Integrations – Qargo API – What, why & how

Learn how to connect Qargo to external systems for orders, load updates, master data, and accounting, including API setup, credentials, testing, and troubleshooting.

Updated November 3, 2026

💬 Email success@qargo.com to start an integration request or to enable any prerequisite configuration.

Introduction

What is the Qargo API?

- A set of connections that let other systems send data to Qargo and receive data from it.
- It covers order capture and statuses, load operations with contract carriers, master data sync, and accounting.

Why use it?

- It removes double entry between Qargo and the other systems a transportation company works with, such as customer systems, contract carrier systems, accounting packages, and fleet or HR tools.
- Data stays up to date in both systems without manual imports.

How does it work?

- Qargo assigns an *integrator ID* to the party building the connection.
- A *Super Admin* creates an *API client* for the right *role* and shares the credentials securely.
- Qargo completes any setup needed on the account. The integration is then tested and goes live.

Prerequisites

- **Super Admin** rights in Qargo.
- An *integrator ID* issued by Qargo (see *Step 1* below).
- For the *Customer* role: **Customer portal access** turned on for the relevant customer.
- For Accounting clients that post invoices: an Accounting API integration set up and linked to the correct *billing entity*.

Integration types

There are three integration types. The type determines how the integration is set up.

- **Tenant**: a direct connection between Qargo and a third-party system used by the transportation company, for example an accounting package.
- **Customer**: a connection from a customer's own system into Qargo, for example to submit orders through the customer portal.
- **Contract carrier**: a connection from a contract carrier's own system into Qargo, for example to send load updates from its own TMS back to Qargo.

Use cases

- A customer wants to send orders straight from its own system into Qargo instead of typing them into the customer portal.
- A contract carrier wants to receive load assignments and send stop updates and PODs back from its own TMS.
- A new driver or vehicle added in an external fleet or HR system should appear automatically in Qargo resources.

- Finance wants invoices and credit memos to flow into the accounting package, with payment statuses sent back to Qargo.
- A group with several legal entities wants each entity's accounting system to post only its own invoices.

Terminology

Term	Definition
API	A way for two systems to exchange data automatically.
Tenant	The Qargo account itself, meaning the transportation company using Qargo.
Integration type	Who the connection is set up for: the tenant, one of its customers, or one of its contract carriers.
API client	A set of login details (a client ID and a *secret*) that lets an external system connect to Qargo for one *role*.
Role	The type of access an API client has: Order management, Customer, Master data management, or Accounting.
Integrator ID	A unique ID from Qargo that identifies who is building the integration, so the right person can be contacted if an issue arises.
Secret	The private key shown once when an API client is created. It cannot be viewed again.
Master data	Core records such as customers, contract carriers, drivers, and vehicles.
Billing entity	The legal or internal company in the group that issues an invoice.
POD	Proof of delivery: the signed document or photo confirming a delivery.
Order Export	An endpoint that returns the full data of an existing order, used as a template for building new order requests.
Order import configuration	Account-level settings, managed by Qargo, that control how orders received over the API enter Qargo.
API check	The first step of an order import. Qargo checks that the message is built correctly (payload validation).
Qargo import	The second step of an order import. Qargo applies the data to the account (tenant validation).

Setting up an API connection

Step 1: Request the integration and an integrator ID

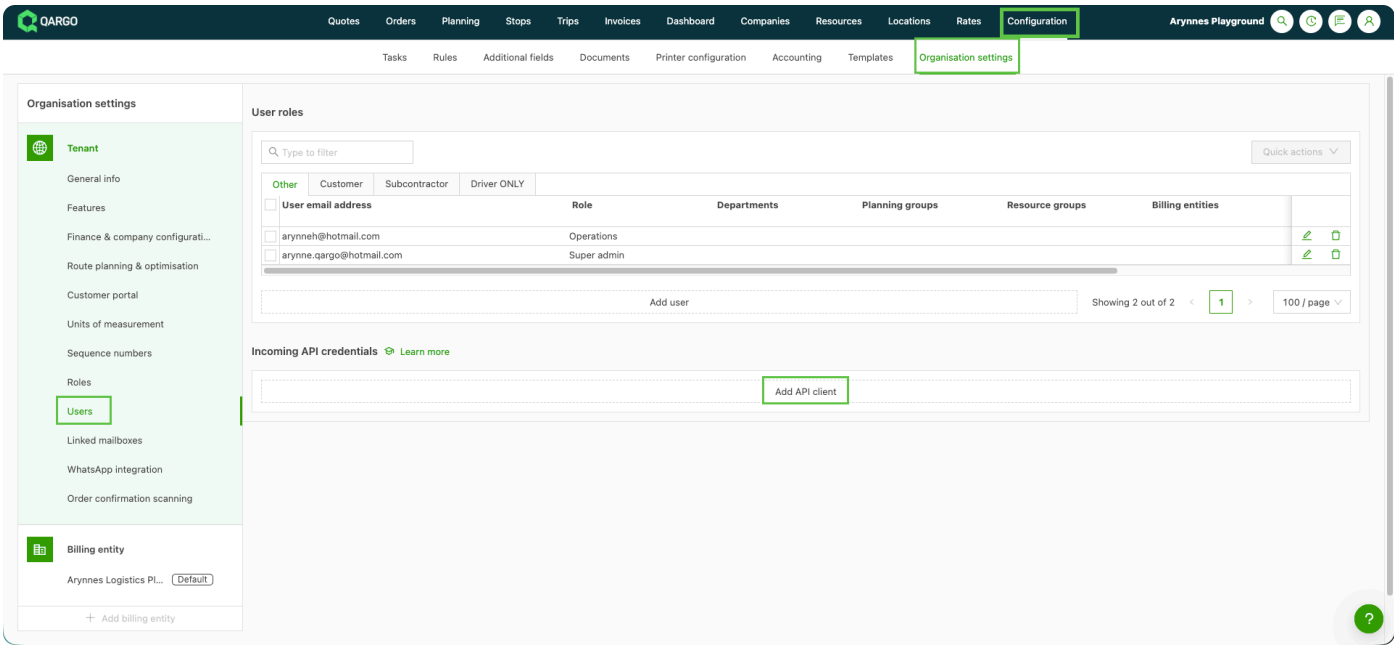
1. **Email** success@qargo.com to start the integration request.
2. **Provide** the following details:
    - **Integration partner**: company details, plus the name and email of the main contact Qargo will work with. If a third party or external integrator is building the connection, give their details rather than the Qargo account holder's.
    - **Type and scope**: the type of integration needed (for example order capture) and the expected functionality. See *Related documentation* at the end of this article.

- **Specific requirements**: any specific functional requirements.

3. **Wait** for an Integrations Delivery Manager (IDM) to get in touch. Qargo then issues the *integrator ID*.

Step 2: Add an API client for the required role

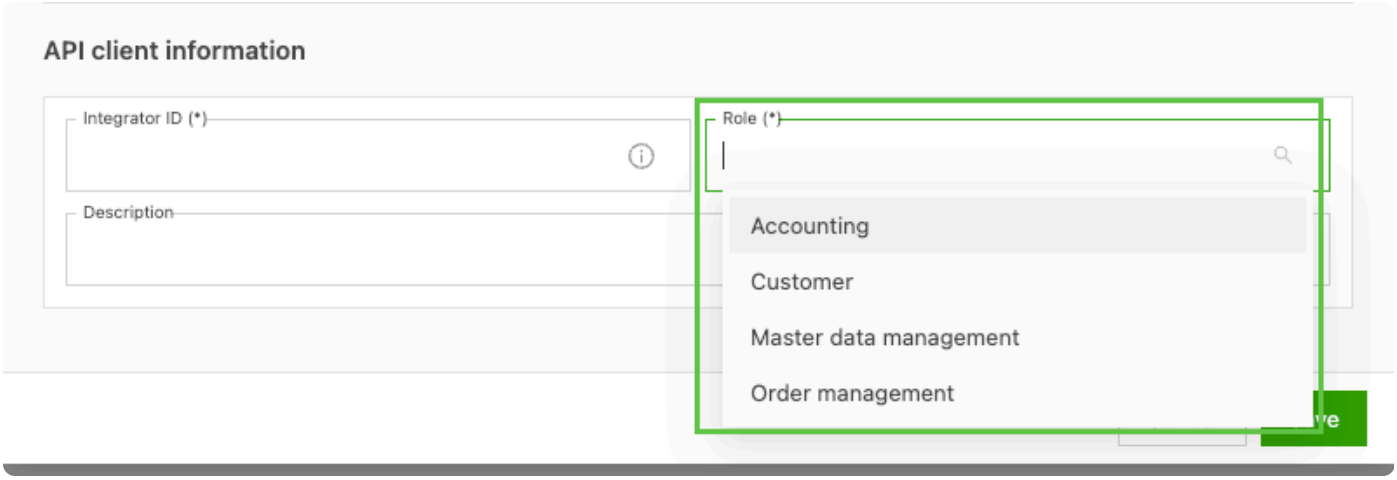
1. **Navigate** to **Configuration > Organization settings > Users** and scroll to the bottom of the page.
2. **Click Add API client**.



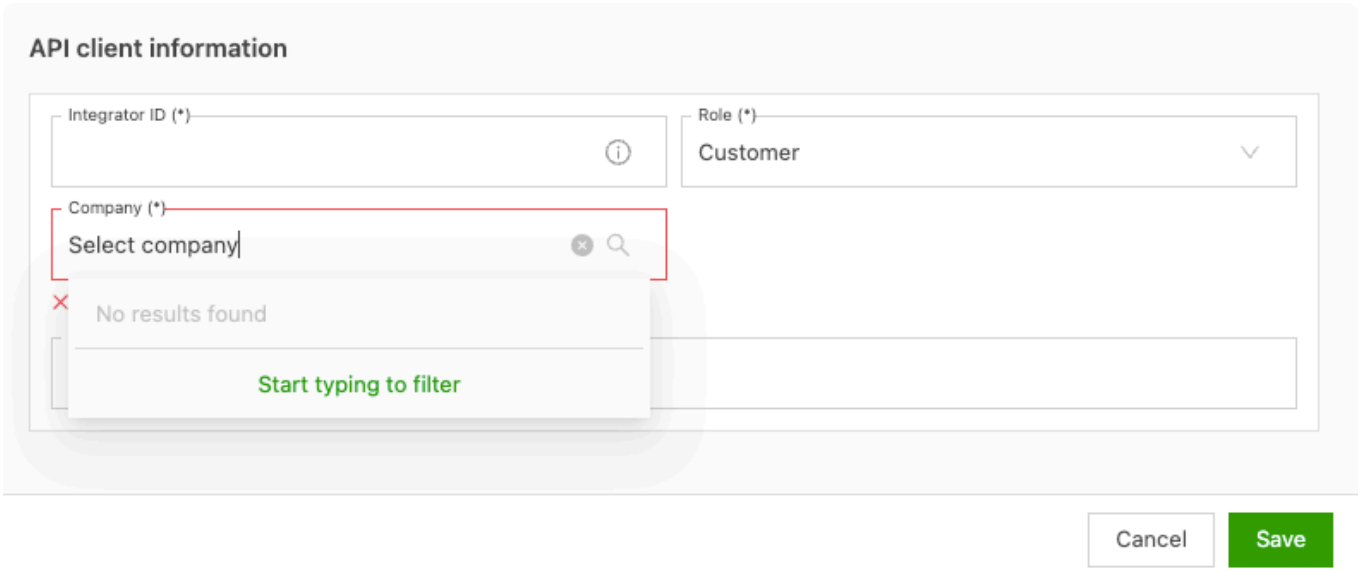
1. **Select** the role needed:

   - Accounting: for posting invoices, see *Accounting API and billing entities* below
   - Customer: used when a customer of the account wants to connect over the API
   - Master data management
   - Order management

Step 3: Fill in the client details

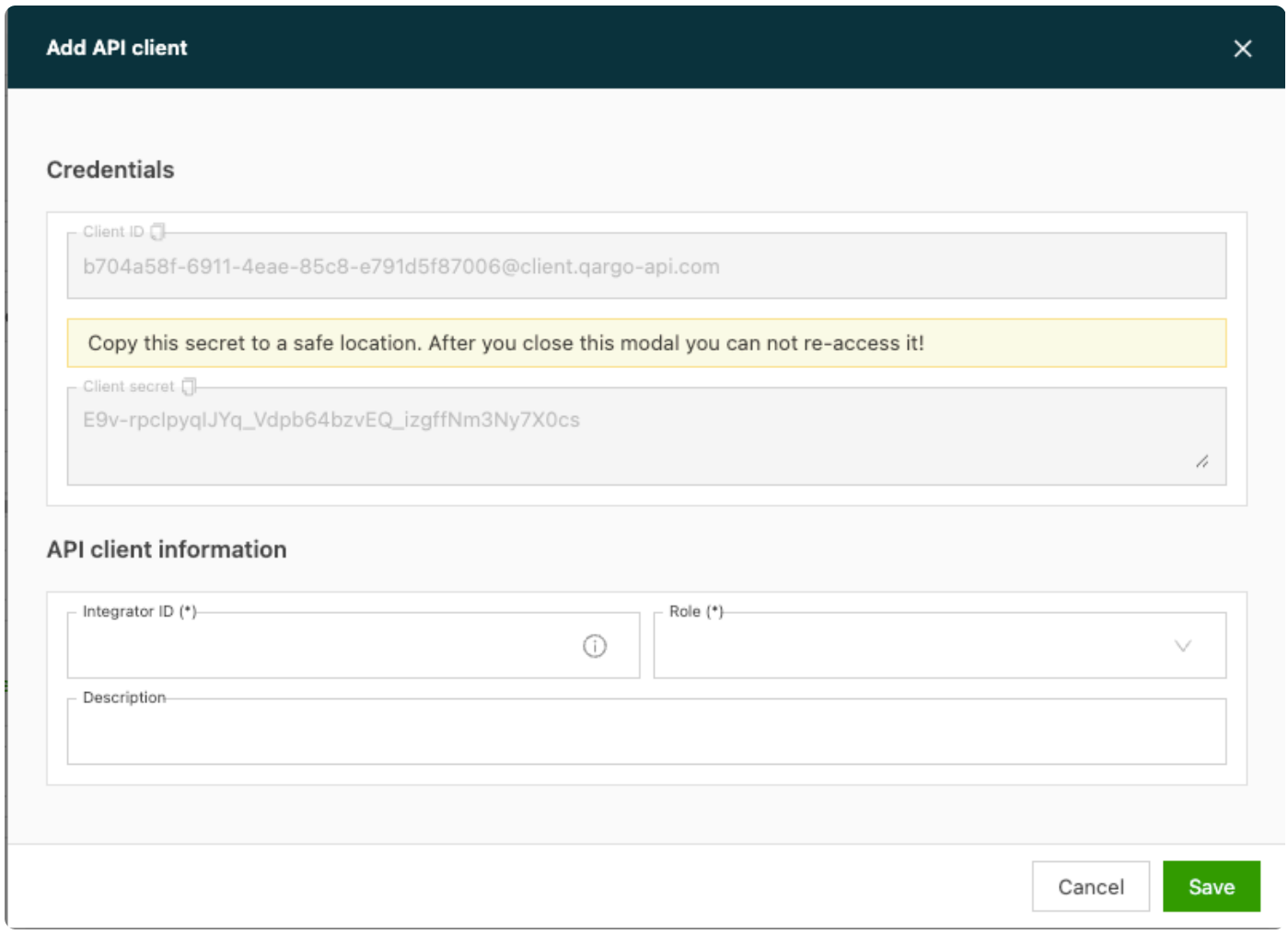


- **Integrator ID**: always required. Enter the ID received from Qargo.
- **Company**: required only for the *Customer* role.



🧰 For the *Customer* role, **Customer portal access** must be turned on for the relevant customer before an API client can be created.

‼️ Step 4: Copy the secret before saving ‼️



‼️ **The *secret* is shown only once.** Copy it to a safe place before clicking **Save** and closing the **Add API client** screen. It cannot be viewed again afterward, and a new client would need to be created.

Step 5: Share the credentials securely

Share the client ID and secret securely with the party building the integration.

Step 6: Let Qargo know

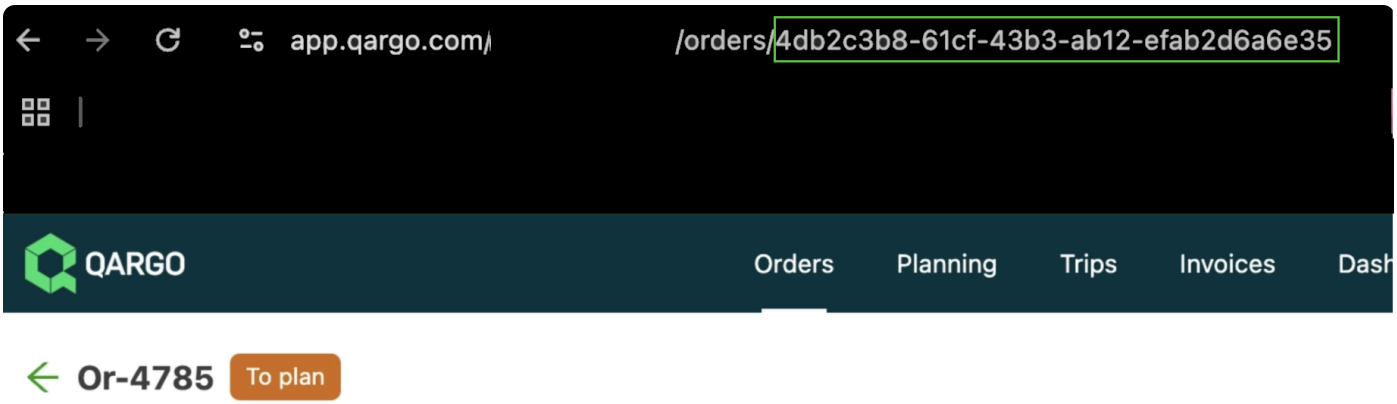
Notify Qargo once the clients are set up. Some extra account-level setup may be needed before testing starts, for example order import mapping for the order API, or Accounting API setup.

Testing the order API

The easiest way to start is to create a few typical test orders manually in Qargo, then use them as a template for building API requests.

Get a template with Order Export

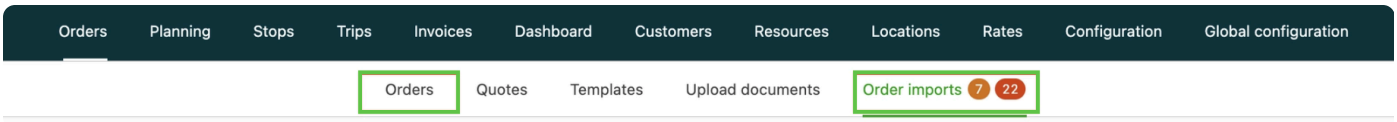
1. **Open** a test order in Qargo and copy the *order_id* from the URL (the part after *orders/*).
2. **Call** the *Order Export* endpoint:
   GET <https://api.qargo.com/v1/orders/order/{order_id}/export>
3. **Review** the returned data. Comparing a real Qargo order with its data structure shows how each field needs to be filled in.



Create a new order

1. **Remove** auto-generated fields such as barcodes.
2. **Include** the import configuration code (by default this is *API*).
3. **Change** unique identifiers, such as the customer reference number and *order_identifier*.
4. **Upload** the order **POST <https://api.qargo.com/v1/orders/order/upload>**

The order then appears in either the **Orders** tab or the **Order imports** tab, depending on the order import configuration.



Order import configuration

Order import configuration is set at the account level and can only be changed by Qargo. These settings control how orders received over the API behave.

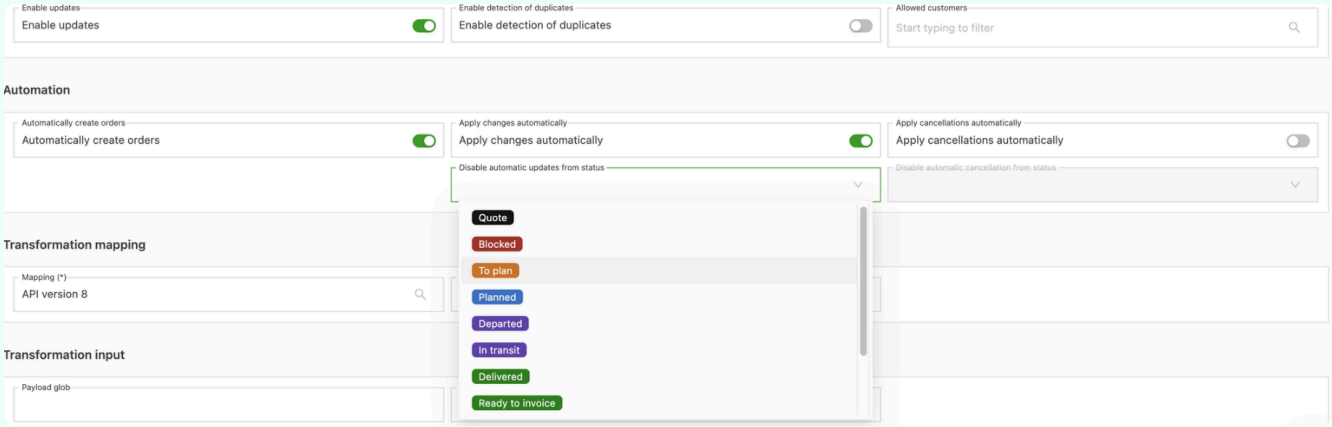
📚 For more detail on mapping and errors, see Integrations – Importing orders into Qargo.

Setting	What it does
Automatically create orders	Sends imported orders straight to the **Orders** tab, or to the **Order imports** tab, where they must be confirmed first.
Enable updates	Controls whether updates are accepted over the API.
Enable detection of duplicates	Flags duplicate imports automatically so they are not imported twice.
Apply changes automatically	Applies changes without manual checks. Can be limited to a chosen order status, after which updates need manual approval again.
Apply cancellations automatically	Works the same way as *Apply changes automatically*, but for cancellations.

📚 During testing, *Automatically create orders* is often turned off. This keeps API test orders separate from the orders other teams create manually.

📚 Orders that pass the API check but fail in Qargo always land in the **Order imports** tab, where a key user or admin can review and fix the error.

📚 **Example:** many companies accept updates automatically until an order is planned. To set this up, turn on *Apply changes automatically* and turn it off from status *Planned* onward.



‼️ **Different rules for the Customer API.** For the *Customer* role, updates and cancellations follow the customer portal rules, not the order import configuration. If customers cannot edit orders in the customer portal, they cannot edit them automatically over the API either. Updates can still be sent, but each one must be approved in the order import screen.

Checking that an order import worked

Qargo checks an order sent over the API in two steps. A pass in step 1 does not mean the order exists in Qargo.

Step	What happens	Where the result shows
1. *API check*	Qargo checks that the message is built correctly.	In the validation response sent back to the external system.
2. *Qargo import*	Qargo applies the data to the account. The order is created or updated. This runs after step 1, in the background.	In the import status, and on the order in Qargo.

What the API check confirms

- The message follows the expected structure (JSON or XML).
- Required fields are present, with the right data types and formats.
- Values such as units and order types are allowed.

What the API check does not confirm

- The customers, locations, products, and resources in the order exist in the Qargo account.
- Business rules and account configuration allow the order to be created or changed.
- The order reference matches correctly, so Qargo creates a new order or updates the right existing one.
- An existing order is still in a status that allows changes.

A Qargo import can fail or be rejected even when the API check passed.

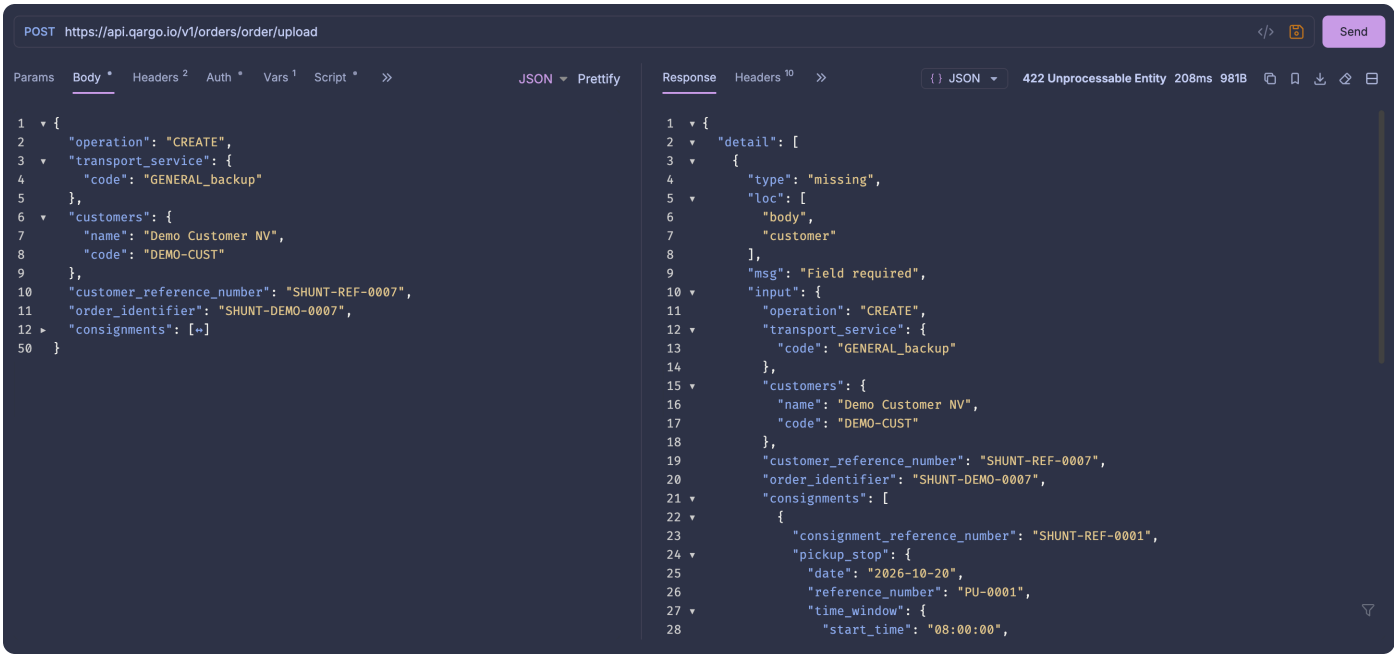
Confirm that an import succeeded

1. **Do not** treat a positive API check as confirmation that the order exists.
2. **Check** the import status of the order. A status of *error* means the Qargo import failed. A manual correction in Qargo is needed.
3. **Open** the **Order imports** tab, or the order in Qargo, to review the result.
4. **Send** Qargo Support the order reference, the timestamp, and the error message if the import still fails.

📚 The *API check* and the *Qargo import* are separate steps. The same order can pass the first and fail the second.

Examples: API check and Qargo import results

API fail (= API validation fail)



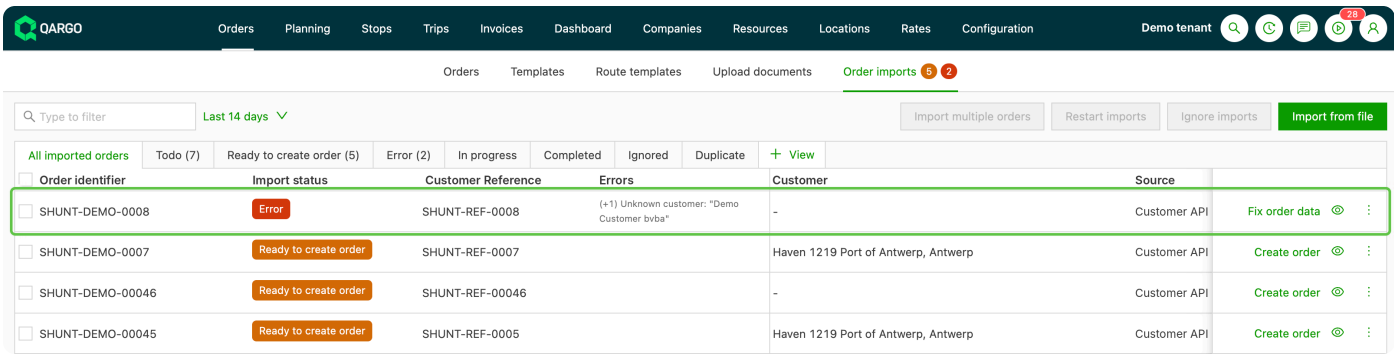
API success



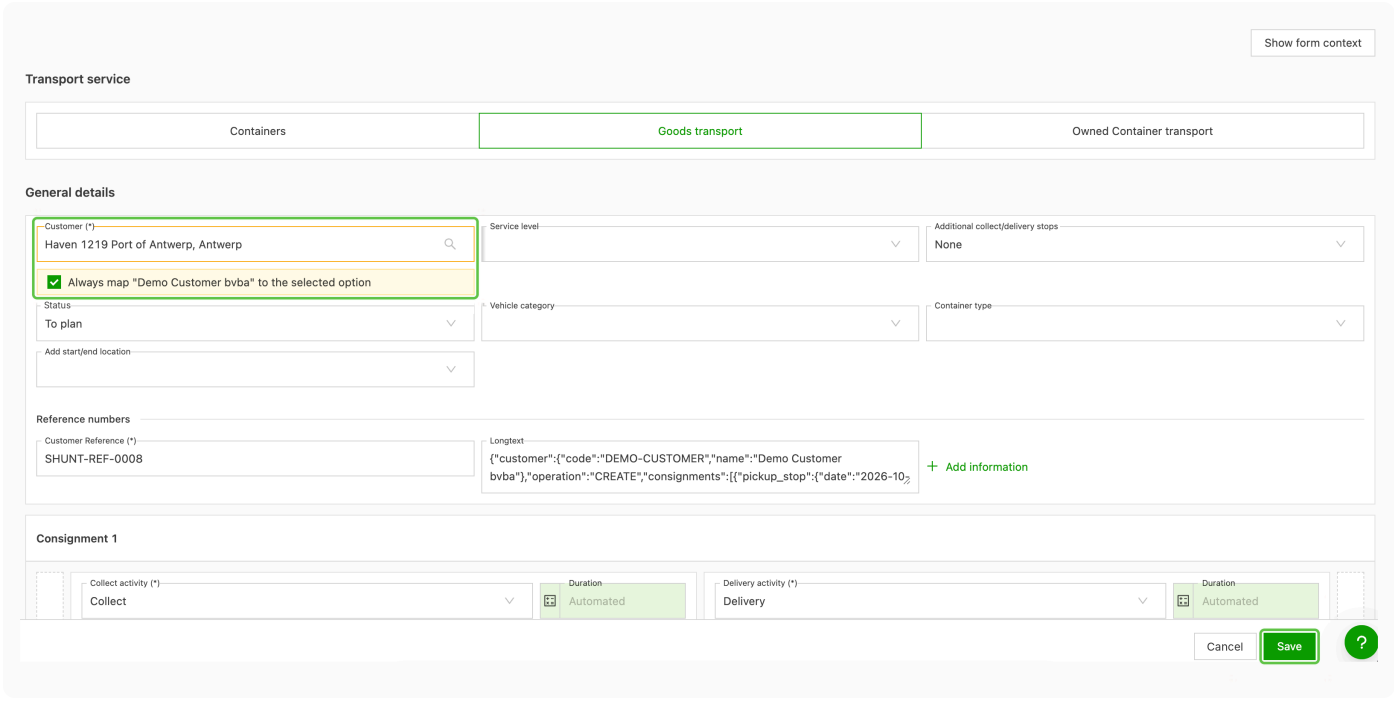
API status upsert error (= UI import fail)



Order import error screen (= UI import fail)



Order import fix (UI)



Order import fix (API)

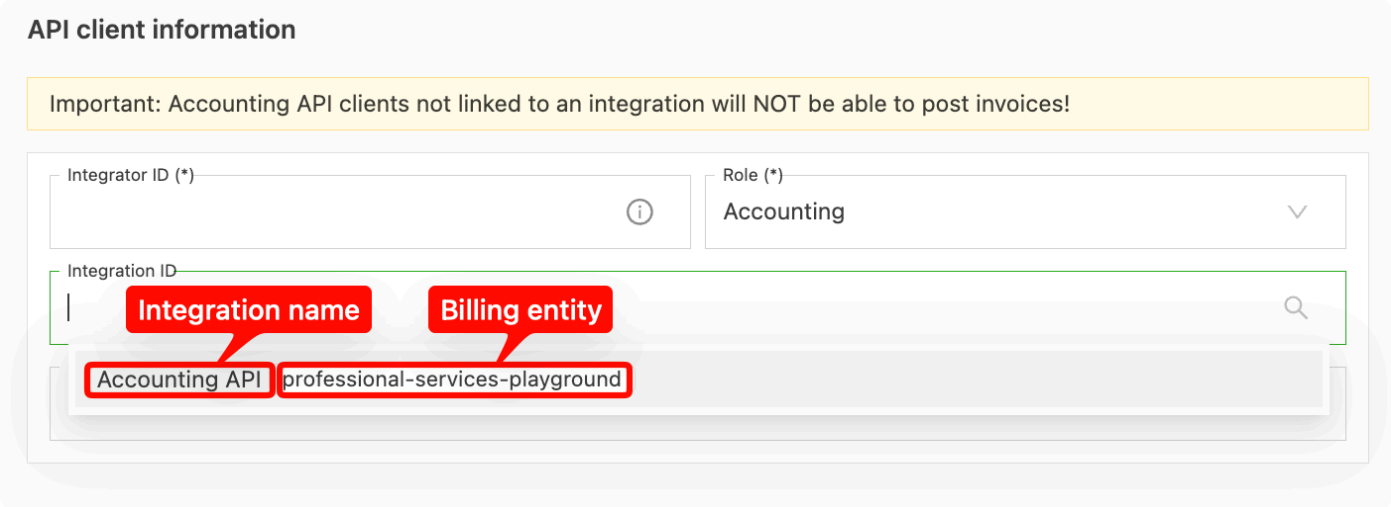


Accounting API and billing entities

The Accounting API supports **multiple billing entities**. This gives tighter security and access control for groups that invoice from more than one company.

What is included

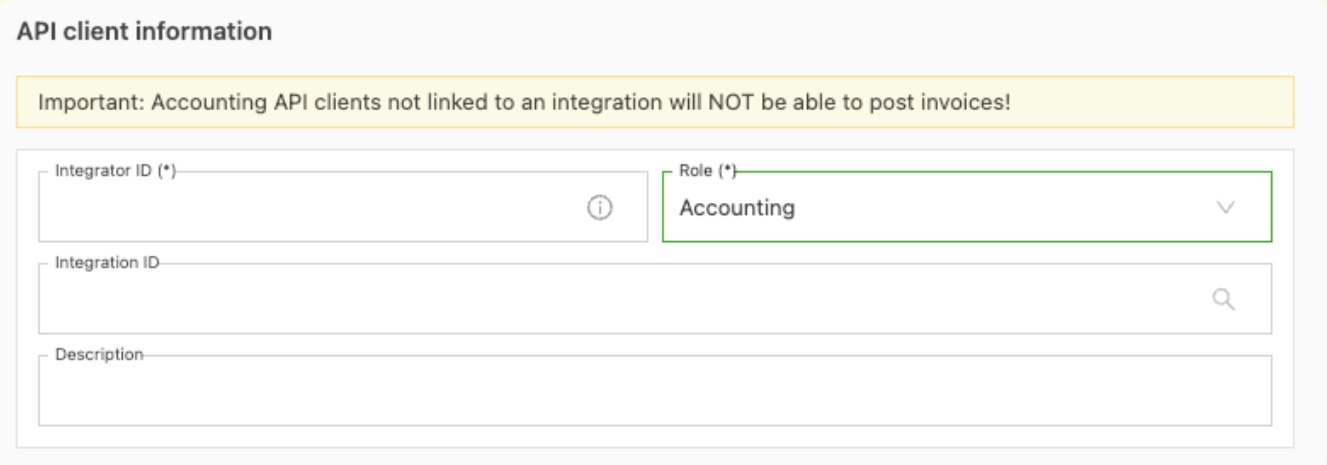
- **Billing entity on invoices**: the invoice endpoints include a *billing_entity* section showing which entity an invoice belongs to.
- **Multiple integrations**: several Accounting API integrations can be created, each linked to one billing entity.
- **Filtering**: invoices and tasks can be filtered by their billing entity.
- **Dedicated API clients**: Accounting API clients are created in the *Accounting API Integration* settings page. These clients are needed for full functionality.



How it affects external integrations

- **Restricted posting**: an integrator can only complete (post) invoices for the billing entity linked to its credentials.
- **Viewing**: invoices for other billing entities stay visible, subject to normal viewing permissions, but cannot be posted.
- **Where credentials are created matters:**
  - To post invoices (set the status to *completed*), credentials must be created in the *Accounting API Integration* settings page and linked to the correct billing entity.
  - Credentials created in *Organization settings* **cannot** set an invoice to *completed* or list invoices still waiting to be synchronized.

‼️ Accounting clients created in *Organization settings* cannot post invoices. Always create clients that post invoices in the *Accounting API Integration* settings page.



Troubleshooting

Issue	Resolution
Test orders do not appear in the **Orders** tab	With *Automatically create orders* turned off, imports go to the **Order imports** tab and must be confirmed there.
The API check passed, but the order is not in Qargo	The Qargo import may have failed. Check the import status and the Order imports tab. See Checking that an order import worked.
The API check passed, but the order is not in Qargo	The Qargo import may have failed. Check the import status and the **Order imports** tab. See *Checking that an order import worked*.
An order imported over the API shows an error in Qargo	Errors that do not block the API land in the **Order imports** tab. A key user or admin can review and fix them there. An import status of error means the Qargo import failed and a manual correction is needed.
The secret was lost after closing the screen	The secret cannot be recovered. Create a new API client and copy the secret before saving.
An API client cannot be created for the *Customer* role	Turn on **Customer portal access** for that customer first.
An Accounting client cannot post invoices	Create the client in `Accounting API Integration settings` and link it to the correct billing entity. Clients from `Organisation settings` cannot post.
An integrator cannot post some invoices	Those invoices belong to a different billing entity. Use credentials linked to that billing entity.
Duplicate orders keep being imported	Ask Qargo to turn on *Enable detection of duplicates*.
Customer API updates are not applied automatically	This is expected. Customer updates follow the customer portal rules and must be approved in the order import screen.

FAQs

Q. Does Qargo offer an API?
A. Yes! Qargo offers APIs for order capture and statuses, load operations, master data sync, and accounting, across the Tenant, Customer, and Contract carrier integration types.

Q. Can an API connection be set up without Qargo?
A. No. Setting up an integration requires **Super Admin** rights and support from the Qargo team. Start by emailing success@qargo.com.

Q. Can one Accounting integration post invoices for every billing entity?
A. No. Each Accounting API client can only post invoices for the billing entity it is linked to. Invoices for other entities stay visible but cannot be posted.

Related documentation

API reference documentation (api.qargo.com)

- Qargo TMS API – Tenant: Order
- Qargo TMS API – Customer specification
- Qargo TMS API – Contract carrier specification
- Qargo TMS API – Master data sync
- Qargo TMS API – Accounting