Logistics: Flexport

From the SourceLace manual: How to connect Flexport, so agents and people can read your freight shipments and freight invoices next to your ERP.

SourceLace version 0.1.0 · Made November 22, 2026
The latest version is at https://sourcelace.ai/docs

Logistics: Flexport

How to connect Flexport, so agents and people can read your freight shipments and freight invoices next to your ERP.

Status: In testing. New. Built on Flexport's published API and covered by automated tests against a simulated version of it. SourceLace has not yet tried it against a real Flexport account, so the first connection is checked together with you. Where Flexport's own documentation does not say something, this page says so and tells you who can confirm it.

At a glance

Source	Kind	Reads	Changes	People sign in with	First test question
Flexport	`flexport`	Shipments and freight invoices (with their line items and credit memos)	None (read-only)	One **shared** API key an admin makes	"Which objects does flexport:main offer?"

Flexport only knows its own freight: the shipments and invoices Flexport handles for your company. Freight carried by other forwarders or carriers is not in it.

It is shared access. Flexport does not sign each person in separately for this API, so everyone who may use the source acts as the one key SourceLace was given. Flexport cannot tell people apart. Narrow who can use the source with access by group, and hide or mask personal data, such as contact names and delivery addresses, with data protection. The key is kept encrypted and never shown again.

SourceLace only reads. The connector sends only the read calls it lists (shipments and invoices, one list and one record each) and refuses anything else before it is sent.

As everywhere in SourceLace, queries are checked before anything is sent, results are held for at most 30 minutes and never written to a database (see How long things are kept).

Flexport

What your admin does in Flexport

Once. About 10 minutes.

Who: someone with access to your company's Flexport account who can open **Settings**. Flexport's documentation does not say which role is needed to make an API key: if you cannot see the **API** tab, ask your Flexport account manager.

1. **Open the API settings.** In the Flexport portal, click **Your Business** in the top bar, then **Settings**, then the **API** tab. Flexport's documentation names the tab **API Credentials**, and calls the other kind of access **API Keys**: SourceLace uses an **API key**.
2. **Make an API key.** Create a key, give it a name such as `SourceLace`, and copy it. It is shown once. Keep it for your SourceLace admin: do not paste it into email or chat.
3. **Decide whether to set an API version.** Your Flexport account has a default API version (the **API** settings page shows it). Leave the **API version** option empty to use it. Set it, for example to `3`, only if Flexport support tells you to.

4. **Give your SourceLace admin** the key. They add the source below.

Check that it worked: ask "Which objects does flexport:main offer?" (you should see `shipments` and `invoices`), then "Show five Flexport invoices that are past due". If you see "Flexport did not accept the API key", look it up under When something goes wrong.

Flexport's own guides: Flexport developer documentation (search for "API credentials") and the API reference.

Add the source (SourceLace admin)

Kind: **Flexport** (`flexport`). There is no address to enter: Flexport has one public API address, which SourceLace already knows.

Option	Type	Default	Example	What it is
`secret`	Secret	(required)		The API key (step 2). Stored encrypted and never shown again. Leave it empty when editing to keep the saved one.
`api_version`	Number	(empty)	3	Optional. The Flexport API version to use instead of your account's default (step 3).

What people can do

`search_schema` lists the two objects, and `describe_object` shows the fields of one: the ones Flexport's documentation names, plus the fields of one real row. `query` (language `flexport_json`) takes one small JSON object:

What	Example	Notes
Invoices past due	`{"resource": "invoices", "filters": {"status": "past_due"}, "sort": "due_date", "direction": "asc", "limit": 100}`	Invoice filters: `status` (outstanding, past_due, void or paid), `shipment_id`, `billed_directly_to_client`, `entity_id`, `entity_ref`, `name`, `issued_after`, `issued_before`, `updated_after`, `updated_before` (dates such as 2026-09-30). Sort: `due_date`.
Shipments	`{"resource": "shipments", "filters": {"transportation_mode": "ocean", "purchase_order": "PO-1001"}, "limit": 50}`	Shipment filters: `transportation_mode`, `status`, `statuses` (a list), `container_number`, `purchase_order`, `sku`, `master_bill_number`, `house_bill_number`, `consignee_external_ref`, `origin_address_ref`, `destination_address_ref`, `archived_after`, `archived_before`. Sort: `id`, `transportation_mode`, `status`, `updated_at` or `archived_at`.

`get_record` reads one shipment or invoice by its Flexport number. Lists are read page by page (100 rows a page) up to the limit you set. Flexport's documentation states no rate limit, so SourceLace starts low for your whole organization together and waits when Flexport asks it to slow down.

An invoice's shipments, line items and credit memos come with the invoice. A shipment's origin and destination are not top-level fields: Flexport keeps them in the shipment's legs and parties.

Flexport in one question across systems

Flexport invoices and shipments can be a table in one question across your systems, so an AI app can put freight cost next to your CRM or finance data in a single answer, for example "which of our customers have past-due Flexport invoices over $10,000?". Name the table as `"flexport:main"."invoices"` or `"flexport:main"."shipments"` and ask for the fields you need. Three things to know:

- **Only a few filters reach Flexport.** An invoice's status or name, a shipment's status or mode, and an invoice's issue or update time are asked of Flexport; every other condition is applied by SourceLace after the rows arrive, so it does not make a large read smaller.
- **Times and dates are real times and dates.** An invoice's `issued_at` is a time and `due_date` a date, so compare them as such (`"issued_at" >= TIMESTAMP '2026-01-01 00:00:00'`, `"due_date" < DATE '2026-06-30'`). Only the issue and update times are asked of Flexport; a condition on a due date or on any shipment date is applied by SourceLace after reading.
- **A very large list is read only up to a limit.** When SourceLace stops at the limit, the answer says it is incomplete and says why, instead of looking whole.

When something goes wrong with Flexport

What you see	What it means	What to do
"Flexport did not accept the API key. ..." (HTTP 401 or 403)	The key is wrong, was deleted or turned off, or belongs to another Flexport account.	Check the key in Flexport (step 2), then enter the current key on the source.
"Flexport does not have that shipment or invoice, or the account cannot see it." (HTTP 404)	The number is wrong, or the shipment or invoice belongs to another account.	Check the number.
"Flexport refused the query. Flexport said: ..." (HTTP 400 or 422)	A filter or value Flexport does not accept.	Check the filter against the table above. The AI usually fixes the query itself.
"Flexport is limiting how fast SourceLace may call it. Wait a minute and try again." (HTTP 429)	Flexport asked SourceLace to slow down.	Wait and ask again, or ask for less at once.
"Flexport returned HTTP ..."	Flexport had a problem or refused for a reason it did not explain.	Try again; if it keeps happening, tell support@sourcelace.com.
"... needs the option secret ..."	The API key is missing.	Enter it (step 2).
"The option api_version of ... is not valid ..."	The version is not a number.	Use a number such as 3, or leave it empty.
"'...' is not a filter of invoices. Use: ..." or "'...' is not allowed. Use resource, filters, sort, direction and limit."	The query used something SourceLace does not accept.	Use the example shapes above.
"SourceLace refused to make that call to Flexport."	A call outside the read calls SourceLace allows.	Use `query` and `get_record`.
"'...' in ... is not a Flexport object. Use shipments or invoices."	A question across systems named a table that is not one of the two.	Use `"flexport:main"."invoices"` or `"flexport:main"."shipments"`.
"Flexport found no ... with the id ...."	No shipment or invoice has that id, or the account cannot see it.	Use an id from a query's results.