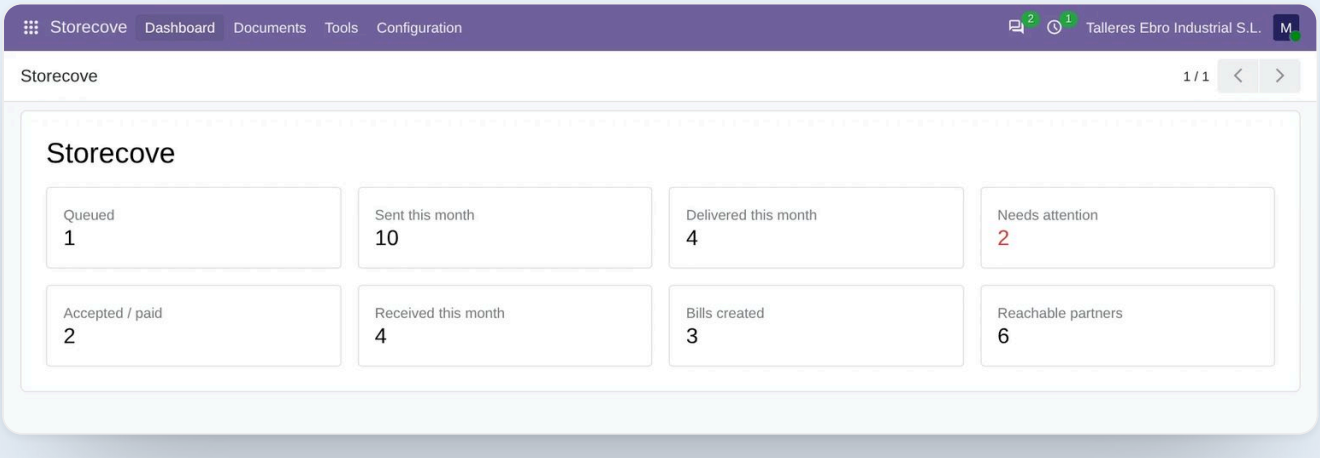
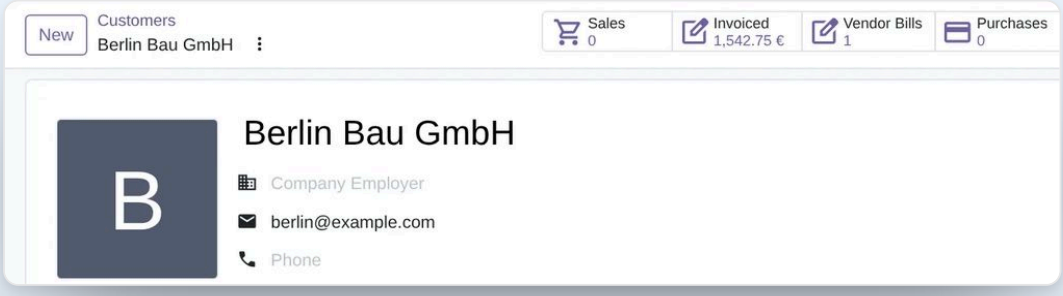
USER MANUAL · E-INVOICING

Storecove 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 20.0.1.2.4 | Odoo 20 screenshots | Edition of 16-11-2026





Real Odoo 20 screenshot. The summary of the app and a contact's card with its reachability on Storecove.

Contents

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

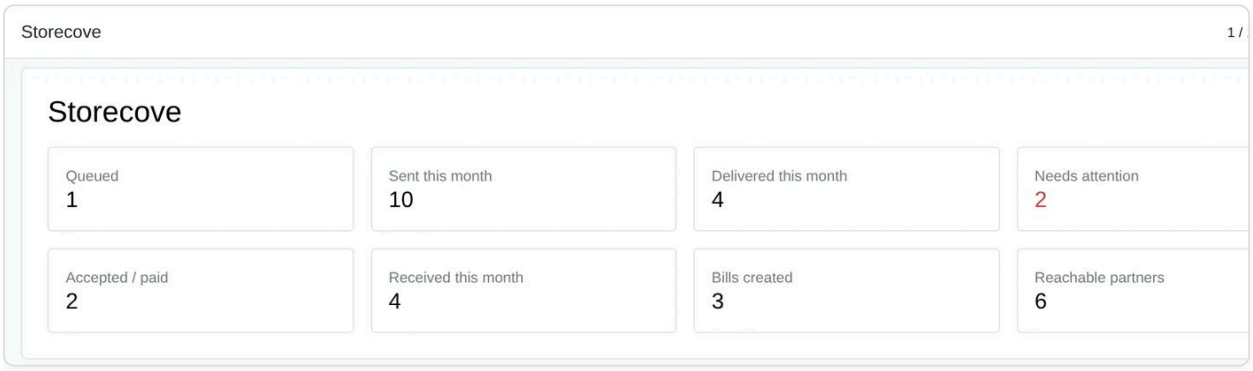
1	What it is and where it is	3
2	Before you start	4
3	Installation	4
4	Activate FlexigoTech Cloud: the 14-day trial	5
5	Connecting your account	7
6	The account options	9
7	Legal entities and identifiers	11
8	Contacts and reachability	12
9	Sending e-invoices	14
10	Tax mappings	16
11	Received Documents	17
12	Test and manual sending tools	19
13	Dashboard and log	20
14	What happens automatically	21
15	If something fails	22
16	Support	23

This manual describes version 20.0.1.2.4 of the module (Odoo branch 20.0, commit a76bb9d4c). 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 Storecove. 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 it is

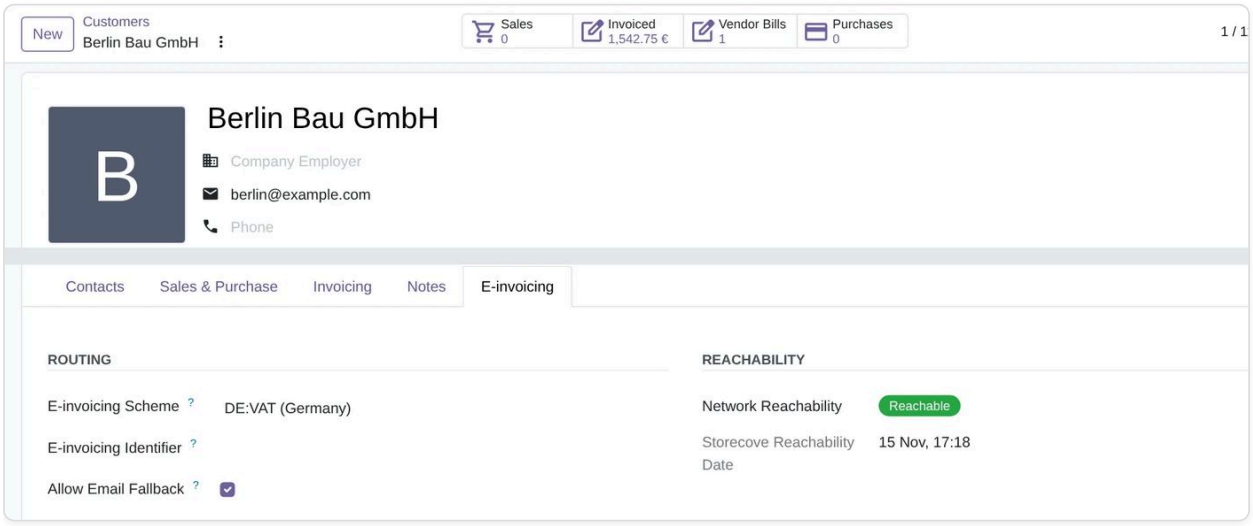
The connector adds an app called **Storecove** to Odoo to send e-invoices through Storecove and receive those from your suppliers. Each connection to Storecove is an account, under **Storecove › Configuration › Accounts**.



- **Real Odoo 20 screenshot.** The app summary, with the counters for submissions and receipts.

Dashboard	The app summary.
Documents	**Submissions**, with the documents you send, and **Received Documents**, with those that reach you.
Tools	**Send Test Invoice**, **Check Partner Reachability** and **Send Raw Document (UBL/CII/IDOC)**.
Configuration	**Accounts**, **Legal Entities**, **Identifier Schemes**, **Tax Mappings** and **Sync Log**. This menu, **Send Test Invoice** and **Send Raw Document (UBL/CII/IDOC)** are only visible to the connector's **Manager** role.

Outside its own app, the connector extends Odoo's invoices, purchase orders, contacts and units of measure. On the contact form it adds an **E-invoicing** tab with routing and reachability, and on the invoice, a **Preview Storecove Route** button.



- **Real Odoo 20 screenshot.** A contact's **E-invoicing** tab, with its reachability on Storecove.

2 Before you start

Before installing the connector you need the following.

- Your own Storecove account (sandbox or production) and its API key. Storecove's fees are not included.
- A FlexigoTech Cloud subscription or its 14-day trial without a card. The API key for each account is stored encrypted in FlexigoTech Cloud and not in your Odoo.
- 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: only an administrator can accept the consent and save the API key.

> The connector does not register you with Storecove or create your account, nor does it obtain the API key for you: you get that yourself from Storecove. Storecove's fees are your responsibility and are not included in the connector.

3 Installation

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

How to get it

From its page on Odoo Apps (**https://apps.odoo.com/apps/modules/20.0/flexigobe_storecove_connector**) or from the connector's page on 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's 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_storecove_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 "Storecove" and click **Activate**.

On your server

1. **Copy the folders**

Place `flexigobe_storecove_connector` and `flexigotech_cloud` in a folder that is in the `addons_path` of your Odoo.

2 **Restart Odoo**

So that it reads the new modules.

3 **Install the app**

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

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

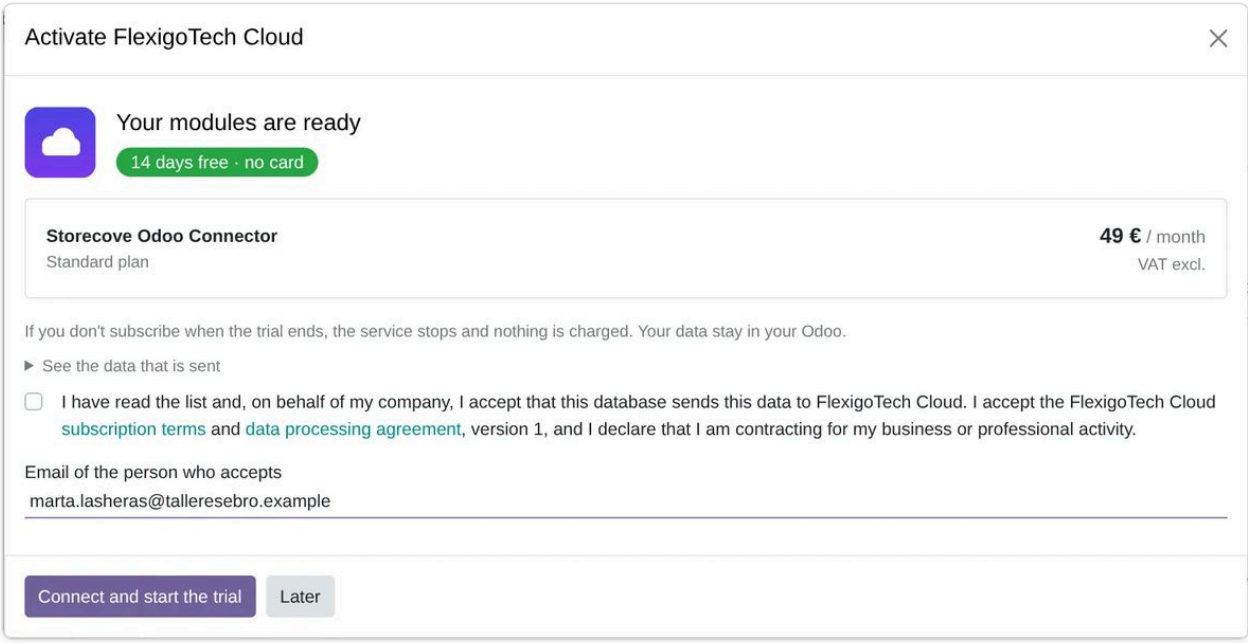
USER MANUAL · STORECOVE

4 Activate FlexigoTech Cloud: the 14-day trial

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

1 **Click Activate in the top bar**

It appears as soon as you install the module. Open 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**

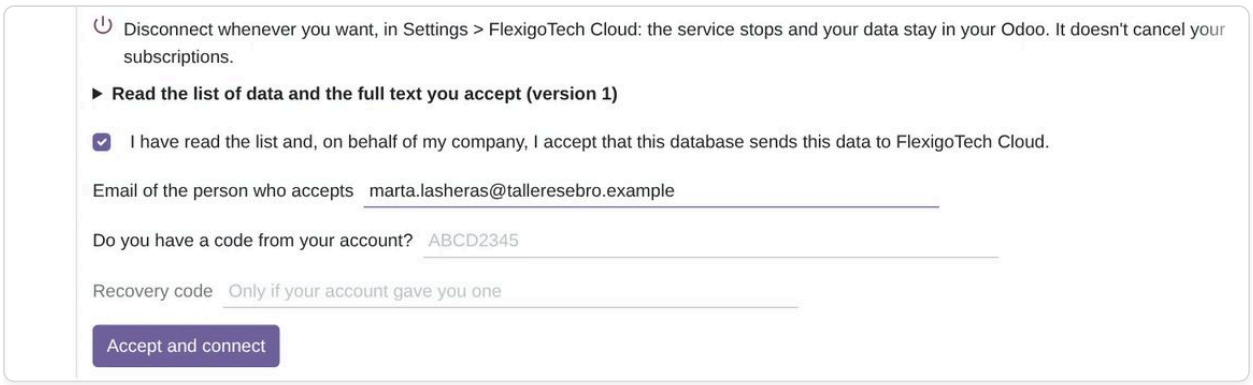
Tick "I have read the list and, on behalf of my company..." (it covers the subscription terms and the data processing agreement) and enter the **Email of the person who accepts**.

3 **Click Connect and start the trial**

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

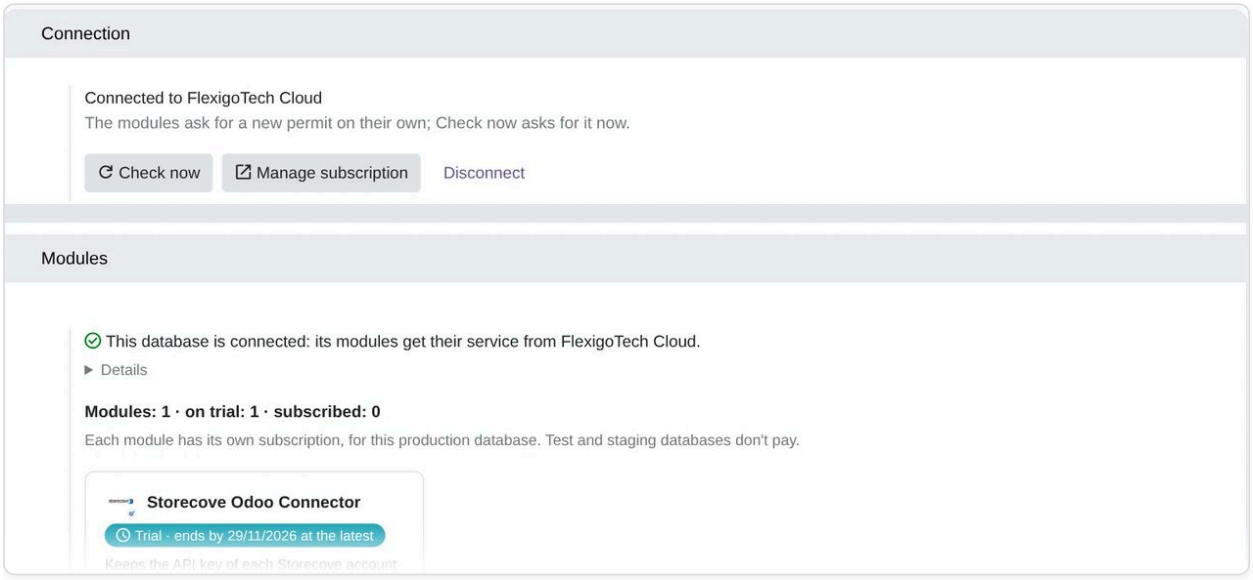
The same, from Settings

In **Settings › FlexigoTech Cloud** is the full list of what the base 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 base 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 base 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 base 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** requests the permission immediately (otherwise the module renews it

automatically every 6 hours).

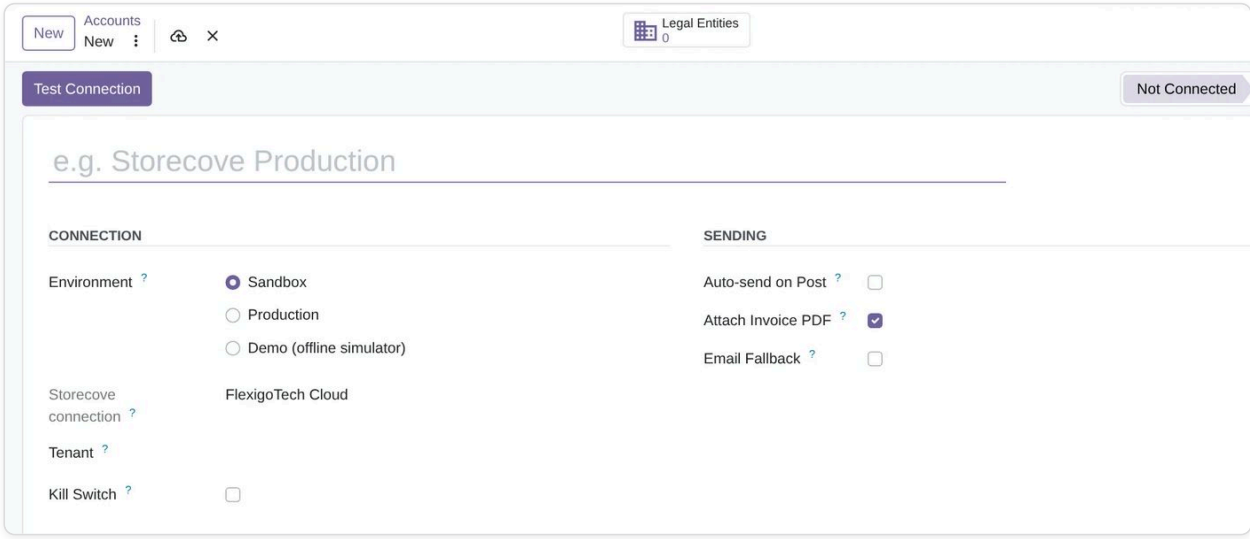
> You cannot start a trial on a neutralised copy (staging) or on a database with demo data: Settings says "This database is a neutralised copy or has demo data, so trials can't be started here."

5 Connecting your account

Each connection to Storecove is an account, and it is created under **Storecove › Configuration › Accounts**.

1 **Open the accounts menu**

Go to **Storecove › Configuration › Accounts** and click **New**. On the new account, **Storecove connection** is already set to **FlexigoTech Cloud** and cannot be changed.



- **Real Odoo 20 screenshot.** The form of a new account, with **Storecove connection** set to **FlexigoTech Cloud**.

2 **Enter the name and environment**

Fill in **Name** and choose the **Environment**: **Sandbox**, **Production** or **Demo (offline simulator)**. Sandbox and production are separate Storecove accounts, each with its own keys. **Demo (offline simulator)** never contacts Storecove.

3 **Adjust sending**

In the **Sending** group you can switch on **Auto-send on Post**, **Attach Invoice PDF** and **Email Fallback**. **Attach Invoice PDF** is on by default.

4 **Accept the consent**

In the **FlexigoTech Cloud (requires FlexigoTech Cloud subscription)** group, tick the box **I accept that my Storecove API key and my calls to Storecove go through FlexigoTech Cloud**.

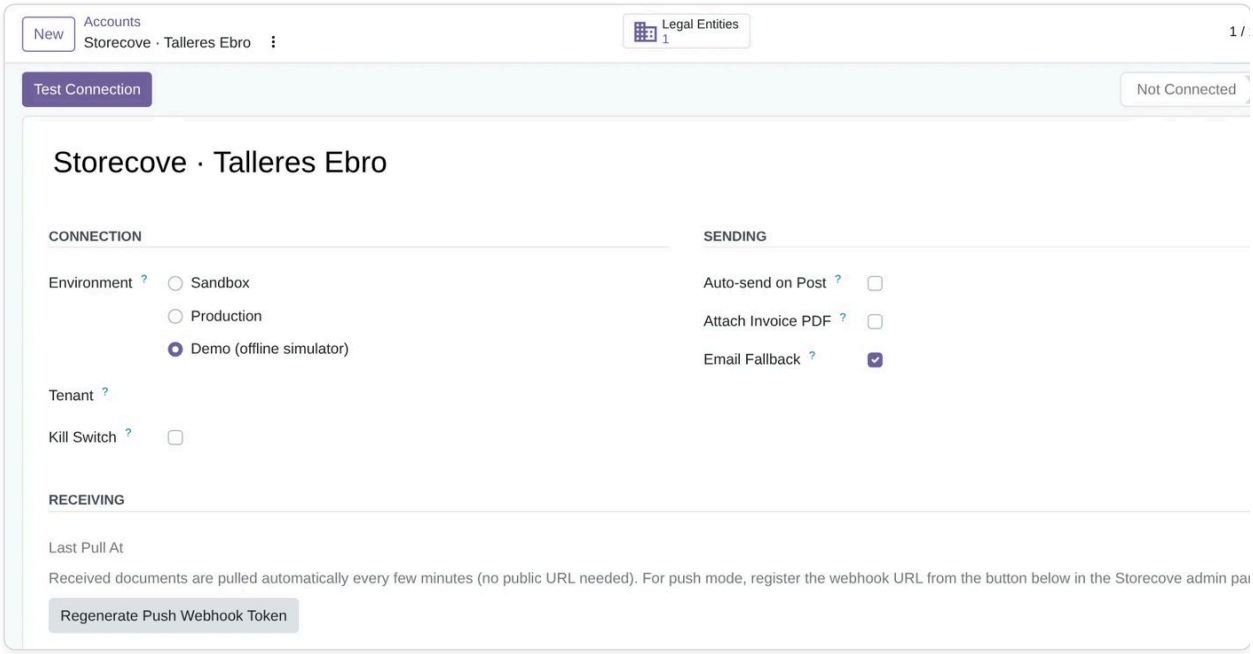
5 **Enter the API key**

Fill in **API key (FlexigoTech Cloud)** and save. The key travels once, encrypted, to FlexigoTech Cloud and is not kept in Odoo. An active trial or subscription is required: without the service, Odoo does not send

the key. Afterwards the date appears under **Saved in FlexigoTech Cloud**.

6 Test the connection

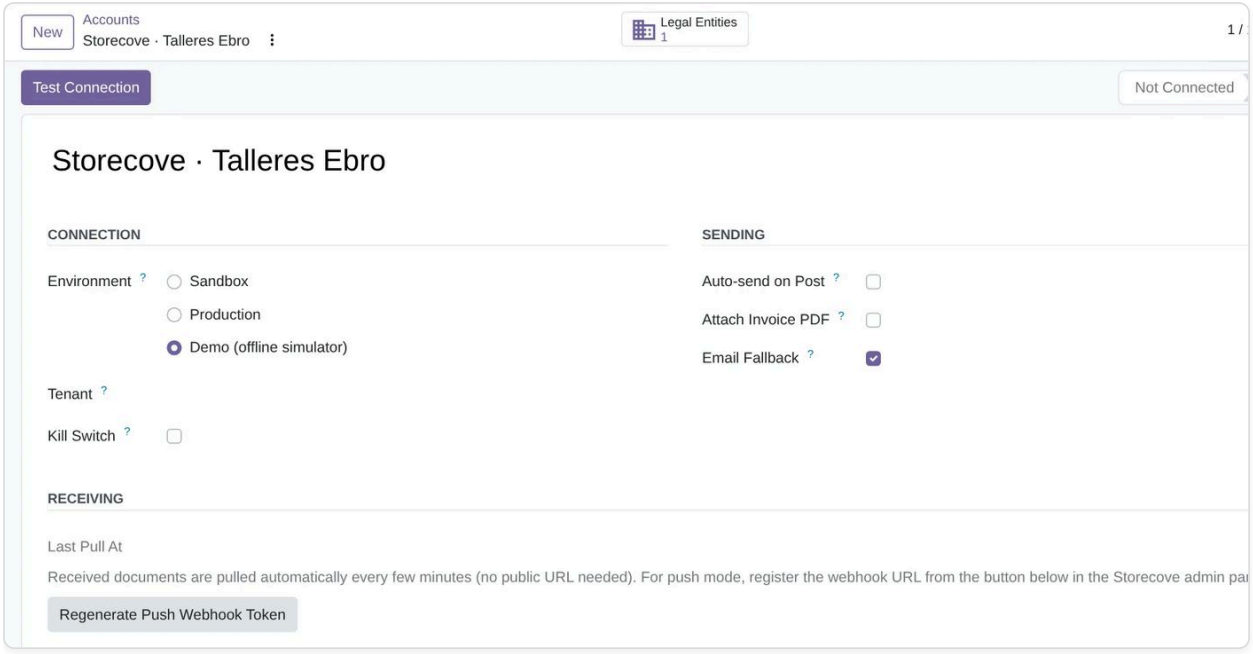
Click **Test Connection**. If everything is fine, the account moves to **Connected**; if not, to **Connection Error**.



- **Real Odoo 20 screenshot.** An account with the **Test Connection** and **Regenerate Push Webhook Token** buttons.

> Only an Odoo administrator can accept the consent and enter the API key. Without the consent, the connector does not send the key and warns you with "Before saving the API key, an administrator has to accept that the key and the calls to Storecove go through FlexigoTech Cloud. Nothing was sent.". A new account does not keep the key in Odoo: if you try to enter it outside the FlexigoTech Cloud section, you get "A new Storecove account keeps its API key in FlexigoTech Cloud, never in Odoo: save it in the FlexigoTech Cloud section of the account.".

After connecting, data arrives by two routes. Queued invoices are sent by the **Storecove: send queued e-invoices** task, every 5 minutes. Received documents are retrieved by the **Storecove: poll webhook queue (pull mode)** task, also every 5 minutes, and no public URL is needed. On the account you see the **Last Pull At** date. The code also has a push mode, in which Storecove calls a URL on your Odoo, but the form does not show that URL. **Regenerate Push Webhook Token** changes the token that goes in it and invalidates the previous one. If **Kill Switch** is ticked, nothing is sent or polled towards Storecove and queued documents wait.

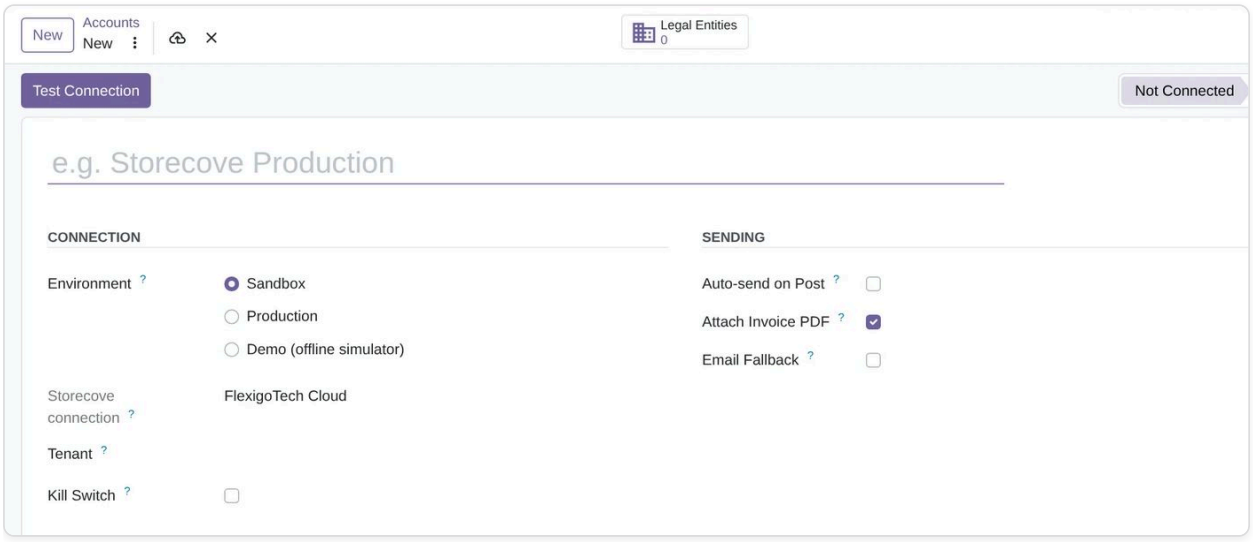


- **Real Odoo 20 screenshot.** The Talleres Ebro account, with its environment, its sending and its receiving.

USER MANUAL · STORECOVE

6 The account options

The options are chosen on the **Storecove › Configuration › Accounts** form and apply to all invoices that go through that account.



- **Real Odoo 20 screenshot.** The form of a new account, with the **Connection** and **Sending** groups.

OPTION	WHEN CREATING	WHAT IT DOES
Name	Empty, required	The name by which you recognise the account in lists.
Environment	**Sandbox**	The other options are **Production** and **Demo (offline simulator)**. Sandbox and production are separate Storecove accounts, each with its own keys. Demo does not contact Storecove and simulates responses.

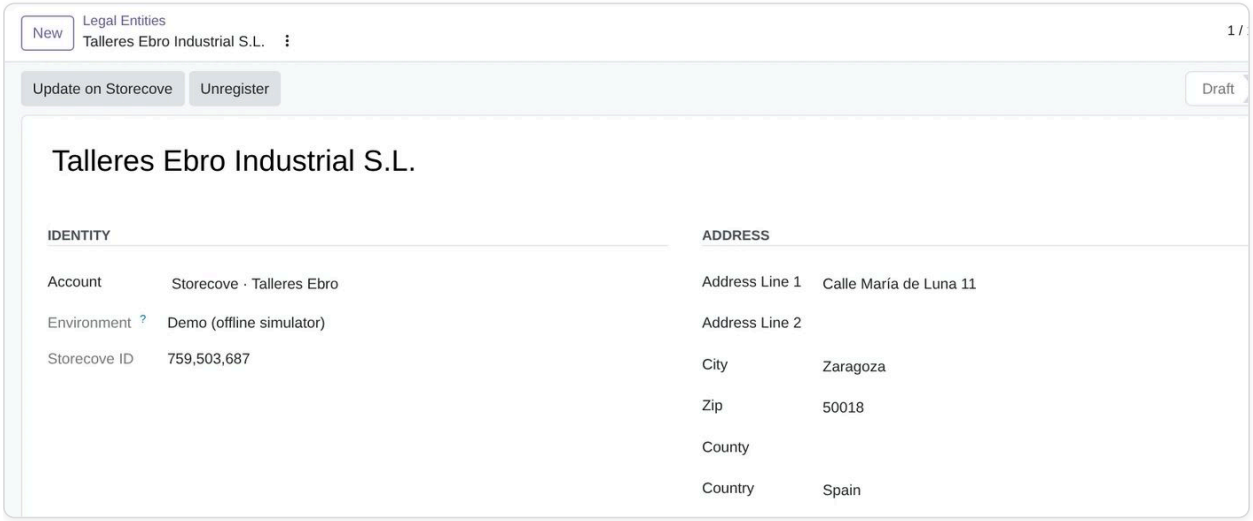
OPTION	WHEN CREATING	WHAT IT DOES
Storecove connection	**FlexigoTech Cloud**	Read only, not changeable: every account you create connects through FlexigoTech Cloud.
Tenant	Empty	Optional identifier passed to Storecove's legal entities. Storecove returns it in each notice, to tell tenants apart.
Kill Switch	Off	Suspends all outbound traffic to Storecove for this account. Queued documents stay queued and nothing is sent or polled.
Main Company	The active company	The company the account belongs to. It is required.
Auto-send on Post	Off	On: every customer invoice or credit note you confirm, from a company with a connected legal entity, enters the send queue automatically.
Attach Invoice PDF	On	Embeds the Odoo invoice PDF in the e-invoice as the main attachment.
Email Fallback	Off	If the recipient has no reachable electronic identifier, Storecove delivers the invoice by email to the contact's billing address.

The API key is not written in Odoo. It is stored encrypted in FlexigoTech Cloud, from the **FlexigoTech Cloud (requires FlexigoTech Cloud subscription)** section of the account, and only an administrator can do so. If you try to save it in any other way, Odoo prevents you with a message. The connection type of an account also cannot be changed after it is created.

7 Legal entities and identifiers

Each Odoo company that sends or receives invoices through Storecove is registered as a legal entity, and its Peppol identifiers are provisioned from it.

Legal entities are under **Storecove › Configuration › Legal Entities**. The catalogue of countries and schemes is under **Storecove › Configuration › Identifier Schemes**. You create the legal entities yourself. The scheme catalogue comes preloaded on installation, with 86 schemes, and from its list you cannot create or delete schemes.



- **Real Odoo 20 screenshot.** Legal entity registered, with the **Update on Storecove** button.

Register on Storecove	Registers the entity on Storecove. It is hidden when the entity is already in **Registered** state.
Update on Storecove	Sends changes for an already registered entity. Only visible in **Registered** state.
Unregister	Deletes the entity on Storecove, along with its identifiers, after asking for confirmation. Only visible in **Registered** state.
Provision	Located on each line of the **Identifiers** tab and registers that identifier on the exchange network. It is hidden when the identifier is already **Provisioned**.
De-provision	Removes it from the network and returns it to **Draft**. Only visible with the identifier **Provisioned**.

The **Identifiers** tab lists the entity's Peppol identifiers.

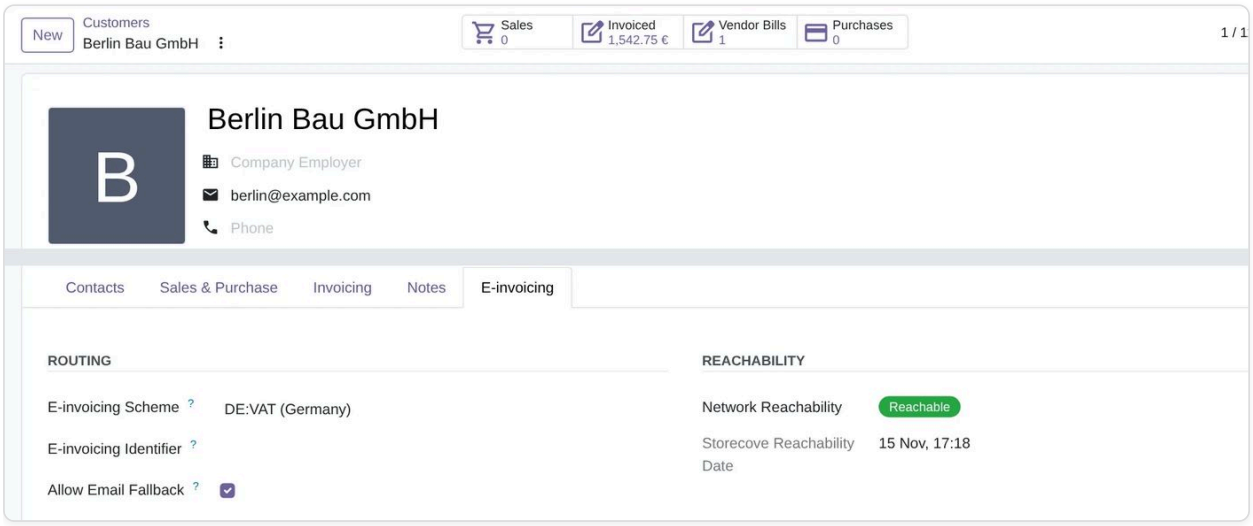
Entity states

- **Draft**: the entity is not yet on Storecove. This is the initial state.
- **Registered**: Storecove already has the entity and has assigned it its **Storecove ID**.
- **Error**: the last registration attempt failed.

> An entity must act as sender, receiver or both. To register it you need the party name, address line 1, city, postcode and country. An identifier cannot be deleted while it is provisioned: first de-provision it.

8 Contacts and reachability

Each contact's record stores the data needed to route its e-invoices and the result of checking whether it can receive them.



- **Real Odoo 20 screenshot.** **E-invoicing** tab of a reachable contact.

Where each piece of data comes from

You fill in the scheme, the identifier and the email fallback checkbox. The **Guess Identifier** button sets the scheme that matches the contact's country and, if the identifier is empty, extracts it from the tax ID. In countries whose documented delivery chain ends in email, it also enables the email fallback. Reachability and its date are written by the connector when you press **Check Reachability**, on the record or from **Storecove › Tools › Check Partner Reachability**. There is no scheduled task that updates them.

Record buttons

Guess Identifier	It fills in the scheme and the identifier from the country and the tax ID. If the country has no known scheme, it warns you with «No known scheme for country "...".».
Check Reachability	Asks Storecove whether the contact can receive e-invoices and saves the result and the date. If there is no account, it warns with "Connect a Storecove account first.".

Network Reachability states

STATE	MEANING
Not checked	The contact has not been checked yet.
Reachable	Storecove finds it on an exchange network.
Not reachable	Storecove does not find it on any network.

Checking several contacts at once

1. **Open the wizard**
   Go to **Storecove › Tools › Check Partner Reachability**.



- **Real Odoo 20 screenshot.** **Check Partner Reachability** wizard.

2. **Choose the contacts**

   In **Contacts**, select the ones you want to check.

3. **Decide whether to guess**

   Guess Missing Identifiers is switched on: it fills in scheme and identifier for contacts that lack them.

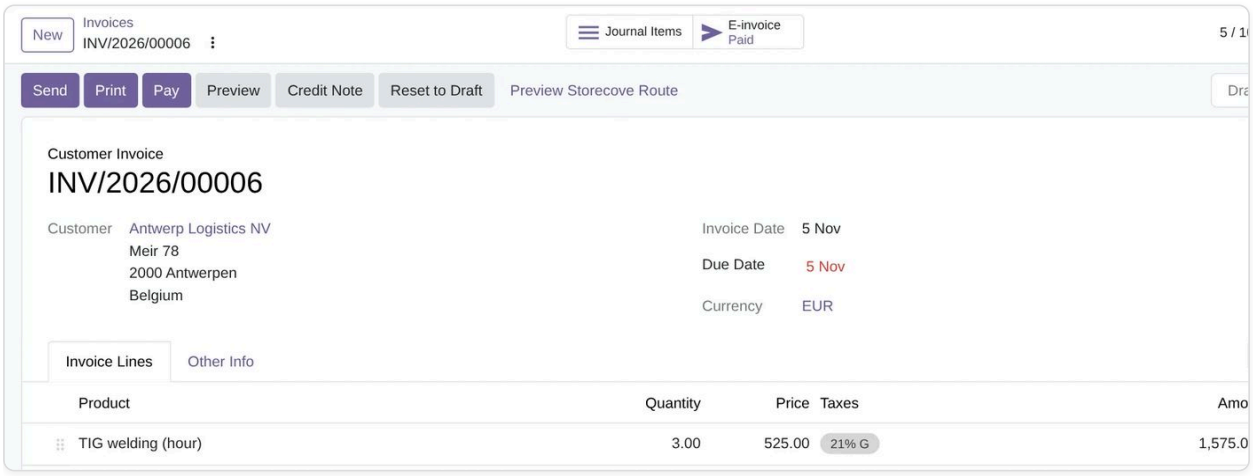
4. **Press Check**

   Each contact is checked and the result is stored on its record. Afterwards the **Check** button disappears and **Close** remains.

> If no partners are selected, the wizard warns you with "Select at least one partner.". A failure on one partner does not stop the others from being checked.

9 Sending e-invoices

The connector converts the Odoo invoice or credit note into the Storecove document, sends it over the network that matches your customer, and tracks its status until delivery.



- **Real Odoo 20 screenshot.** Confirmed invoice with the sending buttons and the e-invoice status.

Where each send comes from

- When you confirm an invoice, the connector queues it if the account has automatic sending enabled. If something needed to send it is missing, such as the route or the customer's country, the invoice is still confirmed but is not queued, with no warning.
- The **Send via Storecove** button sends it to Storecove immediately. If FlexigoTech Cloud is paused, it stays in the queue and the scheduled task retries it.
- The scheduled task **Storecove: send queued e-invoices** delivers to Storecove whatever is in the queue, every 5 minutes.
- The task **Storecove: poll webhook queue (pull mode)** collects, every 5 minutes, the notifications from Storecove that update the status.

Invoice buttons

Send via Storecove	Appears on confirmed customer invoices and credit notes that have no active submission.
Self-Bill via Storecove	The same, for confirmed vendor bills and refunds: the invoice is generated by the buyer on behalf of the supplier.
Preview Storecove Route	Only on confirmed customer invoices and credit notes. It is indicative and does not call Storecove: it shows the planned sending route, even if there is no account or legal entity yet.
E-invoice	The button at the top showing the submission status. It opens the submissions for that invoice and only appears if it already has one.

Submission buttons

Send Now	Sends a queued submission to Storecove without waiting for the task.
Cancel	Cancels a queued submission.
Re-queue	Available on failed, cancelled or **No Route Found** submissions. Prepares a new submission with the current data from the invoice and the contact.
Fetch Evidence	Requests from Storecove the proof of transmission. It does not appear in **Queued**, **Failed**, **No Route Found** or **Cancelled** state.

States

The state goes from **Queued** to **Sent**, **Delivered** and, if the customer's network allows it, **Acknowledged**, **In Process**, **Under Query**, **Accepted** and **Paid**. It can also end in **Rejected**, **Failed**, **No Route Found** or **Cancelled**.

Steps

1. **Confirm the invoice**

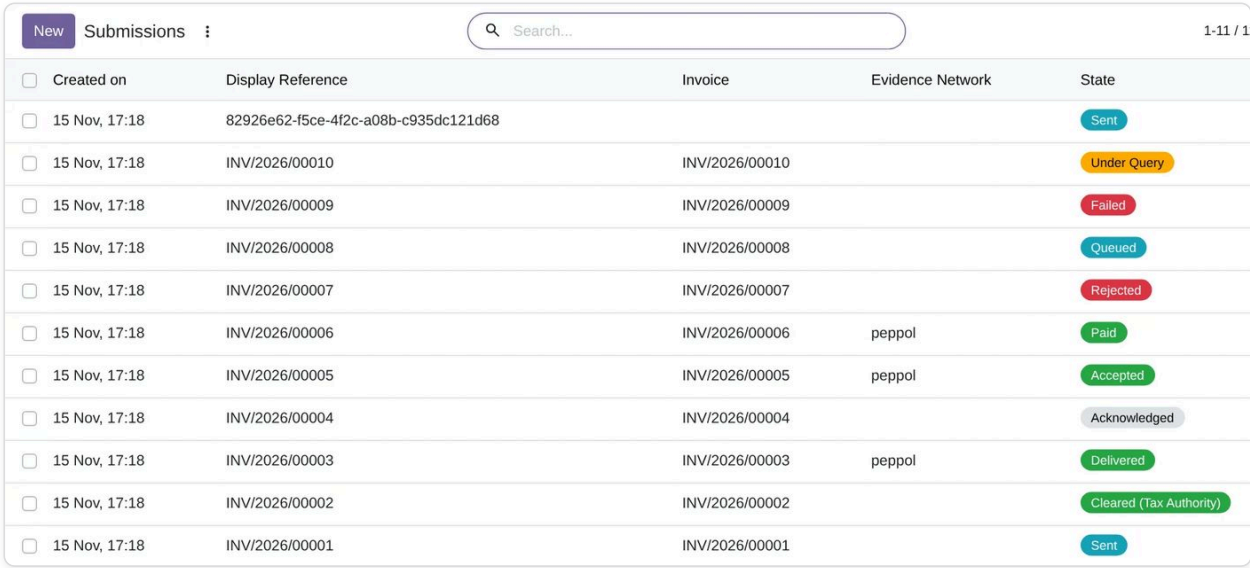
   If automatic sending is on, it is queued by itself.

2. **Press Send via Storecove**

   Only if it was not queued automatically. The invoice goes to Storecove at that moment.

3. **Track the status**

   Check the **E-invoice** button on the invoice, which shows the submission status, or the **Submissions** list.



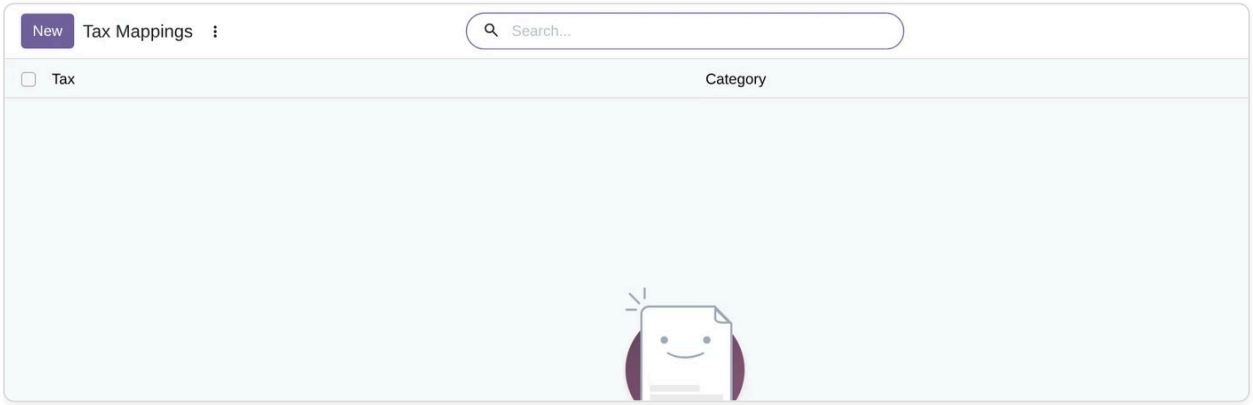
- **Real Odoo 20 screenshot.** Submissions list with their status.

> You cannot send a draft invoice or an invoice that already has an active submission. Only queued submissions can be cancelled: «Only queued submissions can be cancelled; "..." was already handed to Storecove.»

10 Tax mappings

Here you force the tax category that the connector uses for a specific tax, instead of the one it deduces itself.

Go to **Storecove › Configuration › Tax Mappings**. By default the list is empty and the connector deduces each tax's category on its own.



- **Real Odoo 20 screenshot.** The **Tax Mappings** list, still with no overrides.

You create the mappings yourself, one per tax: **New** adds a line to the list itself, which has no separate form.

What you fill in for each mapping

Tax	The Odoo tax for which you want to force the category. Required.
Company	The tax's company. It is copied automatically from the tax.
Category	The category to use for that tax. It defaults to **Standard rate**. The others are **Zero rated**, **Reverse charge**, **Intra-community**, **Exempt**, **Export** and **Outside scope**. Required.

How it is applied

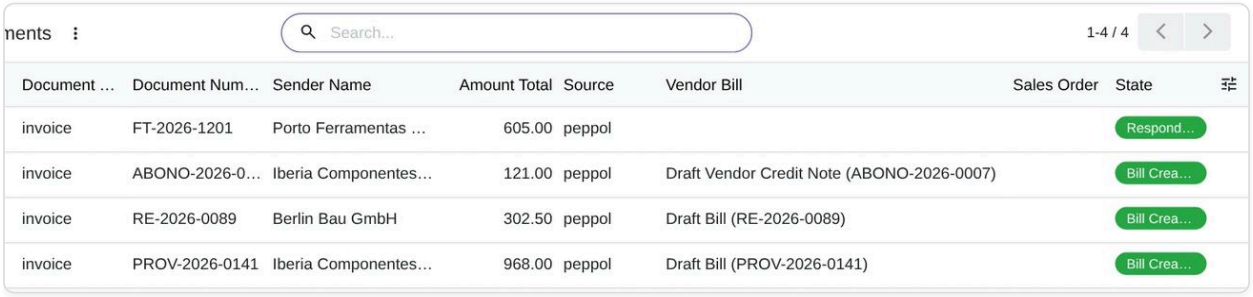
For each tax on an invoice line, the connector first checks whether there is a mapping. If there is, it uses that category. If not, it deduces it from the tax itself.

> A mapping can only force one of the seven categories in the list. The Storecove API has other, country-specific ones, such as India's tax categories (IGST, CGST and SGST), and the connector does not offer them.

11 Received Documents

Here you see the documents that Storecove has received for your legal entities and decide what to do with each one.

Open **Storecove › Documents › Received Documents**. Each row is a document: the type, the number, the sender, the amount, the origin, the vendor bill or sales order created from it, and its state.



- **Real Odoo 20 screenshot.** The **Received Documents** list with its states.

Where the documents come from

- Storecove leaves a notice in its queue for each new document, and the connector records it.
- The scheduled action **Storecove: poll webhook queue (pull mode)** checks the notice queue every 5 minutes.
- If Storecove cannot deliver a document, the connector requests it again on the next run. You can request it yourself with **Fetch**.

The document buttons

Fetch	Asks Storecove for the content of the document. Visible when the document is in **New** or **Failed**.
Create Vendor Bill	Creates a draft vendor bill. Visible when the document is **Fetched**, is not an order, and has no bill yet.
Create Sales Order	Creates the sales order. Visible only for **Fetched** orders that do not yet have a sales order.
Accept (AP)	Tells the sender that you accept the invoice. Visible when the document is **Fetched** or **Bill Created**, and is not an order.
Reject (RE)	Tells the sender that you reject the invoice, under the same conditions as the accept button.

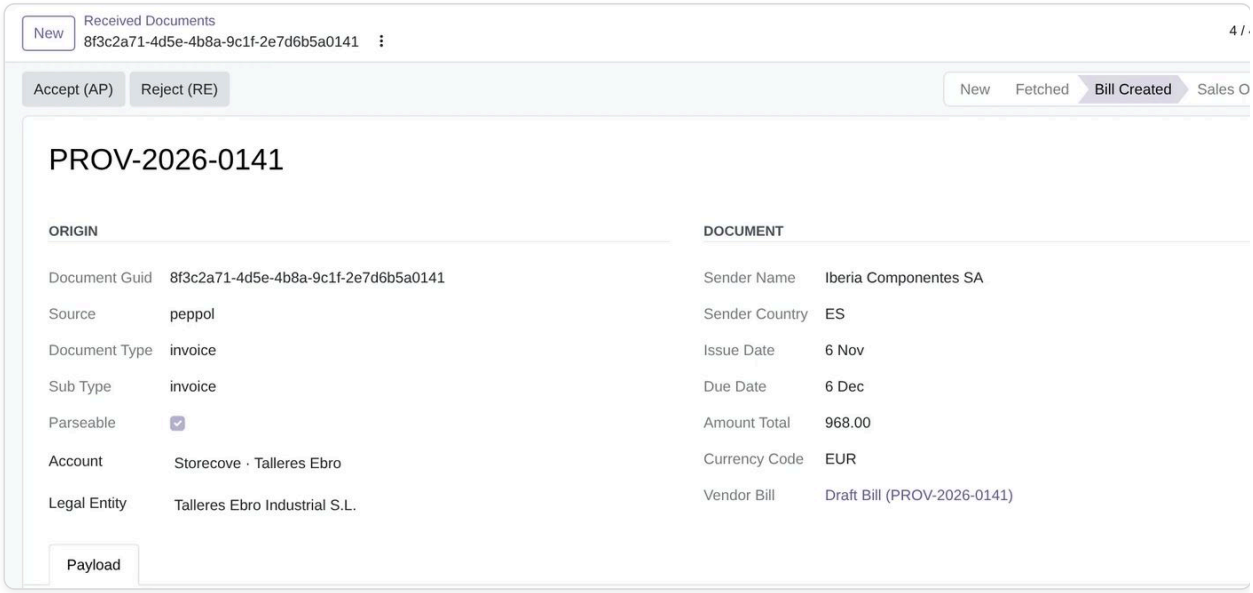
States

- **New**: Storecove has announced the document and it has not been fetched yet.
- **Fetched**: the document is already in your Odoo.
- **Bill Created** and **Sales Order Created**: the draft bill or order already exists.
- **Responded**: you have accepted or rejected the document.
- **Failed**: it could not be fetched or processed.

How to use it

1 Open the document

Check the form: **Origin** and **Document** show the sender, the dates and the amount.



- **Real Odoo 20 screenshot.** The form of a received document, with **Accept (AP)** and **Reject (RE)**.

2 Create the bill or the order

Click **Create Vendor Bill** or **Create Sales Order** if it was not created automatically.

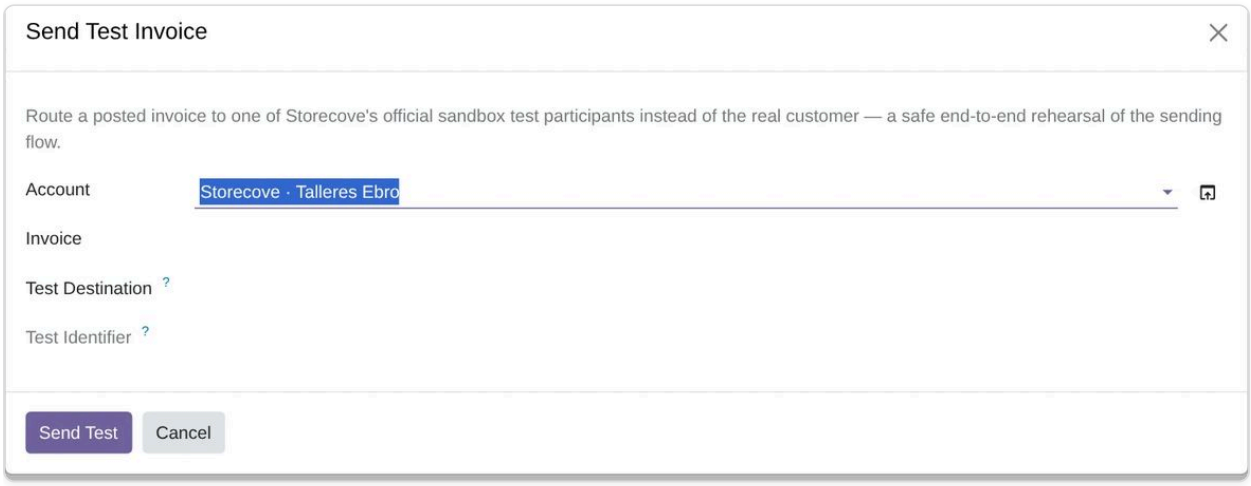
3 Respond to the sender

Click **Accept (AP)** or **Reject (RE)**. The document moves to **Responded**.

> You cannot create a second bill or a second order for the same document. Orders cannot be accepted or rejected. To create the bill or the order, the document must be fetched first.

12 Test and manual sending tools

From **Storecove › Tools** you can rehearse sending to a test participant in the sandbox, and send a document you have already generated.



- **Real Odoo 20 screenshot.** The **Send Test Invoice** wizard with the account, the invoice and the test destination.

Each field comes from what you choose in the wizard: the account, the invoice, the test destination or the file you upload. Both wizards send the document to Storecove immediately, without waiting for the scheduled action, and open the submission they create, which is also kept in **Storecove › Documents › Submissions**.

Send Test Invoice

Account	The Storecove account the invoice is sent from. By default the first connected account is shown.
Invoice	The customer invoice or credit note, already confirmed, that you want to rehearse.
Test Destination	The official sandbox test participant the invoice is routed to, instead of your real customer.
Test Identifier	The identifier of the destination you chose, as the scheme holds it.
Send Test	Sends the invoice to the test destination.
Cancel	Closes the wizard without sending anything.

1. **Open the wizard**
   Go to **Storecove › Tools › Send Test Invoice**.
2. **Choose the account and invoice**
   Fill in **Account** and **Invoice**.
3. **Choose the destination**
   Select the **Test Destination**; the **Test Identifier** fills in automatically.
4. **Send**
   Click **Send Test**.

Send an unprocessed document

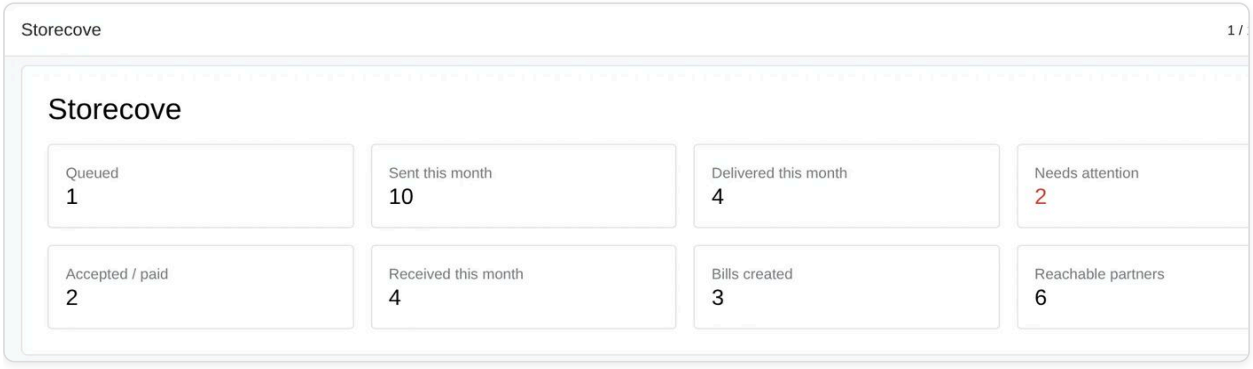
Invoice	It is only used to resolve the legal entity and the routing. What is sent comes from the file you upload: Storecove interprets it and regenerates the document.
Parse Strategy	The file format: **UBL**, **CII**, **IDOC** or **SETU14 (India e-way-bill)**. It defaults to **UBL**.
Document	The file that is sent.
Send	Sends the document.
Cancel	Closes the wizard without sending anything.

> Test invoices can only be sent from accounts in **Sandbox** or **Demo (offline simulator)**. From an account in **Production** the error "Test invoices can only be sent from sandbox or demo accounts - never from production." appears. In addition, the company on the invoice needs a legal entity registered in that account.

USER MANUAL · STORECOVE

13 Dashboard and log

The dashboard summarises Storecove's submissions and receptions, and the sync log keeps every API call and every notice received.



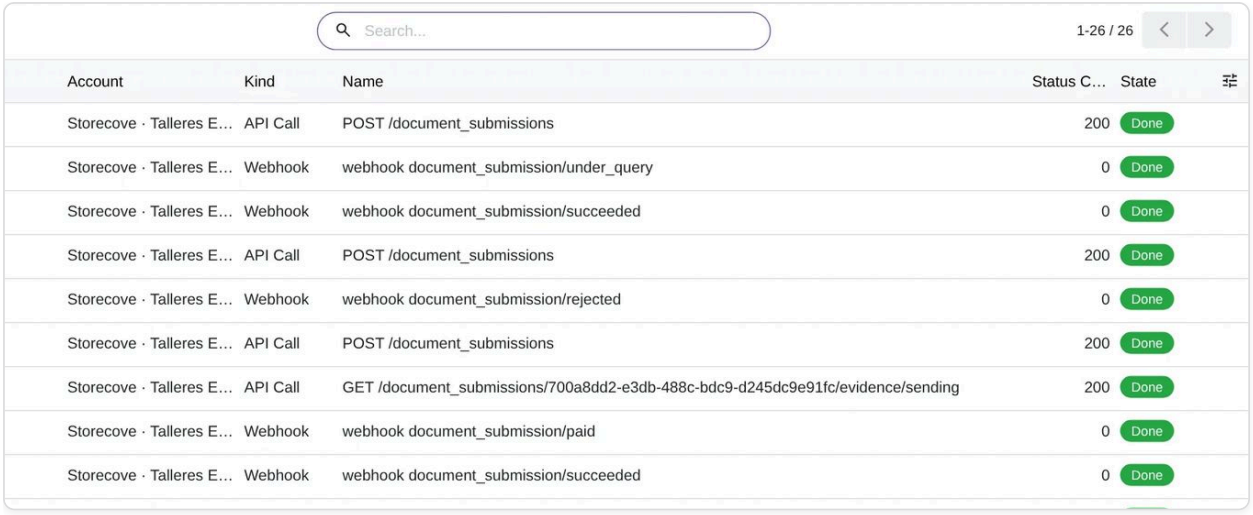
- **Real Odoo 20 screenshot.** The Storecove dashboard with its counters.

You enter the dashboard from **Storecove › Dashboard**. Each tile is a counter that opens the list behind it: **Queued**, **Sent this month**, **Delivered this month**, **Needs Attention**, **Accepted / paid**, **Received this month**, **Bills created** and **Reachable partners**.

While a connected account, a registered legal entity, a provisioned identifier or a reachable partner is missing, the dashboard does not show the counters. Instead four setup steps appear: **1. Connect your Storecove account**, **2. Register your company as a legal entity**, **3. Provision its Peppol identifiers** and **4. Verify a customer's reachability**. Each one opens the screen where that step is done.

Where the data comes from

- Two scheduled actions run every 5 minutes: **Storecove: send queued e-invoices**, which takes submissions out of the queue, and **Storecove: poll webhook queue (pull mode)**, which collects Storecove's notices.
- Each API call and each notice received is recorded in **Storecove › Configuration › Sync Log**.



- **Real Odoo 20 screenshot.** The sync log with API calls and webhooks.

The sync log

Reprocess Webhook	Reprocesses the stored body of a notice. Only visible on **Webhook** notices in **Error** state, and only for the Storecove administrators group.

Each entry has a state: **Pending**, **Done** or **Error**. In the list you can filter by **Errors**, **Webhooks** and **API Calls**, or group by **State** and **Kind**.

> A notice can only be reprocessed if it is of type **Webhook** and stores its body. Otherwise, Odoo shows "Only webhook log entries with a stored body can be reprocessed.". If you are not a Storecove manager, it shows "Only a Storecove Manager can reprocess a webhook log entry.".

USER MANUAL · STORECOVE

14 What happens automatically

The connector has two scheduled actions, which are enabled when it is installed.

ACTION	FREQUENCY	WHAT IT DOES
Storecove: send queued e-invoices	Every 5 minutes, enabled	Sends the queued submissions to Storecove.
Storecove: poll webhook queue (pull mode)	Every 5 minutes, enabled	Checks Storecove for pending notices, records them in the sync log and processes them one by one.

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

15 If something fails

The most common cases and where to look in each.

The key is not saved on the account	Odoo does not store the API key on the account. You enter it in the FlexigoTech Cloud section of the account, after ticking **I accept that my Storecove API key and my calls to Storecove go through FlexigoTech Cloud**. If it has not been saved, you get "Save the API key of … in FlexigoTech Cloud first: nothing was sent."
No FlexigoTech Cloud service	Without a trial or a subscription, the connector sends nothing. You get "Storecove has no FlexigoTech Cloud service on this database: nothing was sent, the queue waits and nothing is marked as failed or deleted. See Settings > FlexigoTech Cloud." Activate the trial or the subscription and the queue resumes by itself.
Only an administrator can do this	Accepting the consent and saving the key are administrator tasks. Other users receive a permissions notice.
The invoice does not reach Storecove	Check its submission under **Submissions**: the invoice may have an active submission, the customer's country may be missing, or a line may have more than one tax. If it was confirmed with automatic sending and has no submission, something was missing when it was confirmed: fix it and click **Send via Storecove**.
A received document does not create the vendor bill	Open the document in **Received Documents** and check its state. If a bill has already been created, another one is not created. If the purchase journal or the legal entity is missing, they must be configured.
You want to see what Storecove replied.	Check the **Sync Log**: every call and webhook is recorded with its status code.

If FlexigoTech Cloud pauses, the send waits. Nothing is marked as failed or deleted, and the **Storecove: send queued e-invoices** task retries it every 5 minutes. A received document that Storecove cannot serve stays as **New** and is requested again on the next run. If a day has passed since it arrived, it is marked as failed, and you can try again with the fetch button.

16 Support

Contact us to install it, configure it, or with any questions. On the Standard plan we reply within 1 working day; on the Pro plan, within 4 working hours, and if needed we can look at it by 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/storecove-connector-odoo.html

On Odoo Apps

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

Storecove 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).