayment_Code 321
TermsOfPayment_Description 321
TEXT 30, 140, 150, 214, 215, 420
TextRow 276
TimeBasisReference 192, 197, 203, 208, 245, 249
timeLocked 323, 328, 337
Timeout (sec) 225
TIMEOUT_SEC 225
TimeReportingArticleRegistrations 323
TimeReportingArticleRegistrationSalesPrices 328
TimeReportingArticles 333
TimeReportingRegisteredArticleSalesPrices 337
TimeReportingRegistrations 341
TimeReportingRegistrationServiceSalesPrices 345
Times 67, 170, 282
todate 17, 19, 20, 22, 23, 25, 27, 28, 30, 31, 32, 34, 35, 36, 39, 41, 43, 44, 45, 47, 50, 52, 53, 55, 57, 59, 60, 62, 63, 66, 67, 69, 70, 74, 78, 80, 82, 84, 85, 86, 88, 89, 90, 92, 93, 94, 95, 100, 101, 106, 111, 115, 120, 124, 128, 133, 137, 139, 140, 141, 143, 144, 146, 147, 149, 150, 151, 153, 154, 156, 157, 158, 159, 161, 162, 163, 165, 166, 167, 169, 170, 172, 173, 175, 177, 179, 181, 186, 192, 197, 203, 208, 214, 215, 216, 219, 220, 221, 223, 224, 227, 231, 235, 239, 241, 245, 249, 254, 256, 257, 259, 260, 261, 263, 264, 266, 267, 269, 270, 271, 273, 274, 276, 277, 279, 280, 282, 284, 285, 286, 288, 289, 292, 293, 295, 298, 301, 303, 306, 308, 311, 313, 315, 317, 318, 320, 321, 322, 323, 328, 333, 337, 341, 345, 349, 350, 352, 353, 354, 355, 357, 361, 363, 364, 366, 367, 369, 371, 372, 373, 375, 376, 377, 379, 381, 382, 384, 386, 387, 390, 392, 394, 396, 397, 399, 401, 402, 404, 406, 408, 410, 411, 415, 417, 420, 421, 425, 429, 434, 437, 439, 441, 442, 444, 445, 447, 449, 450, 452, 454, 455, 458, 459, 461, 463, 464, 465, 467
todate1 52, 323, 328, 333, 337, 341, 345
tofinalpaydate 216
toStockLocationCode 461
toStockLocationId 461
toStockLocationName 461
toStockPointCode 461
toStockPointId 461
toStockPointName 461
Total 67, 74, 78, 170, 192, 197, 203, 208, 216, 231, 235, 239, 245, 249, 254, 276, 282, 298, 301, 303, 306
Total_2 74, 197, 208, 235, 249, 298
TotalDebtEmployee 357
TotalDebtEmployerContribution 357
TotalExcludingVAT 74, 197, 208
totalQuantity 323, 328, 337
totalQuantityInStock 450, 452, 454, 455
totalQuantityRequired 402, 404, 408
totalReceivedQuantity 411, 417, 421, 425, 429, 434
TotalToPay 74, 192, 197, 203, 208, 231, 235, 245, 249
TotalVAT 74, 192, 197, 203, 208, 231, 235, 245, 249
totp-secret 2
Transaction ID 225
TRANSACTION_ID 225
transactionDate 364, 366, 367, 371, 450, 454
TransactionInformation 25, 67, 170, 179, 282, 293, 298, 359, 364
TransactionInformationSettings 25
transferDate 461, 463
translatedResponseState 411, 417, 421, 425, 429, 434
TrustedSenders 352

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

type 47, 50, 52, 55, 156, 279, 345, 375, 377, 379, 381, 382, 384, 392, 410
Type__url 53
Type_AccountAsset 53
Type_AccountAssetId 53
Type_AccountDepreciation 53
Type_AccountDepreciationId 53
Type_AccountRevaluation 53
Type_AccountRevaluationId 53
Type_AccountSaleLoss 53
Type_AccountSaleLossId 53
Type_AccountSaleWin 53
Type_AccountSaleWinId 53
Type_AccountValueLoss 53
Type_AccountValueLossId 53
Type_AccountWriteDown 53
Type_AccountWriteDownAck 53
Type_AccountWriteDownAckId 53
Type_AccountWriteDownId 53
Type_Description 53
Type_Id 53
Type_InUse 53
Type_Notes 53
Type_Number 53
Type_Type 53
TypeId 47, 50, 52

- U -

unit 39, 74, 82, 88, 197, 208, 235, 249, 298, 345
Unit__url 353
Unit_Code 353
Unit_CodeEnglish 353
Unit_Description 353
UnitByCode 353
unitCost 323, 328, 337, 341, 345
unitPrice 323, 328, 337, 341, 345
Units 354
unmatchedValue 390
updated 154
updatedBy 341, 345
URL 225
URLConnection 43
Use HTTP Disk Cache 225
Use HTTP Memory Cache 225
use-batch-insert 2
use-http-disk-cache-read 2
use-http-disk-cache-write 2
use-http-memory-cache-read 2
use-http-memory-cache-write 2
userId 47, 341, 345
UserName 47
userType 163, 169
usingCompanyAddress 437, 439, 442, 444, 445, 447
usingStockPoints 452, 455, 458, 459

- V -

VacationDaysPaid 106, 111, 115, 120, 124, 128, 133
VacationDaysPendingPaid 106, 111, 115, 120, 124, 128, 133
VacationDaysPendingPrepaid 106, 111, 115, 120, 124, 128, 133
VacationDaysPendingSaved 106, 111, 115, 120, 124, 128, 133
VacationDaysPendingSavedYear1 106, 111, 115, 120, 124, 128, 133
VacationDaysPendingSavedYear2 106, 111, 115, 120, 124, 128, 133
VacationDaysPendingSavedYear3 106, 111, 115, 120, 124, 128, 133
VacationDaysPendingSavedYear4 106, 111, 115, 120, 124, 128, 133
VacationDaysPendingSavedYear5 106, 111, 115, 120, 124, 128, 133
VacationDaysPendingSavedYear6Plus 106, 111, 115, 120, 124, 128, 133
VacationDaysPendingUnpaid 106, 111, 115, 120, 124, 128, 133
VacationDaysPrepaid 106, 111, 115, 120, 124, 128, 133
VacationDaysRegisteredPaid 106, 111, 115, 120, 124, 128, 133
VacationDaysRegisteredPrepaid 106, 111, 115, 120, 124, 128, 133
VacationDaysRegisteredSaved 106, 111, 115, 120, 124, 128, 133
VacationDaysRegisteredSavedYear1 106, 111, 115, 120, 124, 128, 133
VacationDaysRegisteredSavedYear2 106, 111, 115, 120, 124, 128, 133
VacationDaysRegisteredSavedYear3 106, 111, 115, 120, 124, 128, 133
VacationDaysRegisteredSavedYear4 106, 111, 115, 120, 124, 128, 133
VacationDaysRegisteredSavedYear5 106, 111, 115, 120, 124, 128, 133
VacationDaysRegisteredSavedYear6Plus 106, 111, 115, 120, 124, 128, 133
VacationDaysRegisteredUnpaid 106, 111, 115, 120, 124, 128, 133
VacationDaysSaved 106, 111, 115, 120, 124, 128, 133

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

VacationDaysSavedEmploymentRateYear1 106, 111, 115, 120, 124, 128, 133
VacationDaysSavedEmploymentRateYear2 106, 111, 115, 120, 124, 128, 133
VacationDaysSavedEmploymentRateYear3 106, 111, 115, 120, 124, 128, 133
VacationDaysSavedEmploymentRateYear4 106, 111, 115, 120, 124, 128, 133
VacationDaysSavedEmploymentRateYear5 106, 111, 115, 120, 124, 128, 133
VacationDaysSavedEmploymentRateYear6Plus 106, 111, 115, 120, 124, 128, 133
VacationDaysSavedYear1 106, 111, 115, 120, 124, 128, 133
VacationDaysSavedYear2 106, 111, 115, 120, 124, 128, 133
VacationDaysSavedYear3 106, 111, 115, 120, 124, 128, 133
VacationDaysSavedYear4 106, 111, 115, 120, 124, 128, 133
VacationDaysSavedYear5 106, 111, 115, 120, 124, 128, 133
VacationDaysSavedYear6Plus 106, 111, 115, 120, 124, 128, 133
VacationDaysUnpaid 106, 111, 115, 120, 124, 128, 133
VacationDebtBasis_LastDay 355
VacationDebtBasis_Month 355
VacationDebtBasis_VoucherNumber 355
VacationDebtBasis_VoucherSeries 355
VacationDebtBasis_VoucherYear 355
VacationDebtBasis_Year 355
VacationDebtBasisByYearAndMonth 355
VacationDebtBasisEmployeesByYearAndMonth 357
VariableEarned 357
VariableUnused 357
VAT 39, 74, 197, 208, 235, 249, 276, 298, 306
VATCode 25, 27, 197, 208, 235, 249
VATIncluded 67, 74, 170, 192, 197, 203, 208, 231, 235, 245, 249, 282
VATType 298, 306
version 323, 328, 333, 337, 461, 463
voided 377, 379, 381, 382, 384, 386, 387, 390, 392, 394, 396, 397, 399, 411, 417, 421, 425, 429, 434, 461, 463
Voucher Lines 359
Voucher__url 369
Voucher_ApprovalState 369
Voucher_Comments 369
Voucher_CostCenter 369
Voucher_Description 369
Voucher_Project 369
Voucher_ReferenceNumber 369
Voucher_ReferenceType 369
Voucher_TransactionDate 369
Voucher_VoucherNumber 369
Voucher_VoucherSeries 369
Voucher_Year 369
VOUCHERAPPROVALSTATE 359, 360
VoucherDescription 359, 360, 363
VoucherFileConnection__url 361
VoucherFileConnection_FileId 361
VoucherFileConnection_VoucherDescription 361
VoucherFileConnection_VoucherNumber 361
VoucherFileConnection_VoucherSeries 361
VoucherFileConnection_VoucherYear 361
VoucherFileConnectionByFileId 361
VoucherFileConnections 363
VoucherLines 359
VoucherLinesByVoucherSeriesAndVoucherNumber 364
VoucherNumber 47, 179, 192, 197, 203, 208, 216, 293, 298, 306, 313, 357, 359, 360, 363, 364, 366, 367, 369, 371
VOUCHERREFERENCENUMBER 359, 360
Vouchers 360, 366
VouchersByVoucherSeries 367
VouchersByVoucherSeriesAndVoucherNumber 369
VoucherSeries 47, 179, 192, 197, 203, 208, 216, 259, 293, 298, 306, 313, 357, 359, 360, 363, 364, 366, 367, 369, 371, 372
VoucherSeries__url 373
VoucherSeries_Approver_Id 373
VoucherSeries_Approver_Name 373
VoucherSeries_Code 373
VoucherSeries_Description 373
VoucherSeries_Manual 373
VoucherSeries_NextVoucherNumber 373
VoucherSeries_Year 373
VoucherSeriesByCode 373
VouchersForCurrentFinancialYear 371
VOUCHERTRANSACTIONDATE 359, 360
VoucherYear 47, 179, 192, 197, 203, 208, 216, 293, 298, 306, 313, 357, 363

- W -

WageEarned 357
WageSaved 357
WageUnused 357
warehouseReady 192, 197, 203, 208, 245, 249, 377, 379, 381, 382, 384
WayOfDelivery 74, 82, 192, 197, 203, 208, 216, 231, 235, 245, 249

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

WayOfDelivery__url 465
WayOfDelivery_Code 465
WayOfDelivery_Description 465
WayOfDelivery_DescriptionEnglish 465
webshop 39
WebshopArticle 39
WithdrawnDays 106, 111
workedDate 341, 345
workedHours 341, 345
WorkType 313
WriteOffExist 177, 292

- Y -

Year 25, 27, 137, 303, 306, 355, 357, 364, 366, 367, 371, 372
YEARACCOUNTINGMETHOD 359, 360
YEAREND 359, 360
YEARID 359, 360
YEARSTART 359, 360
YourOrderNumber 74, 192, 197, 203, 208, 216, 245, 249
yourReference 74, 192, 197, 203, 208, 216, 231, 235, 239, 245, 249, 254, 298, 306, 411, 417, 421, 425, 429, 434
YourReferenceNumber 231, 235

- Z -

ZipCode 100, 192, 197, 203, 208, 231, 235, 245, 249, 311

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.



Invantive B.V.
Biesteweg 11
3849 RD Hierden
the Netherlands

Tel: +31 88 00 26 500
Fax: +31 84 22 58 178
info@invantive.com
invantive.com

IBAN NL25 BUNQ 2098 2586 07
Chamber of Industry and Commerce
13031406
VAT NL812602377B01
RSIN 8122602377
Managing Director: Guido Leenders
Registered office: Roermond

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.
24.0.92
29/06/2026, 19:20
ISBN 978-94-6302-000-8: