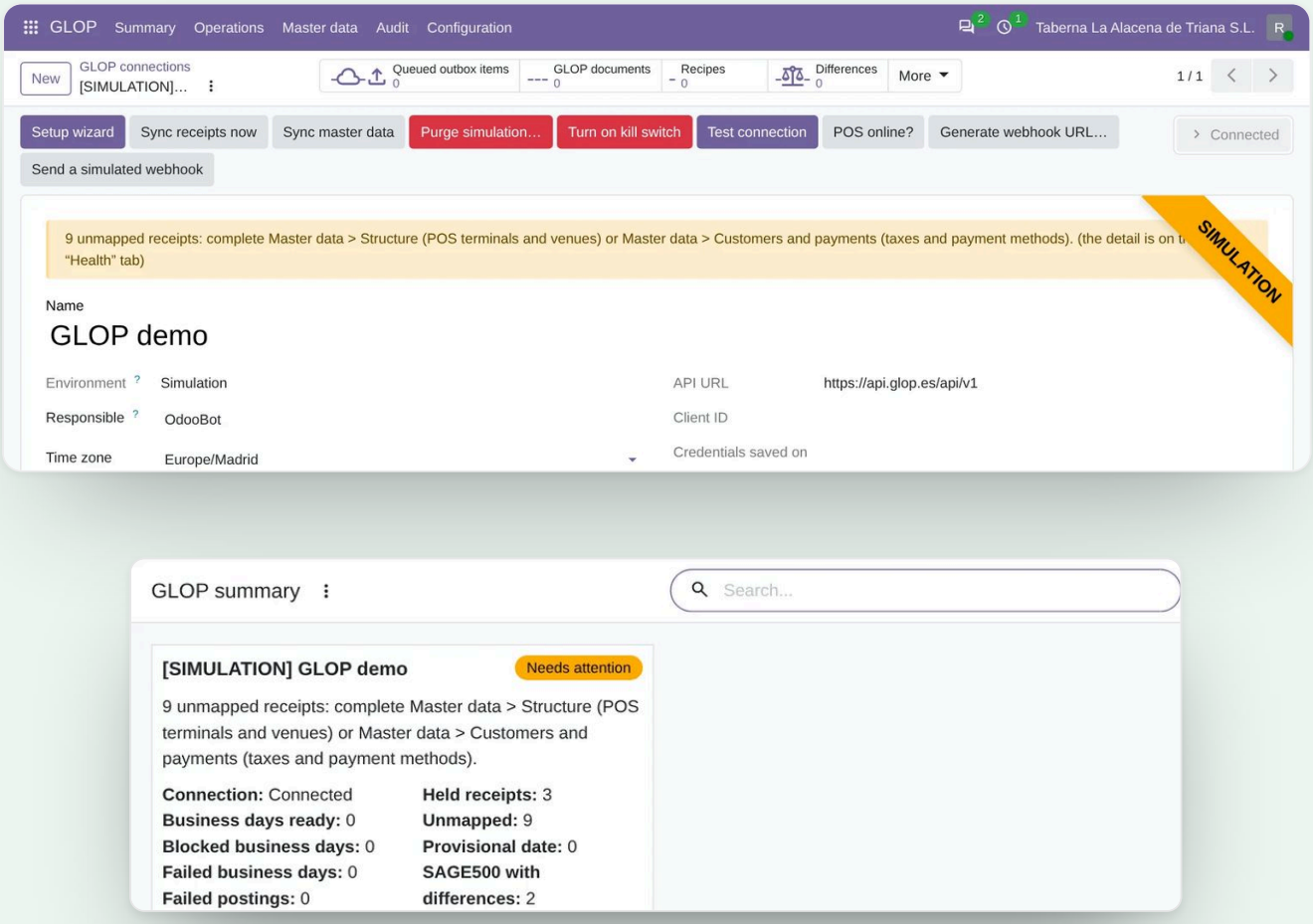
USER MANUAL · POINT OF SALE AND HOSPITALITY

GLOP point-of-sale connector for Odoo

For those who already have the connector: install it, activate it, connect it, set it up and use it, step by step.

Module version 20.0.1.2.1 | Odoo 20 screenshots | Edition of 20-11-2026



Real Odoo 20 screenshot. The [SIMULATION] GLOP demo connection with its access and warnings, and the summary card for that same connection.

Contents

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

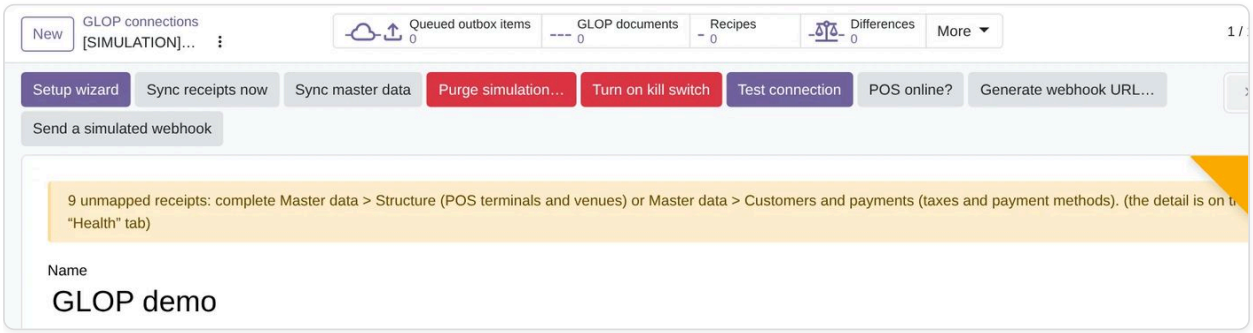
1	What it is and where to find it	3
2	Before you start	4
3	Installation	4
4	Activate FlexigoTech Cloud: the 14-day trial	6
5	Connecting your GLOP account	8
6	The connection options	10
7	Sales and posting	12
8	Payments and customers	15
9	Master data	17
10	Purchases and documents	19
11	Stock and recipes	21
12	Orders to GLOP	22
13	Reconciliation and checks	24
14	Setup and monitoring	26
15	What happens by itself	27
16	If something fails	28
17	Support	29

This manual describes version 20.0.1.2.1 of the module (Odoo branch 20.0, commit e888b59be). Everything it says comes from its code: menus, buttons, fields, scheduled actions and error messages.

The screenshots come from a test Odoo 20 database, with a made-up company and data and no real connection to GLOP. In Odoo 17, 18 and 19 the module has its own version: if something on your screen does not match, write to us.

1 What it is and where to find it

The connector sends GLOP sales to Odoo by business day for posting, and brings in and sends master data, purchases, stock and recipes. Everything is managed from its own **GLOP** app. The screens in this manual come from the module's simulation, with no real GLOP: that is why the connection is called "[SIMULATION] GLOP demo" and the data it brings in carries "[SIMULATED]".



- **Real Odoo 20 screenshot.** The [SIMULATION] GLOP demo connection in **GLOP connections**, with its buttons and counters.

GLOP › Summary	One card per connection with its status and anything that needs attention.
GLOP › Operations	**Day to day** with **Postings**, **Business days**, **Receipts** and **Receipt lines**. **Documents** with **GLOP tables**, **Orders to GLOP** and **GLOP documents**. **Outgoing to GLOP** with **Outbox items to GLOP**, **SAGE500 reconciliation** and **Send stock count**.
GLOP › Master data	**Structure**: **Venues**, **POS terminals** and **GLOP warehouses**. **Catalogue**: **Product families**, **Products**, **Pricelists**, **Sale formats**, **Recipes** and **Sizes and colours**. **Purchases**: **GLOP vendors**, **Purchase families** and **Purchase packages**. **Customers and payments**: **Taxes**, **Payment methods** and **Customers**.
GLOP › Audit	**Sync**: **Runs**, **Cursors** and **Received webhooks**. **Issues**: **API calls (errors)** and **Stock comparison**.
GLOP › Configuration	**GLOP connections**, **Settings** and **Create a simulation**.

Outside its own app, the connector adds its fields to Odoo's purchase orders, delivery orders, journal entries and journals, and to the general settings.

2 Before you start

Before installing the connector and creating the first connection, have the following ready.

- A GLOP account, with the Client ID and secret that GLOP gives you for its API.
- The full dump of your point of sale to the GLOP cloud. The GLOP cloud only receives what is created or changed once you switch it on: VAT types and items created earlier only arrive with that dump ("Envío de datos" › "Enviar todo", and "Enviar ventas" for the receipt history). GLOP support or your distributor does it, only during go-live and giving GLOP notice beforehand. It cannot be started from the API.
- A FlexigoTech Cloud subscription, or its 14-day trial with no card required. Without it the connector does not talk to GLOP.
- 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.
- An Odoo administrator user, who accepts the cloud consent and saves the credentials.

> The connector does not set up your GLOP account: it does not give you the Client ID or the secret and does not touch the point-of-sale configuration. It also does not register terminals or venues in GLOP: it reads them from GLOP. And it does not post on its own while **Post automatically** is off, which is the default.

3 Installation

The module is installed like any third-party Odoo module. An administrator is required.

How to get it

From its Odoo Apps page (**https://apps.odoo.com/apps/modules/20.0/flexigobe_glop_connector**) or from the connector's page at flexigotech.com. Choose the Odoo version you use: each version is a separate module.

The module depends on `flexigotech_cloud`, the connection to FlexigoTech Cloud. On Odoo Apps, the module page lists it under "Community Apps Dependencies" and its price is already included; it is also free and has its own page (**https://apps.odoo.com/apps/modules/20.0/flexigotech_cloud**). Put both folders together in the same modules path and Odoo installs `flexigotech_cloud` at the same time.

On Odoo.sh

1. **Upload the folders to your repository**
   Copy `flexigobe_glop_connector` and `flexigotech_cloud` into your project branch, in your modules folder, and push.
2. **Wait for Odoo.sh to build the branch**
   When it finishes, both modules are in the branch.
3. **Install the app**
   In **Apps**, click **Update Apps List** (with developer mode on), search for "GLOP" and click **Activate**.

On your own server

1. **Copy the folders**

   Place `flexigobe_glop_connector` and `flexigotech_cloud` in a folder that is in your Odoo `addons_path`.

2. **Restart Odoo**

   This makes it read the new modules.

3. **Install the app**

   In **Apps**, **Update Apps List**, search for "GLOP" and click **Activate**.

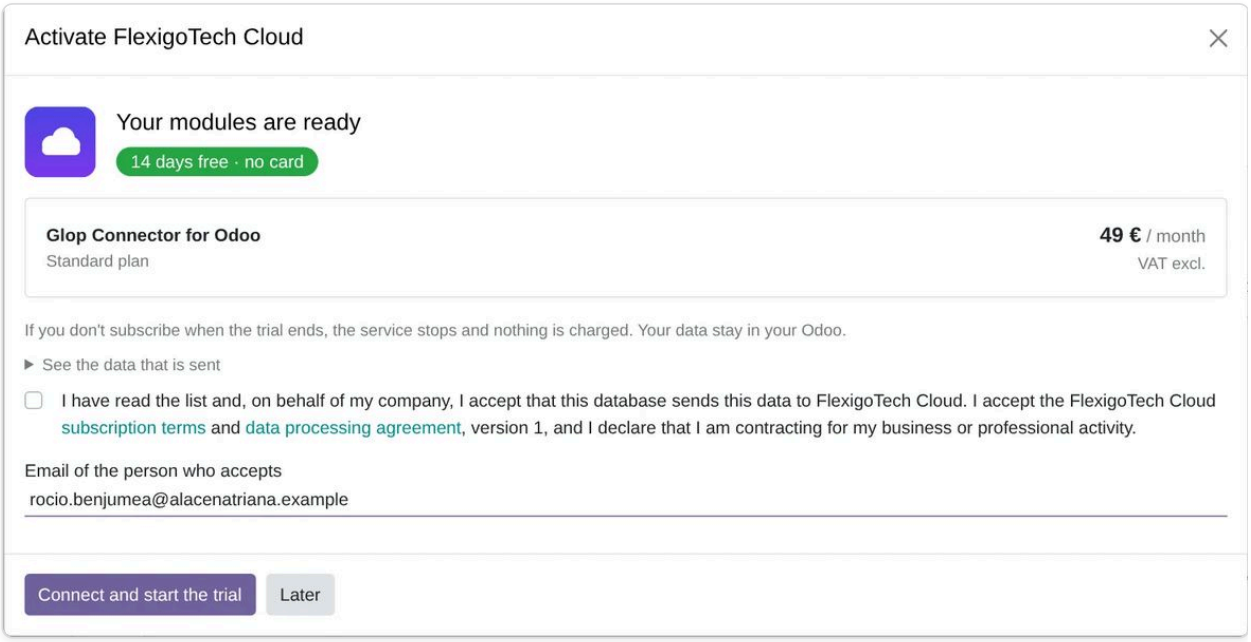
> When installed, Odoo also installs the apps the module depends on: Invoicing, Inventory and Purchase. If you already use them, nothing you have changes.

4 Activate FlexigoTech Cloud: the 14-day trial

Calls to GLOP go through FlexigoTech Cloud, FlexigoTech's service that stores your credentials encrypted. Until the database is connected and the module has a trial or subscription, the connector does not talk to GLOP.

1 **Click Activate in the top bar**

It appears as soon as you install the module. It opens the **Activate FlexigoTech Cloud** window, with the Standard plan and its 14 free days, 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**

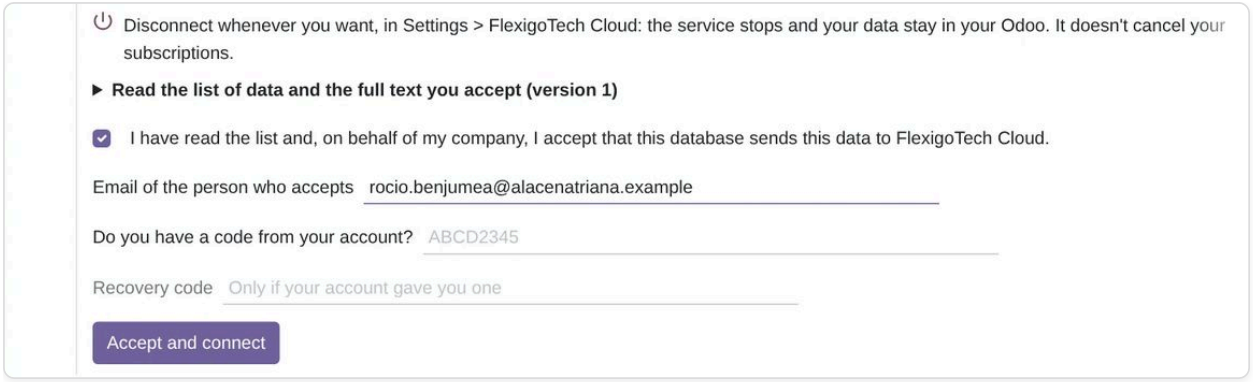
Tick "I have read the list and, on behalf of my company..." (it includes the subscription terms and the data processing agreement) and enter the **Email of the person who accepts**.

3 **Click Connect and start the trial**

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

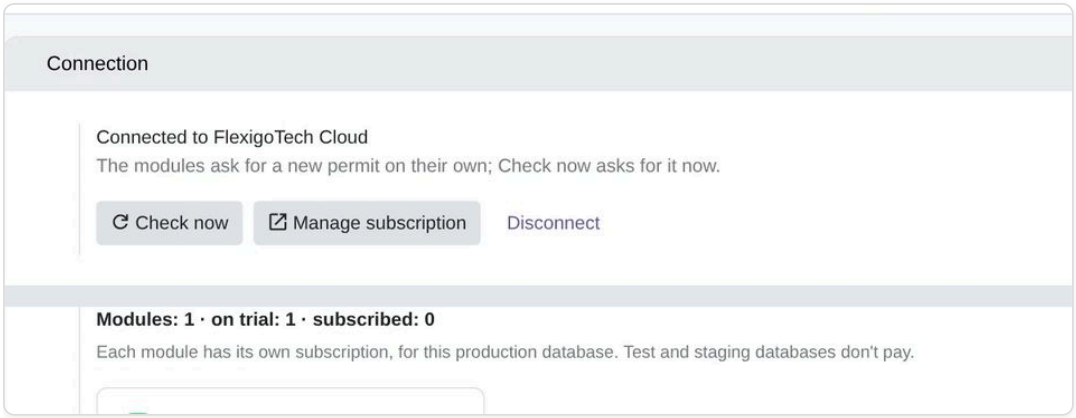
The same, from Settings

Settings › FlexigoTech Cloud has the full list of what the database sends. Tick the box, enter the email and click **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 ticked box, the email and **Accept and connect**.

With the database connected, the **Modules** block shows each module with its status: **On trial** and the date it ends, **Subscribed**, etc. From there you choose the plan (Standard or Pro, monthly or yearly) and click **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.

PLAN	PRICE	INCLUDES
Standard	€49 per month or €490 per year	Support response within 1 business day, up to 3 accounts
Pro	€98 per month or €980 per year	Response within 4 business hours, up to 10 accounts and video call if needed

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** asks for permission immediately (otherwise the module renews it by itself every 6 hours).

> On a neutralised copy (staging) or on 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."

5 Connecting your GLOP account

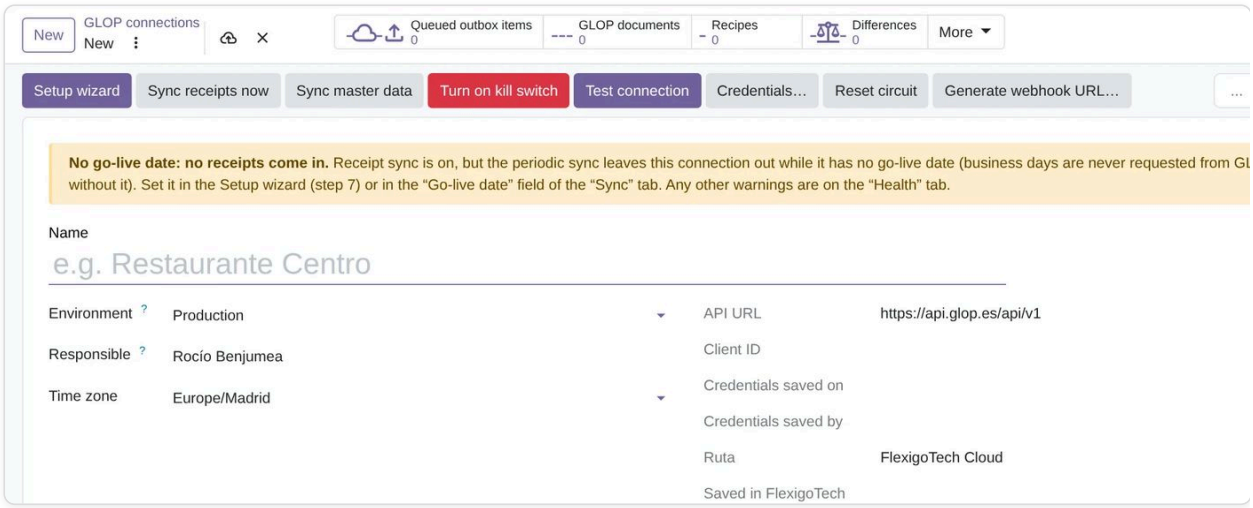
Each connection to GLOP is created from the **GLOP connections** menu, inside the GLOP configuration.

1 Open the connections menu

Go to **GLOP › Configuration › GLOP connections**. The connections that already exist are listed there.

2 Click New

The form for an empty connection opens. Until you save it, nothing is connected.



- **Real Odoo 20 screenshot.** A new connection, with the **Test connection** and **Reset circuit** buttons in the header.

3 Fill in the name and environment

Enter a **Name** and choose the **Environment**: **Production** or **Simulation**. The environment cannot be changed once the connection has venues or receipts.

4 Check the API address and time zone

The **API URL** is filled in with GLOP's. If there is no reason to change it, leave it as it is. Also choose the **Time zone**.

5 Click Credentials…

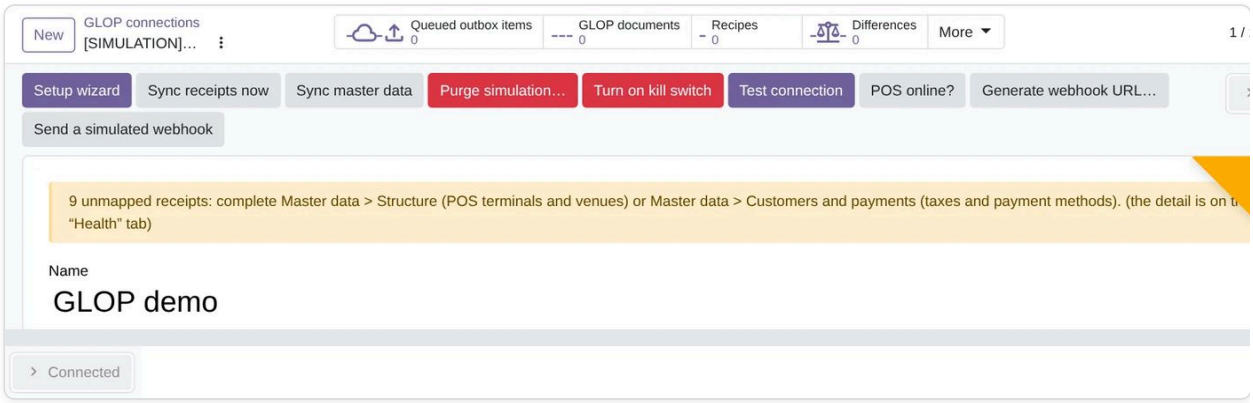
It is in the header and is the only place where the **Client ID** is entered or changed. On a simulation connection it does not appear.

6 Enter the credentials

Enter the **Client ID** and the **Client secret** that GLOP gave you. If the connection goes through FlexigoTech Cloud, also tick "I accept that the GLOP credentials and the calls to GLOP go through FlexigoTech Cloud": the Client ID and the secret travel encrypted once to FlexigoTech Cloud, which stores them and calls GLOP with them, and they are not kept in Odoo. An Odoo administrator saves them. Without the tick, you get "Before saving the secret, an administrator has to accept that the GLOP credentials and the calls to GLOP go through FlexigoTech Cloud. Nothing was sent." For security, the saved secret is never shown again.

7 Click Save and test the connection

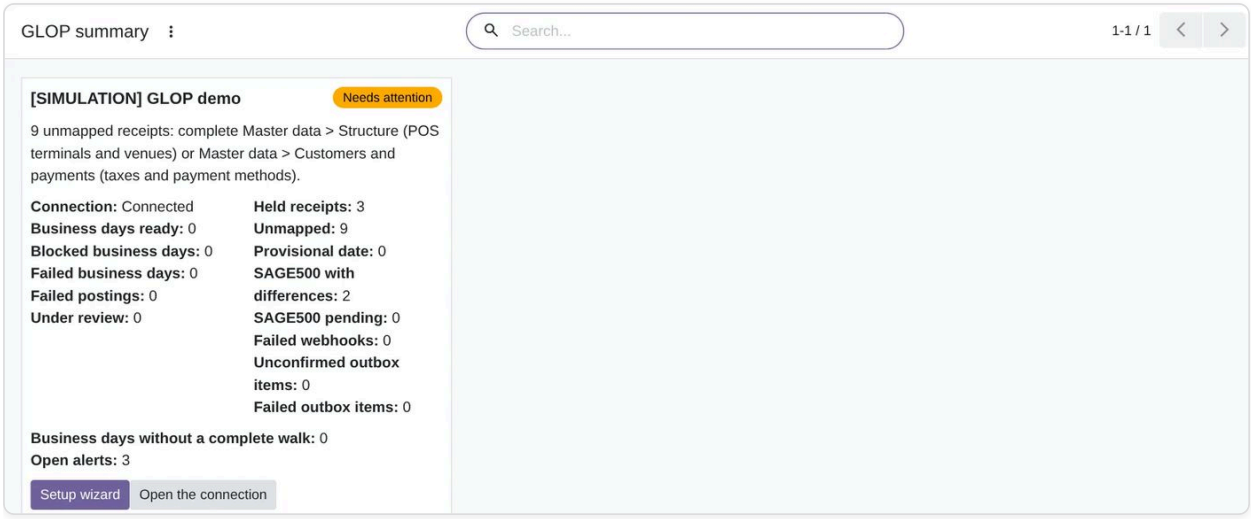
Odoo asks GLOP, through FlexigoTech Cloud, for the list of terminals. If all goes well, you get "Connection is fine: valid token and reachable point-of-sale terminals." with the terminals that GLOP sees, and the connection changes to **Connected**. If GLOP does not have any terminal yet, it warns you that there will be no receipts to import until the point of sale sends data to its cloud. You can repeat the test whenever you like with **Test connection**.



- **Real Odoo 20 screenshot.** The simulation connection in **Connected** status, with the **Test connection** and **Sync receipts now** buttons. In simulation, the test warns that it did not call api.glop.es.

> GLOP credentials are saved by an Odoo administrator, who is the one who accepts that they pass through FlexigoTech Cloud. GLOP administrator permission is enough to click **Test connection**, **Credentials...** and the sync buttons. **Check in FlexigoTech Cloud** and **Delete from FlexigoTech Cloud** are only visible to the Odoo administrator.

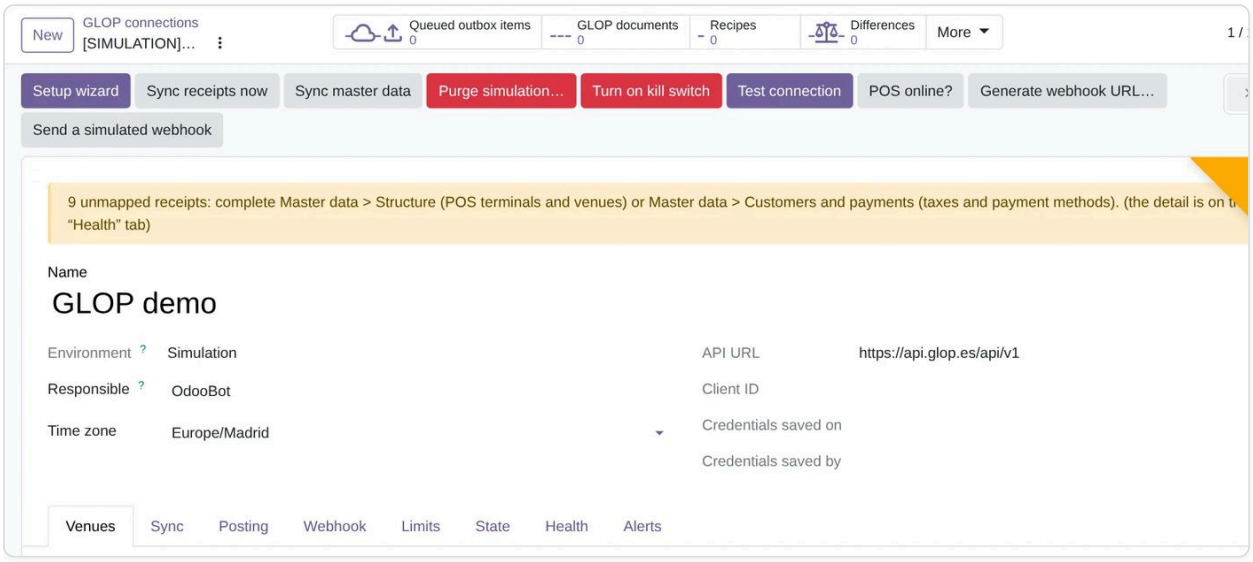
After that, the data arrives by itself. The scheduled task **GLOP: sync receipts by business day** reads the receipts every 10 minutes and **GLOP: master data** brings in terminals, taxes, payment methods and products once a day. You can also request it manually with **Sync receipts now** and **Sync master data**. If the connection has the kill switch on, **Sync master data** does not run. Receipts are only requested from the **Go-live date** onwards: without it none come in, and the connection warns you of this.



- **Real Odoo 20 screenshot.** The **GLOP summary** for the simulation, with one card per connection and its status.

6 The connection options

Each connection stores its own business options, which are chosen on its form.



- **Real Odoo 20 screenshot.** The simulation connection form, with its header buttons and tabs.

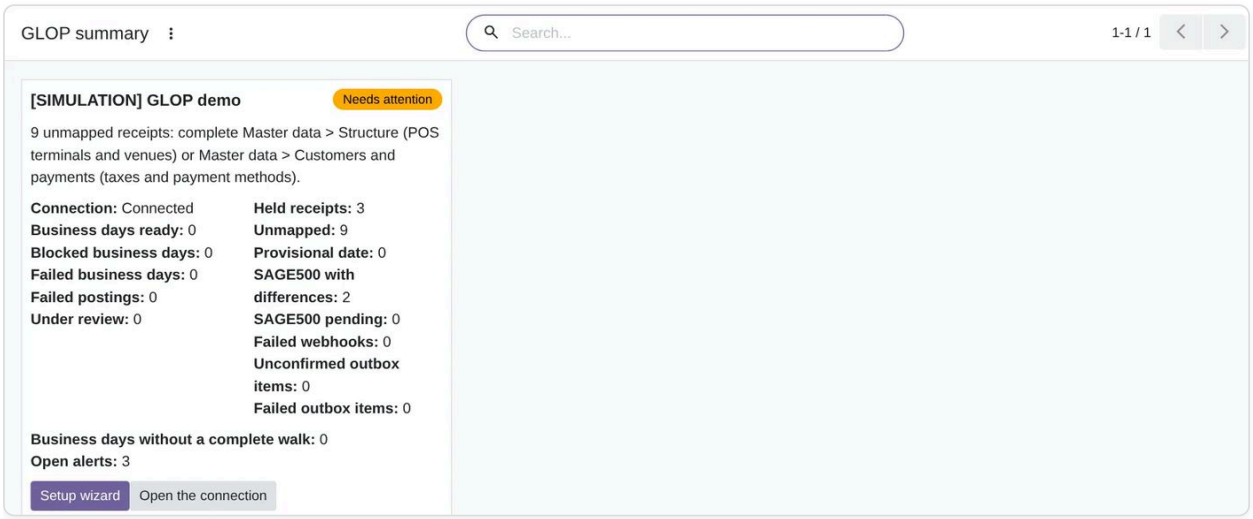
OPTION	WHEN CREATED	WHAT IT DOES
Environment	**Production**	Choose between **Production** and **Simulation**. It cannot be changed once the connection has venues or receipts.
Sync receipts	On	Allows reading receipts from GLOP.
Sync master data	On	Allows bringing structure, catalogue and tax data from GLOP.
Sync customers	On	Allows bringing customers from GLOP.
Sync vendors and purchases	Off	Reads from GLOP the vendors, purchase families, packages and pending purchase documents. It does not write anything to GLOP.
Sync sale formats and recipes	Off	Reads sale formats and recipes from GLOP. Writes nothing to GLOP.
Compare the stock with GLOP	Off	Reads stock levels from GLOP and sets them alongside those in Odoo. It only shows the difference and does not correct any stock.
Send customers to GLOP	Off	Lets Odoo create and update customers in GLOP.
Send purchase orders to GLOP	Off	Lets Odoo send purchase orders to GLOP.
Turn receipts of goods into GLOP delivery notes	Off	Lets a receipt that has already been validated be turned into a GLOP delivery note.
Send stock counts to GLOP	Off	Lets stock counts be sent to GLOP and stock documents be deleted.
Send recipes to GLOP	Off	Lets the recipe of a product be written to GLOP.
Send table and delivery orders	Off	Lets orders be sent to the point of sale, to a table or for delivery, and cancelled.

OPTION	WHEN CREATED	WHAT IT DOES
Go-live date	Empty	Date from which business days are requested. Business days before it are never requested.
Pass window (days)	3	Number of days looked back over.
Deep pass (days)	35	Number of days covered by the deep pass.
Hours until the business day is settled	30	Hours after which the business day is considered settled.
Maximum wait for open receipts (h)	48	Maximum number of hours an open receipt may wait.
Cut-off hour (provisional date)	6	Hour of the day used as the cut-off for the provisional date.
Post automatically	Off	With this off, each business day is posted when you press **Post**.
Full invoices	**Per document**	It can be changed to **Journal**.
Returns	**Correction labels**	The other option is **Net (check the 303 first)**.
Inconsistent VAT	**Hold**	The other option is **Recalculate**.
Extras	**Automatic**	You can choose **Add up to** or **Included**.
Credit note with positive amounts	**Hold**	The other option is **Invert sign**.
Editing after posting	**Review**	The other option is **Publish adjustment**.
SAGE500 check	**Warn**	You can set it to **Off** or **Block**.
Rounding tolerance	0.01	Rounding difference that is accepted.
Maximum rounding per journal entry	0.10	Maximum rounding accepted in one journal entry.
Stock	**No stock**	The other option is **Products sold**: Odoo moves the stock of what GLOP sells.
Owner of the ingredient stock	GLOP	Who deducts the ingredients of a compound product: the GLOP point of sale or the Odoo bills of materials.
Webhook active	Off	Enables the reception of notifications from GLOP.

The checkboxes for what is synchronised and sent, and the business day timings, are on the **Sync** tab; the documents, checks and stock are under **Posting**. **Environment** is at the top, next to the name, and **Webhook active** is on the **Webhook** tab. The checkboxes for sending to GLOP start off and are switched on one at a time. While one is off, the connector rejects that send before it leaves the network. These sends have only been tested with the module's simulation and with isolated calls to a test account in the GLOP cloud, not through FlexigoTech Cloud.

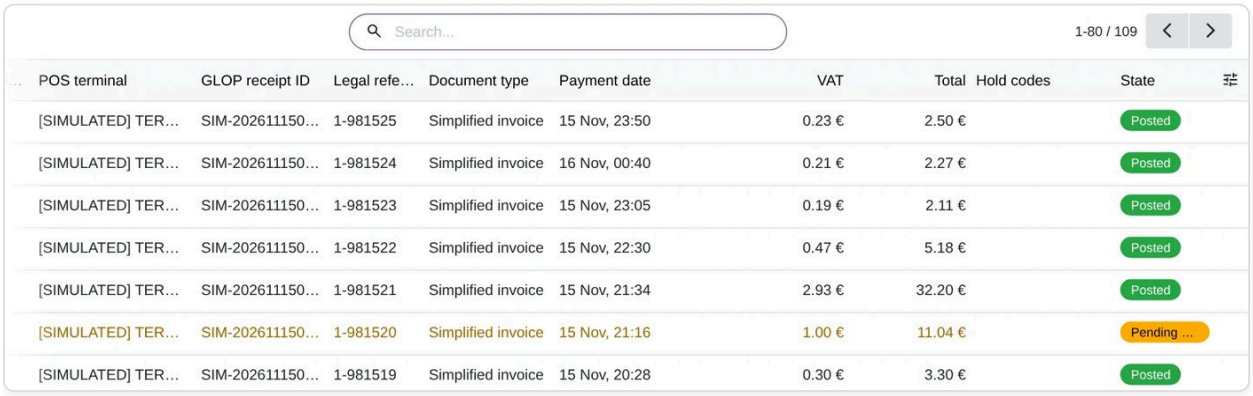
7 Sales and posting

GLOP receipts become one journal entry per terminal and business day. The business day and the posting each have their own buttons to review and correct it.



- **Real Odoo 20 screenshot.** The **GLOP summary** from the simulation, with the business days ready, blocked or in error.

Receipts arrive from GLOP with the scheduled action **GLOP: sync receipts by business day**, every 10 minutes. The action **GLOP: business days and posting**, every 15 minutes, evaluates each business day and leaves it ready to post. The receipts are in **GLOP › Operations › Day to day › Receipts** and the business days in **GLOP › Operations › Day to day › Business days**.

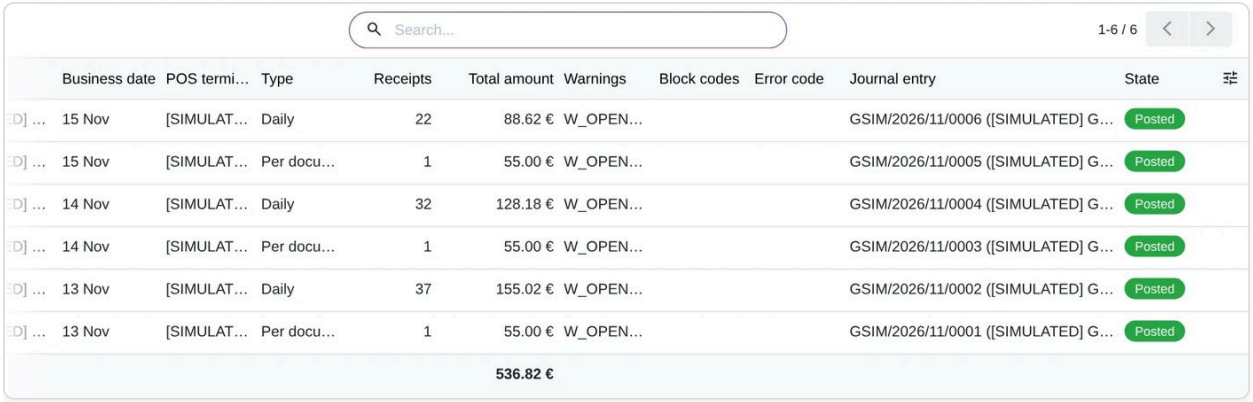


- **Real Odoo 20 screenshot.** The **Receipts** list from the simulation, showing the document type, payment date, VAT, total and status of each one.



- **Real Odoo 20 screenshot.** The simulation's **Business days**: three days in **Posted with held receipts**, with their receipts, the held ones and the difference.

Each business day for a terminal gives rise to one entry with the base and VAT as GLOP recorded them. The entries appear under **GLOP › Operations › Day to day › Postings**.



- **Real Odoo 20 screenshot.** The simulation's **Postings**, daily and per document, with their entry and status.

Business day buttons

Post	Posts the business day. Only shown if the day is **Ready to post** or **Blocked**.
Preview	Shows what would be posted without creating any entry. Only shown if there are receipts waiting to be posted.
Evaluate	Checks again now whether the business day is ready.
Resync business day…	Schedules a new read of the business day from GLOP.
Check SAGE500	Hidden while the day is still in **Collecting receipts**.
Check integrity	Only shown if the business day already has postings.

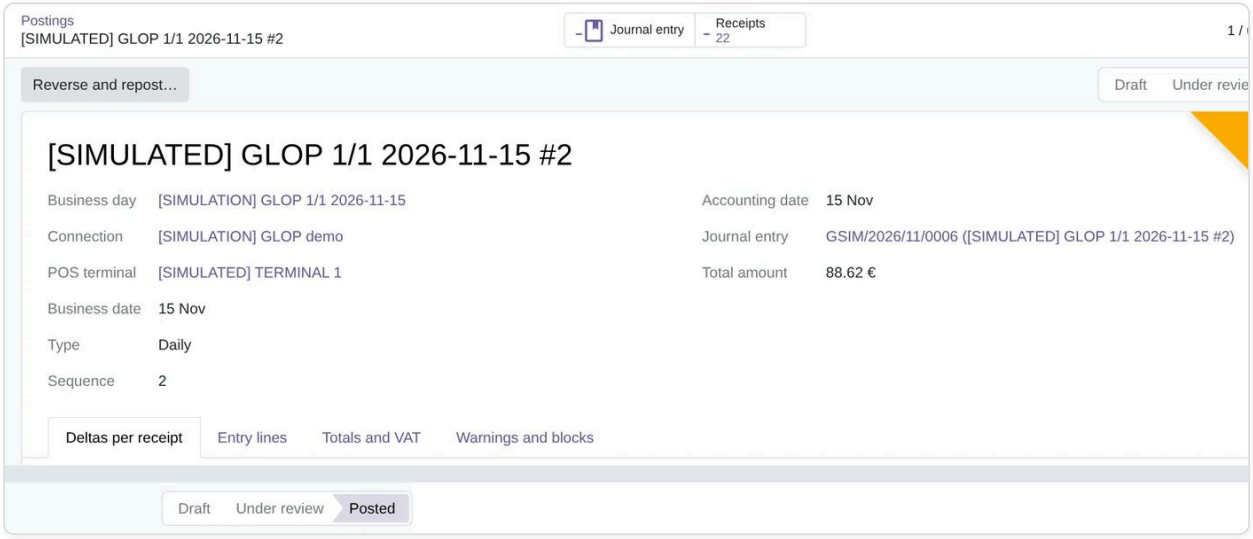


- **Real Odoo 20 screenshot.** A simulated business day in **Posted with held receipts**, with **Evaluate**, **Resync business day…**, **Check SAGE500** and **Check integrity**, and the warning that the held receipts will enter as an adjustment.

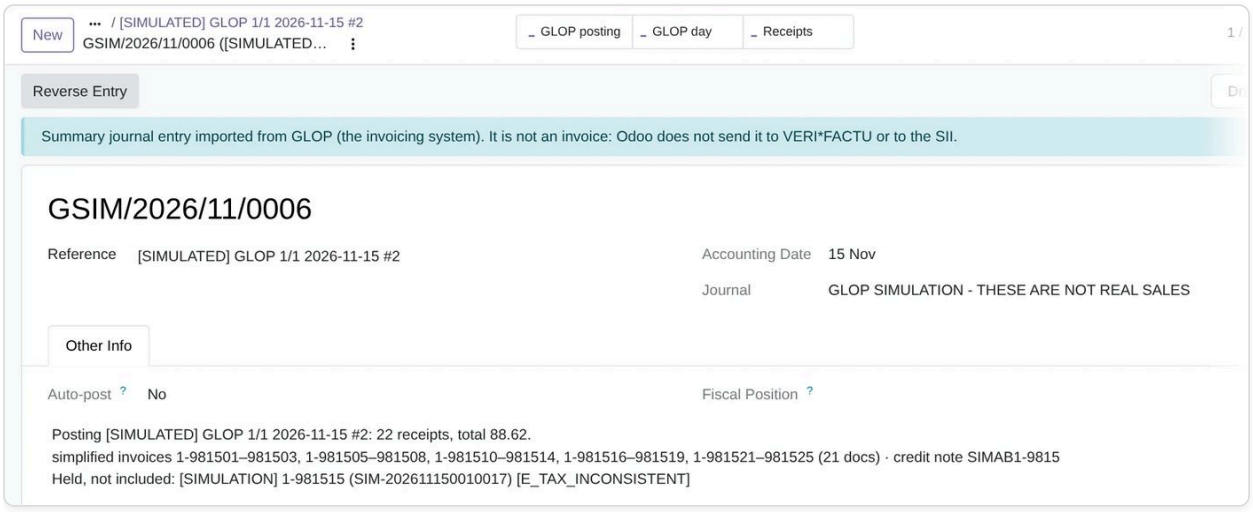
Posting buttons

Post	Posts a posting in **Draft** or **Blocked**.
Retry	Repeats a posting in **Error**.
Approve review	Approves a posting that is **Under review**.
Reject review…	Rejects the review and records the reason.
Post on another date…	For a posting that is **Blocked** by the closing of a period.
Preview	Shows the entry lines it would create.
Reverse and repost…	Reverses the journal entry of a posting that is **Posted**.

Journal entry	Opens the journal entry, if one exists.



- **Real Odoo 20 screenshot.** A daily posting from the simulation, **Posted**, with **Reverse and repost...** and the tabs **Deltas per receipt**, **Entry lines**, **Totals and VAT** and **Warnings and blocks**.



- **Real Odoo 20 screenshot.** Its journal entry, in the simulation journal, with the notice that it is not an invoice and that Odoo does not send it to VERI*FACTU or to the SII.

Day statuses

Collecting receipts, **Ready to post**, **Posted**, **Posted with held receipts**, **Blocked** and **Error**.

Posting statuses

Draft, **Under review**, **Posted**, **Reversed**, **Blocked** and **Error**.

1. **Open the business day**
   Go to **GLOP › Operations › Day to day › Business days** and choose the day from the terminal.

2. **Click Preview**
   Check what would be posted.

3. **Click Post**
   The posting is created with its journal entry.

4 **Correct if needed**

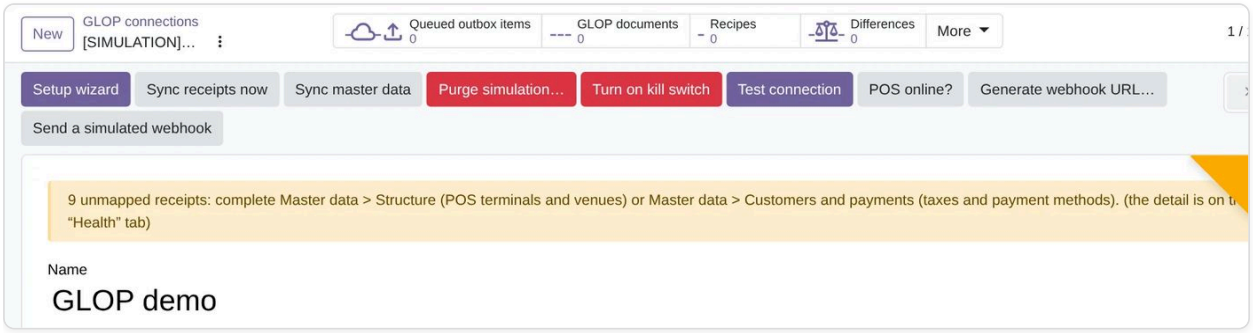
If something fails, use **Retry**, **Approve review** or **Reverse and repost...**.

> A posted journal entry cannot be edited. To change it, you must reverse it and post again. Versions of a receipt cannot be modified either.

USER MANUAL · GLOP

8 Payments and customers

Here you link each GLOP payment method, each tax and each GLOP customer with its account, its tax and its Odoo contact.

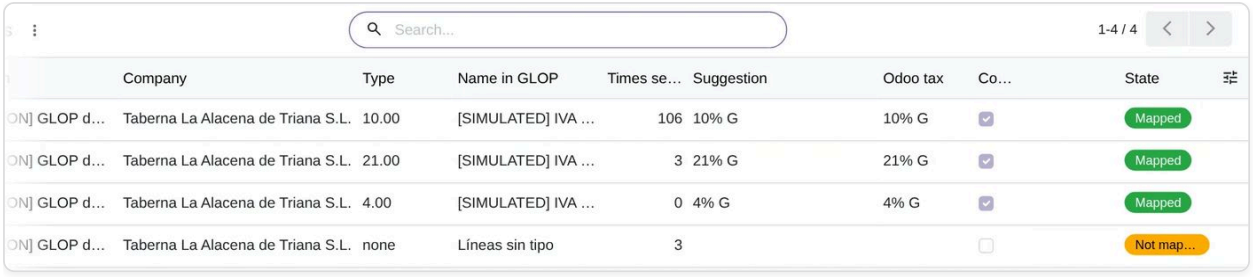


- **Real Odoo 20 screenshot.** The simulation connection, with the notice of 9 receipts not mapped that tells you to complete **Customers and payments**.

These lists are in **GLOP › Master data › Customers and payments**, with the menus **Taxes**, **Payment methods** and **Customers**. Tax and payment method rows are created automatically when GLOP sends receipts that use them. Customers are fetched by the scheduled task **GLOP: master data**, once a day. Customers are only sent to GLOP if you switch on **Send customers to GLOP** in the connection, which is off by default. Then, the one you send with **Send to GLOP** goes out with the task **GLOP: sending customers**, within 15 minutes at most. That sending has only been tested with the module's own simulation and with direct, standalone calls to a test account in the GLOP cloud with no point of sale, on 18 September 2026. It has not been tested through FlexigoTech Cloud or with a real point of sale.



- **Real Odoo 20 screenshot.** The **Payment methods** in the simulation: cash and card, **Mapped**; the other two, **Not mapped**.



- **Real Odoo 20 screenshot.** The **Taxes** in the simulation: three VAT rates **Mapped** and the lines without a rate, **Not mapped**.



- **Real Odoo 20 screenshot.** The GLOP **Customers** in the simulation with their tax ID; one already has an Odoo company and the **Send to GLOP** button.

Use suggestion	On a tax, copies into **Odoo tax** the one that **Suggestion** proposes. Only shown if there is a suggestion and you have not yet chosen a tax.
Confirm	On a tax, approves the chosen tax. Hidden if it is already confirmed or if you have not chosen one.
Send to GLOP	On a customer, sends its **Contact** details to GLOP. Only shown if the customer already has a company.

Statuses

Not mapped	For payment methods and taxes: still needs to be linked to Odoo.
Mapped	It already has its account or its tax.
Unconfirmed	For taxes: a tax is chosen, but nobody has confirmed it.
Not valid	For taxes: the chosen tax does not apply to that VAT rate.

Steps

1. **Open the list**

   Go to **Payment methods** or **Taxes** and filter by **Not mapped** or **Pending**.

2. **Fill in the row**

   On a payment method, choose **Type** and **Account**. On a tax, choose **Odoo tax**.

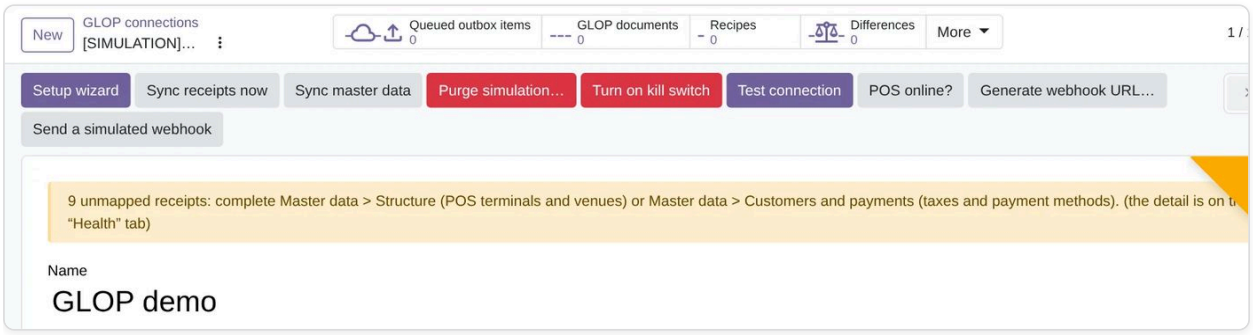
3. **Confirm**

   On a tax, click **Confirm**. The receipts that were waiting for this mapping are checked again.

> A tax cannot be confirmed without first choosing the Odoo tax: you get the message "Choose an Odoo tax before confirming GLOP rate ....". If you change the tax on a row, it stops being confirmed and you have to confirm it again.

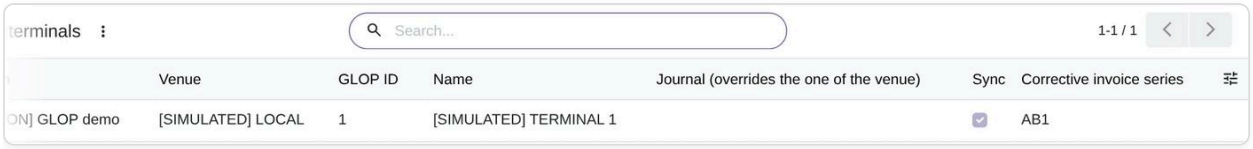
9 Master data

Master data brings in from GLOP the structure and the catalogue you need in order to post.



- **Real Odoo 20 screenshot.** The simulation connection, with the **Sync master data** button and the warning about unmapped receipts.

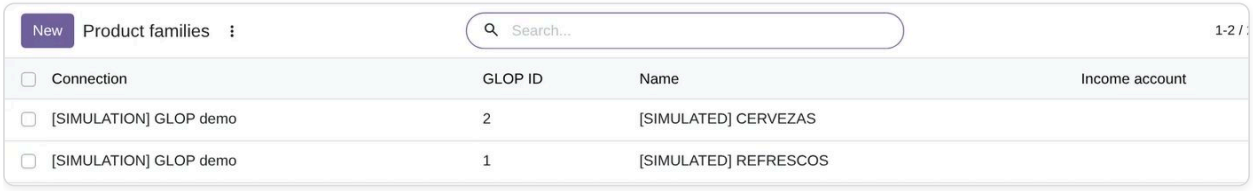
The scheduled action **GLOP: master data** runs every day and goes through all the active connections that have master data switched on. You can also run it by hand from the connection with **Sync master data**. It reads the venues, terminals, tables, warehouses, product families, products, price lists, sale formats, and sizes and colours. The data arrives in **GLOP › Master data › Structure** and **GLOP › Master data › Catalogue**.



- **Real Odoo 20 screenshot.** The simulation terminal, with its venue and its **Corrective invoice series**, AB1, which the connector works out by itself.



- **Real Odoo 20 screenshot.** The GLOP **Products** in the simulation, still **Not linked**, with the **Ignore** button.



- **Real Odoo 20 screenshot.** The GLOP **Product families** in the simulation, still without an **Income account**.

Sync master data	Runs the GLOP read now, without waiting for the daily action. When it finishes, it shows you what it has read, what came back empty and what needs attention.
Ignore	On a GLOP product, it leaves it out of the warnings. It shows while the product is not ignored.
Stop ignoring	Puts the product back on the normal list. It only shows if the product is ignored.

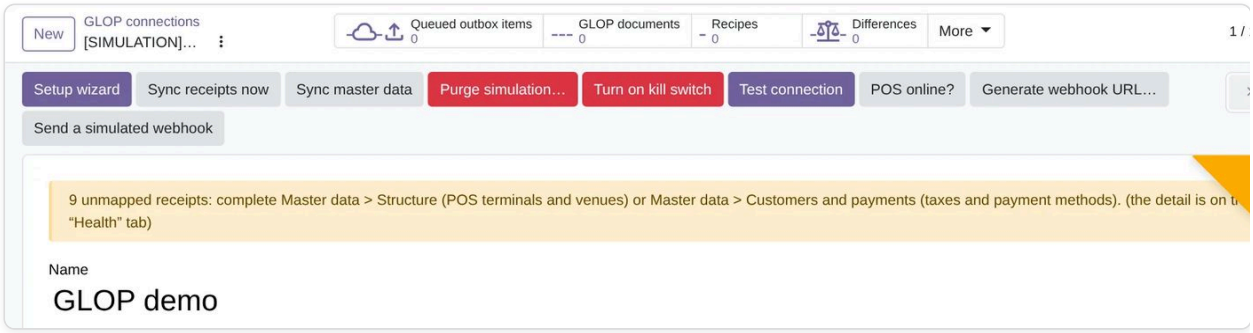
Use the suggestion	On a warehouse, a price list or a sale format, it copies the GLOP suggestion into your decision. It only shows if there is a suggestion and it does not yet match your choice.
Fetch the photo from GLOP	On a product, it downloads by hand the photo that GLOP has for it.
Set the photo on the product	Copies that photo onto the Odoo product. It only shows if a photo has been fetched and the product is linked.

Product statuses

- **Not linked**: GLOP has the product and Odoo does not yet know which product it corresponds to.
- **Linked**: the product is matched with an Odoo one, by **Barcode**, by **Internal reference** or by hand.
- **Ignored**: you have deliberately left it out.

1 Sync

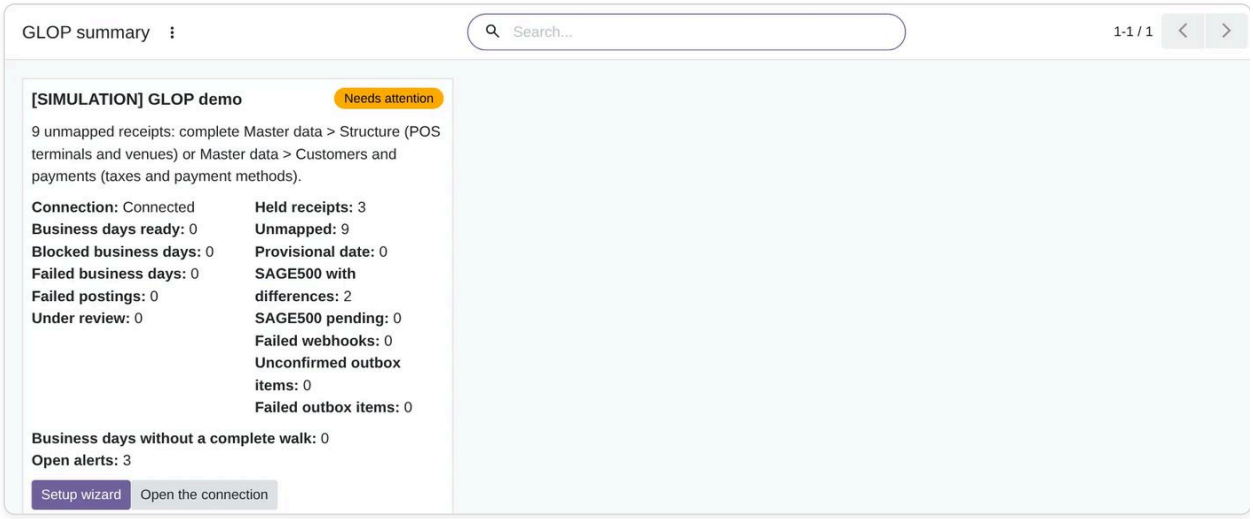
Click **Sync master data** on the connection.



- **Real Odoo 20 screenshot.** The **Sync master data** button on the simulation connection.

2 Check the products

Open **GLOP › Master data › Catalogue › Products** and link the ones that are **Not linked**.



- **Real Odoo 20 screenshot.** The **GLOP summary** in the simulation, with 9 unmapped receipts.

3 Complete the structure

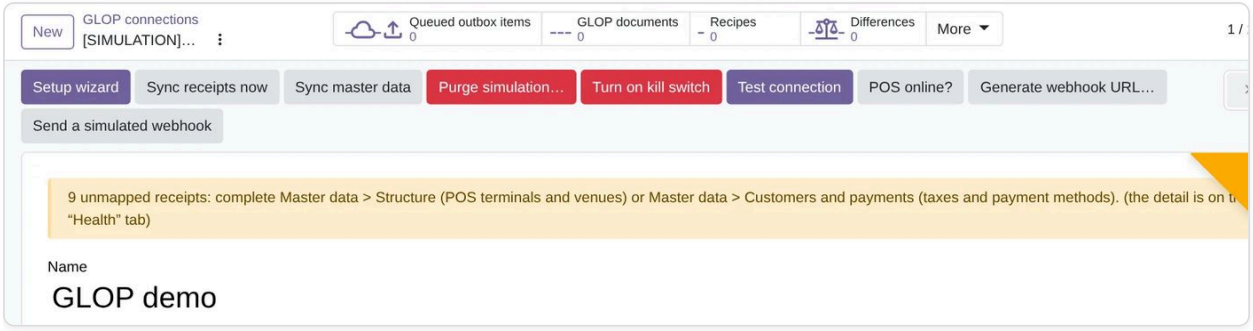
Go to **GLOP › Master data › Structure › Venues** and set the journal and the accounts for each venue.

The sync never creates products or contacts in Odoo, never assigns a warehouse or a price list by itself and never archives or deletes anything. New venues and terminals arrive with **Sync** unticked: nothing is posted until someone configures and activates them.

10 Purchases and documents

The connector links GLOP vendors with Odoo contacts, brings in the purchase orders and delivery notes pending delivery, and can send Odoo purchase orders and goods receipts to GLOP.

Everything in this section is switched off by default. **Sync vendors and purchases**, **Send purchase orders to GLOP**, **Turn receipts of goods into GLOP delivery notes** and **Send stock counts to GLOP** are switched on in the **Sync** tab of the connection. What Odoo writes to GLOP (purchase orders, conversions into delivery notes and the deletion of stock documents) has only been tested with the module's own simulation and with isolated, direct calls to a test account in the GLOP cloud with no point of sale, on 18 September 2026. It has not been tested through FlexigoTech Cloud or with a real point of sale.



- **Real Odoo 20 screenshot.** The simulation connection, with the **GLOP documents** counter at the top.

Where each piece of data comes from

- The purchase orders and delivery notes pending delivery are brought in by the scheduled action **GLOP: pending purchase documents**, every hour.
- Vendors and purchase families and containers are read with the scheduled action **GLOP: master data**, every day, or with the **Sync master data** button on the connection.
- You bring in a specific document by its number with the **Fetch a document from GLOP** wizard.

Record buttons

Fetch from GLOP (document)	Reads that document again and saves what GLOP holds now. It does not show if the document has no **Number in GLOP**.
Turn into a delivery note (document)	Queues the conversion of the order with the quantities of its lines. It only shows on a purchase document that is **In GLOP** and whose type in GLOP is 1. GLOP treats the order as fully converted, even if only part of it has been received, and does it later: its response only confirms that the order was sent to the point of sale.
Delete in GLOP (document)	Queues the deletion of a stock document. It only shows if it is **In GLOP**.

Purchase order and **Goods receipt** (document)	They open the linked Odoo purchase order or receipt. They do not show if there is none.
Send the order to GLOP (purchase order)	Queues the creation of the order in GLOP. Only products and quantities are sent: GLOP applies its own prices and taxes. It only shows with **Send purchase orders to GLOP** switched on and while the order has not been sent.
Fetch template from GLOP (GLOP vendor)	Reads the items that GLOP suggests ordering from that vendor.
Use the suggestion (purchase family or container)	Copy Odoo's suggestion into your decision. You can't tell if there is no suggestion or if it already matches it.

Document statuses

- **Draft**
- **Queued for GLOP**
- **In GLOP**
- **No longer in GLOP**
- **Error**

Linking a vendor

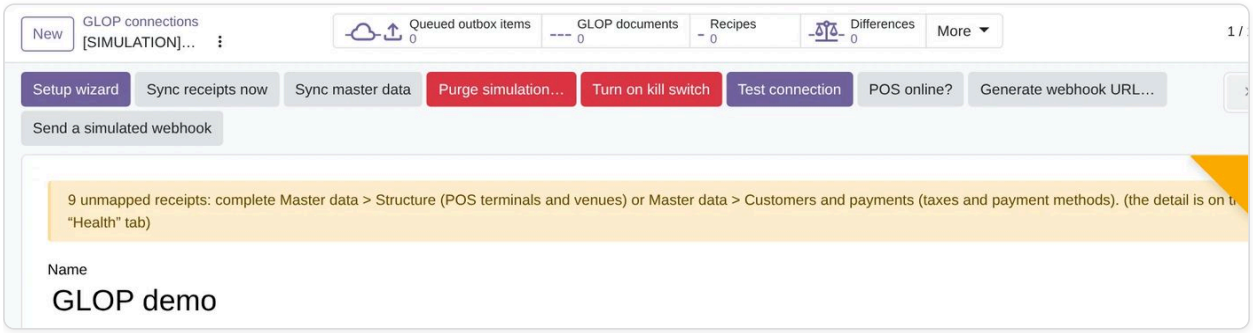
1. **Open the list**
   Go to **GLOP › Master data › Purchases › GLOP vendors**.
2. **Find the pending ones**
   Use the filter **No Odoo vendor**.
3. **Choose the contact**
   Fill in **Vendor** with the Odoo contact.

> The connector never creates a vendor, a category or a unit on its own: every link is your decision or an unambiguous VAT ID match. To send an order to GLOP, the vendor must be linked.

11 Stock and recipes

The connector can move Odoo stock with what GLOP sells, read recipes from GLOP and send them to it, compare GLOP's stock with Odoo's, and send GLOP an inventory count.

> Everything in this section is turned off: **Stock** is set to **No stock**, and **Sync sale formats and recipes**, **Compare the stock with GLOP**, **Send recipes to GLOP** and **Send stock counts to GLOP** are unticked. What Odoo writes to GLOP (recipes and counts) has only been tested with the module's own simulation and with direct, one-off calls to a GLOP cloud test account with no point of sale, on 18 September 2026. It has not been tested through FlexigoTech Cloud or with a real point of sale.



- **Real Odoo 20 screenshot.** The simulation connection, with its buttons and the **GLOP documents**, **Recipes** and **Differences** counters.

Where each piece of data comes from

- With **Stock** set to **Products sold**, the connector generates stock moves for what was sold from each posting. They come out as a delivery order and, if there are returns, as a return order. On the order you can see the **GLOP direction** (**Outgoing** or **Return**) and the **GLOP posting**.
- With **Sync sale formats and recipes** enabled, recipes arrive through the scheduled task **GLOP: master data**, every day, or with the **Fetch from GLOP** button on each recipe.
- GLOP's stock to compare with Odoo's is read from GLOP and stored in **Stock comparison**. The comparison is read-only: it writes no quantities either in Odoo or in GLOP.

Recipe buttons

Recipes are in **Recipes**, under **GLOP › Master data › Catalogue**.

Fetch from GLOP	Reads the recipes for that sale format again and updates what GLOP says now.
Create a bill of materials	Turns the recipe into an Odoo kit bill of materials.
Fetch from the bill of materials	Fills the lines with the product's kit bill of materials in Odoo.
Send to GLOP	Queues the whole recipe for that product and then reads it back to check it, because GLOP replies that it went fine even when the product does not exist.
Send this line	Sends the change of quantity or cost for a single ingredient.
Outbox items	Opens the outbox items for that recipe.

Stock comparison statuses

Each row of the comparison has one of these values in **Comparison**: **They match**, **They differ**, **Not linked** or **No warehouse chosen**. The row stores **Compared on** and **Read from GLOP on**. The **Recalculate** button reads Odoo's stock again without asking GLOP.

Sending a count to GLOP

1. **Open** Send stock count
   It is under **GLOP › Operations › Outgoing to GLOP**. Choose the connection and the **GLOP POS terminal**; its warehouse is the one that gets adjusted.
2. **Press** Load the count
   GLOP's stock is read at that moment and placed next to Odoo's.
3. **Check the lines and press** Send to GLOP
   A single inventory adjustment is queued with the lines that change: GLOP applies each quantity as a difference against its stock at that moment. The **Send to GLOP** button only appears once the count is loaded.

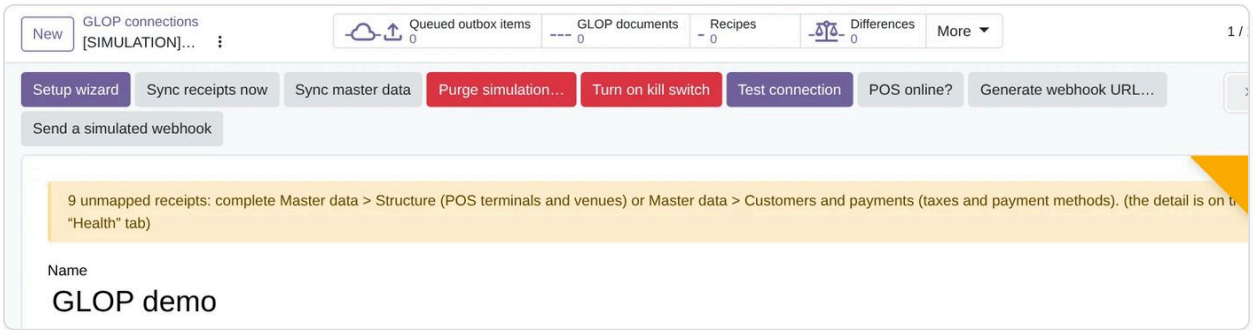
> With GLOP as the stock master, a product with a kit bill of materials does not move in Odoo: the move fails and names the product, because the ingredients would be deducted twice. In that case the recipe cannot be turned into a bill of materials. The link on a delivery order to its GLOP posting cannot be changed, and a second count cannot be sent while another one for the same warehouse is unresolved.

USER MANUAL · GLOP

12 Orders to GLOP

Sends an Odoo order to a GLOP table or to the delivery queue, once, and marks it as confirmed when the matching receipt arrives.

> Turned off: **Send table and delivery orders** is switched on in the **Sync** tab of the connection. It has only been tested with the module's own simulation and with a test order, direct, to a GLOP cloud test account with no point of sale, created and voided on 18 September 2026. It has not been tested through FlexigoTech Cloud or with a real point of sale.



- **Real Odoo 20 screenshot.** The simulation connection, with the **Setup wizard** button and the **Queued outbox items** and **GLOP documents** counters.

Orders are in **GLOP › Operations › Documents › Orders to GLOP**. You create each order yourself. It is sent when you press **Send to GLOP**: it enters the send queue and the scheduled task **GLOP: sending customers** sends it every 15 minutes. That same task handles the rest of the connection's outgoing items. You can follow it in **GLOP › Operations › Outgoing to GLOP › Outbox items to GLOP**.

If you enter a **Table**, GLOP adds the lines to that table's open ticket. If you leave it empty, the order is parked as a delivery order.

Send to GLOP	Puts the order in the send queue. It only appears while the order is in **Draft**.
It did reach the POS	Marks the order as confirmed by hand. It only appears while the order is in **Sent, unconfirmed**.
It did not reach the POS	Closes the order as **Error**. Nothing is resent on its own. It only appears while the order is in **Sent, unconfirmed**.
Void in GLOP	Voids in GLOP an order already sent that has not been paid yet. It only appears while the order is in **Sent, unconfirmed**.
GLOP receipt	Opens the receipt that proves the lines arrived. It appears when the order has that receipt.

Statuses

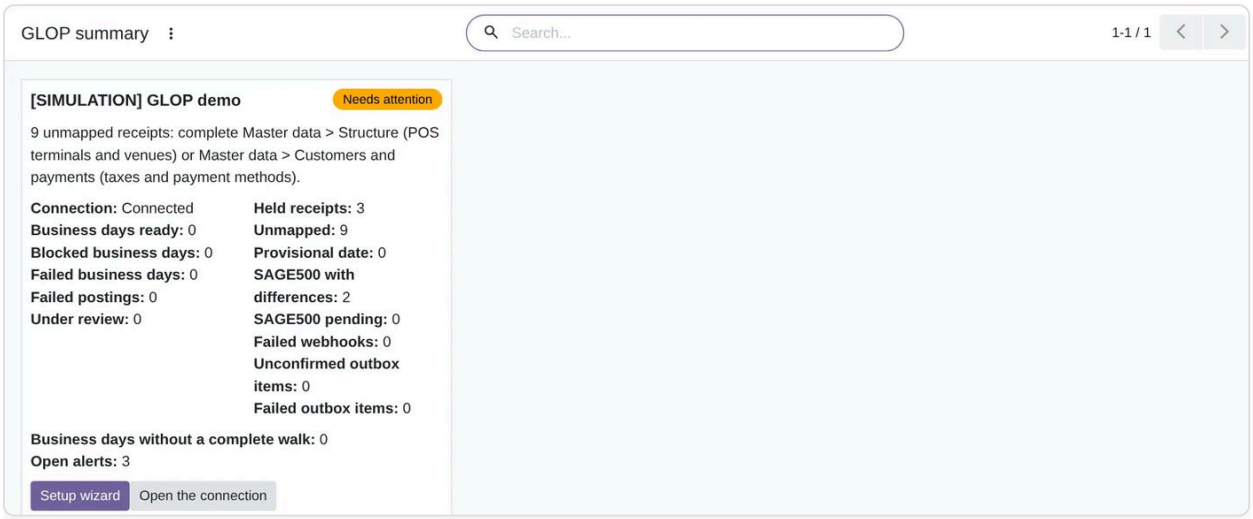
Draft	The order has not been sent.
Queued	The send is waiting for its turn.
Sent, unconfirmed	The send has gone out and the receipt is still missing.
Confirmed in GLOP	The receipt for this order has arrived, or you confirmed it yourself.
Error	The order did not arrive.
Cancelled	The order has been voided.

1. **Create the order**
   Choose terminal, order type and lines, and table if it is for a table.
2. **Press Send to GLOP**
   The order moves to **Queued** and the sending task sends it.
3. **Wait for confirmation**
   When the receipt arrives, the order moves to **Confirmed in GLOP**. If it does not arrive, use **It did reach the POS** or **It did not reach the POS**.

> A sent order is never sent again: resending it would duplicate the lines on the table's ticket. Only a sent order that has not been paid yet can be voided.

13 Reconciliation and checks

The connector matches receipts, GLOP's accounting export (SAGE500) and the postings for each terminal and date, and checks the integrity of the postings every day.

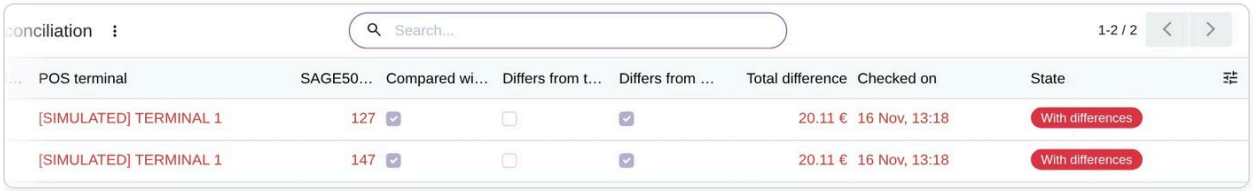


- **Real Odoo 20 screenshot.** The GLOP summary of the [SIMULATION] GLOP demo connection, with the open alerts.

Where the data comes from

- The scheduled task **GLOP: SAGE500 reconciliation** runs every day and checks each terminal and business date.
- The scheduled task **GLOP: integrity of the journal entries** runs every day and makes no calls to GLOP.
- The **Check again** button repeats a check by hand.

The reconciliations are in **GLOP › Operations › Outgoing to GLOP › SAGE500 reconciliation**.



- **Real Odoo 20 screenshot.** The **SAGE500 reconciliation** in the simulation: two days **With differences**, with the total difference for each.

Button and record fields

Check again	Repeats the check for that day even if the SAGE500 check is turned off. Only the accounting group can see it.
Total difference	SAGE500 total minus what was posted, or minus the receipts if there is no posting yet.
Compared with what was posted	Ticked if the day had postings published when it was checked.
Differs from the receipts and **Differs from what was posted**	Checkboxes showing which side SAGE500 does not match.

Days in a row without SAGE500	How many days in a row the export has been unavailable.

Statuses

Pending	Not checked yet, or a reply is still missing.
It balances	SAGE500 matches what was compared.
With differences	There are differences.
Not available	The merchant cannot obtain the SAGE500 export.
Off	The check is switched off on the connection.

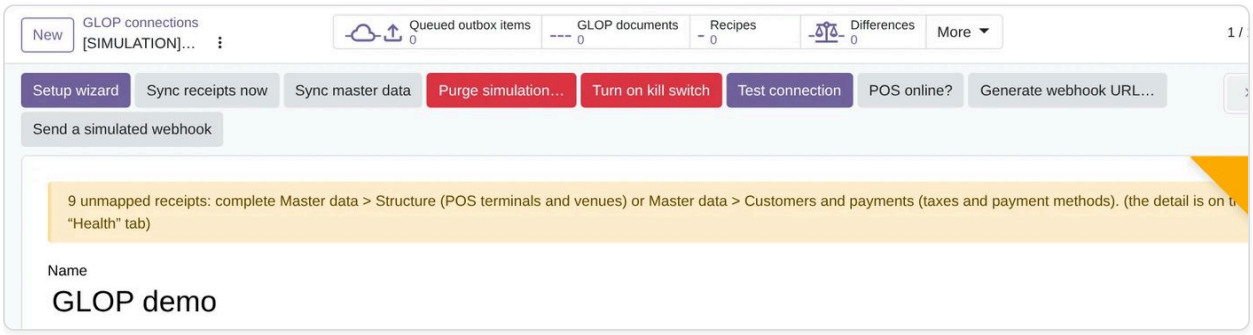
How to review a day with differences

1. **Filter the list**
   Use the **With differences** filter.
2. **Open the record**
   Look at the **By VAT rate** and **By sale type** tabs.
3. **Run the check again**
   Click **Check again** once you have fixed the cause.

> Reconciliation records are read-only: they cannot be created or edited by hand, only the connector generates them.

14 Setup and monitoring

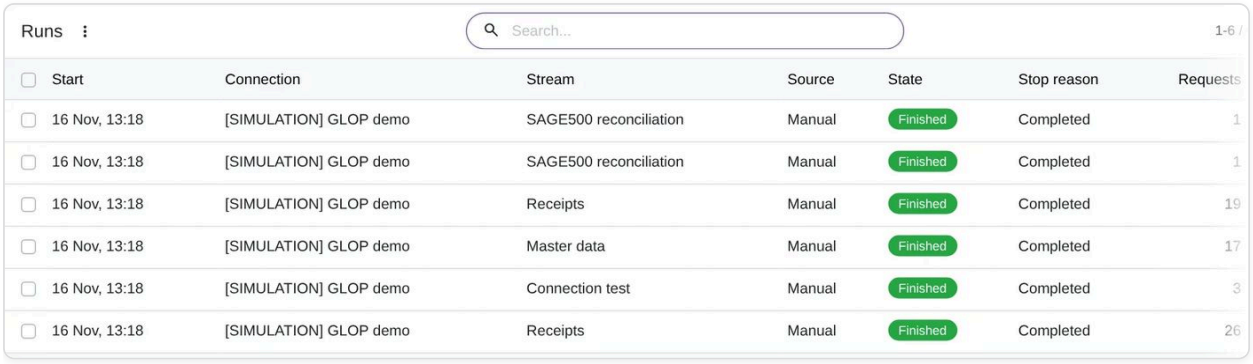
The wizard guides you through setting up a connection with GLOP, the simulation lets you test without real data, and the lists under **GLOP › Audit › Sync** keep a record of what has happened.



- **Real Odoo 20 screenshot.** The [SIMULATION] GLOP demo connection, using the **Setup wizard** button.

Where each piece of data comes from

DATA	WHERE IT COMES FROM
Runs	One for each pass of a flow. Created by the scheduled tasks and by the connection buttons. They are visible under **GLOP › Audit › Sync › Runs**.
Cursors	Updated automatically as each flow progresses. Visible under **GLOP › Audit › Sync › Cursors**.
Calls to GLOP	Saved each time a call is made. Visible under **GLOP › Audit › Issues › API calls (errors)**.
Received webhooks	They arrive when GLOP sends a notification. The scheduled task **GLOP: process received webhooks** processes them every 5 minutes.



- **Real Odoo 20 screenshot.** The **Runs** of the simulation, showing the flow, the origin, the status and the **Stop reason** of each one.

Wizard buttons

Test connection	Checks the connection and leaves the result inside the wizard.
Sync master data	Brings in establishments and terminals, which arrive switched off.
Activate POS terminals	Switches on the terminals of the chosen establishment.
Run a test without posting	Imports the last 3 business days of each activated terminal, without posting.

Save policies	Saves the policies chosen in step 7 to the connection.
Start	Completes the setup at step 9.
Create a simulation connection...	Opens the simulation wizard.

Run statuses

- **In progress**
- **Finished**
- **Partial**
- **Failed**
- **Skipped**

Steps

1. **Open the connection**
   Go to **GLOP › Configuration › GLOP connections** and click **Setup wizard**.

2. **Go through the steps**
   Use **Next** and **Previous**. There are nine steps: step 6 is the dry run without posting and step 9 is the go-live.

3. **Check the monitoring**
   Review **Runs**, **Cursors** and **Received webhooks** in **GLOP › Audit › Sync**.

> While there is no **Go-live date**, no receipt comes in, and business days before that date are never requested. It is set in step 7 of the wizard or in the **Sync** tab of the connection.

USER MANUAL · GLOP

15 What happens by itself

The connector fetches from GLOP, sends and reviews data on its own with ten scheduled actions, all switched on when you install it.

ACTION	FREQUENCY	WHAT IT DOES
GLOP: sync receipts by business day	Every 10 minutes	Fetches the receipts for each business day.
GLOP: process received webhooks	Every 5 minutes	Processes the GLOP notifications that have arrived.
GLOP: business days and posting	Every 15 minutes	Evaluates the business days and their posting.

ACTION	FREQUENCY	WHAT IT DOES
GLOP: sending customers	Every 15 minutes	Sends to GLOP what is in the queue: customers, purchase and stock documents, recipes and table or delivery orders. It only works with connections that have at least one send box ticked.
GLOP: pending purchase documents	Every hour	Fetches the purchase documents pending delivery.
GLOP: master data	Every day	Updates the master data.
GLOP: SAGE500 reconciliation	Every day	Compares the receipts and the journal entries with the SAGE500 export.
GLOP: integrity of the journal entries	Every day	Checks the posted journal entries.
GLOP: deep pass over old business days	Every day	Goes over old business days to detect missing receipts.
GLOP: night maintenance	Every day	Checks the health of the connections and raises or closes warnings.

The frequencies are changed in **Settings › Technical › Automation › Scheduled Actions**, with developer mode switched on.

USER MANUAL · GLOP

16 If something fails

Common cases and where to look in each one.

The connection buttons do not call GLOP	Without the trial or a FlexigoTech Cloud subscription, the connector does not talk to GLOP. Check the status in **Settings**, in the FlexigoTech Cloud block, and click **Check now** if you have just activated it.
The credentials are not valid	Open **Credentials...** on the connection and enter the Client ID and the secret again, without spaces. For security, Odoo never shows the saved secret. Then click **Test connection**.
An action is missing because of permissions	Some actions require a specific group, and the person who handles the accounting is the one who posts or retries. If Odoo shows "This action requires the “%(group)s” group.", ask an administrator for that permission.
A posting ended with an error	Click **Retry** on the posting: the process runs again from the pre-check. If the error is of the "LEGAL_GUARD" type, only a GLOP administrator can retry it, and its cause must be fixed first.
An order sent to GLOP did not arrive	If GLOP did not receive it, the order is closed as an error with **It did not reach the POS**. The connector does not resend it on its own.
You want to see what GLOP replied	Look at **GLOP › Audit › Issues › API calls (errors)** and **GLOP › Audit › Sync › Received webhooks**. A webhook with an error is processed again with **Retry**; API calls are only a log. You decide how long they are kept in **Settings**, in the **Retention** block.

If the database has no FlexigoTech Cloud trial or subscription, or the connection does not yet have its credentials saved there, the connector does not call GLOP: each run ends with the reason **Paused: FlexigoTech**

Cloud, which you can see in **Runs**. As soon as there is a service and credentials, it starts calling again without you doing anything; to avoid waiting for the module to renew the permission, click **Check now** in **Settings**. In a neutralised copy of the database, such as an Odoo.sh test branch, all connections are left with the kill switch on and the webhook off: the copy neither calls GLOP nor receives its notifications.

17 Support

Write to us to install it, configure it or with any question. With the Standard plan we reply within 1 working day; with Pro, within 4 working hours, and if needed we can look at it over a video call. Tell us your Odoo version, the module version and, if there is an error, its full text.

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/glop-connector-odoo.html

On Odoo Apps

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

GLOP 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).