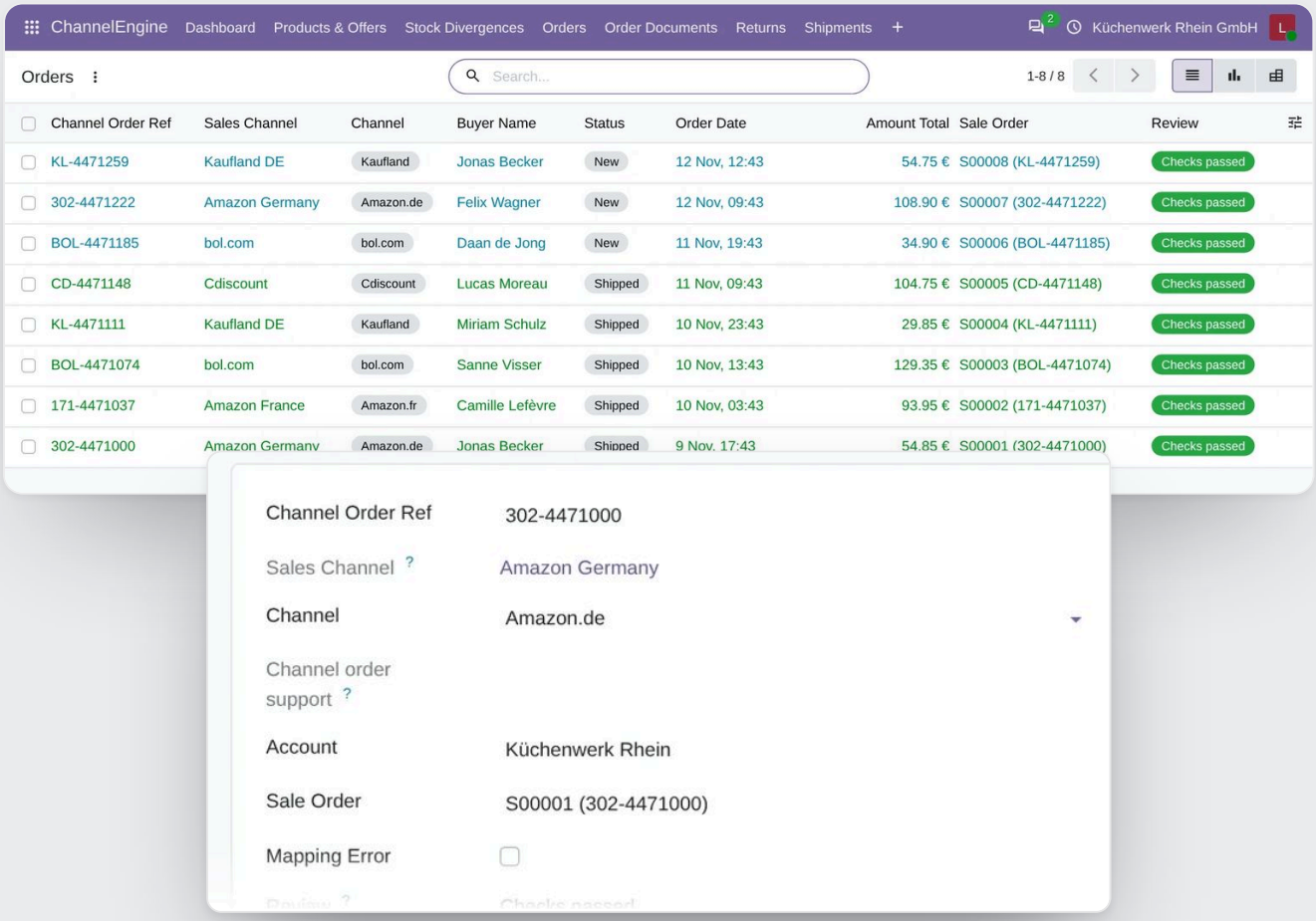
CONNECTOR GUIDE · MARKETPLACES AND E-COMMERCE

ChannelEngine connector for Odoo

Orders from the marketplaces you run in ChannelEngine land in Odoo Sales, and Odoo sends them back price, stock and shipments.

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



Real Odoo 20 screenshot. The list of ChannelEngine orders and the detail of one of them.

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 figures	4
3	Where you see it in your Odoo	4
4	What you need before you start	5
5	The FlexigoTech Cloud trial	5
6	Connecting your account step by step	7
7	The ChannelEngine channels and their synchronisation	10
8	Publishing products and offers to your channels	12
9	Pricing rules	15
10	ChannelEngine orders in Odoo	16
11	Shipments, tracking and cancellations	20
12	Returns and refunds	23
13	Stock checking against ChannelEngine	26
14	The data that goes back and forth	28
15	What runs on its own	29
16	The account options and their defaults	30
17	Security and personal data	32
18	Several accounts and several companies	33
19	If something fails	33
20	Other ways to do it	35
21	Plans and pricing	36
22	Frequently asked questions	37
23	Glossary	39
24	Checklist and contact	41

This guide describes version 20.0.6.1.11 of the module (Odoo branch 20.0, commit af29566e2). Everything in it comes from its published code, the platform's official documentation or real screenshots.

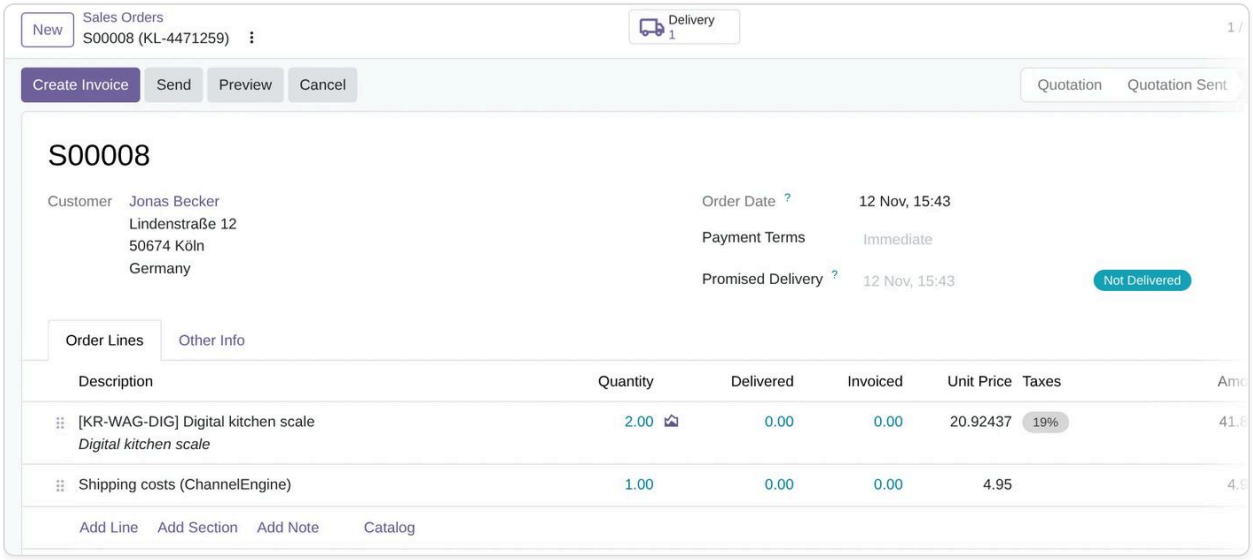
The screenshots come from a test Odoo 20 database, with a made-up company and data and no real connection to ChannelEngine. The module also has a version for Odoo 17, 18 and 19.

1 Who it is for and what it solves

> The connector needs the FlexigoTech Cloud subscription: calls to your ChannelEngine account go through it. Without the trial or the subscription, the connector does not talk to ChannelEngine.

Without the connector, someone copies orders from each channel into Odoo by hand, updates stock and prices in ChannelEngine and records shipments and returns in two places.

The connector brings orders, returns and cancellations from ChannelEngine into Odoo, and sends offers, stock and shipments from Odoo to your channels. Each connection to ChannelEngine is an account, which you add in **ChannelEngine › Configuration › Accounts**. The Standard plan supports up to 3 accounts and the Pro plan up to 10.



Real Odoo 20 screenshot. The sales order for a Kaufland order: customer, lines, tax and shipping costs.

What changes day to day

- New orders reach Odoo every 15 minutes.
- Pending price and stock are sent every 30 minutes.
- Shipments and tracking updates are sent every 15 minutes.
- Returns are imported every 2 hours.
- Every day, Odoo's stock is compared with ChannelEngine's and the differences appear under **Stock Divergences**.

What it does not do

- It does not work without FlexigoTech Cloud.
- It is not real time: orders come in every 15 minutes, price and stock go out every 30 and returns come in every 2 hours.
- It does not set up your channels in ChannelEngine or decide on which channels each product is sold: that is done in ChannelEngine.
- It does not set a different price per channel: ChannelEngine keeps one price per product.
- It does not include configuring the module for you or importing historical data.

- It does not cover migrating your database to another version of Odoo.
- The FlexigoTech Cloud subscription is paid separately from the module, and you contract your ChannelEngine account and your Odoo hosting yourself.

CONNECTOR GUIDE · CHANNELENGINE

2 The connector in figures

4
versions of Odoo (17, 18, 19 and 20)

13
scheduled tasks, all active on installation

14
own menus

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

665
automated tests

10,524
lines of code in Odoo Apps (version 20.0, 14-11-2026)

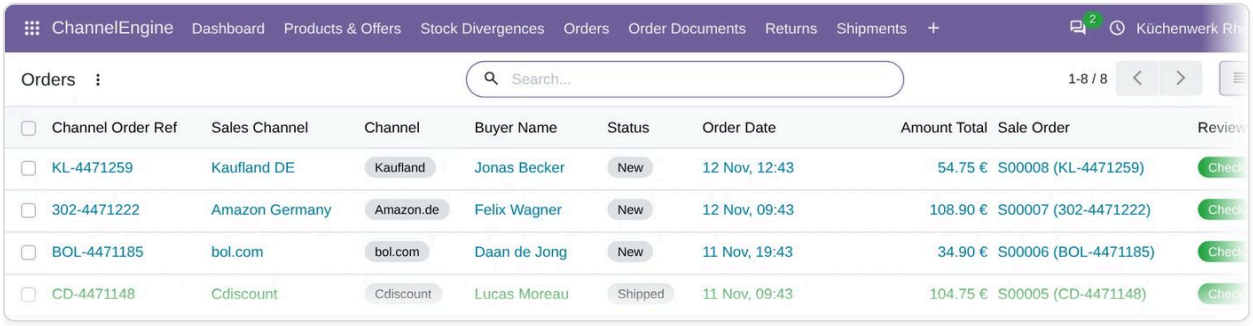
14
days of FlexigoTech Cloud trial, no card required

The connector depends on the Odoo Sales, Inventory, Invoicing and Discuss apps, on delivery with carriers, and on FlexigoTech Cloud. All of that is Odoo Community, so it works on Community and on Enterprise.

The 665 automated tests use a simulator that answers with the shapes of the ChannelEngine Merchant API. The screenshots in this guide come from an Odoo 20 with that simulator and a fictional German shop. It has not yet been tested against a real ChannelEngine tenant.

CONNECTOR GUIDE · CHANNELENGINE

3 Where you see it in your Odoo



- **Real Odoo 20 screenshot.** The **Orders** list of the ChannelEngine app, with its menus at the top.

Dashboard	The day's figures: today's and pending orders, offers with errors, pending returns, shipments to send, 30-day revenue and recent activity.
Products & Offers	The **Offers** list: each published product, with its price, stock and status.
Stock Divergences	The products whose stock in ChannelEngine does not match Odoo's.
Orders	The orders that come in from your channels, each with its sales order.

Order Documents	The invoices, labels and delivery notes that the channel attaches to an order.
Returns	The returns from marketplaces and those you record from Odoo.
Shipments	What has gone out to the buyer, with carrier and tracking.
Cancellations	Order cancellations, full or partial.
Sync Log	What each synchronisation does, with its result and errors.
Configuration	**Accounts**, **Channels** and **Pricing Rules**. Only the **ChannelEngine Admin** group sees it.

Outside its app, the connector extends Odoo's products, sales orders and warehouse moves and pickings. It also adds the **Publish Offers** and **Publish on ChannelEngine** wizards to their screens.

4 What you need before you start

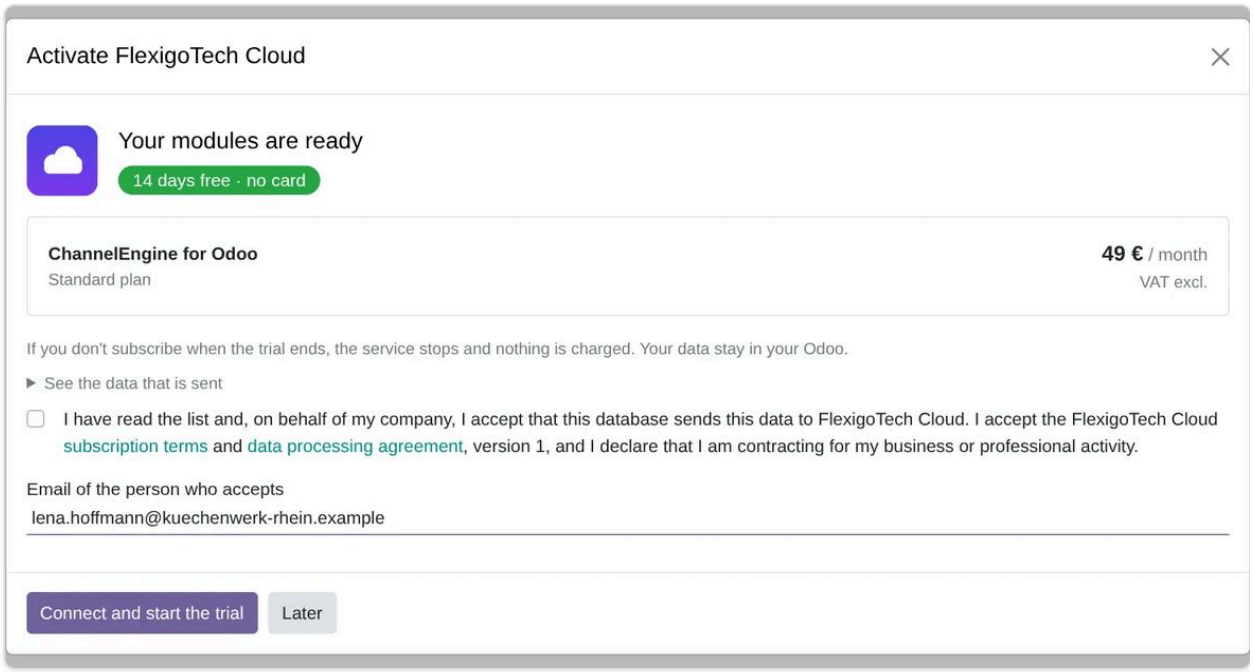
- A ChannelEngine account and a Merchant API key created in ChannelEngine, under Settings > Merchant API keys. A Channel API key will not work: that one is for marketplaces that connect to ChannelEngine, not for merchants.
- The address of your account's tenant, in the form **https://yourcompany.channelengine.net** (or the test tenant, **https://yourcompany-dev.channelengine.net**). Do not use **https://demo.channelengine.net**: that address belongs to the Channel API and rejects Merchant API keys.
- A FlexigoTech Cloud subscription or its trial. Every call to ChannelEngine goes through FlexigoTech Cloud, which stores your key encrypted and adds it to the call. Without a trial or subscription, the connector does not talk to ChannelEngine.
- 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 see and change the API key and the consent.
- A public URL for your Odoo if you want to use the webhook: ChannelEngine can only call an address reachable from the internet.

> You create the Merchant API key in ChannelEngine, and you configure the channels, their settings and which channels each product is sold on there. Odoo reads the channels that already exist in your tenant: it does not create any.

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, Invoicing and Inventory.

5 The FlexigoTech Cloud trial

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**

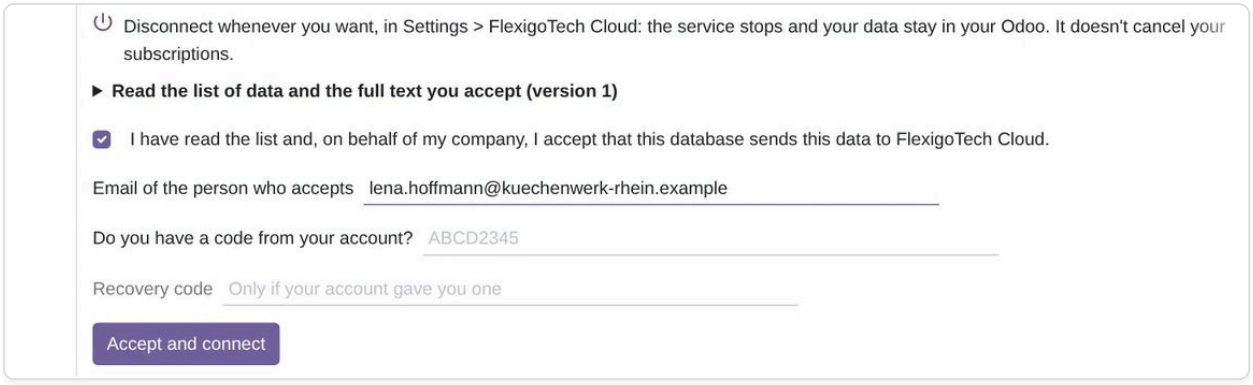
Tick "I have read the list and accept, on behalf of my company..." (it includes the subscription terms and the data processing agreement) and enter the **Email of the person who accepts**.

3 **Click Connect and start the trial**

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

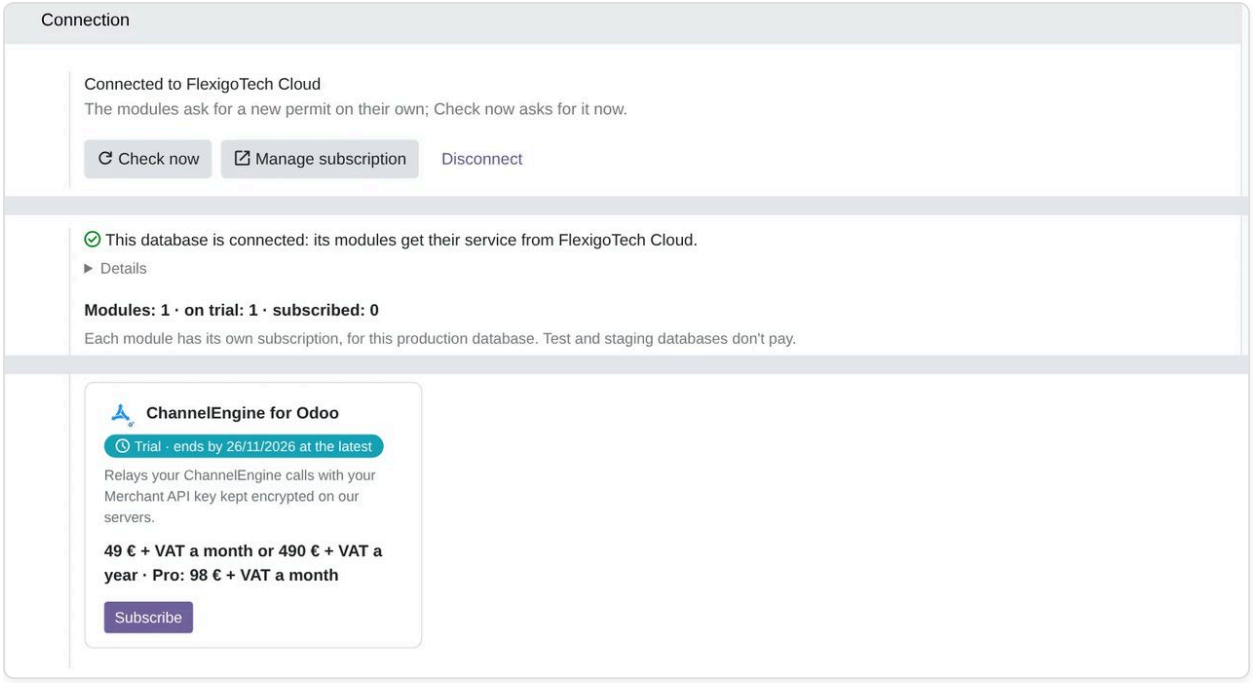
The same, from Settings

In **Settings › FlexigoTech Cloud** is the full list of what the database sends. Tick the box, enter the email and 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: **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.

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 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 can't start a trial: Settings says «This database is a neutralised copy or has demo data, so trials can't be started here.»

CONNECTOR GUIDE · CHANNELENGINE

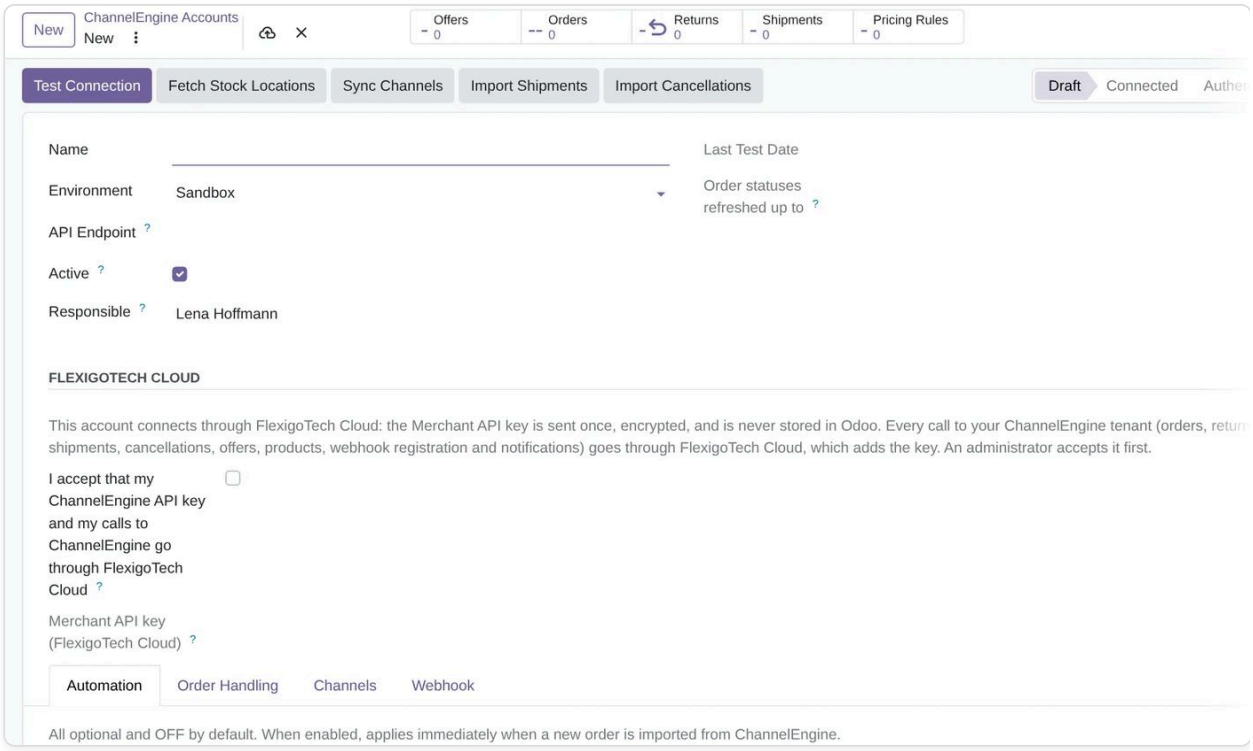
6 Connecting your account step by step

1. **Open the accounts menu**

   Go to **ChannelEngine › Configuration › Accounts**. The **ChannelEngine Accounts** list opens.

2. **Click New**

   The form for a new account opens.



- **Real Odoo 20 screenshot.** A new account: name, environment, endpoint and, in the FlexigoTech Cloud block, the consent and the key.

3 Enter the name and the tenant address

Type the account **Name** and, in **API Endpoint**, the address of your tenant, which must start with https://. Choose the **Environment**: **Sandbox** or **Production**. When you create the account it starts on **Sandbox**.

4 Accept the consent

Tick the box **I accept that my ChannelEngine API key and my calls to ChannelEngine go through FlexigoTech Cloud**. Until you tick it, the key field does not accept input.

5 Enter the Merchant API key

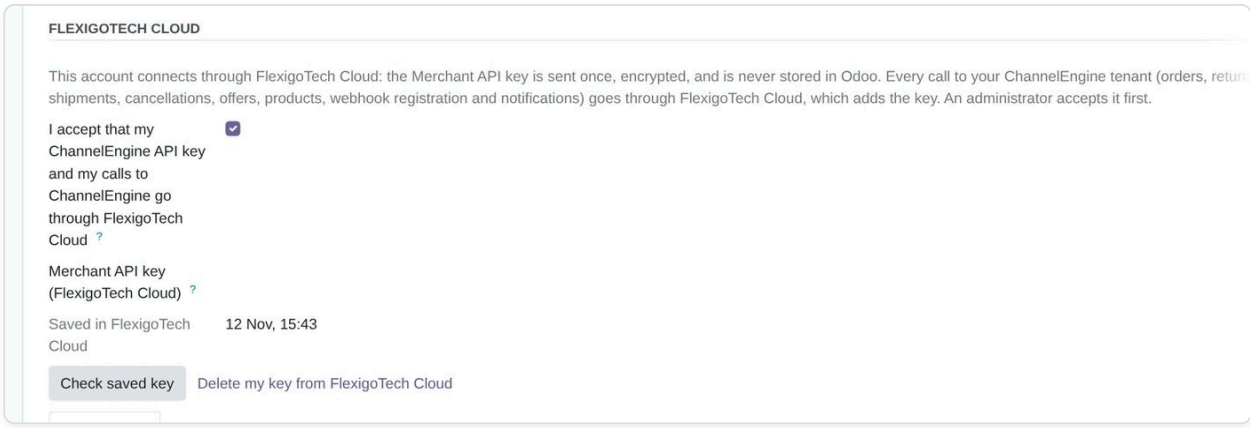
Paste the key into **Merchant API key (FlexigoTech Cloud)**. It is sent once, encrypted, to FlexigoTech Cloud and is not stored in Odoo.

6 Save the account

On saving, the key goes to FlexigoTech Cloud and **Saved in FlexigoTech Cloud** shows the date. The account starts in **Draft**.

7 Click Test Connection

Odoo reads the settings of your tenant in ChannelEngine without changing anything. If the key or the tenant is wrong, the account moves to **Authentication error**. If everything is fine, it moves to **Connected** and **Last Test Date** is filled in.



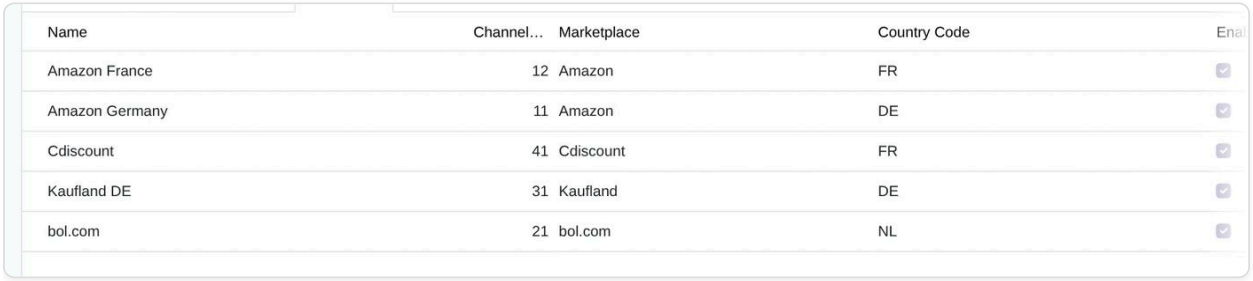
- **Real Odoo 20 screenshot.** The connected account: the key stored in FlexigoTech Cloud, with its date and its two buttons.

8 Sync the channels

Click **Sync Channels** to bring in the channels of your tenant. You see them in the **Channels** tab.

> Only an administrator can enter the key and accept the consent. If there is no consent, Odoo saves nothing and warns that nothing has been sent. To move an account to another tenant you must re-enter the key in the same save.

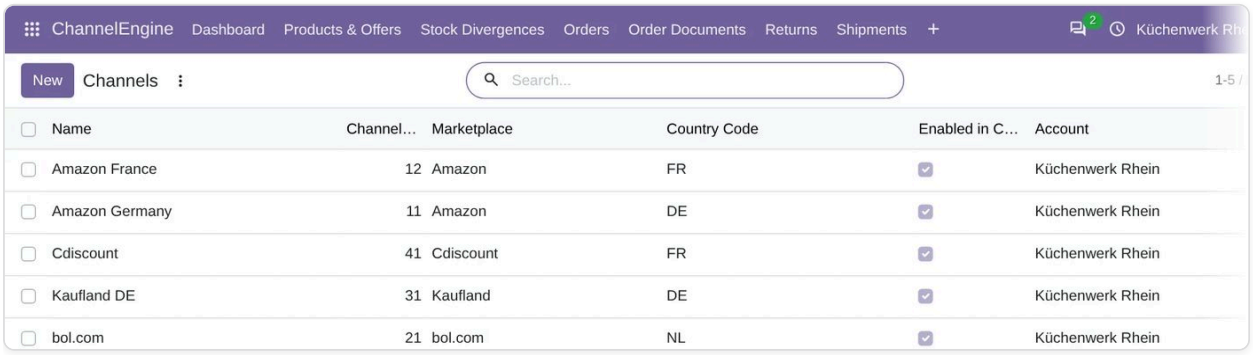
After connecting, data arrives by itself through the scheduled tasks. New orders are imported with **ChannelEngine: import new orders** every 15 minutes, order statuses are updated every 2 hours, pending price and stock are sent every 30 minutes and returns are imported every 2 hours. Channels are synchronised every day. On the account you have buttons to do it by hand, such as **Import Shipments** and **Import Cancellations**.



Name	Channel…	Marketplace	Country Code	Ena
Amazon France	12	Amazon	FR	
Amazon Germany	11	Amazon	DE	
Cdiscount	41	Cdiscount	FR	
Kaufland DE	31	Kaufland	DE	
bol.com	21	bol.com	NL	

- **Real Odoo 20 screenshot.** The **Channels** tab of the account, with the channels brought from ChannelEngine.

7 The ChannelEngine channels and their synchronisation



- **Real Odoo 20 screenshot.** The ChannelEngine channels in Odoo, each with its marketplace and its country.

You enter from **ChannelEngine › Configuration › Channels**. Channels are not created by hand. The scheduled task **ChannelEngine: sync channels** brings them from ChannelEngine every day. If an order arrives from a channel that had not yet been brought in, Odoo creates it at that moment so the order is not left without a channel.

What each channel shows

Name	The name of the channel, for example that of a marketplace together with its country.
Account	The ChannelEngine account the channel belongs to.
Marketplace	The marketplace the channel corresponds to.
Country Code and **Language Code**	The country and language of the channel.
Enabled in ChannelEngine	Checkbox showing whether the channel is enabled in ChannelEngine. It is ticked by default.
Active	Checkbox showing whether the channel is visible in Odoo. It is ticked by default.

In the list you can filter by **Enabled** and **Archived**, and group by **Account** or **Marketplace**.

> Channels are managed by the synchronisation. You can change a channel's details on its form, but the next synchronisation will reset the name, country, language and enabled status to match ChannelEngine. Synchronisation never deletes a channel: if you no longer need one, archive it.

How channels arrive

ChannelEngine returns each marketplace with the channel instances of your account inside it. Odoo goes through that response and creates or updates one channel per instance, so synchronising again does not create duplicates. The **ChannelEngine: sync channels** task does this every day.

There is a second way channels come in. If an order arrives from a channel that has not been synchronised yet, Odoo creates that channel at that moment. When synchronisation runs later, it fills in the details, because the on-the-fly creation does not overwrite a channel that already exists.

Channel states

CHECKBOX	ON	OFF
Enabled in ChannelEngine	The channel is enabled in ChannelEngine.	It is disabled in ChannelEngine.
Active	The channel appears in Odoo's lists.	The channel is archived and only appears with the **Archived** filter.

The two checkboxes are independent: a channel can be enabled in ChannelEngine and archived in Odoo, or the other way round. The **Enabled** and **Archived** filters separate each case.

Channel identifiers

ChannelEngine Channel ID	Required. This is the number ChannelEngine uses to identify this channel in this account. Orders, shipments and documents refer to their channel through it.
Global Channel ID	ChannelEngine's number for the marketplace itself, the same for all merchants.
Company	Taken from the account the channel belongs to.

Rules and conditions

- Each channel belongs to a single account, and the account is required.
- Odoo does not allow two channels with the same **ChannelEngine Channel ID** in one account; in different accounts, it does.
- Synchronisation does not delete channels: if a channel disappears from ChannelEngine or you no longer need it, archive it.

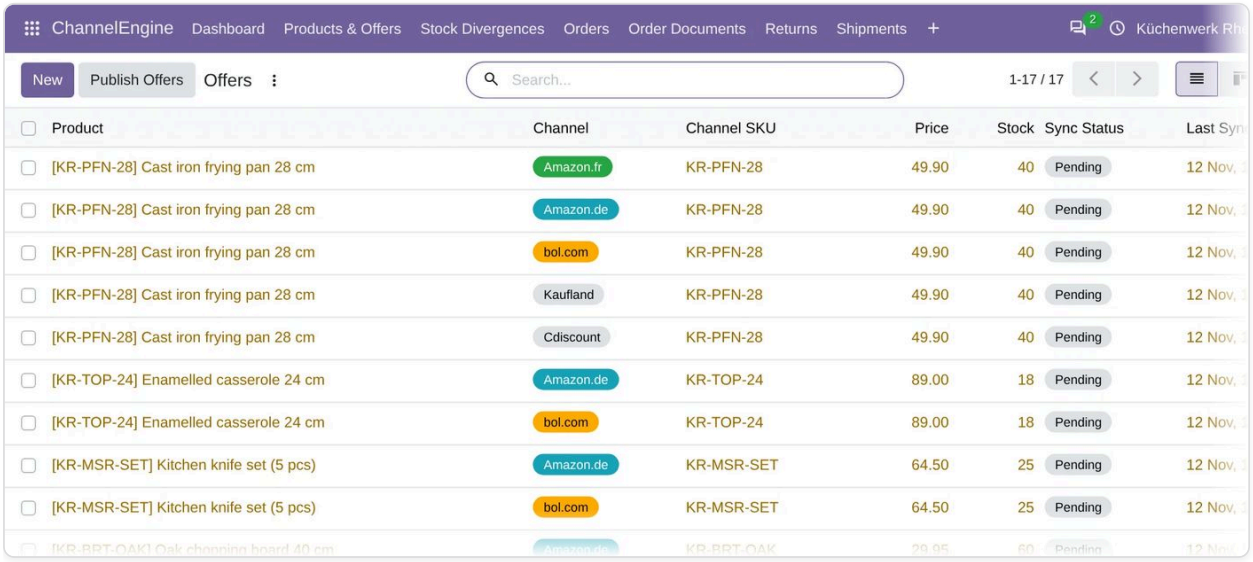
End-to-end example

1. **Channels are synchronised**
   Once a day the **ChannelEngine: sync channels** task fetches your channels, and you can see them in **ChannelEngine › Configuration › Channels**. To avoid waiting, press **Sync Channels** on the account.

2. **An order arrives**
   The **ChannelEngine: import new orders** task, which runs every 15 minutes, fetches the order and links it to its channel.

3. **You review a channel's activity**
   In **ChannelEngine › Configuration › Channels** you group by **Account** or by **Marketplace**.

8 Publishing products and offers to your channels



- **Real Odoo 20 screenshot.** The offers list: product, channel, price, stock and synchronisation status.

You enter via **ChannelEngine › Products & Offers**, which opens the **Offers** list. An offer is a product on a channel of an account. The price and stock are pushed by the scheduled task **ChannelEngine: push pending price/stock**, every 30 minutes. The **ChannelEngine: queue changed prices and stock** task checks every hour which offers have changed and queues them. You can also push manually using the buttons.

Publish Offers	Opens the wizard to create offers for several products and channels at once.
Sync Now	Pushes the price and stock of that offer without waiting for the task.
Push content + images	Pushes the product's content and images. Only visible if the account has Odoo as its content manager.
Apply Pricing Rule	Shows on the offer the price the next push will send.

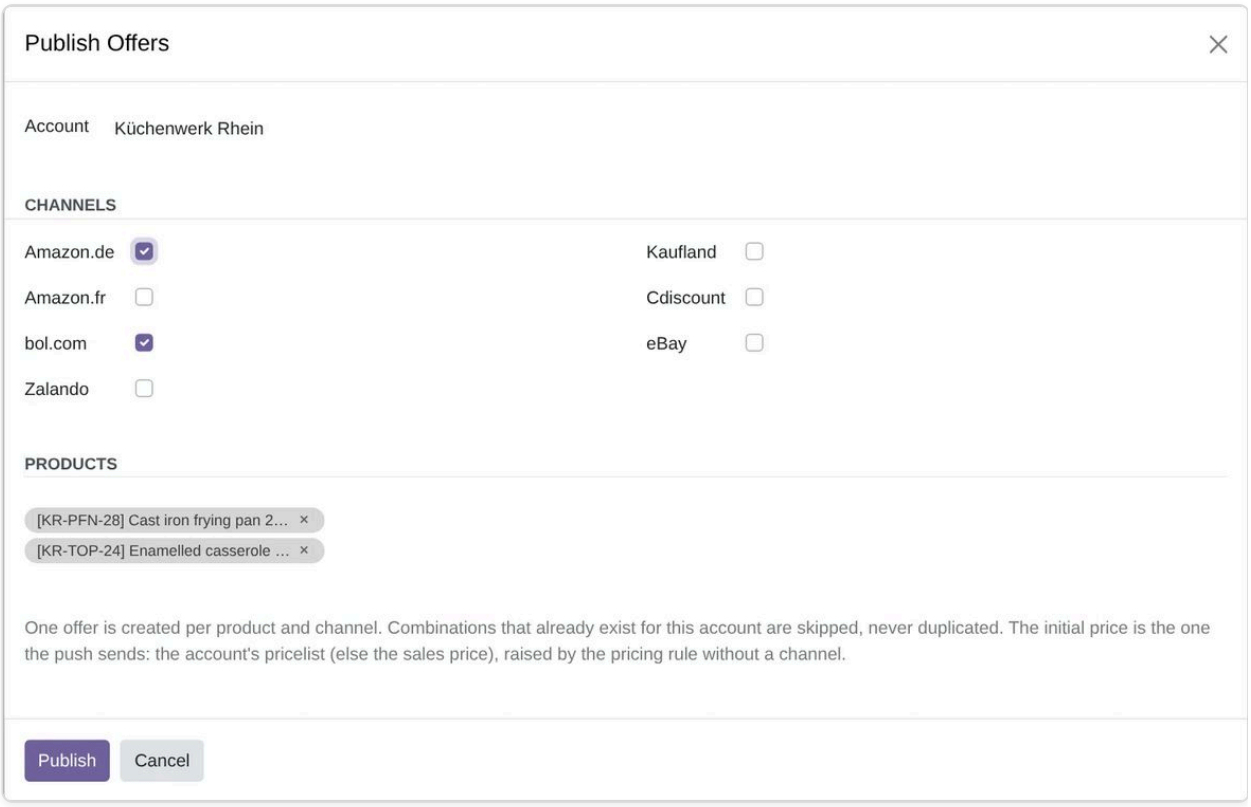
Statuses

Pending	The offer is waiting for the next push.
Synced	ChannelEngine accepted the last push.
Error	The push failed. The reason is shown in **Error Message**.

Publishing in bulk

1. **Select products**
   From the product list, open the **Publish on ChannelEngine** action, or press **Publish Offers** on the offers list.

2. **Choose account, products and channels**
   Tick the channels in the **Channels** group. You can choose Amazon.de, Amazon.fr, bol.com, Zalando, Kaufland, Cdiscount and eBay.



- **Real Odoo 20 screenshot.** The publish wizard, with the chosen products and the ticked channels.

3 **Press Publish**

One offer is created per product and channel. Those that already existed are skipped. The new ones are set to **Pending**.

> You cannot publish without choosing at least one product and one channel. Products without an internal reference are not published either. The offer's price is read-only: it is recalculated on each push.

In ChannelEngine, a product is a single offer, with one price and one stock for all its channels. Each push sends those two figures along with the product's internal reference. The channel on the Odoo offer helps you sort and filter; which channels the product appears on is decided in ChannelEngine.

What each status means

Pending	This is the initial state of every new offer. Already synchronised offers also return to it when their stock changes or an order is imported. The 30-minute task pushes pending offers first and then those in error.
Synced	ChannelEngine accepted the last send. Odoo then stores **Last Sent Price** and **Last Sent Stock**.
Error	The send failed. **Error Message** stores the reason and the task retries it on the next run.

Where the price and stock come from

The price comes from the account's price list. If it has none, it comes from the product's sales price. It is then raised to the minimum of the pricing rule that has no channel. ChannelEngine accepts only one price per

product, so a rule for a specific channel does not change the price sent: **Applicable Pricing Rule** is for information only.

The stock is the real stock in the account's warehouse, or across all warehouses of the company if none is chosen. Stock already committed is subtracted: open deliveries and orders still in quotation. It never drops below 0.

Rules and conditions

- A product with several channels is a single offer in ChannelEngine. Each push carries only price and stock, and anything not sent is left untouched.
- **Push content + images** replaces the whole product in ChannelEngine. It only works with accounts where Odoo manages the content. If the account does not manage it, Odoo refuses to push it.
- Images are sent as public links. Your Odoo must be reachable from the internet, and if the base address points to localhost, the content will not arrive.
- Only one push per account at a time. If another is running, the offer waits for the next run.
- ChannelEngine limits offer calls, so each run pushes in batches and leaves the rest for the next run.
- **Sync Now** on a large selection queues whatever does not fit into that run.

Offer fields

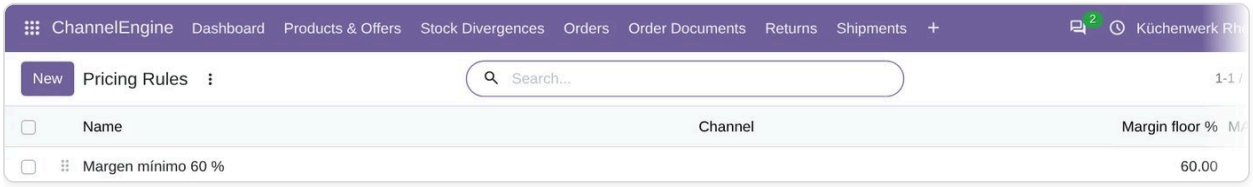
Account, **Product**, **Channel**	Required. An offer is the combination of these three.
Channel SKU	Required. The push uses the product's internal reference as the key in ChannelEngine and, if it has none, this field.
EAN	The product's barcode.
Price	Read-only. The **Apply Pricing Rule** button shows you the price that will go out.
Stock	Whole number, 0 by default.
Last Sync	Date and time of the last send.

Each product also shows the button with the number of **ChannelEngine Offers** it has.

End-to-end example

1. **Choose products**
   In the product list you select several and open **Publish on ChannelEngine**.
2. **Tick channels**
   You choose the account, tick Amazon.de and bol.com and press **Publish**. Odoo creates one offer per product and channel, and tells you how many it created and how many it skipped.
3. **Wait or force**
   Offers start in **Pending**. Within about 30 minutes the task transmits them, or you press **Sync Now**.
4. **Check**
   In **Products & Offers** you see **Synced** and the latest values sent. If you see **Error**, read **Error Message**.

9 Pricing rules



Real Odoo 20 screenshot. Pricing rules: each rule is a row in the list and you write it there.

Go to **ChannelEngine › Configuration › Pricing Rules**. Each rule is a record in the list. There is no scheduled task or button that creates them: you write them yourself, and the connector reads them each time it transmits prices.

Rule fields

Name	Name of the rule. Required.
Sequence	Order of the rules. It defaults to 10. The first one applies.
Active	Switched on by default. An inactive rule is ignored.
Channel	Leave it empty so the rule changes the price sent. A rule with a channel is saved but not applied during transmission.
Margin floor %	Minimum margin over the product cost. With 20, the price never drops below cost x 1.20. With 0, no margin floor applies.
MAP (Minimum Advertised Price)	Absolute minimum price, regardless of margin. With 0, it does not apply.
Company	The company the rule belongs to. Required, and it defaults to the active company.

Steps

1. **Open the list**

   Go to **ChannelEngine › Configuration › Pricing Rules**.

2. **Create the rule**

   Press **New**: the rule is written directly in the list. Fill in **Name** and leave **Channel** empty.

3. **Set the floor**

   Enter **Margin floor %**, **MAP (Minimum Advertised Price)**, or both.

> ChannelEngine supports a single price per product for all channels. A rule with a **Channel** does not change the price sent. Each transmission that skips such a rule notes it in the sync log.

Which rule applies

The price transmission uses a single rule: the first one, by **Sequence**, from the account's company that has no **Channel**. The other channel-less rules are not added together. To change which one takes priority, change the order in **Sequence**.

Rules with a channel

A rule with a **Channel** is saved as a reference, but the transmission does not apply it. ChannelEngine keeps one price per product for all channels. If the connector applied a channel-specific floor, it would have to raise the price everywhere. When a transmission skips such rules, it names them in the sync log.

How the floor is calculated

- With **Margin floor %** other than 0, the price does not fall below the product cost multiplied by 1 plus that percentage. With 20, the floor is cost times 1.20.
- With **MAP (Minimum Advertised Price)** other than 0, the price does not fall below that amount, whatever the margin.
- With both filled in, the floor is the higher of the two.
- With both at 0, the rule imposes no floor.

Which company each rule belongs to

Each rule carries its **Company**, and the transmission uses the rule from the account's company. That company may not be among those of the user pressing the button or of the task doing the transmission, and the rule is still read and only the floor it sets is used.

An example

To ensure no product goes below its cost plus 20% or below €15, create a rule with **Name** "General floor", **Channel** empty, **Margin floor %** 20 and **MAP (Minimum Advertised Price)** 15. For a product costing €10, the margin requires €12 and the MAP requires €15, so the MAP wins: if its selling price is €19.90, €19.90 is sent; if it is €13, €15 is sent. The next transmission sends that price to ChannelEngine. If there is also a rule tied to a channel, the sync log names it as skipped.

CONNECTOR GUIDE · CHANNELENGINE

10 ChannelEngine orders in Odoo

Orders are in **ChannelEngine › Orders**. Odoo brings in new orders with the scheduled task **ChannelEngine: import new orders**, every 15 minutes. Later changes, such as a shipment or a cancellation made on the channel, are picked up by **ChannelEngine: refresh order statuses**, every 2 hours. Each order's documents arrive with **ChannelEngine: import order documents**, every hour, or with the **Fetch Documents** button.

Each imported order creates a sale order, which you see in the **Sale Order** field of the record. The **Lines**, **Returns**, **Shipments**, **Cancellations** and **Documents** tabs hold what hangs off the order. **Raw payload** stores the order exactly as ChannelEngine sent it.

Fetch Documents	Downloads the order's documents, including ones already received, in case a file was lost.
Create Return	Opens the return wizard. Only visible if the order has a sale order.
Cancel Order	Cancels the whole order in ChannelEngine. Visible while the status is **New** or **Accepted**.
Cancel Part of Order	Cancels only some units. Visible under the same statuses.

Mark Cancellation Reviewed	Clears the warning for a cancellation that Odoo could not apply on its own and releases the deliveries. Visible if there is a cancellation to review or a pending request.
Re-check	Repeats the auto-confirmation checks. Visible if the **Review** is **To review**.
Confirm anyway	Confirms the order as it is. Visible under the same condition.

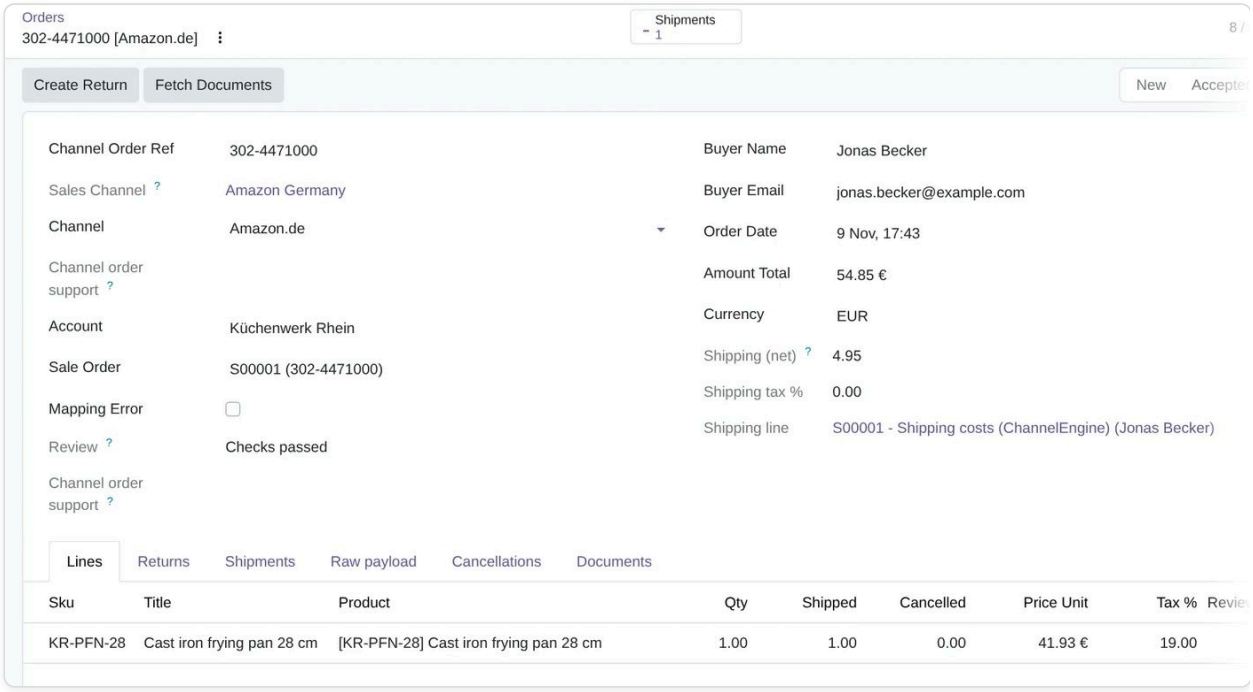
Statuses

The **Status** field can be **New**, **Accepted**, **Shipped**, **Cancelled** or **Returned**. All are list filters, as are **Mapping Error**, **To review** and **Test orders**.

If an order is left to review

1. **Open the order**

   The **Review Reason** field says which check failed.



- **Real Odoo 20 screenshot.** A ChannelEngine order: channel, buyer, amounts and its sale order.

2. **Fix the cause**

   For example, restock the item or adjust the cost price.

3. **Press Re-check**

   If it now passes the checks, Odoo confirms the order. If you prefer to accept it as it is, press **Confirm anyway**.

> A shipped or returned order cannot be cancelled from here: record a return instead. It is also not cancelled if it has a refunded return or a shipment in ChannelEngine, or if the channel does not support automated cancellations.

What each status means

New	The order has just arrived. This is the status it is created with.
Accepted	Order in progress. With the account's auto-confirmation, the sale order is confirmed.
Shipped	The order counts as shipped when every unit not cancelled is in a shipment that ChannelEngine holds. A shipped status never confirms a quotation, because delivery would open stock that has already left.
Cancelled	Cancelled on the channel or from Odoo. If units had already shipped, Odoo does not cancel the sale order behind your back: it marks the order for review and blocks validation of the open deliveries.
Returned	The buyer has returned the order. Returns are on their tab.

Auto-confirmation review

To review	A check failed and the sale order was left unconfirmed.
Checks passed	Auto-confirmation ran its checks and they passed.
Confirmed anyway	A person accepted the order as it was.

The checks look at, among other things, whether stock is sufficient, whether the account's minimum margin is met, and whether shipping charges can be invoiced. If the order carries shipping charges and the account has no shipping product, that check fails, because the order would be below what the buyer paid. When you confirm, Odoo continues with the account's invoice and delivery automation. If a step of that automation fails, the order stays imported and confirmed, and the failure is recorded as an error line in the sync log and as an activity on the order.

Cancellations

What each order supports depends on the **Channel order support** field. With **None** there is no automated cancellation and you must cancel on the marketplace. With **No split** or **Orders** only the whole order can be cancelled. With **Split orders** whole lines can be cancelled. With **Split order lines** any quantity can be cancelled. If part has already shipped and the channel only cancels whole orders, Odoo asks you to record a return for what was shipped.

When cancelling units, Odoo removes them from the sale order and, if already invoiced, generates the credit note. If the credit note fails, the cancellation stands and the order is marked for review. When the buyer requests cancellation of units, **Cancellation Request Pending** is ticked and the deliveries wait until it is processed. If ChannelEngine rejects a confirmation, it is retried on the next refresh.

Record fields

Account, **Channel** and **Sales Channel**	The account that imported it and the originating channel.
Channel Order Ref	The order reference on the channel.
Buyer Name and **Buyer Email**	Buyer details, which Odoo uses to find or create the customer and their addresses.
Order Date	Stored in UTC. If ChannelEngine does not send it, Odoo uses the import date.

Amount Total and **Currency**	The order is recorded in the currency the buyer paid in.
Shipping (net), **Shipping tax %** and **Shipping line**	Shipping charges collected from the buyer, excluding tax.
Mapping Error	Flags orders with a problem matching them to Odoo.
Test order	ChannelEngine marks it as a test. On a **Production** account it is not imported; on a **Sandbox** account, it is.

On the sale order you see the linked ChannelEngine order, the **Marketplace** and the review reason. The product does not decide the VAT: Odoo reuses a sales tax with the same percentage the channel charged, or creates one if none exists.

Documents

The marketplace attaches documents to the order, such as the channel invoice, labels or delivery notes. Odoo stores the file on the sale order and copies invoices to the Odoo customer invoice when one exists. Each document shows its **Status** (**No file yet** or **Stored**), the **Acknowledged** flag and, if ChannelEngine no longer has the file, **File No Longer Available**. The **Download** button opens the stored file. Odoo only confirms a document to ChannelEngine after its file is stored, and if that fails, it retries on the next run.

An order from start to finish

1. ### The order arrives

   Within about 15 minutes it appears in **ChannelEngine › Orders** as **New**, with its sale order. Odoo acknowledges the order to ChannelEngine only after storing it.

2. ### It is checked

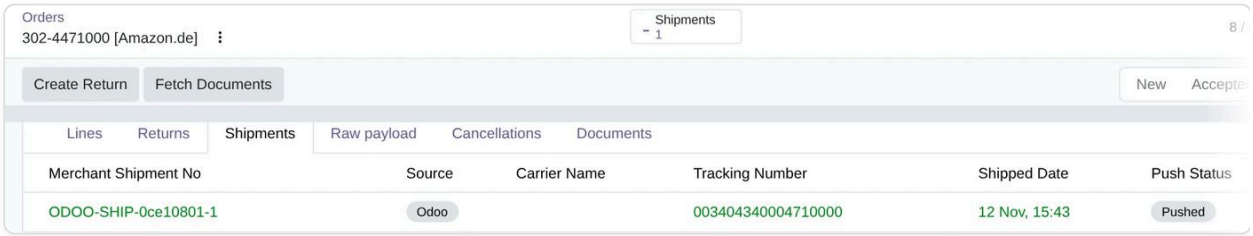
   With automatic confirmation enabled, the **Review** moves to **Checks passed** and the sale order is confirmed, or it stays **To review** with the reason.

3. ### Open the sale order

   From **Sale Order** you reach the sale order, which links back to the ChannelEngine order.

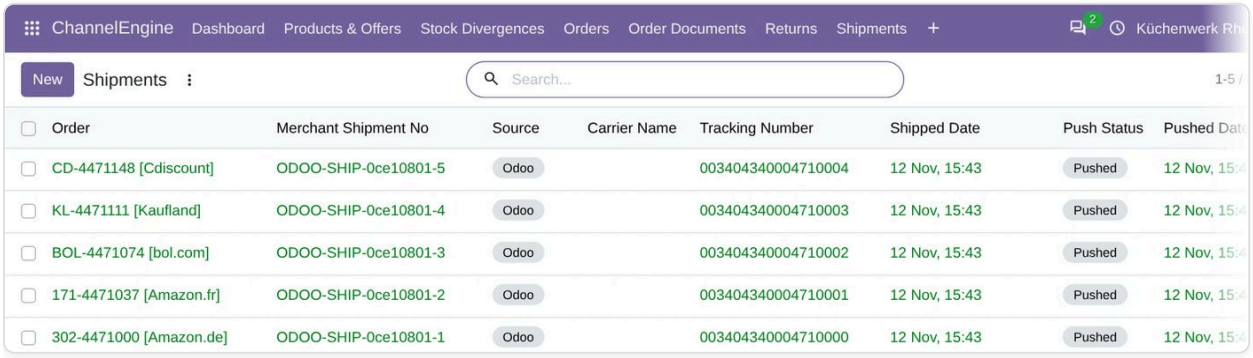
4. ### The shipment goes out

   When every uncancelled unit is in a shipment, the order moves to **Shipped**.



- **Real Odoo 20 screenshot.** An order already **Shipped** and its **Shipments** tab, with the shipment pushed.

11 Shipments, tracking and cancellations



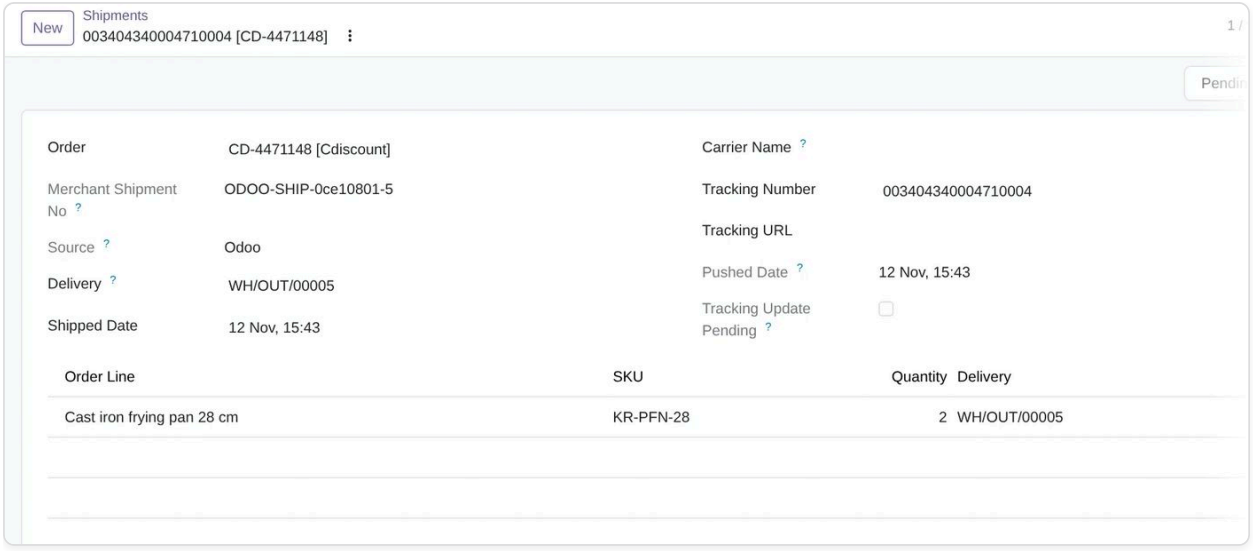
- **Real Odoo 20 screenshot.** Shipments: which order they belong to, their origin and their push status.

Where the data comes from

DATA	WHERE IT COMES FROM	WHEN
Shipments created in Odoo	Validating a delivery for a ChannelEngine order creates the shipment and pushes it straight away	On validation. The task **ChannelEngine: push shipments and tracking updates** retries what is pending every 15 minutes
Shipments created in ChannelEngine	Task **ChannelEngine: import shipments created in ChannelEngine**	Every 30 minutes
Cancellations created in ChannelEngine	Task **ChannelEngine: import cancellations created in ChannelEngine**	Every 30 minutes
Cancellations made in Odoo	**Cancel Order** or **Cancel Part of Order**, on the order	When confirmed

Shipment buttons

Push Shipment	Sends the shipment to ChannelEngine. Not shown when the shipment is already **Pushed**.
Send Tracking Update	Sends the changed carrier or tracking to ChannelEngine. Only shown on a **Pushed** shipment with the **Tracking Update Pending** box ticked.



- **Real Odoo 20 screenshot.** A pushed shipment: merchant number, delivery, carrier, tracking and push date.

Push statuses

- **Pending**: the shipment is waiting to be pushed.
- **Pushed**: ChannelEngine already has it.
- **Error**: the push failed; the **Error Message** gives the reason. These shipments wait until you press **Push Shipment**.

Cancelling part of an order

1. **Open the ChannelEngine order**
   Press **Cancel Part of Order**. The cancellation wizard opens.
2. **Choose the Reason**
   It is required and is sent to the marketplace. **Note to the marketplace** is optional.
3. **Enter the quantity on each line**
   A quantity greater than what can be cancelled is rejected.
4. **Press Cancel and Send to ChannelEngine**
   The cancellation is recorded under **Cancellations** and sent to ChannelEngine.

> What has already been shipped cannot be cancelled: for those units you must register a return. You also cannot register a shipment on a cancelled order, nor cancel without a reason or with negative quantities.

Shipment statuses

The **Push Status** goes through three values. In **Pending** the shipment exists in Odoo but ChannelEngine does not know it yet. In **Pushed** ChannelEngine accepted it, and **Pushed Date** records when. In **Error** the push failed: the **Error Message** and the **Push Attempts** are stored. If you change the carrier or tracking of an already pushed shipment, **Tracking Update Pending** is ticked, the error and attempts are cleared, and the next push sends the new values.

Cancellation statuses

Applied means the units were removed from the sale order. **Needs review** means those units had already been shipped and the quantities were not touched. The cancellation's status comes from its lines. If the sale order was already invoiced, a **Credit Note** is posted for the cancelled units.

Rules and conditions

- A shipment created by hand without **Lines** takes everything still open on the order. If nothing is left, it says "Nothing to ship: every unit of order %s is already shipped or cancelled."
- A shipment is not registered on a cancelled order: "Cannot register a shipment on cancelled ChannelEngine order %s."
- If ChannelEngine does not accept splitting the order into several shipments, the lines wait until a shipment covers the whole order.
- Tracking updates need a carrier and a tracking number. If either is missing, it says "ChannelEngine needs the carrier (Method) and the tracking number (TrackTraceNo) to update a shipment. Missing: %s."
- If the carrier or tracking is unknown when pushing, the shipment is sent without them and they are added later.
- When validating a delivery, if the buyer asked to cancel something, that cancellation must be processed first. With the order cancelled, the delivery is not validated: "ChannelEngine order %s is cancelled: do not ship the rest of it. Reduce or cancel this delivery; the order's cancellation review says what already left."
- Shipments created in ChannelEngine may validate the open delivery of the order in Odoo, if the account allows it. Failures show as "None of the shipped products is open on delivery %s." or "Delivery %s could not be validated."

Shipment fields

Order	The ChannelEngine order it belongs to. Required.
Origin	**Odoo** if it was created by validating a delivery here. **ChannelEngine** if it was created there and the connector read it afterwards.
Merchant Shipment No	The unique reference by which ChannelEngine knows the shipment. Tracking updates are sent with it.
Delivery	The validated delivery the shipment comes from, or the one the connector validated for a shipment created in ChannelEngine.
Carrier Name, **Tracking Number** and **Tracking URL**	Taken from the delivery if it has a carrier. The name is shown on the marketplace. Overly long texts are cut and an overly long URL is omitted.
Lines	What this shipment carries: the quantities that actually left the warehouse.

On cancellations, **Origin** distinguishes **Odoo**, **ChannelEngine** (requested by the buyer or the marketplace) and **Merchant, in ChannelEngine** (created in the ChannelEngine portal). The **Created by the channel** filter separates those the merchant did not create.

An example from start to finish

1. **An order arrives with two lines**
   You see it in **ChannelEngine › Orders**; the import task brings it in every 15 minutes.

2 **You validate the delivery**

A **Pending** shipment is created with the quantities that left and is pushed when validation finishes.

3 **You change the tracking**

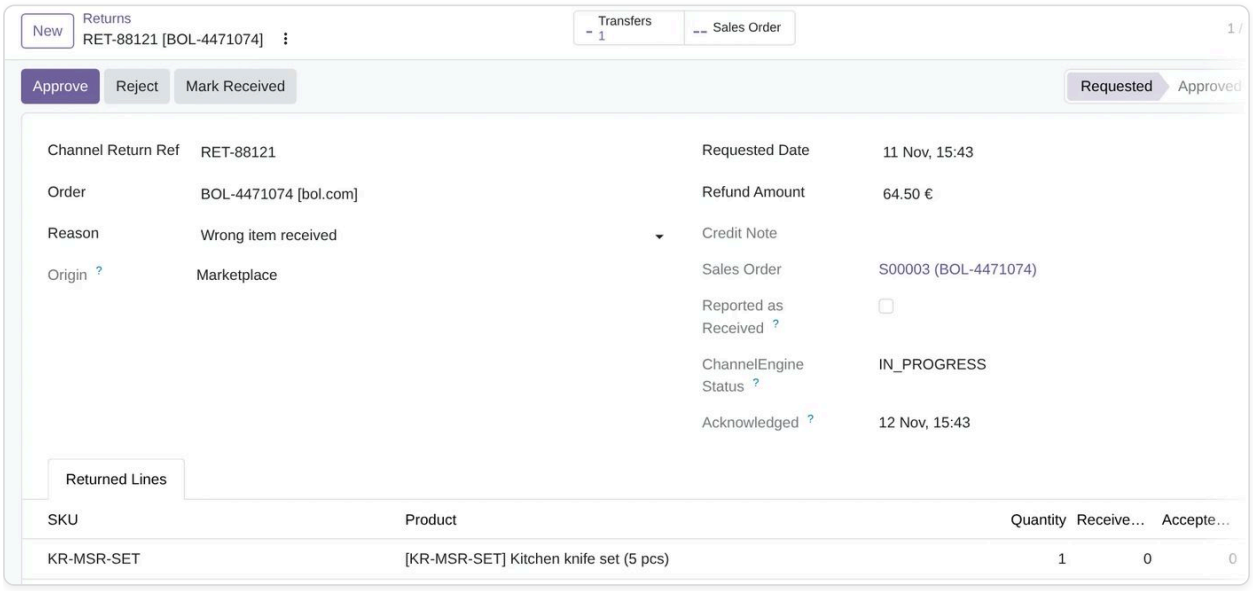
If you later change the carrier or tracking number on the delivery, the already **Pushed** shipment is marked **Tracking Update Pending**. The 15-minute task sends it, or you press **Send Tracking Update**.

4 **You cancel an unshipped line**

With the wizard you choose the **Reason**, enter the quantity and press **Cancel and Send to ChannelEngine**. It becomes **Applied** under **Cancellations**.

CONNECTOR GUIDE · CHANNELENGINE

12 Returns and refunds



- **Real Odoo 20 screenshot.** A bol.com return in **Requested**, with **Approve**, **Reject** and **Mark Received** at the top.

Marketplace returns arrive through the scheduled task **ChannelEngine: import returns**, which runs every 2 hours. Those you start yourself are created when you confirm the return wizard from an order.

Record buttons

Approve	Approves the return and creates the credit note in draft. Only shown while the return is **Requested**.
Confirm Refund	Confirms the credit note and marks the return as refunded. Only shown while the return is **Approved**.
Reject	Rejects the return. Shown while the return is **Requested** or **Approved**.
Mark Received	Tells ChannelEngine the goods have arrived. Only shown on marketplace returns with a ChannelEngine ID that have not yet been notified.
Create Return Transfer	Creates in Odoo the incoming transfer for the returned goods. Not shown if transfers already exist or the return is rejected.
Credit Note	Opens the credit note. Only shown if it exists.

Sale Order	Opens the return's sale order, if it has one.

Statuses

The **Status** can be **Requested**, **Approved**, **Rejected** or **Refunded**. A new return enters as **Requested**.

Starting a return from Odoo

1. **Open the order and press Create Return**
   The wizard opens with the order already filled in.
2. **Fill in the details**
   Enter the **Return Number**, **Reason**, an optional **Comment** and the quantity for each line.
3. **Press Create Return**
   It is registered in ChannelEngine first and then in Odoo.

> Only marketplace returns are reported as received, and ChannelEngine accepts that notification just once per return. A return started in Odoo cannot be marked as received.

Statuses and what they mean

Requested	The return has just come in and nobody has dealt with it yet. This is the initial status.
Approved	You have approved the return, and the draft credit note exists if the order had an invoice.
Rejected	The return has been rejected. It no longer allows creating the return transfer.
Refunded	The credit note is confirmed and the cycle ends.

If ChannelEngine changes the status of a return you have already read, it only moves forward. An approval, rejection or refund made in Odoo is not undone.

Rules and conditions

- A return can only be saved against an order that already exists in Odoo. If the order has not been imported, the return is not confirmed to ChannelEngine and it arrives again once the order exists.
- The return transfer needs a validated delivery. If there is none, the warning "Return %s: the order has no validated delivery in Odoo, so there is nothing to receive back yet." is shown.
- If the order went out in several transfers, one return transfer is created for each source delivery, and each line is limited to what can still be returned.
- If everything has already been returned, the message "Return %s: everything on it was already returned in Odoo." is shown.
- To mark as received, the accepted plus rejected quantities of each line must add up to the returned quantity; otherwise the warning with the SKU and the three figures is shown.
- Returned quantities cannot be negative, and accepted plus rejected cannot exceed the returned quantity.

Receipt and credit note

When the return is marked as received, ChannelEngine receives the accepted and rejected quantities for each product. If nobody has set them, they are filled in automatically: with a validated transfer, what arrived is accepted and the rest is rejected; without a transfer, everything is accepted. When the last transfer of the return is validated, Odoo notifies it as received without blocking the warehouse; if that fails, the **Mark Received** button remains available.

The credit note is created as a draft from the original invoice and only keeps the returned products. It counts the accepted quantity if the outcome is already known and, if not, the returned quantity. If there is nothing to refund, none is created. A confirmed credit note is not touched; a draft one is rebuilt with the accepted quantities.

Record fields and tabs

Order and **Order Line**	The order is required. The line is optional: if empty, the return covers the whole order.
Channel Return Ref	The reference used by the channel. It is not repeated within the same account.
ChannelEngine Return Id	The identifier in ChannelEngine itself, needed for the acknowledgement and for marking it as received.
Reason	**Damaged in transit**, **Wrong item received**, **Not as described**, **No longer needed** or **Other**.
Origin	**Marketplace** or **Started in Odoo**.
ChannelEngine Status	The status the return had in ChannelEngine at the last read.
Acknowledged	When ChannelEngine was told the return is registered. It is sent only once.
Reported as Received	Checkbox showing that receipt has already been reported.
Refund Amount, **Currency**	Amount of the return and its currency.
Credit Note, **Sale Order**, **Picking Count**	Links to what already exists in Odoo.
Returned Lines tab	The lines, with the accepted and rejected quantities.

In the list you can filter by **Requested**, **Approved**, **Refunded** and **Rejected**, and group by **Status** or by **Reason**.

End-to-end example

1. **The return arrives**
   The task **ChannelEngine: import returns** brings it in and it sits at **Requested**, with an acknowledgement.
2. **Press Create Return Transfer**
   Odoo prepares the goods receipt against the original delivery.
3. **Validate the transfer in the warehouse**
   On validation, Odoo notifies ChannelEngine of what was received.
4. **Press Approve**
   The credit note is created as a draft, with the accepted quantities.

5 **Press Confirm Refund**

The credit note is confirmed and the return moves to **Refunded**.

CONNECTOR GUIDE · CHANNELENGINE

13 Stock checking against ChannelEngine

The list is at **ChannelEngine › Stock Divergences**. Each row is a product of an account with its **SKU**, the **ChannelEngine stock**, the **Odoo stock**, the **Difference** and the time of the last check in **Checked At**. The **Difference** is the ChannelEngine stock minus the Odoo stock.

The **Odoo stock** is what Odoo would send now: the real warehouse stock of the account, minus open deliveries and orders still waiting as quotations.

Where the data comes from

The scheduled task **ChannelEngine: compare stock with ChannelEngine** reads the stock ChannelEngine holds each day and compares it with Odoo's. Each check replaces the previous divergences for the account. If one account fails, the others are still checked.

List buttons

Check stock now	Runs the check for all active accounts without waiting for the daily task. If there is an error, you see it on screen.
Requeue divergent	Queues the offers of the selected divergences for the next sync. If you select none, it queues all of them.
ChannelEngine shows more	Filter: only products where ChannelEngine has more stock than Odoo.
ChannelEngine shows less	Filter: only products where ChannelEngine has less stock than Odoo.
Account	Filter by account.

How to fix a divergence

1 **Open the list**

Go to **ChannelEngine › Stock Divergences** and review the rows.

2 **Filter if needed**

Use **ChannelEngine shows more**, **ChannelEngine shows less** or **Account**.

3 **Select the rows**

Tick the ones you want to fix, or none to handle them all.

4 **Press Requeue divergent**

Those offers are sent to ChannelEngine in the next sync.

> The list only shows the comparison: it does not allow creating or editing rows. A divergence is fixed by sending the stock again.

What the check is for

A stock update can leave Odoo and never arrive. As there is no other way to know, the connector reads back the stock ChannelEngine holds and compares it with what Odoo would send now. Each product that differs is saved in the list so you can see it and requeue it with one click.

Which stock is read from ChannelEngine

The connector reads the stock for the whole account. If the account has a ChannelEngine stock location configured, only that one is read, because it is the one the connector's updates write. Without a configured location, all the rows of each SKU are summed, which is the total ChannelEngine exports to the channels.

How each row is interpreted

SITUATION	WHAT IT MEANS
Positive Difference	ChannelEngine has more stock than Odoo. Seen with the **ChannelEngine shows more** filter.
Negative Difference	ChannelEngine has less stock than Odoo. Seen with the **ChannelEngine shows less** filter.
Product not listed	No divergence: both stocks match.

Rules and conditions

- Each check replaces the divergences the account had. Old checks do not accumulate.
- If ChannelEngine fails during the check of an account, the previous result for that account is kept.
- **Check stock now** goes through all active accounts and, if anything fails, shows the error on screen.
- **Requeue divergent** acts on the selected rows and, with no selection, on all of them.

Data in each row

Account	The ChannelEngine account the row belongs to.
Product	The Odoo product.
SKU	The code by which ChannelEngine knows the product.
ChannelEngine stock	What ChannelEngine has on record.
Odoo stock	What Odoo would send right now.
Difference	ChannelEngine stock minus Odoo stock.
Checked At	Date and time of the check.

End-to-end example

An example with round figures: a product has 10 units in stock in Odoo and there is an open order for 2, so Odoo would send 8. ChannelEngine has 12 on record. The daily check creates a row with **ChannelEngine stock** 12, **Odoo stock** 8 and **Difference** 4, and it appears with the filter **ChannelEngine shows more**. You select that row and press **Requeue divergent**. The product joins the send queue and is pushed on the next run of **ChannelEngine: push pending price/stock**, which runs every 30 minutes. On the following day's check, if both stock figures match, the row no longer appears.

14 The data that goes back and forth

DATA	DIRECTION	WHEN
New orders	From ChannelEngine to Odoo	**ChannelEngine: import new orders**, every 15 minutes
Status changes for orders already imported	From ChannelEngine to Odoo	**ChannelEngine: refresh order statuses**, every 2 hours
Returns	From ChannelEngine to Odoo	**ChannelEngine: import returns**, every 2 hours
Order documents (channel invoices, labels, delivery notes)	From ChannelEngine to Odoo	**ChannelEngine: import order documents**, every hour, or **Fetch Documents** on the order
ChannelEngine notices	From ChannelEngine to Odoo	**ChannelEngine: import notifications**, every 4 hours
Channels	From ChannelEngine to Odoo	**ChannelEngine: sync channels**, daily, or **Sync Channels** on the account
Shipments created in ChannelEngine	From ChannelEngine to Odoo	**ChannelEngine: import shipments created in ChannelEngine**, every 30 minutes, or **Import Shipments**
Cancellations created in ChannelEngine	From ChannelEngine to Odoo	**ChannelEngine: import cancellations created in ChannelEngine**, every 30 minutes, or **Import Cancellations**
Stock held by ChannelEngine, to compare with Odoo's	From ChannelEngine to Odoo	**ChannelEngine: compare stock with ChannelEngine**, daily, or **Check stock now**
Price and stock of the offers	From Odoo to ChannelEngine	**ChannelEngine: push pending price/stock**, every 30 minutes, or **Sync Now**
Product content and images	From Odoo to ChannelEngine	**Push content + images**, only on accounts where Odoo manages the content
Shipments and tracking	From Odoo to ChannelEngine	**ChannelEngine: push shipments and tracking updates**, every 15 minutes, or when the delivery is validated; **Push Shipment** and **Send Tracking Update** by hand
Cancellations made in Odoo	From Odoo to ChannelEngine	**Cancel and Send to ChannelEngine**
Returns created from Odoo and returns received	From Odoo to ChannelEngine	**Create Return** on the order; when the return receipt is validated

Calls to ChannelEngine do not leave your Odoo directly. They go through FlexigoTech Cloud, which forwards each call and stores your Merchant API key encrypted on its servers: Odoo does not store it. Without the FlexigoTech Cloud subscription, or without the trial, the connector does not talk to ChannelEngine. If the cloud

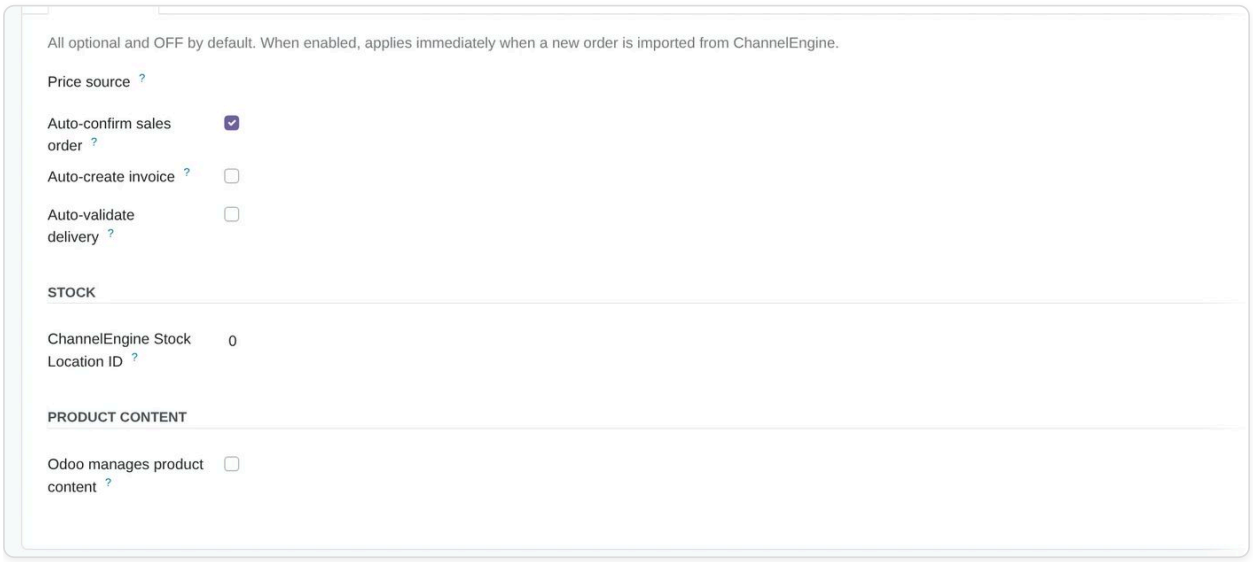
pauses, nothing is marked as failed, nothing is disconnected and nothing is deleted: what is pending waits for the next run.

15 What runs on its own

TASK	FREQUENCY	WHAT IT DOES
ChannelEngine: import new orders	Every 15 minutes	Fetches new orders and confirms them to ChannelEngine once they are saved.
ChannelEngine: refresh order statuses	Every 2 hours	Re-reads already imported orders to pick up status changes.
ChannelEngine: push pending price/stock	Every 30 minutes	Sends the prices and stock of the queued offers and retries the ones that failed.
ChannelEngine: import returns	Every 2 hours	Fetches new returns and the changes to ones already known.
ChannelEngine: clean up old sync logs	Every day	Deletes sync log lines older than 90 days.
ChannelEngine: import notifications	Every 4 hours	Copies ChannelEngine notices to the sync log, each one only once.
ChannelEngine: sync channels	Every day	Updates the list of channels for each active account.
ChannelEngine: import order documents	Every hour	Downloads the documents the channel attaches to an order and saves them.
ChannelEngine: queue changed prices and stock	Every hour	Queues the offers whose current price or stock differs from the last one sent.
ChannelEngine: compare stock with ChannelEngine	Every day	Reads the stock held by ChannelEngine and records the differences from the stock in Odoo.
ChannelEngine: push shipments and tracking updates	Every 15 minutes	Sends new shipments and tracking changes, and retries those that failed for a temporary reason.
ChannelEngine: import shipments created in ChannelEngine	Every 30 minutes	Fetches shipments created in ChannelEngine that Odoo had not generated.
ChannelEngine: import cancellations created in ChannelEngine	Every 30 minutes	Fetches cancellations created in ChannelEngine and applies them to the order.

You change the frequencies in **Settings › Technical › Automation › Scheduled Actions**, with developer mode on.

16 The account options and their defaults

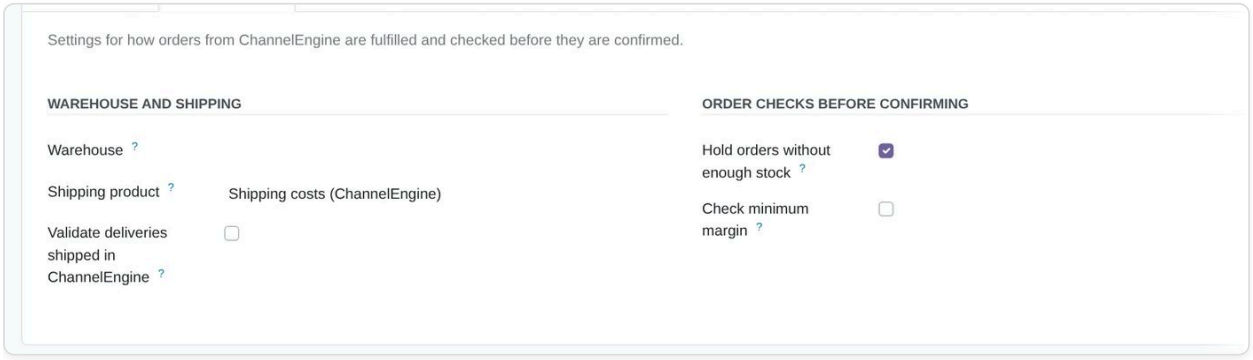


- **Real Odoo 20 screenshot.** The **Automation** tab of an account, with the checkboxes that decide what happens on import.

OPTION	WHEN CREATED	WHAT IT DOES
Active	On	If you switch it off, the account stops making calls to ChannelEngine immediately.
Price source	Empty	If you choose a pricelist, the offers take their price from it. If empty, they use the product's sales price.
Responsible	The user who creates the account	Receives an activity each time a sync of the account logs an error.
Warehouse	Empty	Warehouse that ships the account's orders and whose stock is sent to ChannelEngine. It must belong to the account's company.
Shipping product	Empty	Service product used on the shipping cost line of the sales orders.
ChannelEngine Stock Location ID	Empty	Only if your ChannelEngine account has several stock locations. Leave it empty if it doesn't. The **Fetch Stock Locations** button reads the locations of the account.
Auto-confirm sales order	Off	Confirms the sales order when the ChannelEngine order is imported.
Auto-create invoice	Off	Creates and posts the customer invoice on import. The sales order must be confirmed first.
Auto-validate delivery	Off	Validates the outgoing delivery on import. Requires the sales order to be confirmed.
Hold orders without enough stock	On	If the warehouse has no stock for all lines, the order is not confirmed automatically and is left for review.
Check minimum margin	Off	If an order's margin falls below the minimum, it is not confirmed automatically and is left for review.
Minimum margin (%)	Empty	Minimum margin over cost, as a percentage, when the check is on.

OPTION	WHEN CREATED	WHAT IT DOES
Margin basis	**Before marketplace fees**	Choose whether the margin is measured before or after marketplace fees: **Before marketplace fees** or **After marketplace fees**.
Validate deliveries shipped in ChannelEngine	Off	If ChannelEngine reports a shipment created outside Odoo, validates the corresponding delivery.
Odoo manages product content	Off	Turn it on only if titles, descriptions and images are maintained in Odoo. It allows content and images to be pushed from the offers.

Pushing product content replaces the whole product in ChannelEngine: any field Odoo does not send is cleared. That is why it is off by default and should only be turned on if Odoo is where that content lives.



- **Real Odoo 20 screenshot.** The **Order Handling** tab: the warehouse, the shipping product and the checks before confirmation.

The webhook

Scheduled tasks are enough for everything to arrive. If your Odoo is reachable from the internet, on the **Webhook** tab you can press **Register Webhook**: ChannelEngine notifies your Odoo when an order, a return or a shipment changes, or when there is a new notice, and Odoo runs the relevant task straight away instead of waiting for its scheduled time. The notification carries no data: the task requests it from ChannelEngine through FlexigoTech Cloud, as always. **Remove Webhook** deletes it and **Regenerate Secret** changes the secret part of its address.

What is done by hand

TAB OR GROUP	BUTTONS
Account header	**Test Connection**, **Fetch Stock Locations**, **Sync Channels**, **Import Shipments**, **Import Cancellations**
Webhook	**Register Webhook**, **Remove Webhook**, **Regenerate Secret**
FlexigoTech Cloud	**Check saved key**, **Delete my key from FlexigoTech Cloud** (administrators only)

Webhook	Stores the account's **Webhook URL** and the **Webhook Secret**, which is generated automatically and cannot be edited. **Regenerate Secret** creates a new one if you think it has leaked. **Register Webhook** tells ChannelEngine to call your Odoo, but it does not work if your Odoo address is not public.
Channels	Lists the account's channels, which **Sync Channels** fetches from ChannelEngine.

FlexigoTech Cloud	The account connects through FlexigoTech Cloud: the Merchant API key is sent encrypted only once and Odoo does not store it. To send it, you must first tick the consent checkbox, and only an administrator can do that. **Saved in FlexigoTech Cloud** shows when it was saved.
Status and environment	The **Status** can be **Draft**, **Connected** or **Authentication error**, and it is not editable. The **Environment** is **Sandbox** or **Production**. The **API Endpoint** must be an https:// URL.

17 Security and personal data

- The account connects through FlexigoTech Cloud: the Merchant API key is sent once, encrypted, and FlexigoTech Cloud stores it encrypted. Odoo keeps no copy.
- After saving it, the account only shows the date in **Saved in FlexigoTech Cloud**. With **Check saved key** you confirm what FlexigoTech Cloud holds, without seeing the key, and with **Delete my key from FlexigoTech Cloud** you remove it from there.
- Consent is given by an Odoo administrator with the checkbox **I accept that my ChannelEngine API key and my calls to ChannelEngine go through FlexigoTech Cloud**. If it is not ticked, Odoo shows the error "Before saving the Merchant API key, an administrator has to accept that the key and the calls to ChannelEngine go through FlexigoTech Cloud. Nothing was sent." and sends nothing.
- The key, the consent checkbox and the two buttons above are only used by Odoo administrators. Only an administrator can delete the key.
- The module provides three user groups, each including the previous one: **ChannelEngine User** is read-only; **ChannelEngine Manager** also works with offers, orders, returns, shipments, cancellations, channels and rules, and uses the wizards; **ChannelEngine Admin** also creates and changes accounts and sees **Configuration**. In a multi-company database, each company sees only its own.
- Every call from the account goes through FlexigoTech Cloud to your ChannelEngine tenant: orders, returns, shipments, cancellations, offers, products, webhook registration and notifications, with the data each call carries.
- The connector stores in Odoo the data of the orders it imports, including the buyer's details and their shipping and invoicing addresses, as any sales order does. It does not delete them on its own.
- When the subscription or trial ends, or you disconnect the database, FlexigoTech Cloud deletes your ChannelEngine key and the data of that connection within 30 days at most, or sooner if ChannelEngine asks. The module listing on Odoo Apps states this.

Activity log

Each sync run is recorded in the sync log, with its result. If a run ends in error, the connector opens an activity for the account's responsible user, so the failure shows up in their Odoo activities list. A warning is logged without opening an activity. Every week, the **ChannelEngine: clean up old sync logs** task deletes lines older than 90 days.

18 Several accounts and several companies

Each account in **ChannelEngine › Configuration › Accounts** works with its own company and its own options. Scheduled tasks go through the active accounts one by one, each in its company. If one account fails, for example because of a rejected key, it is logged and the others carry on.

Each FlexigoTech Cloud plan allows a number of accounts per connector: up to 3 with the Standard plan and up to 10 with the Pro plan.

ChannelEngine orders remain available to all accounts of the same tenant until one of them confirms them. For that reason the connector checks whether another account of the same tenant has already saved the order, in any company, and the account of the matching company picks it up instead of creating it twice.

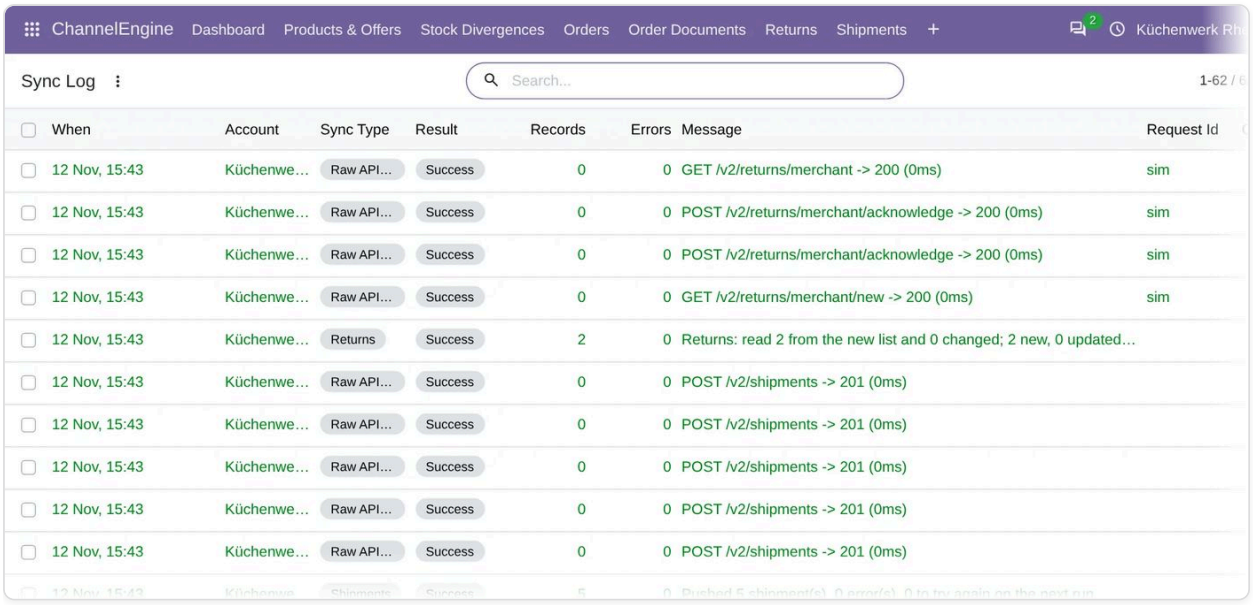
Pricing rules apply according to the account's company.

19 If something fails

The account cannot reach ChannelEngine and nothing is sent	Every call goes through FlexigoTech Cloud. Without a trial or subscription, the connector does not talk to ChannelEngine: scheduled tasks leave pending items as they are and do not mark anything as failed. Activate the trial or subscription and run the action again.
"Before saving the Merchant API key, an administrator has to accept that the key and the calls to ChannelEngine go through FlexigoTech Cloud. Nothing was sent."	An administrator must tick the **I accept that my ChannelEngine API key and my calls to ChannelEngine go through FlexigoTech Cloud** checkbox and save the key again.
"The Merchant API key of %s is not saved in FlexigoTech Cloud: type it again."	Enter the key again in **Merchant API key (FlexigoTech Cloud)**. If **Saved in FlexigoTech Cloud** is empty, the key is not saved.
"ChannelEngine authentication failed (% (status)s) on account '% (account)s'. Check that the API Endpoint is your own tenant (https://<your-account>.channelengine.net) and that the key is a Merchant API key created in ChannelEngine under Settings > Merchant API keys."	Check the account's endpoint and that the key is a Merchant API key created in ChannelEngine, under Settings > Merchant API keys.
An order, return or shipment is not created	Open the account's sync log. Every error is recorded there and an activity is scheduled for the account's responsible user, so also check their pending activities.
You want to know what ChannelEngine replied	ChannelEngine error messages are shown in Odoo with their text: "ChannelEngine client error %(code)s: %(body)s" and "ChannelEngine rejected %(method)s %(path)s: %(message)s%(details)s". That is where the platform's response is.

If FlexigoTech Cloud is paused, the connector does not mark anything as failed, does not disconnect the account and does not delete anything. Scheduled tasks only go through accounts that can make calls, and

whatever was left pending is picked up on the next run.



- **Real Odoo 20 screenshot.** The **Sync Log**: each sync with its account, its type, its result and its message.

Without a subscription, or when uninstalling

Without a trial or subscription no call goes out to ChannelEngine: no orders come in and no prices, stock or shipments go out. Nothing is marked as failed or deleted, and everything already in Odoo (orders, sales orders, shipments and returns) can still be viewed. When uninstalling the module, Odoo asks FlexigoTech Cloud to delete the key for this database; if it does not respond at that moment, the module is uninstalled anyway.

Other notices you may see

"This Odoo's base URL is not reachable from the internet (%s), so ChannelEngine could never deliver a webhook to it. Set a public 'web.base.url' first."	Your Odoo must be reachable from the internet to receive ChannelEngine notifications. Set a public web address for the database and try again.
"API Endpoint must be an https:// URL."	Enter your tenant's endpoint starting with https://.
«ChannelEngine rate limit reached on %(method)s %(path)s: it asks to wait %(seconds)s seconds before the next call. Nothing was lost: the next scheduled run continues where this one stopped.»	ChannelEngine limits calls. Nothing is lost: the next scheduled run continues where this one stopped.
"ChannelEngine returned 404 for %(method)s %(path)s. The endpoint or the tenant does not exist. Check the API Endpoint on account '%(account)s'."	Check the account's endpoint: it must be that of your own tenant.

«ChannelEngine account '%s' is inactive (kill switch). Re-enable it before syncing.»	The account is archived. Reactivate it and sync again.
"ChannelEngine order %s is already cancelled."	Nothing to do: the order is already cancelled.
"ChannelEngine order %(ref)s is %(status)s and can no longer be cancelled. Register a return instead."	If the order has already been shipped or returned, cancellation is not possible. Register a return.
«The channel of order %s does not support automated cancellations (ChannelEngine order support: NONE). Cancel it on the marketplace.»	That channel does not allow cancelling from ChannelEngine. Cancel the order on the marketplace.
«Select at least one channel to publish to.»	Under **Publish**, tick at least one channel before you confirm.
«To move %(name)s to another ChannelEngine tenant, type its Merchant API key again in the same save: the one saved in FlexigoTech Cloud belongs to %(host)s.»	If you change an account's endpoint, also enter the Merchant API key for the new tenant in the same save.

CONNECTOR GUIDE · CHANNELENGINE

20 Other ways to do it

METHOD	WHAT IT INVOLVES
Manually from the ChannelEngine dashboard	Someone logs into ChannelEngine, checks the new orders, copies them into Odoo, records shipments and returns, and keeps checking that stock and prices match in both places.
Odoo alone	In the Odoo 20 code, Community and Enterprise, there is no module with "channelengine" in its name and no manifest that names it (search carried out 13-11-2026). Odoo on its own has no integration with ChannelEngine.
This connector	It brings ChannelEngine orders, returns and shipments into Odoo and sends your offers, with their price and stock, to your channels. It requires a FlexigoTech Cloud subscription, which stores your Merchant API key encrypted on its servers and forwards your calls to ChannelEngine. There are two plans: Standard, with up to 3 accounts, and Pro, with up to 10.

If you already connect ChannelEngine with another tool: the connector imports the orders that ChannelEngine has not yet had acknowledged and acknowledges each one individually as soon as it saves it. Orders the other tool has already received do not come back, and there is no option to start from a given date. If both tools run at the same time, each new order is taken by whichever picks it up first: switch the other one off before connecting the account.

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 does not require a card. Standard covers up to 3 accounts; Pro covers up to 10.

22 Frequently asked questions

Does it work without FlexigoTech Cloud?

No. Calls to ChannelEngine go through FlexigoTech Cloud, so without the trial or the subscription the connector cannot talk to ChannelEngine.

Does it work on Odoo Online?

No. Odoo Online does not allow installing third-party modules.

Does it need Odoo Enterprise?

No. Everything it requires comes from Odoo Community: Sales, Inventory, Invoicing, Discuss and delivery with carriers, plus FlexigoTech Cloud.

Which Odoo versions does it work with?

Published branches exist for Odoo 17, 18, 19 and 20.

How many ChannelEngine accounts can I connect?

Up to 3 on the Standard plan and up to 10 on the Pro plan. Each connection to ChannelEngine is one account.

How much does the subscription cost?

The Standard plan costs €49 a month or €490 a year, and Pro costs €98 a month or €980 a year, excluding VAT.

What is the difference in support between the plans?

With Standard, the first reply arrives within 1 working day. With Pro, within 4 working hours, and a video call is included if needed.

Can I try it before paying for the subscription?

Yes: FlexigoTech Cloud has a 14-day trial with no card required. You buy the module license first, and without the trial or the subscription the connector cannot talk to ChannelEngine.

What key do I need from ChannelEngine?

A Merchant API key, created in ChannelEngine. A Channel API key will not work, because it is for marketplaces.

Where is my Merchant API key stored?

FlexigoTech Cloud stores it encrypted on its servers. Odoo keeps no copy of it.

Who can set up the key?

Only an administrator, who must also accept that the key and the calls to ChannelEngine pass through FlexigoTech Cloud.

Can I delete my key from FlexigoTech Cloud?

Yes, with the **Delete my key from FlexigoTech Cloud** button on the account. **Check saved key** lets you confirm what is kept, without viewing the key.

How often do new orders arrive?

Every 15 minutes, with the task **ChannelEngine: import new orders**.

How often are order statuses updated?

Every 2 hours, with **ChannelEngine: refresh order statuses**.

How often are price and stock sent?

Every 30 minutes, whatever is pending is sent and offers that returned an error are retried.

How often do returns arrive?

Every 2 hours, with the task **ChannelEngine: import returns**.

How often are shipments sent?

A shipment is sent when the delivery is validated, and the shipments task repeats whatever is pending every 15 minutes.

How often is stock checked?

Every day, Odoo's stock is compared with ChannelEngine's and the differences appear under **Stock Divergences**.

Can I change the task frequencies?

Yes, under **Settings › Technical › Automation › Scheduled Actions**, with developer mode activated.

What happens to orders that existed before I installed it?

Those that ChannelEngine still has not had acknowledged come in. Orders another tool already received do not come back, and there is no option to import from a given date.

Does it import historical data or set up the module for me?

No. Module setup and the import of historical data are not included.

Do I create the channels myself?

No. **ChannelEngine: sync channels** brings them in from ChannelEngine every day. Odoo only reads the channels that already exist in your account.

How do I publish products to the channels?

From the product list, with **Publish on ChannelEngine**, or with **Publish Offers** on the offers list. One offer is created per product and channel.

Can products without a reference be published?

No. Products without an internal reference are not published, and you cannot publish without choosing at least one product and one channel.

Can I set a minimum price?

Yes, with the pricing rules: **Margin floor %** and **MAP (Minimum Advertised Price)**. Only the first rule without a channel counts.

Can I have a different price on each channel?

No. ChannelEngine accepts a single price per product, so a rule with a channel does not change the price sent.

What stock is sent to ChannelEngine?

The actual stock in the account's warehouse, minus open deliveries and orders that are still quotations. It never drops below 0.

Are orders confirmed automatically?

It depends on the account's settings. With automatic confirmation, Odoo runs some checks and, if any fail, the order is set to **To review** until you approve it.

Can I cancel orders from Odoo?

Yes, the whole order or just some units, while the order is **New** or **Accepted**. Anything already shipped is not cancelled: a return is recorded instead.

Can I handle returns?

Yes. They arrive from the marketplace or you start them yourself from the order, and you approve, reject and refund them from the return record.

Does it generate credit notes?

Yes. If the order has an invoice, approving the return creates the draft credit note, and confirming the refund confirms it.

What happens if an account fails?

It is logged and the other accounts carry on. The failure opens an activity for the account's owner.

What happens if FlexigoTech Cloud is paused?

Nothing is marked as failed, the account is not disconnected and nothing is deleted. Pending items wait for the next run.

Does my Odoo need to be reachable from the internet?

Only if you use the webhook or send images from Odoo, which go out as public links. Otherwise it does not: the scheduled tasks request data from ChannelEngine through FlexigoTech Cloud.

CONNECTOR GUIDE · CHANNELENGINE

23 Glossary

ChannelEngine	A platform that connects a merchant to many marketplaces and shops from a single account.
Marketplace	A third-party online store where you sell your products, such as Amazon or bol.com.
Channel	A specific marketplace or shop in your tenant, with its country and language. Orders and offers are grouped by channel.
Tenant	Your space within ChannelEngine, with its own address of the form https://yourcompany.channelengine.net.

Account	Each connection between Odoo and ChannelEngine. It is set up under **ChannelEngine › Configuration › Accounts**.
Merchant API	The ChannelEngine interface intended for merchants. The key the connector needs is for this API, not the Channel API.
Merchant API key	The key you create in ChannelEngine and paste into the account so the connector can talk to your tenant.
API endpoint	The address of your tenant in ChannelEngine. It must start with https://.
FlexigoTech Cloud	The FlexigoTech service through which the connector's calls to ChannelEngine pass. Its trial or subscription is required.
Standard plan and Pro plan	The two FlexigoTech Cloud plans for this connector: up to 3 accounts with Standard and up to 10 with Pro.
Offer	A product published on a channel of an account. It carries its price and stock.
Channel SKU	Offer field holding its reference in ChannelEngine. The transmission previously used the product's internal reference.
Pricing rule	Rule that sets a floor on the price sent: a margin over cost, an advertised minimum price, or both.
MAP	Minimum advertised price: the amount below which the price sent will not drop.
Shipment	Record of what has been sent to the buyer, with carrier and tracking. It is transmitted to ChannelEngine.
Return	A buyer's request to return goods. It goes through **Requested**, **Approved**, **Rejected** and **Refunded**.
Cancellation	Withdrawal of units from an order, requested by the buyer, by the marketplace or by you.
Stock divergence	Product whose stock in ChannelEngine does not match what Odoo would send now. They are listed under **Stock Divergences**.
Synchronisation log	List where each run of the connector is recorded with its result.
Scheduled action	Odoo process that runs on its own at a fixed frequency. The connector ships with 13, all active on installation.

24 Checklist and contact

- [ ] You have a ChannelEngine account and a Merchant API key created there, not a Channel API one.
- [ ] You know your tenant address, in the form **https://yourcompany.channelengine.net**, and it is not the demo one.
- [ ] Your Odoo is version 17, 18, 19 or 20, on Odoo.sh or on your own server (not Odoo Online).
- [ ] You have installed the module and can see the **ChannelEngine** menu.
- [ ] You have activated the FlexigoTech Cloud trial or have the subscription.
- [ ] You have created the account under **ChannelEngine › Configuration › Accounts** with its endpoint and key, and an administrator has accepted the consent.
- [ ] The channels appear under **ChannelEngine › Configuration › Channels**.
- [ ] You have reviewed the account options and the pricing rules.
- [ ] An order has arrived in **ChannelEngine › Orders** with its sales order.
- [ ] A published offer has reached the **Synced** status under **ChannelEngine › Products & Offers**.

If you would like us to set it up with you, write to us. 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/channelengine-connector.html

On Odoo Apps

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

ChannelEngine 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).