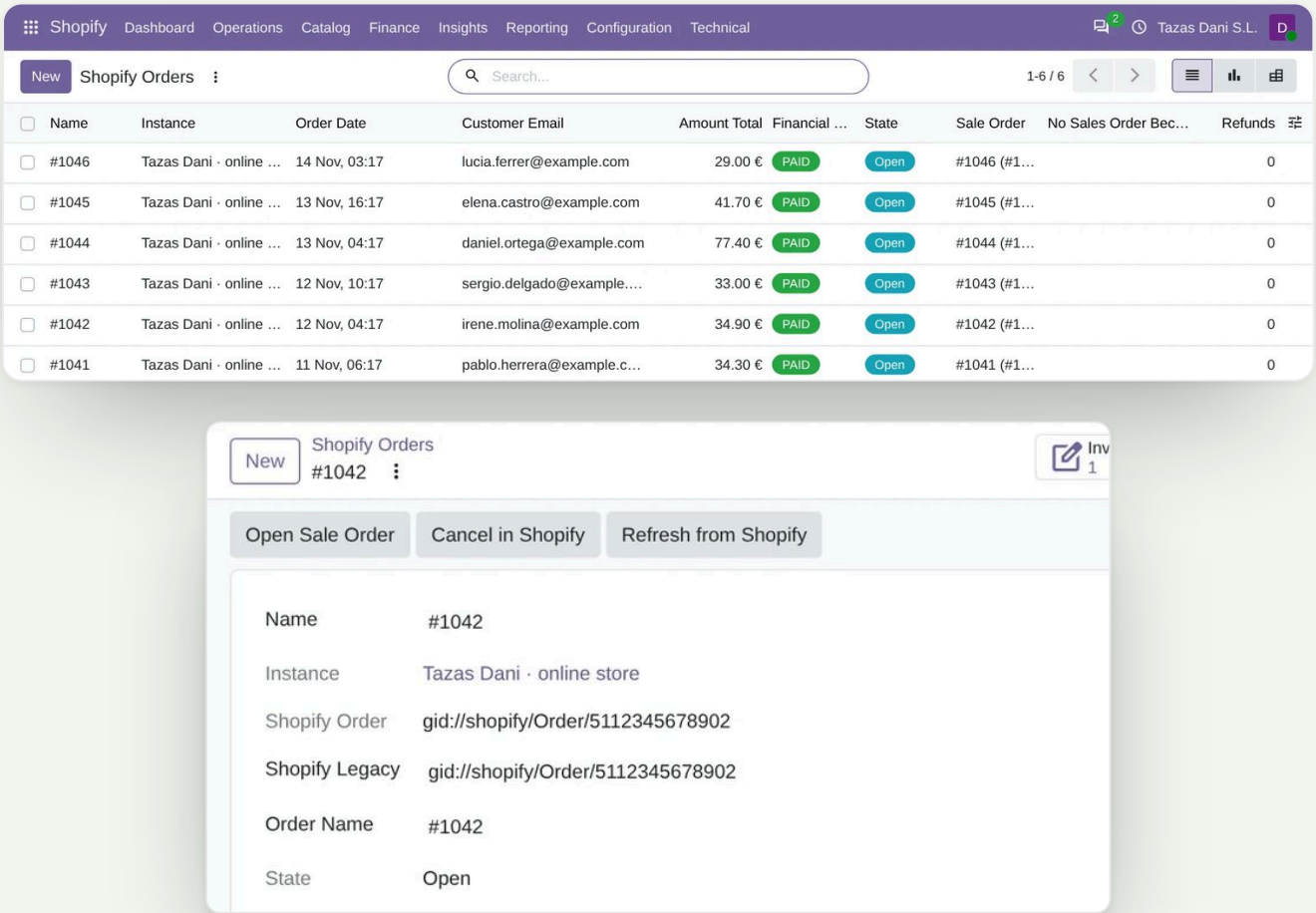
CONNECTOR GUIDE · MARKETPLACES

Shopify connector for Odoo

Your Shopify orders land in Odoo, and stock, prices, shipments and their tracking go back to Shopify.

Odoo 17 · 18 · 19 · 20 | 14-day trial, no card | Edition of 18-11-2026



Real Odoo 20 screenshot. The list of Shopify orders and the order detail with its linked Sale Order.

Contents

On the right, the page where each chapter starts. Menu and button names are those of Odoo in English.

1	Who it is for and what it solves	3
2	The connector in numbers	4
3	Where you see it in your Odoo	5
4	What you need before you start	6
5	The FlexigoTech Cloud trial	7
6	Connecting your store step by step	8
7	Shopify orders in Odoo	11
8	Shipping and tracking	14
9	Linked products and prices	17
10	Stock and locations	20
11	Refunds and returns	23
12	What Shopify pays you	26
13	Dashboard, reports and AI	28
14	The data that flows in and out	31
15	What runs automatically	31
16	The store options and how they come by default	33
17	Security and personal data	36
18	Several stores and several companies	37
19	If something fails	37
20	Other ways of doing it	39
21	Plans and pricing	39
22	Frequently asked questions	40
23	Shop Domain	42
24	Checklist and contact	44

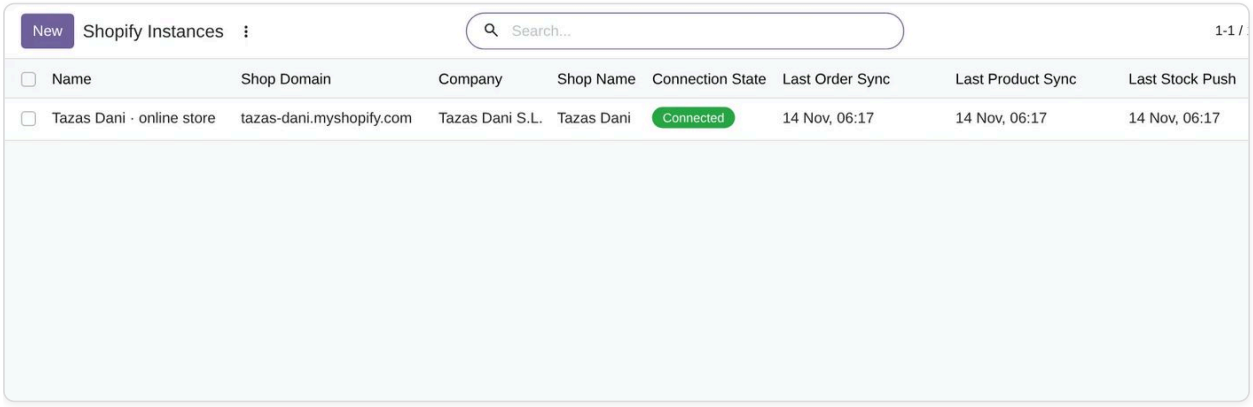
This guide describes version 20.0.26.1.23 of the module (Odoo branch 20.0, commit 1a0dbbfd1). Everything in it comes from its published code, the platform's official documentation or real screenshots.

The screenshots come from a test Odoo 20 database, with a made-up company and data and no real connection to Shopify. The module also has a version for Odoo 17, 18 and 19.

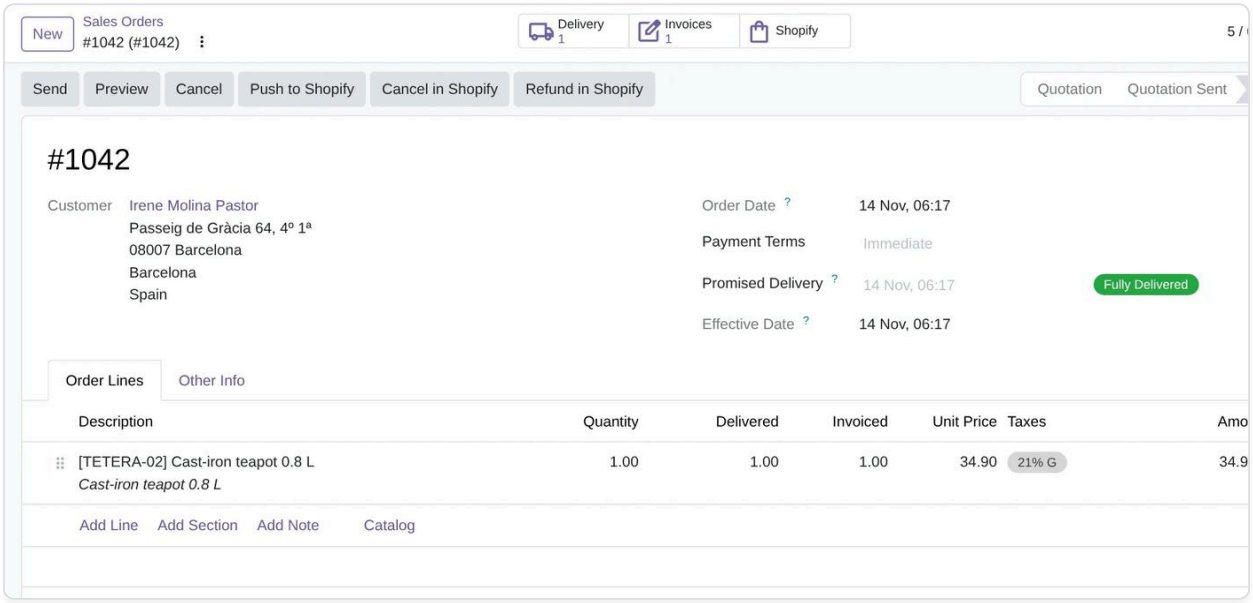
1 Who it is for and what it solves

> The connector needs the FlexigoTech Cloud subscription, with a 14-day trial and no card required. Without the trial or the subscription, the connector does not talk to Shopify.

The connector brings Shopify orders and products into Odoo, sends stock, prices and shipments with their tracking to Shopify, and collects refunds. Each Shopify connection is a store, in **Shopify › Configuration › Stores**. Without the connector, someone copies each order by hand, updates stock in two places and tells Shopify about each shipment.



- **Real Odoo 20 screenshot.** The list of **Shopify Instances** with the connection state of each store.



- **Real Odoo 20 screenshot.** A Shopify sale order with the buttons **Push to Shopify**, **Cancel in Shopify** and **Refund in Shopify**.

What changes day to day

- Shopify orders appear in **Shopify Orders** and link to their Sale Order.
- With **Sync Stock Enabled** on the store, Odoo stock is pushed to Shopify every 20 minutes, per location; and every 6 hours Odoo adjusts its stock to Shopify's.
- Odoo sale prices go to Shopify every hour, for single-variant products.
- From the delivery order you send the fulfilment to Shopify with **Send Shopify Fulfillment**.

- Shopify refunds appear in **Shopify Refunds**. If you enable the store's finance, invoices and credit notes are created by two scheduled tasks.

What it does not do

- It does not work alongside the Shopify Connector (flexigobe_shopify_connector): you must uninstall that one first.
- It is not real time: a store connected through FlexigoTech Cloud does not receive webhooks, so orders come in every 5 minutes and products every hour.
- It does not send Shopify the price of a product with several variants: Shopify sets that price.
- It does not create the Shopify app or its permissions: those are done in Shopify before connecting.
- It books nothing until you enable the store's finance, which is off by default.

CONNECTOR GUIDE · SHOPIFY

2 The connector in numbers

4
Odoo versions: 17, 18, 19 and 20

18
scheduled tasks, 11 active on installation

36
own menus

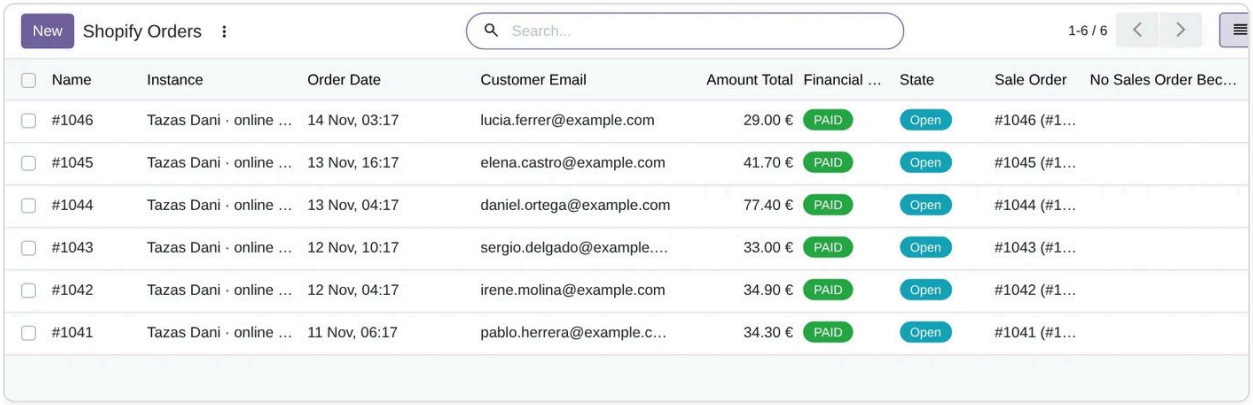
3 and 10
accounts per connector, with Standard and with Pro

14
days of trial, no card required

14,491
lines of code

The connector needs the Odoo apps Sales, Inventory, Invoicing and Delivery, all from Odoo Community, and the FlexigoTech Cloud module. It is published for Odoo 17, 18, 19 and 20 with the same automated tests on all four (502 on Odoo 20), which answer with a simulator of the Shopify Admin GraphQL API. The module's queries were checked, read-only, on a Shopify development store in September 2026.

3 Where you see it in your Odoo



- **Real Odoo 20 screenshot.** List of **Shopify Orders** with the state of each order and its sale order.

Dashboard	The indicators of each store.
Operations	**Orders**, **To Ship** and **Refunds & Returns**.
Catalog	**Linked Products** and **Restock Plan**.
Shopify › Finance	**Control Tower**, **Payment Transactions**, **Fee Analysis**, **Payouts**, **Balance Transactions** and **Disputes**.
Insights	**Ask AI** and **AI Reports**.
Reporting	**Sales Analytics**, **KPI Trends**, **Fulfilment Funnel** and **Stock Forecast**.
Configuration	**Settings**, **Stores**, **Mappings** (with **Carriers**, **Payment Gateways**, **Taxes** and **Locations**) and **Sync Now**.
Technical	**Sync Logs** and **Webhook Events**.

Outside its own app, the connector adds the buttons **Push to Shopify**, **Cancel in Shopify** and **Refund in Shopify** to sale orders. On delivery orders it adds **Send Shopify Fulfillment**. The product form gains a **Shopify** tab with the Shopify data, and the **Settings** screen has a **Shopify Seller** section.

4 What you need before you start

- A Shopify store with its myshopify.com domain and its own app, created in the Dev Dashboard of that store's organisation, from which you will need the Client ID and the Client Secret. It can also be an app distributed to your store, which is authorised once in Shopify.
- A Shopify administration with permission to accept the scopes the app asks for. If the app changes its permissions, it must be authorised again.
- Odoo 17, 18, 19 or 20, on Odoo.sh or on your own server. It does not work on Odoo Online, which does not allow installing third-party modules.
- The Odoo apps it depends on, all from Odoo Community: Sales, Inventory, Invoicing and Delivery, plus FlexigoTech Cloud. Odoo installs them with the connector.
- An Odoo administrator user: only an administrator can accept the consent and save the Client Secret.
- The FlexigoTech Cloud subscription, with a 14-day trial and no card required. Without the trial or the subscription, the connector does not talk to Shopify.
- Do not install Shopify Connector (flexigobe_shopify_connector) at the same time: if you have it, uninstall it first.

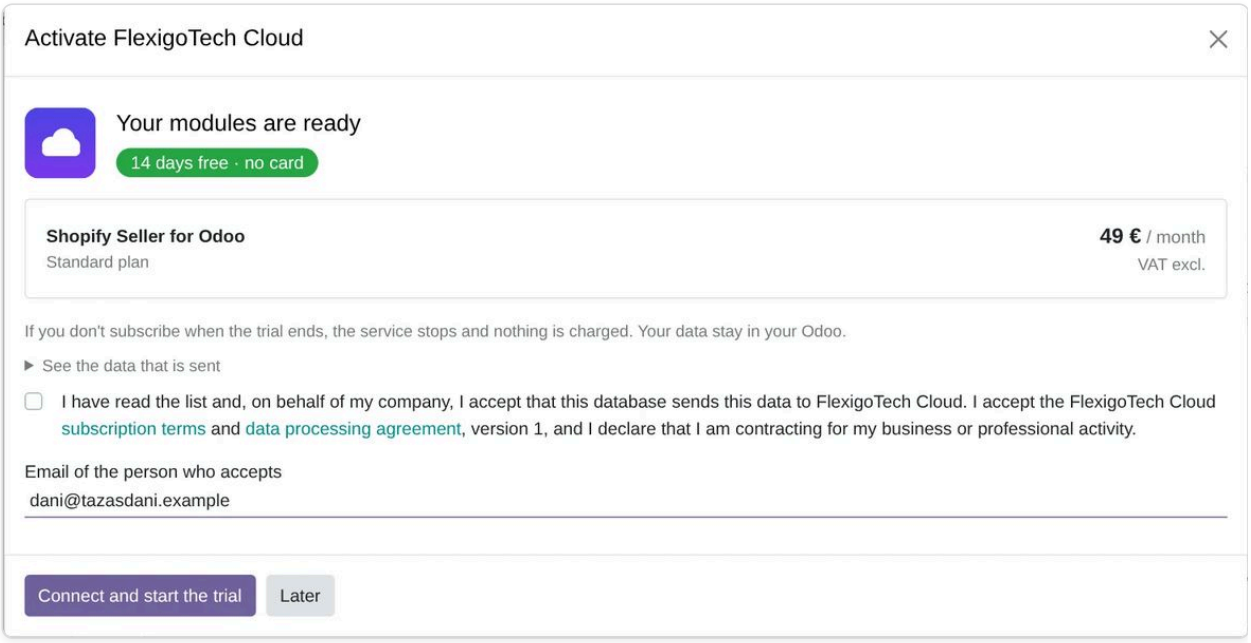
> The connector does not create or configure the app in Shopify. It also does not give you the Client ID or the Client Secret, nor does it grant on your behalf the scopes it needs. All of that is done in Shopify before connecting.

It is installed like any third-party Odoo module, on Odoo.sh or on your server. Odoo installs at the same time `flexigotech_cloud`, the connection to FlexigoTech Cloud, and the apps it depends on: Sales, Inventory and Invoicing.

5 The FlexigoTech Cloud trial

1 **Press Activate in the bar at the top**

It appears as soon as you install the module. It opens the **Activate FlexigoTech Cloud** window, with the Standard plan and its 14 days free, no card required.



- **Real Odoo 20 screenshot.** The activation window: read the list of data, tick the box and enter your email.

2 **Accept and enter your email**

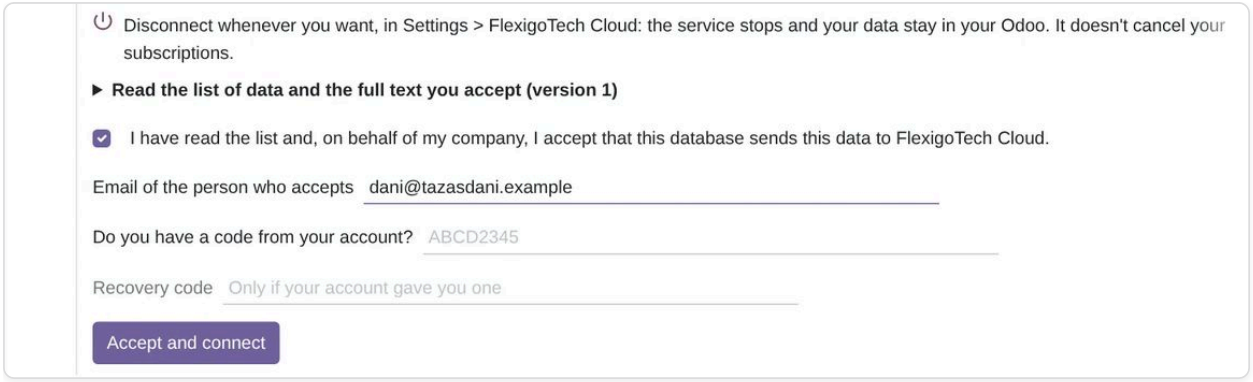
Tick «I have read the list and, on behalf of my company, I accept...» (it covers the subscription terms and the data processing agreement) and type the **Email of the person who accepts**.

3 **Press Connect and start the trial**

Odoo registers the database with FlexigoTech Cloud and starts the module's trial. If you leave it for later, **Later** closes the window.

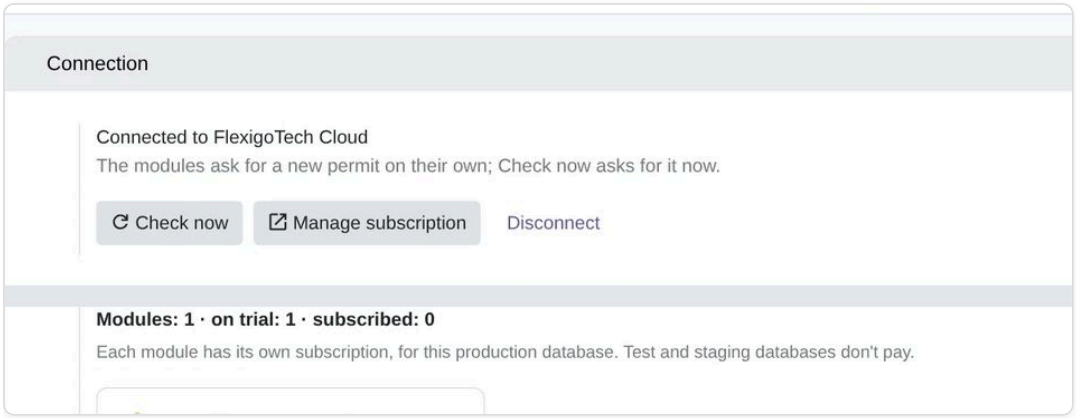
The same, from Settings

In **Settings › FlexigoTech Cloud** is the full list of what the database sends. Tick the box, enter the email and press **Accept and connect**. If you already have an account at app.flexigotech.com, the field **Do you have a code from your account?** links the database to it (the code is valid for 10 minutes).



- **Real Odoo 20 screenshot.** Settings › FlexigoTech Cloud, before connecting: the box ticked, the email and **Accept and connect**.

With the database connected, the **Modules** block shows each module with its state: **Trial** and the date it ends, **Subscribed**, and so on. From there you choose the plan (Standard or Pro, monthly or yearly) and press **Subscribe**: payment is made on the Stripe page, outside Odoo.



- **Real Odoo 20 screenshot.** Settings › FlexigoTech Cloud with the database connected and the connector on trial.

Prices exclude VAT. If you do not subscribe when the trial ends, the service stops and nothing is charged; your data stays in your Odoo. **Check now** requests the permission immediately (otherwise the module renews it by itself every 6 hours).

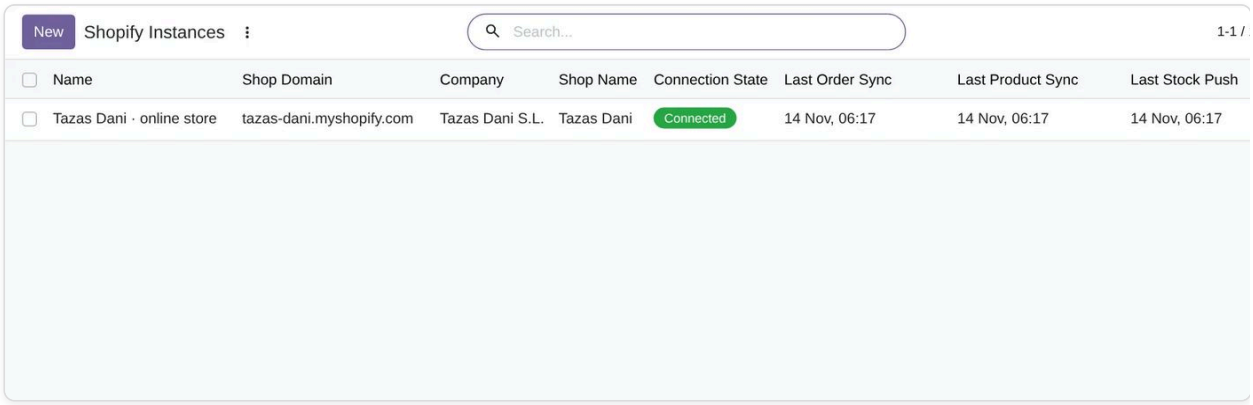
> In a neutralised copy (staging) or in a database with demo data you cannot start a trial: Settings says «This database is a neutralised copy or has demo data, so trials can't be started here.»

CONNECTOR GUIDE · SHOPIFY

6 Connecting your store step by step

1 Open the stores menu

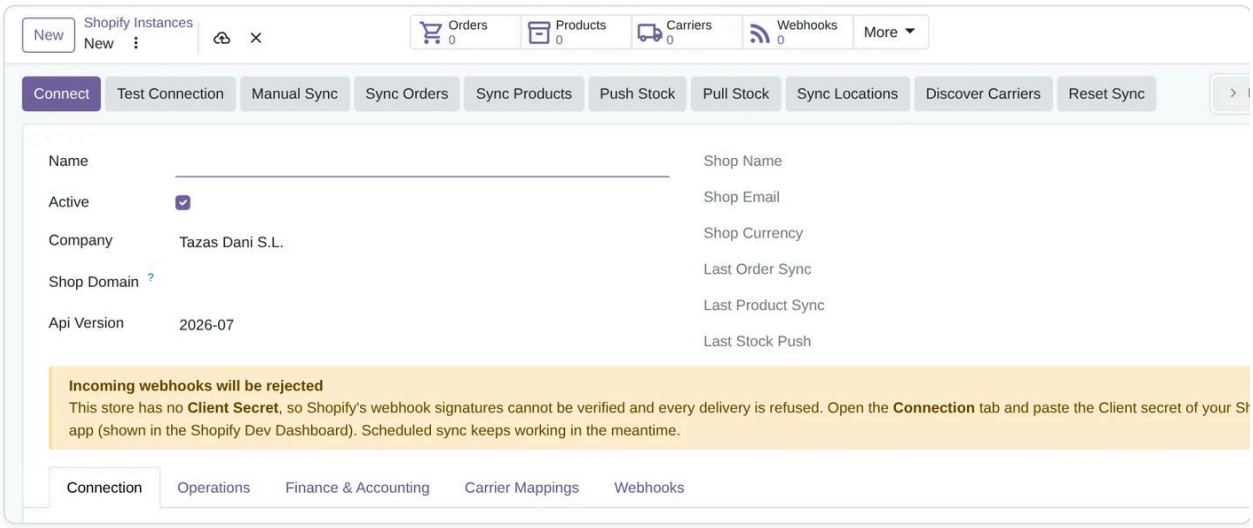
Go to **Shopify › Configuration › Stores**. The **Shopify Instances** list opens.



- **Real Odoo 20 screenshot.** The **Shopify Instances** list with one connected store.

2 Press New

The form for a blank store opens, with the state **Unconfigured**.



- **Real Odoo 20 screenshot.** A new store, with the state **Unconfigured**.

3 Fill in the store details

Enter the **Name** and the exact **Shop Domain** from Shopify, for example tazas-dani.myshopify.com. A custom domain of the store will not do. Check the **Company**.

4 Open the Connection tab

In the **FlexigoTech Cloud** group, the connection route for a new store is always FlexigoTech Cloud. Choose the type of app in **Shopify app**: your own, created in the Dev Dashboard, or one distributed to this store.

5 Enter the credentials

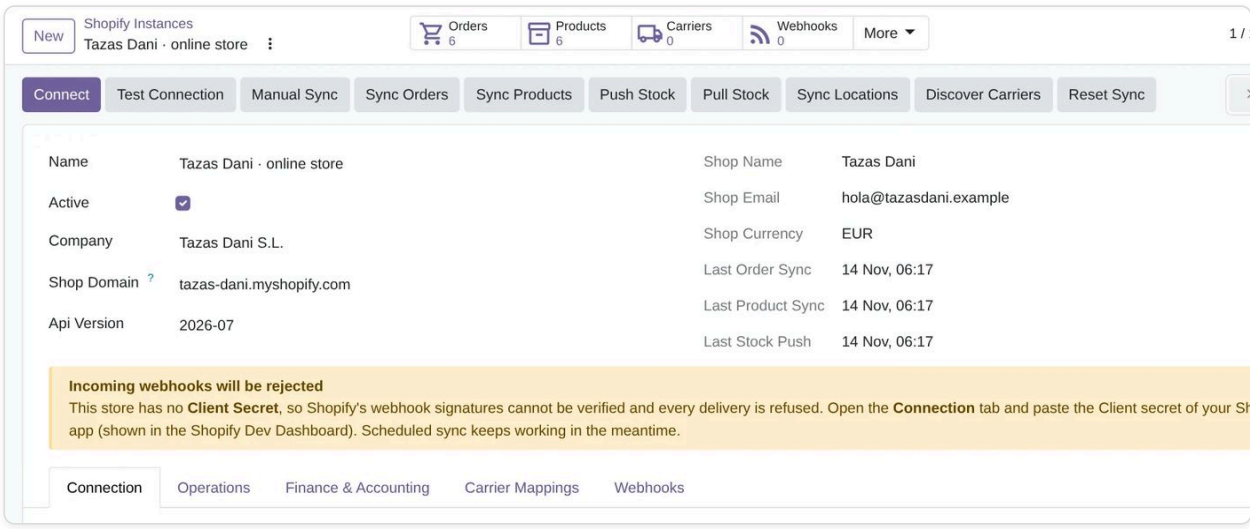
Tick the box **I accept that my Shopify app and my calls to Shopify go through FlexigoTech Cloud**: until then the secret field does not accept input. Then enter the Client ID in **Client ID** and the Client Secret in **Client secret (FlexigoTech Cloud)**. On saving, the secret is sent once and encrypted to FlexigoTech Cloud, and is not stored in Odoo.

6 Connect

With an own app, press **Connect**: there is nothing to authorise. With an app distributed to the store, press **Connect with Shopify** and accept the permissions on the Shopify page, which returns you to Odoo. The dates **Saved in FlexigoTech Cloud** and **Authorized through FlexigoTech Cloud** show that each step has been done.

7 Test the connection

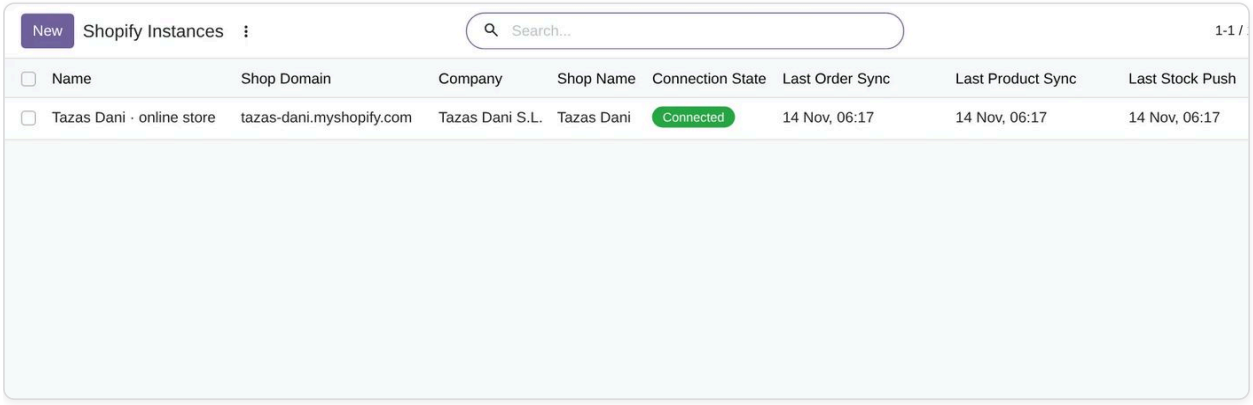
Press **Test Connection**. If all goes well, the state changes to **Connected** and **Shop Name**, **Shop Email** and **Shop Currency** are filled in. If it fails, the reason is kept in **Last Connection Error**.



- **Real Odoo 20 screenshot.** The store in the **Connected** state.

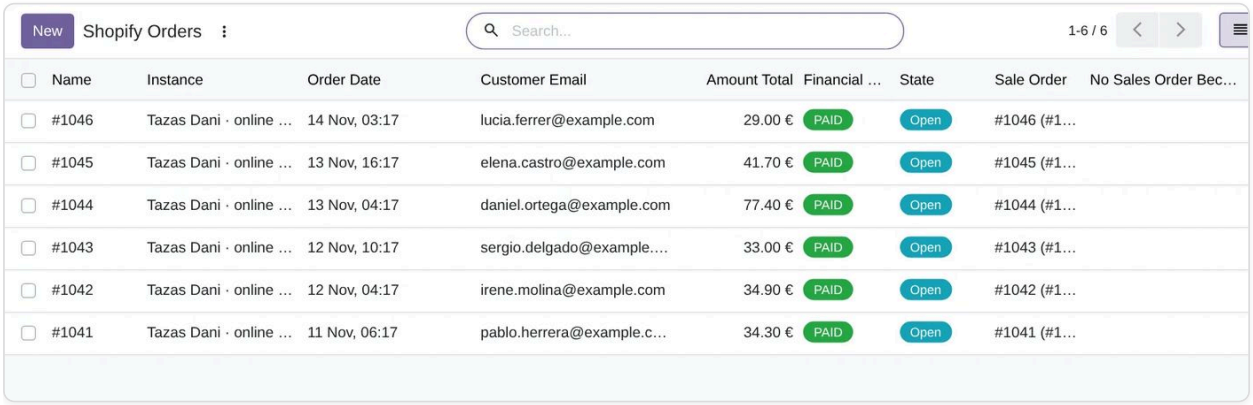
> Only an Odoo administrator can tick the consent and enter the Client Secret. If you are not one, Odoo sends nothing and warns that you must accept first. To authorise a distributed app, whoever does it must have permission in Shopify to accept the permissions the app asks for.

Once connected, the store does not receive real-time notices: orders and products arrive with the scheduled tasks, and the **Register Webhooks** button is not visible. The tasks **Shopify Sync Orders** (every 5 minutes) and **Shopify Sync Products** (every hour) are off on installation, and switch on by themselves when you save an active store. You can also run the import by hand with **Sync Orders**, **Sync Products**, **Push Stock** or **Pull Stock**, or with **Manual Sync**.



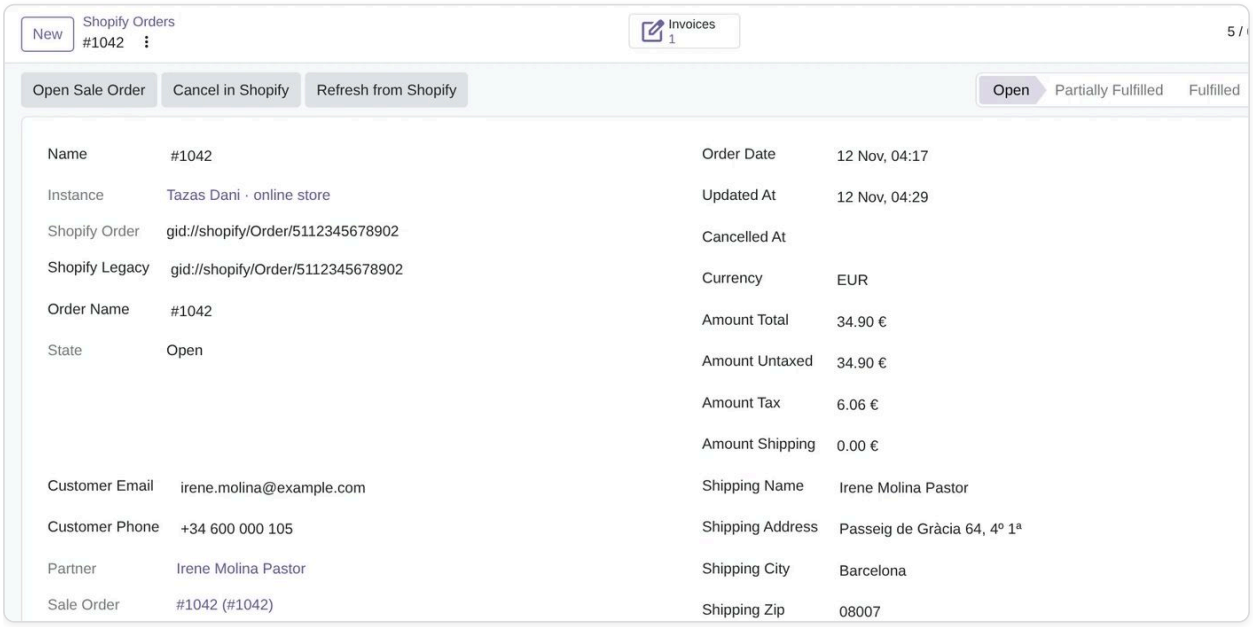
- **Real Odoo 20 screenshot.** The list of stores with the **Connection State** column at **Connected**.

7 Shopify orders in Odoo



- **Real Odoo 20 screenshot.** The list of the store's Shopify orders, with their financial state, their state and their sale order.

Go in through **Shopify › Operations › Orders**. The task **Shopify Sync Orders** brings the orders every 5 minutes: it is off on installation and switches on by itself as soon as you save an active store with **Sync Orders Enabled**, which is on by default. To not wait, press **Sync Orders** on the store. Taxes are assigned with **Shopify › Configuration › Mappings › Taxes**.



- **Real Odoo 20 screenshot.** The record of a Shopify order, with its amounts and the link to its sale order.

Create Sale Orders	From the list, creates the sale order of the selected orders that do not have one yet. Only for Shopify operators.
Open Sale Order	Opens the sale order of this order. Hidden if the order does not have one yet.
Create Sale Order	Creates the sale order by hand. Hidden if the order already has one. Only for the Shopify manager.
Cancel in Shopify	Cancels the order in Shopify. Hidden if the order is already cancelled.
Refresh from Shopify	Fetches the order again from Shopify.
Create Invoice	Creates the invoice for the order. It is hidden if the order has no sale order or already has an invoice.

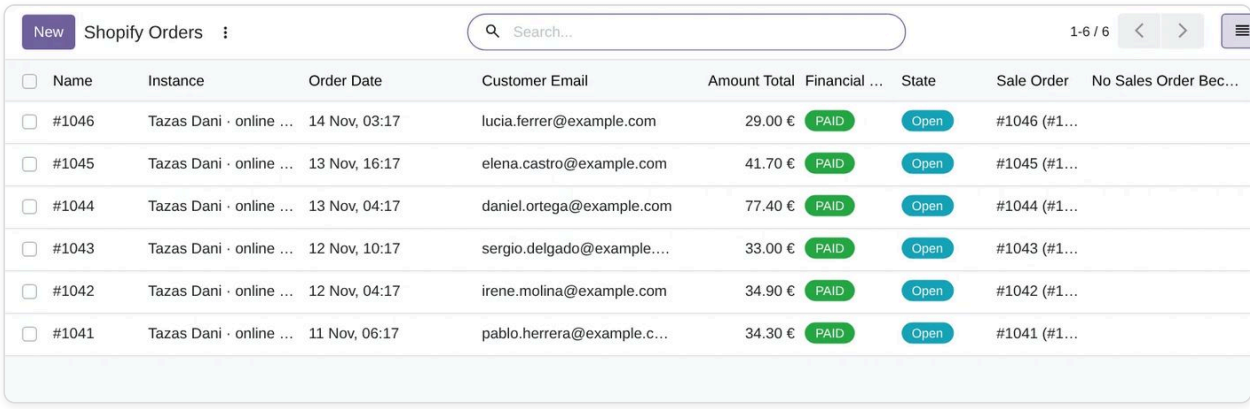
The **State** field can be **Open**, **Partially Fulfilled**, **Fulfilled**, **Cancelled** or **Refunded**. It starts as **Open**.

1 Import the orders

They arrive on their own every 5 minutes. To avoid waiting, click **Sync Orders** on the store.

2 Open the order

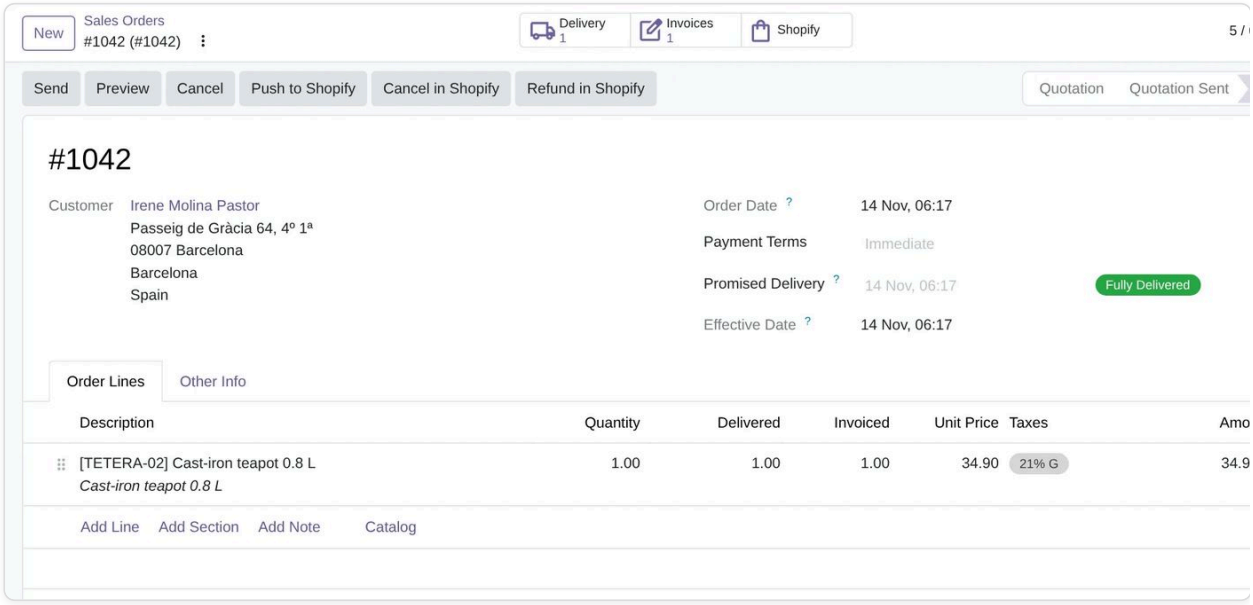
In **Shopify Orders**, click the order you are interested in.



- **Real Odoo 20 screenshot.** Each row is an imported order, with its **State** and its **Sale Order**.

3 Jump to the sale order

Click **Open Sale Order** to see the lines with their taxes.



- **Real Odoo 20 screenshot.** The Odoo sale order created from the Shopify order.

> An order can only be deleted when Shopify has cancelled it and it does not hold any sale order. Changing an order's record requires being a Shopify Manager, and the buttons that call Shopify require being an Operator or Manager.

The Shopify order states

Open	The order is imported and not yet fulfilled. It is the state it enters with.

Partially Fulfilled	Shopify has prepared only part of the order.
Fulfilled	Shopify has prepared the whole order.
Cancelled	The order is cancelled. The **Cancelled At** field records when.
Refunded	The order has been refunded.

When the order does not generate a sale order

A field on the order explains why the connector does not create the sale order, so that it is not sent twice. There are five reasons. The labels of those options appear in English on screen, so they are described here in words.

- Someone deleted the sale order in Odoo.
- Shopify had already fulfilled or cancelled the order when it was imported.
- The order predates the date from which the store creates sale orders.
- The order had no sale order when the connector was updated, either because it was deleted or because its creation failed.
- Its Shopify reference already appears on sale orders that this store did not make, whether by hand or from another store. The connector does not link any of them blindly.

With one of these reasons, **Create Sale Order** still creates it, as a quotation. The connector never confirms it, even if the store has automatic confirmation switched on. The order is marked as created by hand.

Rules and conditions

- Creating the sale order by hand requires being a Shopify Manager. Otherwise it shows "Only a Shopify Manager can create the sales order of a Shopify order by hand."
- Changing the fields that decide whether the sale order is created and shipped requires being a manager. Otherwise it shows "Only a Shopify Manager can change ... of a Shopify order: they decide whether the connector creates (and ships) its sales order."
- The buttons that call Shopify require being an Operator or Manager. Otherwise it shows "You need Shopify Operator (or Manager) access for this action." Anyone who only views can only read.
- With no data downloaded, **Create Sale Order** gives «No Shopify payload available to create sale order. Refresh from Shopify first.»
- An order that is already cancelled gives «This order is already cancelled.» and one with no Shopify identifier gives «No Shopify order ID found.»
- **Create Sale Orders** is a mass action: it skips the held orders, which are only created from their record.
- The order is never matched with the contact of an internal user, an employee, a company or the store's placeholder customer.

The sale order that results

The sale order stores the **Shopify Order Ref**, the **Shopify Instance**, the **Shopify Tags**, the **Shopify Note** and the **Shopify Amount Total**. If the Odoo total differs from the Shopify total beyond rounding, the **Shopify Total Mismatch** is flagged. Once set, the Shopify reference is not changed.

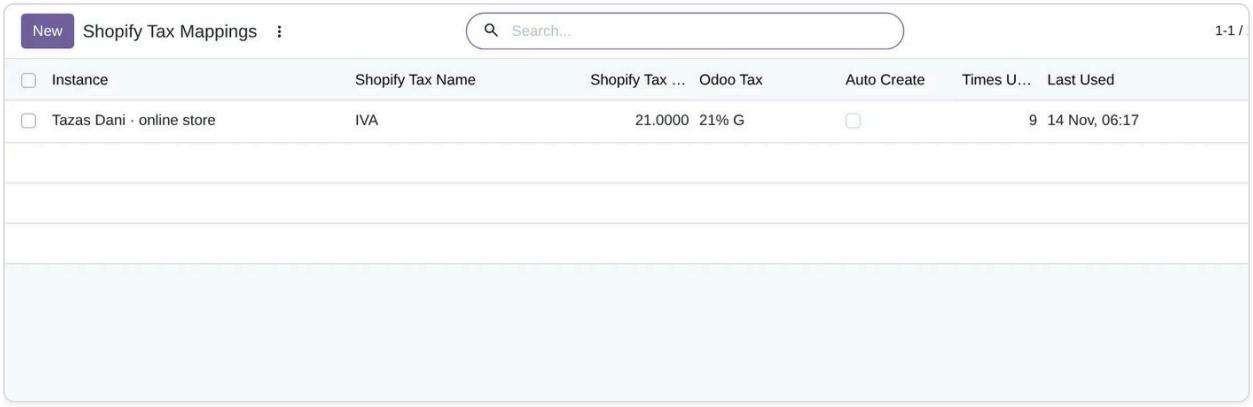
If you delete a sale order, the Shopify order is held with the reason for the deletion and no synchronisation will create it again.

Record fields and tabs

The record shows the name, the store, the dates, the amounts (total, untaxed, taxes and shipping), the customer's email and phone, the shipping details, the tags, the note, the financial and fulfilment states, the company and the sale order. The rest goes under **Shopify Details** and **Notes**. The **Technical** tab stores the **Raw Payload**. To locate orders, the list has the filters **Needs Action**, **Payment Pending**, **No Sale Order**, **Today** and **This Month**.

An order from start to finish

An order from the Tazas Dani store, placed by Irene Molina Pastor, comes in with state **Open**. In the list you can see its financial status, its state and its sale order. When the sale order is created, it appears with the same order number and the cast iron teapot as a line. From it, **Push to Shopify**, **Cancel in Shopify** and **Refund in Shopify** send the changes back to the store. The store's tax is resolved in **Shopify › Configuration › Mappings › Taxes**, where the VAT row notes how many times it has been used.

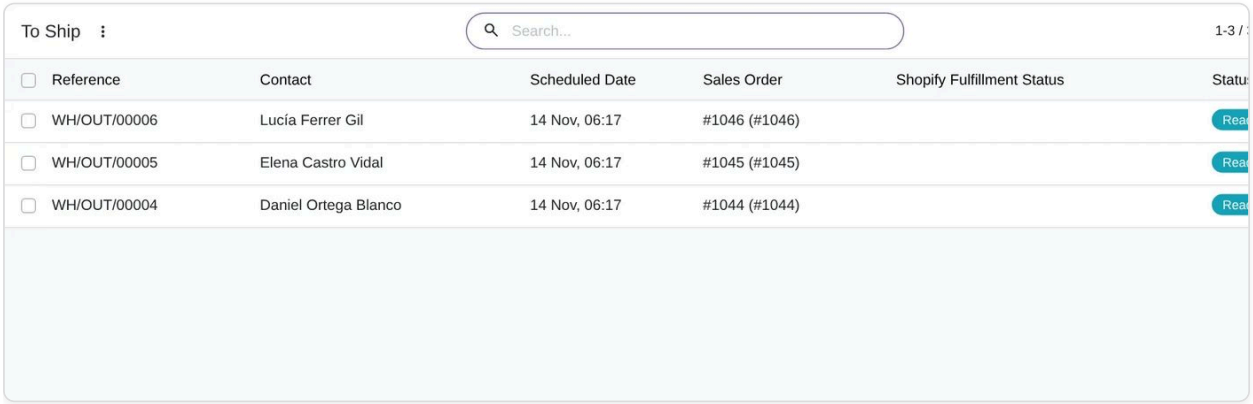


- **Real Odoo 20 screenshot.** The mapping of the Tazas Dani store's VAT tax to the Odoo tax.

The mapping links the **Shopify Tax Name** with the **Odoo Tax**. With **Auto Create** switched on, Odoo creates the tax at that rate the first time it encounters it. You can also create it with **Auto-Create Odoo Tax**. An unmapped tax has its filter, **Unmapped**.

CONNECTOR GUIDE · SHOPIFY

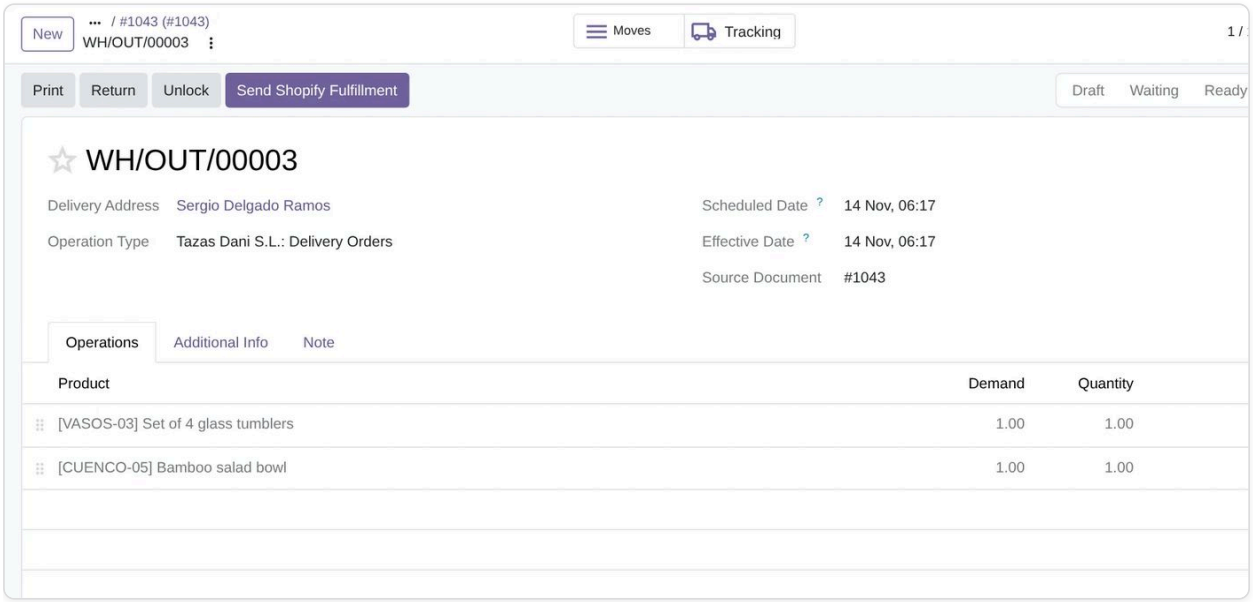
8 Shipping and tracking



- **Real Odoo 20 screenshot.** The **To Ship** list with three delivery orders in **Ready** state.

Pending deliveries are in **Shopify › Operations › To Ship**. The push to Shopify happens from three places: when validating the delivery (with **Auto Send Fulfillment On Validate**, which is switched on in new stores), when

saving a tracking number on a delivery already done, or with the buttons on the delivery order. The carrier sent to Shopify is that of the sale order (taken from the Shopify shipping line, cross-referenced with **Shopify › Configuration › Mappings › Carriers**) or, if it has none, that of the delivery; the tracking number is that of the delivery.



- **Real Odoo 20 screenshot.** Delivery order marked done with the **Send Shopify Fulfillment** button and the error Shopify returned.

Send Shopify Fulfillment	Sends the order's fulfilment. It only appears on a delivery of a Shopify order in **Done** state.
Update Shopify Tracking	Sends the tracking number to Shopify. It only appears if the delivery is done and already has a fulfilment in Shopify.
Retry send fulfillment	Repeats the send when it failed. It appears in the red error notice, with the delivery done.
Shopify Fulfillment Status	How the shipment stands in Shopify.
Shopify Fulfillment Error	The last error Shopify returned, if there was one.

The **Shopify** group on the delivery order stores the fulfilment, the date of the last shipment and the date of the last tracking sent. If FlexigoTech Cloud pauses the send, the delivery is validated, is not marked as failed, and the chatter notes "Not sent to Shopify".

1. **Validate the delivery**
   If automatic sending is enabled on the store, or if there is already a tracking number, Odoo sends the fulfilment to Shopify.

2. **Check the delivery order**
   Look at the fulfilment status and the error, if there is one.

3. **Correct and retry**
   Retry send fulfillment sends it again.

> A fulfilment cannot be sent from an unvalidated delivery, nor from one that does not belong to a Shopify order. If the delivery already has a fulfilment, use **Update Shopify Tracking**. Shopify Operator permission is required.

Fulfilment states

The delivery order shows **Shopify Fulfillment Status**, with the value Shopify returns. In the **To Ship** list you will see the Shopify delivery orders pending shipment, alongside their fulfilment status.

When it will not let you send

SITUATION	MESSAGE
The delivery is not validated	«Picking must be done before sending Shopify fulfillment.»
The delivery is not from a Shopify order	«This picking is not linked to a Shopify order.»
The delivery already has a fulfilment	«This picking already has a Shopify fulfillment. Use Update Shopify Tracking.»
Tracking missing when updating	«Tracking number is required to update Shopify tracking.»
The delivery has no fulfilment when updating tracking	«No Shopify fulfillment id found on this picking.»
The order has no store	«No Shopify instance linked to this picking sale order.»
The user lacks permission	"You need Shopify Operator (or Manager) access to push fulfillment updates to Shopify." (Odoo shows this untranslated)

Permission is checked on the server: a user with normal Inventory access but without the Shopify group cannot send updates with these buttons.

Automatic sending

On validation, Odoo sends the fulfilment if the store has automatic sending on validation enabled, or if the delivery order already carries a tracking number. If you save a new tracking number on a done delivery of a Shopify order, it is also sent. If Shopify rejects the send, it is recorded in **Shopify Fulfillment Error** and the chatter says "Shopify auto-fulfillment failed". If the tracking update fails, the chatter says "Shopify tracking auto-sync failed". If FlexigoTech Cloud pauses the send, the chatter notes "Not sent to Shopify" and no error is marked.

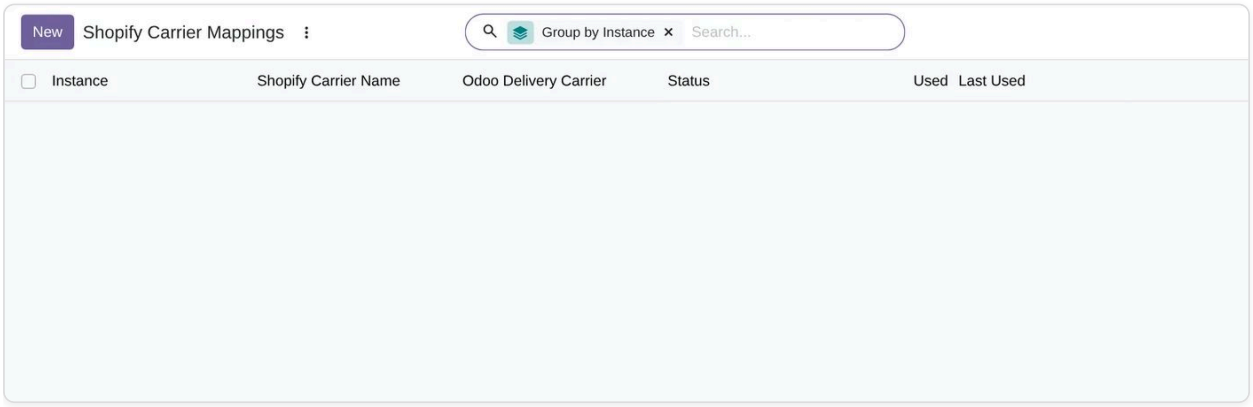
Carriers

In **Shopify › Configuration › Mappings › Carriers** the name of the Shopify carrier is related to one in Odoo. The name must match the shipping lines of Shopify exactly, ignoring case. The **State** of each row can be **Mapped**, **Auto Create** or **Unmapped**.

Odoo Delivery Carrier	The carrier assigned when that name appears on an order.
Auto-Create Carrier	If no carrier is assigned, creates one with that name.
Source	How it was discovered: from orders, from shipping profiles or by hand.

Last Used and Order Count	When and how many times it matched during order synchronisations.
Create Carrier	Creates the Odoo carrier now. It is hidden if the row already has one.
Open Carrier	Opens the assigned carrier.

For each name, Odoo first looks for a mapping with an assigned carrier, then one marked to auto create, then an Odoo carrier with that name belonging to the store's company or shared, and finally the store's default carrier.



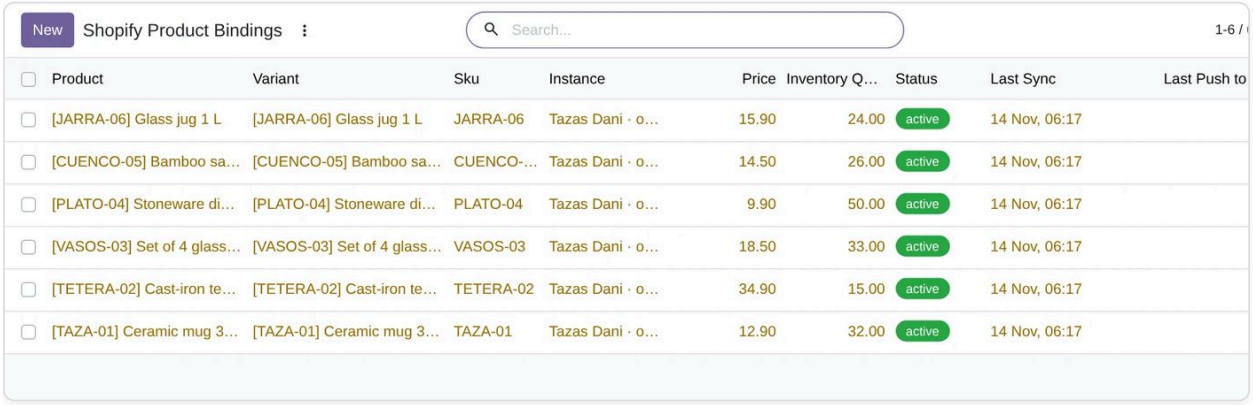
- **Real Odoo 20 screenshot.** List of **Shopify Carrier Mappings**, groupable by instance.

End-to-end example

In **To Ship** there are three delivery orders in **Ready** state, including WH/OUT/00004 for Daniel Ortega Blanco. You validate it. If the store sends on validation, or the delivery order already has tracking, the fulfillment goes to Shopify. If Shopify returns an error, it appears on the delivery order and you click **Retry send fulfillment**. Once a fulfillment exists, new tracking is sent with **Update Shopify Tracking** or simply by saving it.

CONNECTOR GUIDE · SHOPIFY

9 Linked products and prices



- **Real Odoo 20 screenshot.** The **Shopify Product Bindings** list with the store's six linked products.

You enter through **Shopify › Catalog › Linked Products**. Each line is a Shopify variant joined to an Odoo product. The Shopify data arrives with the **Shopify Sync Products** task, every hour (it switches itself on with the first active store), or with the **Sync Products** button on the store. The Odoo price goes out with the **Shopify Push Prices** task, every hour. Stock goes out with **Shopify Push Stock**, every 20 minutes, in stores with **Sync Stock Enabled**, which comes switched off.

Binding buttons

Push to Shopify	Sends this variant's price, compare-at price, SKU and barcode to Shopify. The price is the Odoo sales price on a single-variant product of a store with **Odoo sets the prices**; otherwise it is the one that came from Shopify.
Push Stock	Sends this variant's quantity to Shopify.
Open Product	Opens the Odoo product this variant is linked to.

Statuses in the list

- **Active** and **Archived** separate the bindings in use from the archived ones.
- **Never Pushed** marks the variants that have not yet been sent to Shopify.
- **Stale Sync (>24h)** marks those that have gone more than 24 hours without syncing from Shopify.
- **No Stock Sync** marks those that have not yet received stock.

Publishing an Odoo product to Shopify

1. **Open the product**
   Open the product record in Odoo. If it has no binding yet, you will see **Export to Shopify**.
2. **Click Export to Shopify**
   A wizard opens where you choose the store and the **Initial Status**: **Draft (hidden)** or **Active (visible in store)**.
3. **Confirm with Export to Shopify**
   The product is created in Shopify and linked.
4. **Update it afterwards with Push to Shopify**
   Once it is linked, this button on the product form replaces the export one.

> You cannot export a product that already has a binding in that store: Odoo asks you to use **Push to Shopify**. You also cannot export if there is no connected, ready store.

What each binding stores

Each binding joins an Odoo variant to a Shopify variant, in a specific store. In the form you see five groups: **Odoo**, **Shopify Product**, **Variant Data**, **Shopify IDs** and **Sync Status**. There are also three tabs: **Description**, **Variant Payload** and **Product Payload**, which show what Shopify sent, exactly as received.

Last Sync	When this variant was last read from Shopify.
Last Push to Shopify	When it was last sent from Odoo.
Last Stock Sync	When stock was last synced.
Never Pushed	Checkbox that is ticked automatically if the variant has never been sent to Shopify.

When the price goes out

If you change the sales price of a single-variant product in Odoo, the binding is marked to send its price. The scheduled price task sends it on its next run, every hour. This only happens on stores with **Odoo sets the prices** and that sell in the product's currency. If the price change came from Shopify, it is not sent back.

What changes on the Odoo product

The product record gains the **Shopify** tab, with the groups **Shopify Data** and **Description**. There you see the **Shopify Product GID**, the **Shopify Status** (**Active**, **Draft**, **Archived** or **Unlisted**), the **Shopify Vendor**, the **Shopify Product Type**, the **Shopify Tags** and the **Last Push to Shopify**. When the description is sent, the text goes exactly as it arrived from Shopify as long as you do not edit the sales description. If you edit it, the edited text is sent.

Rules and error messages

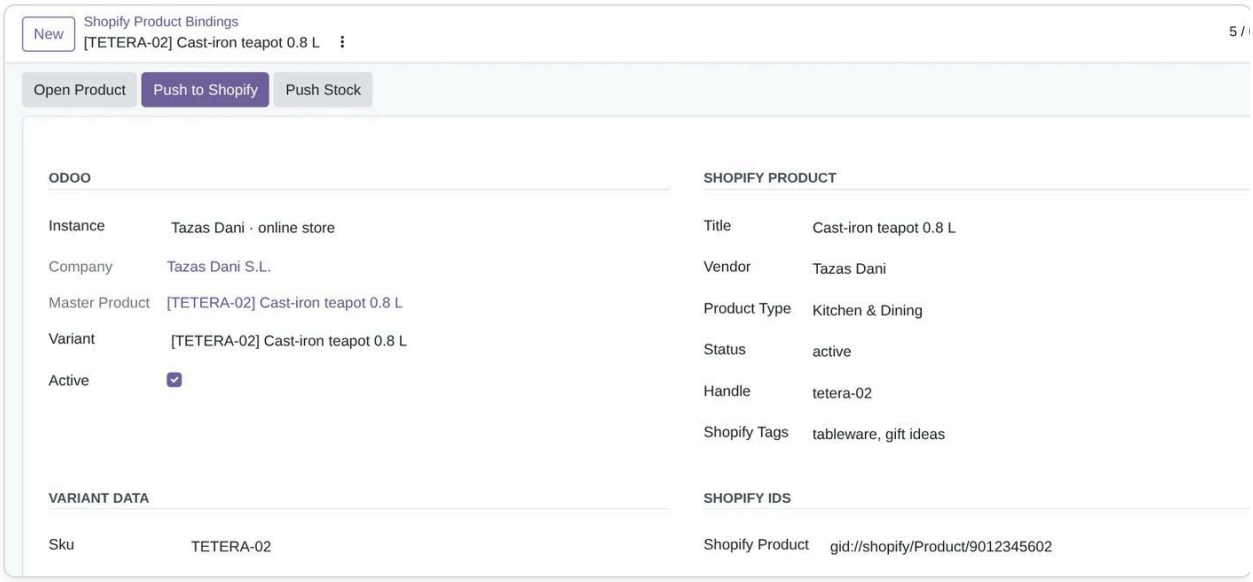
WHEN	WHAT ODOO SAYS
You click Push to Shopify on a product with no binding	«No Shopify bindings found for '...'. Use 'Export to Shopify' to publish it for the first time.»
You try to export a product already linked on that store	«Product '...' already has a Shopify binding on '...'. Use Push to Shopify to update it»
No store is connected and ready	«Connect a Shopify store first (Shopify > Configuration > Stores): there is no store ready to export this product to.»
Shopify rejects the creation	"Shopify export errors: ..."
Shopify does not return the ID of the created product	«Shopify did not return a product ID after export.»

For security, a product cannot be tied to a store of another company: Odoo checks this when creating or saving a product with Shopify data.

End-to-end example

1. **Starting point**
   The Tazas Dani store has six linked products in **Linked Products**, all active and with the last sync on 14 Nov, 6:23.

2. **Opening the teapot**
   You open [TETERA-02] Cast-iron teapot 0.8 L. In the **Shopify** tab you see its product GID and, in the binding, a quantity of 15.00 on hand.



- **Real Odoo 20 screenshot.** The binding for [TETERA-02] Cast-iron teapot 0.8 L with the **Push to Shopify** and **Push Stock** buttons.

3 **You click Push to Shopify**

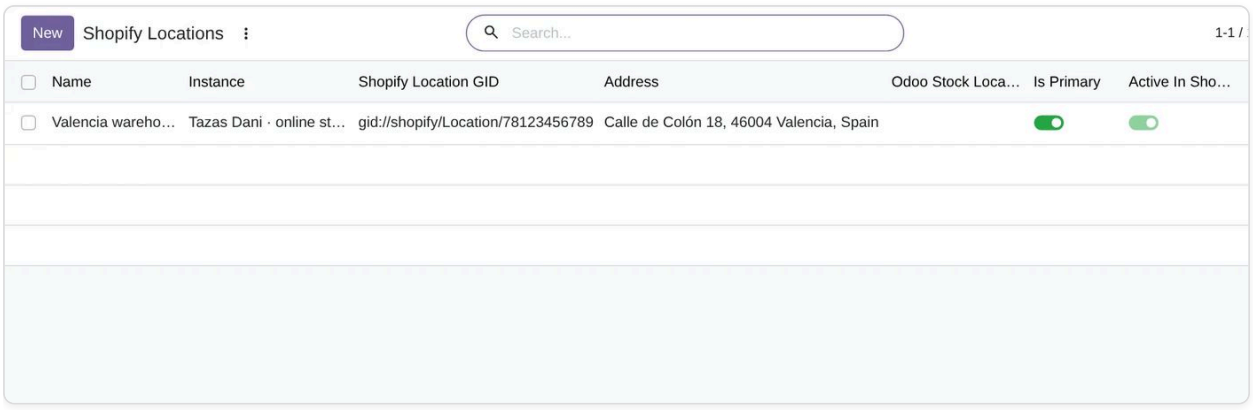
The price, the compare-at price and the SKU of the variant are sent to Shopify. The **Last Push to Shopify** field is updated.

4 **You click Push Stock**

Shopify receives the quantity for this variant and **Last Stock Sync** is updated.

CONNECTOR GUIDE · SHOPIFY

10 Stock and locations



- **Real Odoo 20 screenshot.** The **Shopify Locations** list with the Valencia warehouse location of the Tazas Dani store.

Where each piece of data comes from

DATA	HOW IT ARRIVES	WHEN
Stock from Odoo to Shopify	**Shopify Push Stock** task	Every 20 minutes, in stores with **Sync Stock Enabled** (it comes switched off)

DATA	HOW IT ARRIVES	WHEN
Shopify stock to Odoo (adjusts Odoo's stock)	Scheduled task that imports stock from Shopify	Every 6 hours, on the same stores
Replenishment plan	**Shopify Refresh Product Analytics** task	Daily, active

You see it in **Shopify › Configuration › Mappings › Locations**, where the **Shopify Locations** list opens. The plan is in **Shopify › Catalog › Restock Plan** and in **Shopify › Reporting › Stock Forecast**, which open the same list.

The location record

Odoo Stock Location	The Odoo internal location that corresponds to this Shopify location. It receives the free stock of that location and of those inside it, minus that of locations that already have their own Shopify location. An unassigned Shopify location receives nothing.
Is Primary	If it has no Odoo location assigned, the primary one receives the rest of the company's stock (and all of it, if none is assigned).
Set Primary and **Set as Primary**	Make this location the primary one of its store. They are hidden when it already is the primary.

Restock plan states

Each product has a **Restock State**: **OK**, **Watch**, **Reorder now** or **Stockout**. The **Reorder now**, **Stockout** and **Watch** filters leave you with only the ones that matter.

> The **Shopify Location GID**, **Address** and **Active In Shopify** fields come from Shopify and cannot be edited. Locations are fetched with **Sync Locations**, which is on the store form.

Locations and their mapping

Each Shopify location comes with its name, its **Shopify Location GID**, its **Address** and whether it is **Active In Shopify**. This data is read-only. The only things you decide are the **Odoo Stock Location** and which one is primary. In the list you can filter by **Primary**, **Unmapped** and **Inactive in Shopify**, or group by store with **Group by Instance**.

The form separates two groups: **Shopify Location** and **Odoo Mapping**.

What each restock state means

STATUS	WHAT IT IS
OK	The product does not sell, or its stock lasts longer than the supplier lead time plus the target cover days.
Watch	The stock lasts longer than the supplier lead time, but does not also cover the target cover days.
Reorder now	The stock lasts the same as or less than the supplier's delivery time (7 days when the store was created).
Stockout	The product sells and has no stock left.

The rotation class

Each product also has a **Mover Class**: **Winner**, **Steady**, **Slow** or **Dead**. The **Winners** and **Dead stock** filters pull out the extremes. You can group with **Group by Restock state** or **Group by Mover class**.

What you see on each line of the plan

Qty Available	The product's available stock.
Units 30d and **Velocity 30d (u/day)**	The units sold in the last measured period and their daily rate.
Revenue 30d	What that product was invoiced in the same period.
Cover and **Days Of Cover**	How many days the stock lasts. In the column it reads "Never", "Out of stock" or "N d".
Projected Stockout Date	The day the stock would run out.
Recommended reorder qty	The quantity it suggests you reorder.
Last Refresh	When the row was last recalculated.

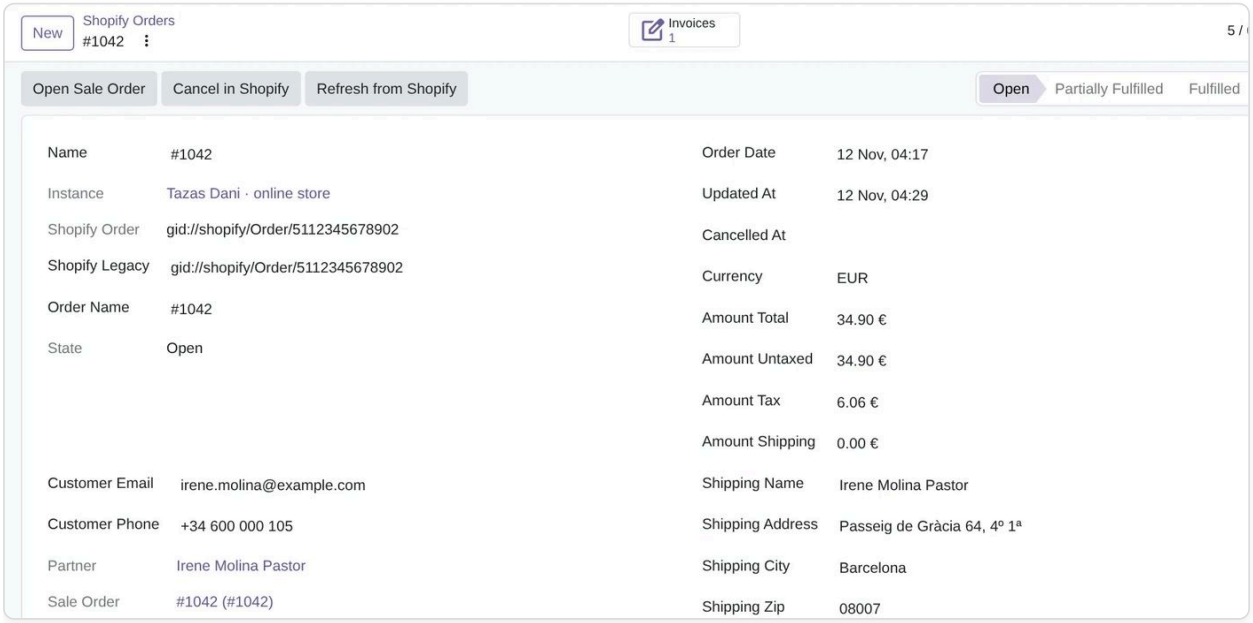
Rules

- The plan is recalculated daily for the products sold or linked in each store. There is also a task every 4 hours that refreshes the dashboard.
- The stock import task, which adjusts Odoo stock to Shopify's, runs every 6 hours. The stock push task runs every 20 minutes, and only for stores with **Sync Stock Enabled**.
- Shopify scheduled tasks can only be run manually by an administrator. If another user tries, Odoo shows "Only an administrator can run this scheduled action."

An end-to-end example

The store Tazas Dani has a location, Valencia warehouse, which appears in **Shopify Locations** with its Calle de Colón address. For stock to travel, you enable **Sync Stock Enabled** on the store and assign it an **Odoo Stock Location**. Every 20 minutes Odoo sends the available stock of that location to Shopify. When you need it right away, you go to **Shopify › Configuration › Sync Now**, choose the store and tick **Push Stock**. For the product [TETERA-02] Cast-iron teapot 0.8 L, which has 15 units, the link stores the date of its last push to Shopify and the date of its last stock sync.

11 Refunds and returns



- **Real Odoo 20 screenshot.** The Shopify order, with the **Cancel in Shopify** button at the top.

Where each piece of data comes from

The refund is recorded with the status **Draft**. The credit note is created by the scheduled task **Shopify Create Credit Notes**, every 3 hours, on stores with **Enable Finance & Accounting** and **Auto-create Credit Notes** turned on. You can also create it with the **Create Credit Note** button. The stock return is triggered by the **Return Stock** button.

Record buttons

Create Credit Note	Creates the credit note for the order's invoice. Visible with status **Draft** or **Failed**.
Retry	Returns a **Failed** refund to **Draft** so it can be tried again.
Ignore	Marks the refund as **Ignored**. Visible with status **Draft** or **Failed**.
Open Sale Order	Opens the linked sale order. Visible if the refund has a sale order.
Return Stock	Creates and validates the return transfer. Visible if the **Stock Return** is in **Pending Return**, **To Review** or **Return Failed**.

Statuses

- **Draft**: the refund is recorded and has no credit note yet.
- **Credit Note Created**: the credit note already exists in Odoo.
- **Ignored**: you have discarded it manually.
- **Failed**: the credit note could not be created. The reason is in **Error**.

How to use it

1 Open the list

Go to **Shopify › Operations › Refunds & Returns** and use the **Needs Action** filter.

2. **Review the refund**
   Check **Amount Refunded**, **Note** and the linked order.

3. **Create the credit note**
   Click **Create Credit Note**.

4. **Return the stock**
   If the **Stock Return** requires it, click **Return Stock**.

> The credit note cannot be created if the store does not have Finance enabled, if the refund has no sale order, or if the order has no validated invoice.

Refund statuses

Draft	Newly recorded refund, with no credit note. From here you can create the credit note or ignore it.
Credit Note Created	The credit note is created and linked in **Credit Note**.
Ignored	Discarded manually. It can be ignored from **Draft** or **Failed**.
Failed	Creating the credit note failed. **Retry** returns it to **Draft**.

Stock return statuses

N/A	The refund has no stock return.
Pending Return	There is stock to return and it has not been done.
To Review	The connector cannot tell whether the stock should come back. You decide.
Stock Returned	The return transfer is done and remains in **Return Transfer**.
Return Failed	The return gave an error. The reason is in **Return Error**.

Credit note rules

The credit note is created by reversing the order's validated invoice. A full refund is validated. A partial one is left in draft with a note, because the connector does not validate a partial amount on its own. The payment is settled against the same clearing account as the original payment, without recording a second payment.

These cases prevent the credit note from being created and Odoo shows an error:

- "Enable Finance on the store before creating credit notes." The store does not have Finance enabled.
- «No sale order linked to this refund. Cannot create credit note.»
- «No posted invoice found for sale order .... Invoice the order first before creating a credit note»
- "Failed to create the credit note: ..." The credit note could not be created.

Stock return rules

The connector decides on its own whether the stock should come back. It leaves the item in **To Review** if no completed delivery of the order shipped the product, if the warehouse already returned part of it with an Odoo return of its own, or if there is less left to return than Shopify restocked. When you click **Return Stock**, if the order has a completed delivery, the return comes out of that delivery. If it does not, a manual receipt is created from the customers location.

- «No restockable products found for this refund. Make sure products are linked by SKU or Shopify variant ID.»
- «No products could be resolved for the stock return.»
- «No warehouse configured for Shopify instance ....»
- «No incoming picking type found for warehouse ....»
- «The return transfer ... could not be validated.»
- «Failed to create stock return: ...»

Record fields and tabs

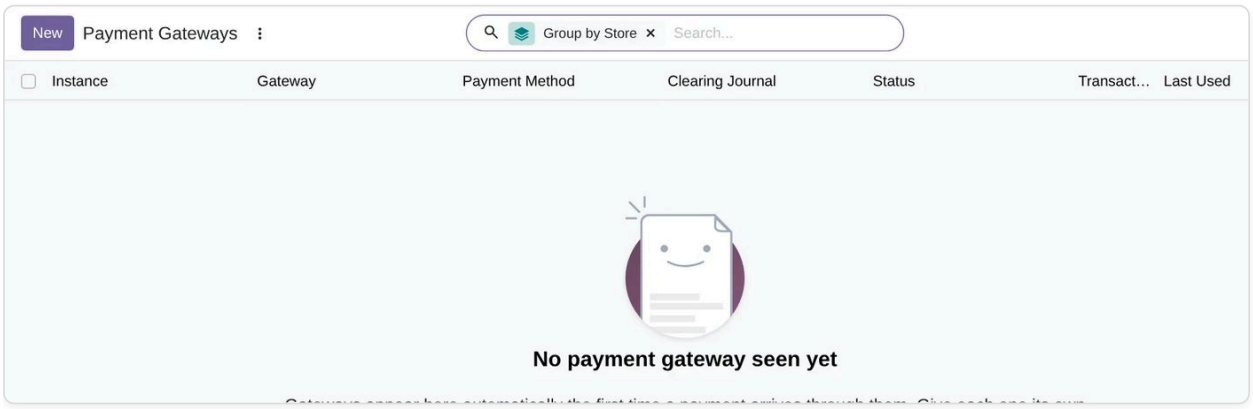
Shopify	Group with the **Instance**, the **Shopify Refund**, the **Shopify Order** and the **Sale Order**.
Amounts	Group with **Amount Refunded**, **Amount Shipping Refunded** and **Currency**.
Note	Group with the text that accompanies the refund in Shopify.
Raw Payload	Tab with the data exactly as it arrived from Shopify.

The list can be filtered by **Draft**, **Credit Note Created**, **Failed**, **Ignored**, **Needs Action**, **Stock Pending**, **Stock To Review** and **Stock Returned**. It can be grouped by instance, status or month.

An end-to-end example

The store Tazas Dani sells order #1042 to Irene Molina Pastor. A refund for that order appears in **Shopify › Operations › Refunds & Returns** in **Draft**. You open the refund and check the order with **Open Sale Order**. You click **Create Credit Note** and the status changes to **Credit Note Created**. If the **Stock Return** is in **Pending Return**, you click **Return Stock** and it ends up in **Stock Returned**.

12 What Shopify pays you



- **Real Odoo 20 screenshot.** The payment gateways list empty, before the first payment arrives.

Where each piece of data comes from

- Payment transactions are stored with each order that is synced. You see them in **Shopify › Finance › Payment Transactions**.
- Payouts, balance movements and disputes are pulled in by the **Shopify Sync Payouts** task, every 6 hours, which is switched on when you tick **Enable Finance & Accounting**, or immediately with the **Sync payouts** button on the store's **Finance & Accounting** tab. They are consulted in **Shopify › Finance › Payouts**, **Shopify › Finance › Balance Transactions** and **Shopify › Finance › Disputes**.
- The cards in **Shopify › Finance › Control Tower** are recalculated by the **Shopify Refresh Finance Dashboard** task, every 4 hours, which is also switched on with **Enable Finance & Accounting**. The **Refresh** button on each card does it immediately.
- Each gateway appears automatically in **Shopify › Configuration › Mappings › Payment Gateways** the first time a payment comes in through it.

Record buttons

Book & Reconcile Settlement	On a payout. Generates the settlement entry and reconciles the pending collection with the payments for each order. Only visible if the payout is paid and does not yet have an entry.
View Transactions	On a gateway. Opens the transactions that have gone through it.
Refresh	On the Control Tower card. Recalculates the figures for that store.

Statuses

Gateway: Mapped or **Unmapped**	It stays **Unmapped** until you give it a **Clearing Journal**.
Payout: Unmatched, **Booked** or **Reconciled**	Shows how far the payout has got in accounting.

How to reconcile a payout

1 Switch on finance

On the store's **Finance & Accounting** tab, tick **Enable Finance & Accounting** and press **Set up finance accounts**, which prepares the Shopify Payments clearing journal.

2. **Assign a journal to each gateway**
   In **Shopify › Configuration › Mappings › Payment Gateways**, choose the **Clearing Journal** for the **Unmapped** rows. Each journal has its own clearing account.

3. **Bring in the payouts**
   Press **Sync payouts** on the store or wait for the **Shopify Sync Payouts** task. The payouts appear in **Shopify › Finance › Payouts**.

4. **Open a paid payout**
   Press **Book & Reconcile Settlement**. The entry is posted and the clearing balance is reconciled.

5. **Check the result**
   The payout moves to **Reconciled**, and **Reconcile Diff** shows what is still unreconciled.

> This section does not charge or pay anything on its own: it does not move money and does not create entries until you press **Book & Reconcile Settlement**. It only covers Shopify Payments, because it is the only gateway that provides fees and payouts.

Gateway statuses

Unmapped	The gateway has been seen in a payment but has no clearing journal. It is created this way on purpose so it stands out and is not silently mixed into the Shopify Payments account by default.
Mapped	It has a journal. Its payments go to that journal.

If you leave the journal empty, the store's default clearing journal is used. When you assign or change the journal, the module gives it its own reconcilable clearing account. Two gateways whose journals share an account would defeat the point of separating them, because reconciliation would see entries from both on the same account.

Payout statuses

Unmatched	Initial status. The payout is imported and has no entry.
Booked	The settlement entry already exists.
Reconciled	The payout's clearing balance has been reconciled with the payments of each order.

The Shopify status of the payout is one of scheduled, in_transit, paid, failed or canceled. Only a paid payout can be booked.

Rules and conditions

- **Book & Reconcile Settlement** is hidden if the payout is not paid or if it already has a **Settlement Entry**.
- The entry debits the bank for the net amount and the fees for their amount, and credits the clearing balance for the gross amount. It is posted immediately.

- Pressing it again does not duplicate the entry.
- Each payout should balance: **Bt Diff** is the sum of the net of its balance transactions minus the payout's net and should be close to zero.
- **Reconcile Diff** is what remains unreconciled in the clearing account for that payout's orders.
- Scheduled tasks can only be run by an administrator. If you are not one, Odoo says "Only an administrator can run this scheduled action."

Record fields and tabs

Payout: Payout and Amounts	Payout details and its amounts: **Amount Net**, **Total Fees** and **Gross**, which is the net plus the fees.
Payout: Balance Transactions	Tab with the balance transactions that make up the payout.
Payment transaction: Transaction and Amounts & Fees	Type and status of the transaction, amount, currency and fee. Fees only come from Shopify Payments.
Control Tower: Payouts & Fees and Invoicing & Risk	Store figures: paid, fees, fee percentage, payouts to reconcile, open disputes, invoices and refunds.

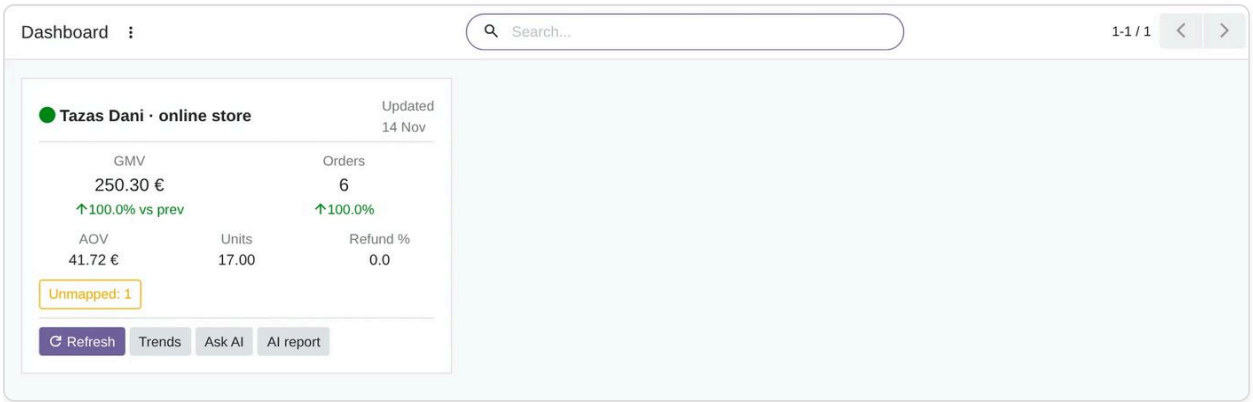
Transactions are filtered by **Shopify Payments**, **With Fees**, **Sales/Captures** and **Refunds**, and grouped by **Gateway**, **Kind**, **Store** or **Processed Month**.

An end-to-end example

At the Tazas Dani store, payments come in through Shopify Payments. The gateway appears in **Shopify › Configuration › Mappings › Payment Gateways** as **Unmapped**. You give it its **Clearing Journal** and it becomes **Mapped**. With finance switched on, the next paid payout arrives in **Shopify › Finance › Payouts** with its **Balance Transactions**. You open it and press **Book & Reconcile Settlement**. The entry is posted and the payout moves to **Reconciled**.

CONNECTOR GUIDE · SHOPIFY

13 Dashboard, reports and AI



- **Real Odoo 20 screenshot.** The dashboard with the indicators for the Tazas Dani store and the Trends, Ask AI and AI report buttons.

You enter through **Shopify › Dashboard**. There is one dashboard per store, and they are created automatically so the screen is never empty. The indicators are recalculated by the scheduled task **Shopify Refresh KPI**

Dashboard, which runs every 4 hours, or with the **Refresh** button. The **Shopify AI Weekly Digest** is generated every week. **AI report** generates one immediately.

Refresh	Recalculates the indicators for that dashboard.
AI report	Generates an AI report for this store and places it on the **AI Summary** tab.
Ask AI	Opens the question assistant already limited to this store.
Orders	Opens the store's orders.
Sales Analytics	Opens sales analytics.
Trends	Opens the evolution of the indicators.
Top products	Opens the best-selling products.
Fulfilment Funnel	Opens the order fulfilment funnel.
Reorder list	Opens the list of products to reorder.
Refunds	Opens the refunds.

The dashboard has two groups of figures, sales and stock with its issues, and the tabs **Trend**, **Top products** and **AI Summary**. Under **Period** you choose between **Last 7 days**, **Last 30 days**, **Last 90 days**, **Month to date**, **Last month**, **Year to date** and **Custom**. The default is **Last 30 days**.

Each AI response is saved with one of these statuses: **OK**, **No key**, **Refused**, **Error**, **AI disabled**, **Cost cap** or **AI notice not accepted**.

1. **Open Ask AI from the dashboard**
   The store is already selected.
2. **Type the Question and press Ask AI**
   The **Answer** appears in its section, with the **Cost**.
3. **Press Recent questions**
   The archive of questions for that store opens.

> The AI only counts figures the connector has already calculated: it does not access other data in your store. If the monthly cost cap is reached, the assistant does not respond.

When each thing is updated

On installation, the connector creates a dashboard per store and fills in the history with the orders you already have, so the trends are useful from day one. Only the sales metrics are rebuilt. The stock metrics build up from then on. Each day keeps a single snapshot per store, and if two updates fall on the same day only one is kept.

What each field shows

GMV, **Order Count**, **Average order value**, **Units Sold**	The store's sales in the chosen **Period**.

Open Orders, **Refund Count**, **Refund rate (%)**	Orders still open and refunds in the period.
GMV vs prev (%), **Orders vs prev (%)**	Comparison with the previous period.
Tracked products, **Reorder now**, **Watch**, **Dead stock**	Stock status of the products.
Unmapped, **Failed syncs 24h**, **Orders without SO**	Open issues.
Last Refresh	When the dashboard was last recalculated.

AI response or report statuses

STATUS	WHAT IT MEANS
OK	The AI responded and the answer is saved.
No key	No AI key is configured.
Refused	The AI model rejected the request.
Error	The call failed. The reason is recorded in **Error Note**.
AI disabled	AI is switched off.
Cost cap	The monthly cost limit has been reached.
Missing AI notice acceptance	The AI notice has not been accepted yet.

Rules and conditions

- The weekly report can only be run by an administrator. If another user runs it, the message "Only an administrator can run this scheduled action." appears.
- Generating a report or answering a question does not fail: if something goes wrong, the result is stored with its status and a note.
- When you ask a question, **Remaining AI budget this month** and **Stock data as of** tell you how much budget is left and how old the stock data is.

AI analysis archive

Every report and every question is saved in **Shopify › Insights › AI Reports**, with **Scope Weekly report**, **Question** or **Manual**, the **Model Used**, the **Provider** and the **Cost**. The **Source data** tab stores the figures the AI relied on. Trends are in **Shopify › Reporting › KPI Trends**.

End-to-end example

On the Tazas Dani dashboard, with the default period, the dashboard shows the indicators **GMV**, **Order Count**, **Average order value**, **Units Sold** and **Refund rate (%)**, plus the **Unmapped** warning. You click **Ask AI**, type your question and read the answer. Then you click **AI report** and the report is stored in **AI Summary** and in **AI Reports**.

14 The data that flows in and out

DATA	DIRECTION	WHEN
Orders	From Shopify to Odoo	**Shopify Sync Orders**, every 5 minutes, or the **Sync Orders** button
Products	From Shopify to Odoo	**Shopify Sync Products**, every hour, or the **Sync Products** button
Stock	From Odoo to Shopify	**Shopify Push Stock**, every 20 minutes if the store has **Sync Stock Enabled**, or the **Push Stock** button
Stock	From Shopify to Odoo	Scheduled task every 6 hours, or the **Pull Stock** button, if the store has **Sync Stock Enabled**
Prices	From Odoo to Shopify	Scheduled task every hour, for prices changed in Odoo
Locations	From Shopify to Odoo	**Sync Locations** button
Carriers	From Shopify to Odoo	**Discover Carriers** button
Payouts, balance transactions and disputes	From Shopify to Odoo	**Shopify Sync Payouts**, every 6 hours with **Enable Finance & Accounting** switched on, or the **Sync payouts** button
Order line edits, cancellation and refund of an order	From Odoo to Shopify	**Push to Shopify**, **Cancel in Shopify** and **Refund in Shopify** buttons on the sales order
Fulfilment and tracking	From Odoo to Shopify	When the delivery is validated, if the store has it enabled, or with **Send Shopify Fulfillment** and **Update Shopify Tracking**
New product	From Odoo to Shopify	**Export to Shopify** button on the product

Each store connects through FlexigoTech Cloud. The client secret of your Shopify app is stored there encrypted, the platform obtains and renews access, and it passes each store call on to your Shopify Admin API. Odoo stores neither the secret nor the token.

The AI summary sends sales data to an external language model provider, and it is only sent if the administrator has accepted the AI notice in the settings.

15 What runs automatically

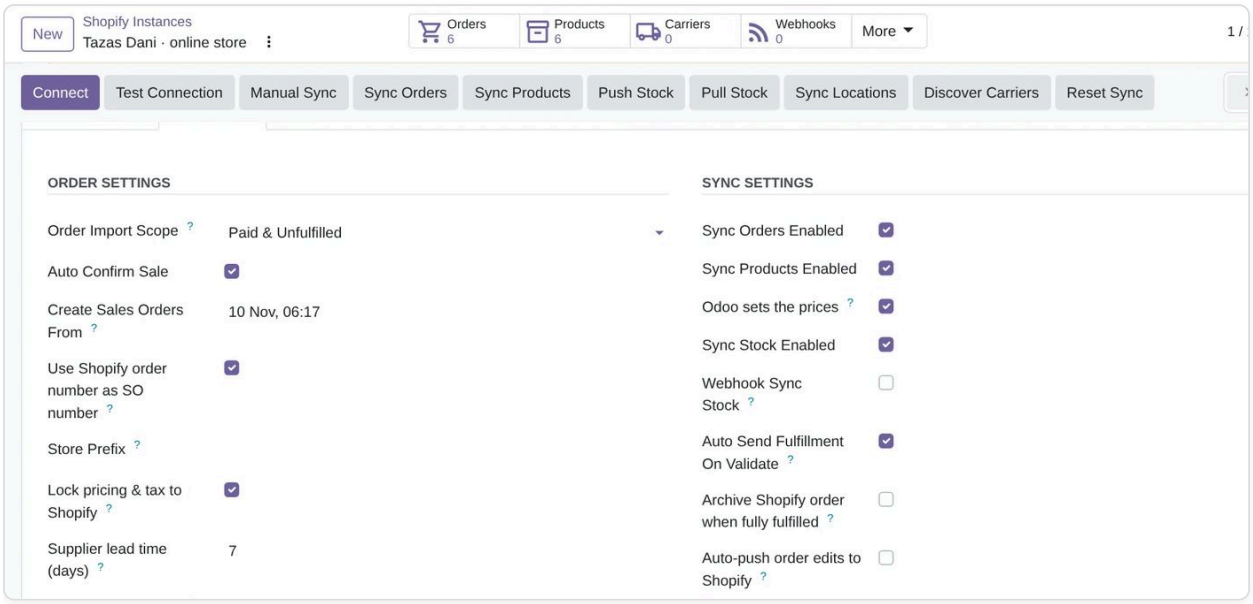
TASK	FREQUENCY	WHAT IT DOES
Shopify Sync Orders	Every 5 minutes (switches on automatically with the first active store)	Brings in new orders from the stores.

TASK	FREQUENCY	WHAT IT DOES
Shopify Sync Products	Every hour (switches on automatically with the first active store)	Brings in products from the stores.
Shopify Push Stock	Every 20 minutes	Sends stock from Odoo to Shopify, for the stores with **Sync Stock Enabled**.
Shopify Pull Stock (from Shopify to Odoo)	Every 6 hours	Adjusts Odoo stock to Shopify quantities at the assigned locations, on stores with **Sync Stock Enabled**.
Shopify Push Prices	Every hour	Sends prices changed in Odoo to Shopify.
Shopify Process Webhooks	Every minute	Processes received webhook notifications. A store connected through FlexigoTech Cloud does not receive webhooks.
Shopify Retry Failed Webhooks	Every 30 minutes	Reprocesses notifications that failed.
Shopify Notify Sync Errors	Every hour (switched off)	Warns about sync errors.
Shopify Cleanup Old Webhook Events	Every day	Deletes notifications already processed or ignored that are more than 30 days old.
Shopify Cleanup Old Sync Logs	Every 7 days	Keeps only the last 90 days of sync logs.
Shopify Refresh Product Analytics	Every day	Recalculates product analytics.
Shopify Refresh KPI Dashboard	Every 4 hours	Updates the dashboard indicators.
Shopify Refresh Instance Stats	Every hour	Resets the counters for each store.
Shopify AI Weekly Digest	Every week	Generates the weekly AI report, if AI is enabled in the settings.
Shopify Create Invoices	Every 2 hours (switches on with **Enable Finance & Accounting**)	Invoices confirmed orders from the stores with **Auto-create Invoices**, and validates the invoice if the order is paid in Shopify.
Shopify Create Credit Notes	Every 3 hours (switches on with **Enable Finance & Accounting**)	Creates the credit note for each refund in the stores with **Auto-create Credit Notes**: validates those for a full refund and leaves those for a partial refund in draft.
Shopify Sync Payouts	Every 6 hours (switches on with **Enable Finance & Accounting**)	Brings in payouts, balance movements and disputes from Shopify Payments.
Shopify Refresh Finance Dashboard	Every 4 hours (switches on with **Enable Finance & Accounting**)	Updates the finance dashboard.

Frequencies are changed in **Settings › Technical › Automation › Scheduled Actions**, with developer mode enabled.

> A store connected through FlexigoTech Cloud has no webhooks. Its orders and products arrive only through the sync tasks, which are switched off: they switch on automatically when you activate a store of that type with its switch turned on.

16 The store options and how they come by default



- **Real Odoo 20 screenshot.** The store opened on the **Operations** tab.

OPTION	WHEN CREATING IT	WHAT IT DOES
Order Import Scope	**Paid & Unfulfilled**	Decides which Shopify orders are imported as sales orders: **All orders**, **Paid only** or **Paid & Unfulfilled**.
Create Sales Orders From	The moment the store is created	Shopify orders dated before that point are imported but do not generate a sales order, so they are not sent again. **Create Sale Order** on one of them still creates it anyway. If empty, sales orders are created for all of them.
Auto Confirm Sale	On	The sales order is confirmed on import.
Use Shopify order number as SO number	On	The sales order carries the same number as the Shopify order.
Store Prefix	Empty	Identifies the source store on each imported sales order, for example ZA or EU.
Warehouse	Empty	The store's warehouse.
Sales Team	Empty	The store's sales team.
Fallback Customer	Empty	The store's fallback customer.
Default Pricelist	Empty	The pricelist applied to all sales orders imported from this store.

OPTION	WHEN CREATING IT	WHAT IT DOES
Default Fiscal Position	Empty	The fiscal position applied to imported orders.
Lock pricing & tax to Shopify	On	Line prices and taxes come only from the Shopify order: no Odoo pricelist or fiscal position is applied.
Odoo sets the prices	On	The Odoo sale price of a single-variant product is pushed to Shopify, if the store sells in the product's currency. Product sync only takes the Shopify price for a new product that arrives from Shopify.
Sync Stock Enabled	Off	While off, Odoo stock is not pushed to Shopify: the stock task skips the store.
Auto Send Fulfillment On Validate	On	When a delivery of a Shopify order is validated, the fulfilment is created in Shopify.
Notify Customer On Fulfillment	Off	Shopify notifies the customer when the order is fulfilled.
Auto-push cancellations to Shopify	Off	If you cancel a linked order in Odoo, it is also cancelled in Shopify.
Auto-push order edits to Shopify	Off	Line changes on a confirmed order are sent to Shopify.
Refund as store credit	Off	Refunds are issued as store credit instead of to the original payment method.

The customer notifications on cancelling, editing the order and refunding are on by default; so are restocking on cancel and refund on cancel. The invoicing and accounting options are explained in "Manual steps".

Manual steps

TAB OR GROUP	BUTTONS
Store header	**Connect** or **Connect with Shopify**, **Test Connection**, **Manual Sync**
Header, sync	**Sync Orders**, **Sync Products**, **Push Stock**, **Pull Stock**, **Sync Locations**, **Discover Carriers**, **Reset Sync**
FlexigoTech Cloud (tab Connection)	Check the saved app and delete it from FlexigoTech Cloud (administrators only)
Finance & Accounting	Prepare the finance accounts and sync payouts (only with finance enabled)
Carrier Mappings	**Discover Carriers from Orders & Profiles**, **Auto Create**, **Open**
Webhooks	**Regenerate Webhook Token**

Connect is hidden if the store uses an app distributed to that store. With a distributed app, the authorise-in-Shopify button is hidden if the store uses your own app. **Test Connection** works in both cases.

The **Webhooks** tab and **Register/Update Webhooks** do nothing on a store connected through FlexigoTech Cloud: that store receives orders and products only through the scheduled tasks, which are off by default and are switched on once the store is active and its sync is enabled.

Finance & Accounting

All finance options are off by default: enable finance, auto-create invoices, auto-register payments, auto-create credit notes, and auto-post and reconcile payouts. Until you switch on the first one, Odoo posts nothing. The journals and accounts you choose are the sales journal (by default, the company's sales journal), the Shopify Payments pending receivables journal (created when preparing the finance accounts), the bank journal for payouts, the processing fee account and the settlement journal. The date from which automatic accounting applies is filled in only for stores that predate an update: if empty, it covers everything.

Connection

The **Connection** tab stores the **Shop Domain**, the **Client ID** and the app secret, which is sent encrypted to FlexigoTech Cloud and is not stored in Odoo. The consent checkbox and the secret are visible only to an administrator. Two read-only fields show when the app was saved to FlexigoTech Cloud and when it was authorised. **Connection State** is **Unconfigured** when the store is created.

Advanced

Inside the store, **Page Size** defaults to 50 and **Max Pages** to 20. The page size must be between 1 and 100.

17 Security and personal data

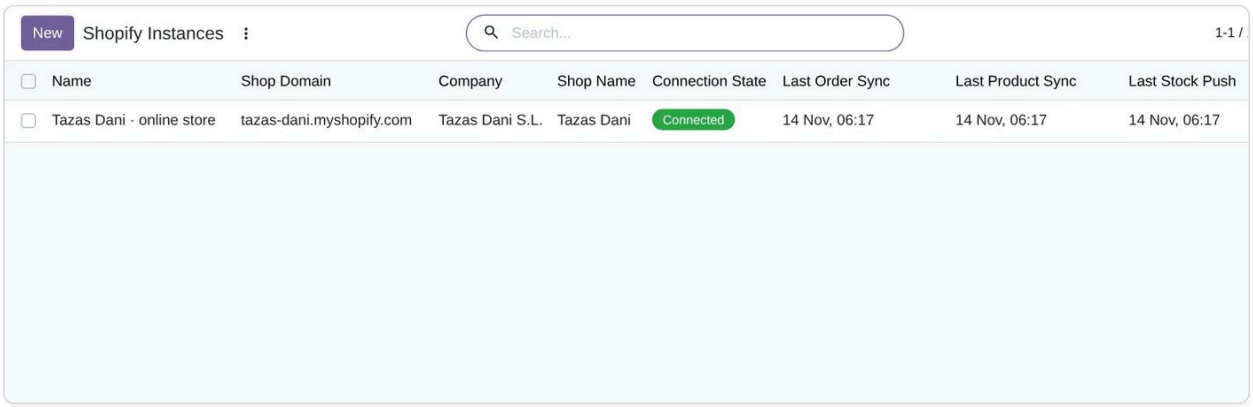
- When you connect a store, your Shopify app's client secret is sent once, encrypted, to FlexigoTech Cloud together with the client identifier and the store domain. Odoo stores neither the secret nor the access token.
- Only an Odoo administrator can write the client secret, and only after ticking the consent checkbox on the **Connection** tab. Without that checkbox, the secret is not saved and nothing is sent.
- On the same tab, an administrator sees the date the app was saved to FlexigoTech Cloud and the date it was authorised.
- The buttons on that tab that check the saved app and delete it from FlexigoTech Cloud are for administrators only. The check shows what FlexigoTech Cloud holds for the store, without the secret. The deletion removes the app and its tokens there.
- If you delete a store in Odoo, its app is also deleted from FlexigoTech Cloud.
- The connector has three user groups: **Viewer**, **Operator** and **Manager**. For example, **Create Sale Order** on a Shopify order is for **Manager** only, and **Cancel in Shopify** and **Refresh from Shopify** are for **Operator**.
- Each user works only with the stores of the companies they are working in. If they try to use a store from another company, Odoo replies "This Shopify store belongs to a company you are not working in."
- The store domain is validated before every call. The connector rejects internal, local or private-network addresses instead of treating them as a Shopify store.
- Your app's calls to Shopify and their responses pass through FlexigoTech Cloud: orders, products, stock, locations, shipments with their tracking, refunds, cancellations, carriers, payments and disputes. Its servers are in Germany. When the trial or subscription ends, or when the database is disconnected, FlexigoTech Cloud deletes your app's secret and that connection's data within 30 days at most.
- Activating FlexigoTech Cloud keeps a record of who accepts: the company name, that person's email, and the date and version of the accepted text.
- The AI summary sends sales data to an external language model provider, and only if an administrator has accepted the AI notice in the settings.

History and audit

The store and the delivery each have their history at the bottom, where the connector records what happens, such as "Not sent to Shopify" when FlexigoTech Cloud pauses a send. After saving, the connector rechecks the company of a store, an order or a pending event, so that a user from one company cannot move those records into another.

18 Several stores and several companies

Each store has its own **Company** field. Its orders, products, carriers and tasks are handled with that store's options. The **Shopify › Configuration › Stores** list shows each one with its domain, its company and the connection state, and you can group the list with **Group by Company**.



- **Real Odoo 20 screenshot.** The list of stores with their domain, company and connection state.

Each store uses its own app. A store's app is never used for another store, not even for another company in the same database. The same Shopify store cannot be connected to two companies at once: Odoo warns "The Shopify store … is already connected in another company: archive it there first."

Each FlexigoTech Cloud module supports a number of accounts depending on the plan: up to 3 with Standard and up to 10 with Pro.

If you move a store to another company, Odoo rechecks the permission on the moved store, so that a user from one company cannot hand stores over to another. If you change a store's domain or client identifier while the app is already saved, re-enter the client secret in the same save, because the saved app belongs to the previous store.

Products from another company that are still linked to a store are unlinked from it. If a link or an order remains, Odoo warns: the next product sync rebuilds the link, and an order remains a document of its company.

19 If something fails

The store does not call Shopify or shows "FlexigoTech Cloud asks this store to wait: try again later."	This is a FlexigoTech Cloud pause, for example for lack of trial or subscription. Nothing is marked as failed and nothing is disconnected or deleted. The scheduled tasks for that store leave everything pending and carry on when the service returns.
It shows "Save the client secret of … in FlexigoTech Cloud first." or "Fill in the shop domain and the client ID before saving the client secret."	Fill in the store domain and the Client ID in **Shopify › Configuration › Stores**, enter the Client Secret and save. An administrator must first accept the consent checkbox about FlexigoTech Cloud.

The app is one of the apps distributed to the store and will not connect: "Authorise ... in Shopify first (Connect with Shopify)."	Authorise the app in Shopify once, after saving the store, using the connect-with-Shopify button.
Shopify denies access to fulfilment orders	Add the read_merchant_managed_fulfillment_orders and write_merchant_managed_fulfillment_orders scopes to the app and authorise it again.
A Shopify order does not create a sales order, or a delivery order is not sent to Shopify	Open the order or the delivery order record and read the warning it shows. If Shopify rejects the fulfilment, the error gives the reason, for example that the fulfilment order is closed. Fix the cause and use the retry button on the delivery order.
You want to see what Shopify replied	Open **Shopify › Configuration › Stores** and check the sync logs: each run shows its status, how many records it processed and how many errors it had.

That pause message appears when FlexigoTech Cloud rejects a call. It is only a wait: the store is not disconnected and nothing is lost.

Other messages you may see

"Delete the app from FlexigoTech Cloud before choosing the other kind of app."	To switch between your own app and a distributed app, first delete the saved app in FlexigoTech Cloud.
"Your own app needs no authorisation: press Connect."	With your own app there is nothing to authorise: press **Connect**.
"Invalid OAuth state token."	The authorisation was not started from this store or has already been used. Press **Connect with Shopify** again.
"OAuth callback shop does not match this store."	Shopify returned a different store. Check the **Shop Domain** and repeat the authorisation.
"Set your shop domain (e.g. my-store.myshopify.com) first."	First enter the store's myshopify.com domain.
"Only a Shopify Manager can change ... of a Shopify order: they decide whether the connector creates (and ships) its sales order."	That change is only for users in the Shopify Manager group. Ask a manager.
"You need Shopify Operator (or Manager) access to push changes to Shopify."	Your user does not have Shopify Operator or Manager permission. Ask for it to be assigned to you.
"This Shopify store belongs to a company you are not working in."	Switch to the store's company in the Odoo company selector.

20 Other ways of doing it

METHOD	WHAT IT INVOLVES
By hand from the Shopify dashboard	Someone looks at the orders in Shopify and retypes them in Odoo. The same goes for stock, shipments and refunds.
Odoo alone	The Odoo 20 code, in Community and in Enterprise, includes no module with «shopify» in its name and no manifest that mentions it (searched on 14-11-2026).
Plans and price	It brings Shopify orders into Odoo and sends stock, shipments with their tracking and refunds back. You connect each store in **Shopify › Configuration › Stores** and it needs a FlexigoTech Cloud subscription.

The connector relies on a FlexigoTech Cloud subscription: there is a 14-day trial with no card, then a paid plan. If you already bring in your Shopify orders with another tool, **Create Sales Orders From** stops earlier ones from being sent again: the new store sets the moment you create it there, and earlier orders are imported as Shopify orders without creating a sale order.

21 Plans and pricing

€9

License per Odoo version.

+ FlexigoTech Cloud

STANDARD

€49/month

or €490 a year

- First response within 1 working day
- Up to 3 accounts

PRO

€98/month

or €980 a year

- First response within 4 working hours
- Up to 10 accounts
- Video call if needed

14-day trial, no card · Prices exclude VAT

The module license is per Odoo version. FlexigoTech Cloud is paid monthly or yearly, and the 14-day trial asks for no card. Standard allows up to 3 stores; Pro, up to 10.

22 Frequently asked questions

What does the connector do?

It brings Shopify orders and products into Odoo, and sends stock, prices and shipments with their tracking to Shopify. It also picks up refunds.

Does it work without FlexigoTech Cloud?

No. Without the trial or a FlexigoTech Cloud subscription, the connector does not talk to Shopify.

What does FlexigoTech Cloud do for my store?

It stores the client secret of your Shopify app encrypted, obtains and renews access, and passes the store's calls to Shopify.

How much does it cost?

The module license costs €9 per Odoo version. The FlexigoTech Cloud subscription is Standard, €49 a month or €490 a year, or Pro, €98 a month or €980 a year, excluding VAT.

Is there a free trial?

Yes, 14 days with no card.

What happens if I do not subscribe when the trial ends?

The service stops and nothing is charged. Your data stays in your Odoo.

Does it work on Odoo Online?

No. Odoo Online does not allow third-party modules to be installed. It works on Odoo.sh or on your own server.

Which Odoo versions does it work with?

It is published for Odoo 17, 18, 19 and 20.

Does it need Odoo Enterprise?

No. All its dependencies are Odoo Community apps (Sales, Inventory, Invoicing and Delivery), plus the FlexigoTech Cloud module: it works on Community and on Enterprise.

How many stores does each plan allow?

Each FlexigoTech Cloud module allows up to 3 accounts with Standard and up to 10 with Pro.

Can the same store be connected in two companies?

No. A Shopify store cannot be connected in two companies of the same database at the same time.

What do I need in Shopify before connecting?

A store with a myshopify.com-type domain and an app with its Client ID and its Client Secret. The connector does not create or configure that app.

Who can connect a store?

Only an Odoo administrator can tick the consent and type the Client Secret.

Does Odoo store my app's secret?

No. For a new store it is sent encrypted to FlexigoTech Cloud, and Odoo stores neither the secret nor the token.

How often do orders arrive?

Every 5 minutes, with the **Shopify Sync Orders** task, which switches itself on when you save your first store. To avoid waiting, click **Sync Orders** on the store.

How often do products arrive?

Hourly, with the **Shopify Sync Products** task, which also switches itself on with your first store.

How often does stock go out to Shopify?

Every 20 minutes, with the **Shopify Push Stock** task, as soon as you switch on **Sync Stock Enabled** on the store (it comes switched off). You also have the **Push Stock** button.

How often does stock come from Shopify into Odoo?

Every 6 hours, on stores with **Sync Stock Enabled**: a scheduled task adjusts Odoo stock to the Shopify quantities in the assigned locations. You can also use **Pull Stock**.

How often do prices go out?

Hourly, with the **Shopify Push Prices** task: it sends the prices changed in Odoo for single-variant products, on stores with **Odoo sets the prices** (switched on by default on new stores) that sell in the product's currency.

How do shipments reach Shopify?

When you validate the delivery, with **Auto Send Fulfillment On Validate**, which is switched on by default on new stores, or with **Send Shopify Fulfillment** on the delivery order. New tracking on a delivery that has already been sent is also sent.

What if Shopify rejects a shipment?

The error is kept on the delivery order and you repeat the send with **Retry send fulfillment**.

Do refunds create credit notes automatically?

They arrive in **Draft** status. If you switch on **Enable Finance & Accounting** and **Auto-create Credit Notes** on the store, a task turns them into credit notes every 3 hours. You can also click **Create Credit Note**.

Does it post anything on its own?

Not unless you switch on the finance options, which all come switched off. Payouts are posted when you click the button to post and reconcile the settlement on the paid payout.

What happens to orders that were already in Shopify before you installed it?

They are imported, but they do not create a sale order if they predate the store's start date, which is the moment the store is created. With that field empty, they are created for all.

Which orders are imported as sale orders?

The ones set by **Order Import Scope**: **All orders**, **Paid only** or **Paid & Unfulfilled**. When you create the store it defaults to **Paid & Unfulfilled**.

Are sale orders confirmed automatically?

Yes, with **Auto Confirm Sale**, which is switched on by default: the sale order is confirmed when it is imported. One created by hand with **Create Sale Order** always stays as a quotation and the connector never confirms it.

Does it work alongside another Shopify connector?

Not alongside the Shopify Connector (flexigobe_shopify_connector): you need to uninstall that one first.

Does the store receive real-time notifications?

No. A store connected through FlexigoTech Cloud receives orders and products only through the scheduled tasks.

Where can I see the stores I have connected?

In **Shopify › Configuration › Stores**, with their domain, their company and the connection status.

What happens if the FlexigoTech Cloud service pauses a store?

It is only a wait: nothing is marked as failed, the store is not disconnected and nothing is lost. The tasks leave everything pending and carry on when the service comes back.

And on a test or pre-production database?

Test and pre-production databases do not pay. On a neutralised copy or one with demo data, a trial cannot be started.

What does the AI do and what data does it send?

It comes switched off. If an administrator switches it on in **Settings › Shopify Seller** with **Enable AI Insights**, accepts its notice, chooses the **AI provider** and enters their own **AI API key**, it generates a weekly summary, reports and answers about the sales figures the connector has already calculated. Those figures go to the provider you choose.

What support is included?

First response within 1 working day with Standard and within 4 working hours with Pro, which includes a video call if needed.

CONNECTOR GUIDE · SHOPIFY

23 Shop Domain

Store	Each connection with Shopify in Odoo. It is created and managed in **Shopify › Configuration › Stores**.
Store domain	The store address of the myshopify.com type. A custom store domain does not work for connecting.
FlexigoTech Cloud	The online service the connector needs. It stores the client secret of your Shopify app encrypted, obtains and renews access, and passes every store call on to Shopify.
Free trial	The 14 days without a card that you start with. Without the trial or the subscription, the connector does not talk to Shopify.

Client ID and Client Secret	The details of the Shopify app you need to connect. The connector does not create them: you get them from Shopify.
Webhook	A real-time notice that Shopify sends to Odoo. A store connected through FlexigoTech Cloud does not receive them: its orders and products arrive with the scheduled tasks.
Scheduled task	A process that Odoo runs on its own at a fixed frequency, such as **Shopify Sync Orders**. It is managed in **Scheduled Actions**.
Shopify order	The order exactly as it arrives from Shopify. You can see it in **Shopify Orders** and it links to its sales order.
Sales order	The Odoo order created from a Shopify order.
Fulfilment	The preparation and dispatch of an order in Shopify. It is sent from the delivery order with **Send Shopify Fulfillment**.
Delivery order	The Odoo delivery. The shipment, with its tracking number, goes from it to Shopify.
Product link	The link between a Shopify variant and an Odoo product. You can see it in **Linked Products**.
Shopify location	A warehouse of the store in Shopify. It is associated with an Odoo stock location in **Shopify Locations**.
Mapping	The assignment between a Shopify data item and an Odoo one, such as taxes, carriers and gateways. It is in **Shopify › Configuration › Mappings**.
Payment gateway	The means by which an order is paid in Shopify. Each gateway appears on its own the first time a payment comes through it.
Payout	The amount Shopify Payments pays into your bank. It is recorded from the reconcile button on the paid payout.
Credit note	The refund of an order's invoice, created from a Shopify refund.
Replenishment plan	The list that calculates, per product, how much stock is left, when it runs out and how much it is worth reordering.
AI summary	The text the AI generates from a store's already-calculated sales figures. It is only sent if an administrator has accepted the AI notice.

24 Checklist and contact

- [ ] Your Odoo is version 17, 18, 19 or 20, on Odoo.sh or on your own server. Odoo Online does not work, because it does not allow third-party modules.
- [ ] The Odoo Sales, Inventory, Invoicing and Delivery apps and the FlexigoTech Cloud module are installed: Odoo installs them with the connector.
- [ ] If you had the Shopify Connector (flexigobe_shopify_connector) installed, you have uninstalled it first.
- [ ] You have a Shopify app with its Client ID and Client Secret, and an administration account that can accept the permissions it asks for.
- [ ] You have accepted the FlexigoTech Cloud notice and started the 14-day trial without a card.
- [ ] You have created the store in **Shopify › Configuration › Stores** with the exact domain of type myshopify.com.
- [ ] You have pressed **Connect** (your own app) or **Connect with Shopify** (an app distributed to the store).
- [ ] You have reviewed the store options: **Sync Stock Enabled**, which comes switched off, and the finance ones, if you want them.
- [ ] You have run **Sync Orders** and **Sync Products** from the store.
- [ ] A test order appears in **Shopify Orders** with its sales order linked.

If you would like us to set it up with you, get in touch. With the Pro plan, if needed, we can go through it by video call.

Email

To install it, set it up or for any question.
comercial@flexigotech.com

On the web

The connector page, with the guide and the prices.
flexigotech.com/en/odoo-shopify-connector.html

On Odoo Apps

The module page on the Odoo store.
apps.odoo.com

Shopify is a trademark of its owner. FlexigoTech publishes this connector on its own, with no endorsement or sponsorship from the trademark owner. FlexigoTech is a brand of Flexibles y Accesorios Gobe, S.L. (Barcelona, Spain).