API documentation

Version v1.20261104 · November 2026

The Cargopedia API lets your TMS/ERP publish, update, up and delete load and truck ads on Cargopedia automatically, without using the site.

CONTENTS

What's new in this version 2
Getting started 2
Type glossary 3
Adding/updating a load 5
Deleting loads 9
Upping loads 10
Listing your loads 11
Adding/updating a truck 13
Deleting, upping and listing trucks 15
Own IDs 16
Listing company users 16
Errors 18

What's new in this version

Version v1.20261104 changes only how duplicates are handled (see Duplicates). Everything else keeps working as before.

- **Other users' ads are not changed**: if you publish a load or a truck similar to an active ad of another user of your company, that ad is no longer updated with your data: you just get the `duplicate` error. Publishing again a load similar to one of your own ads still updates your ad, as before.
- **Edits don't delete ads**: an edit that would make an ad similar to another active ad of your company is refused with the `duplicate` error, and both ads stay as they were. Before, the other ad was deleted.
- **Duplicate responses contain the url** of the existing ad.
- **Sandbox ads are compared only with sandbox ads**, so your tests no longer affect your live ads.

Version v1.20261027 added:

- **Multiple loading places**: `from` accepts several localities separated by „|", like `to`. A place that can't be recognized now returns an error, instead of being silently skipped.
- **Stable error codes**: every error response has a `code` field (see Errors). Validation errors also contain `ok` and `message` now.
- **Ad links**: `loads/publish` and `trucks/publish` return the `url` of the ad. Duplicate responses contain `"duplicate": true`.
- **API key in a header**: you can send the key in the `X-Api-Key` HTTP header, instead of the `key` parameter.
- **New endpoints** `loads/list` and `trucks/list`: your ads and their status (active, expired, deleted).
- **Truck ads** (`trucks/publish`, `trucks/delete`, `trucks/up`) are now documented.
- **Request limits**: when you send many requests at once, the extra ones are queued, not rejected (see Request limits).

Getting started

Account and API key

To use the Cargopedia API, you need a Cargopedia account and a valid **API key / user id** combination. You can find them in your Cargopedia account, at Settings → Company settings → API access data.

Sandbox and approval

By default, any account can use the API, but the ads published via API are **sandboxed**: they are visible only to you, not to the other users. This allows you to safely test your integration. When you are ready, write us at api@cargopedia.net to lift this limitation. Before that, make sure you master the API: thoroughly test every feature you'll use (add, edit, delete etc.).

Requests

Each request can be made either by **GET** (parameters in the URL) or by **POST** (parameters sent as form data or as a JSON body).

Send your API key in the `X-Api-Key` **HTTP header** (recommended) or as the `key` parameter. With the header, the key doesn't appear in URLs or in request bodies, which may end up in the logs of the servers in between. If both are given, the header is used.

Responses

Each response is a JSON object, like this:

```
{
  "ok": true,
  "message": "The load #710717 has been successfully added.",
  "id": 710717,
  "url": "https://www.cargopedia.net/load-to-transport-from-frankfurt-de-to-warsaw-pl-l710717"
}
```

If the request is unsuccessful, `ok` is false, `code` contains a stable error code and `message` a human-readable description (see Errors). If the account is banned, the response also contains `banned` (true) and `ban_reason`:

```
{
  "ok": false,
  "code": "banned",
  "message": "The account is banned!",
  "banned": true,
  "ban_reason": "Security reasons"
}
```

Request limits

Requests are processed in the order they arrive. If you send many requests at once (more than 10 per second with the same API key, or from the same IP address), the extra ones are **queued** and processed as soon as possible: they just wait a little (up to about 10 seconds), without errors.

If the queue is full, you get **HTTP status 429** and the error code `too_many_requests`: wait the number of seconds given in the `Retry-After` header, then try again. We recommend sending the requests sequentially, or with a low degree of parallelism.

If we change these limits, we will notify you by e-mail at least 30 days in advance.

Fair use

The API shall be used responsibly, in good faith. Do not abuse it to spam the site with fake ads or for other malicious activity: besides banning your account, we reserve the right to take legal action against you and your company.

Contact

For questions or suggestions to improve the API, please write us at **api@cargopedia.net**.

Type glossary

PLACE

A locality, region or country, if not specified otherwise. The system is very flexible regarding place names; for example, all these are accepted:

EXAMPLE	FORMAT
`Bucharest`	city (simplest, but not recommended)

EXAMPLE	FORMAT
`Bucharest, Romania`	city, country name
`Bucharest, RO`	city, country code
`Curtea de Arges, Arges, Romania`	city, region, country
`Curtea de Arges, Arges, RO`	city, region, country code
`RO-115300`	postal code only
`RO-115300, Curtea de Arges, RO`	postal code, city, country code
`RO-115300, Curtea de Arges, Arges, RO`	postal code, city, region, country code

You can omit the spaces, for example „Bucharest,RO" instead of „Bucharest, RO". Several notes:

- For best results, if you have the postal code, provide it together with the city name (like `RO-115300, Curtea de Arges, RO`). We don't recommend providing only the postal code, as it is not always reliable. Without a postal code, use `city, region, country code` if you know the region, otherwise `city, country code`. If there are several localities with the same name, the biggest/most important one is matched first – usually the right one.
- You can use any language variant of a name – Bucharest (English), București (Romanian), Bucarest (Spanish) – but it is best to use the same language as the `lang` parameter: English names with `lang=en`, Romanian names with `lang=ro`, Spanish with `lang=es`, French with `lang=fr` and so on.

Other types

TYPE	DESCRIPTION
`TRUCK TYPE`	One of: tilt (or tent), box, cartransporter, flatbed, openbody, tipper, tractorunit, container, animaltransporter, crane, tank, liquidfoodtank.
`DATE`	YYYY-MM-DD format. For example, 2026-10-30 means 30 October 2026.
`DATETIME`	ISO 8601 format, with time zone. For example 2026-10-27T09:59:31+00:00. Used only in responses.
`INTEGER`	An integer number.
`DECIMAL`	An integer or decimal number. Both decimal separators – comma and dot – are supported. Examples: 5, 10.5, 20,7
`BOOLEAN`	0 (false) or 1 (true).
`TEXT`	A text block, of any size if not specified otherwise.
`STRING(LENGTH)`	A string, of any size if not specified otherwise.
`ARRAY(TYPE)`	Multiple values, separated by comma. For example: 100,101 or blue,white.

Adding/updating a load

GET / POST `https://www.cargopedia.net/api/v1/loads/publish`

PARAMETER	TYPE	DESCRIPTION
key	STRING(15) **required**	Your Cargopedia API key – or send it in the `X-Api-Key` header.
user_id	INTEGER **required**	Your Cargopedia user id (see Listing company users).
from	PLACE **required**	Loading/departure place. Must be a locality. Several loadings, in different cities: separate them with „|", e.g. `Brasov,Romania|Rasnov,Romania`. The first one is the departure place.
to	PLACE **required**	Unloading/arrival place. Must be a locality. Several unloadings: separate them with „|", e.g. `Wien,Austria|Schwechat,Austria`. The last one is the final destination.
lang	STRING(2) optional	The language of the place names in `from` and `to` (see PLACE), e.g. `from=București,RO&to=Varșovia,PL&lang=ro` or `from=Bucharest,RO&to=Warsaw,PL&lang=en`. Default: en
valid_from	DATE optional	The start date of the ad. Default: today.
valid_to	DATE optional	The end date of the ad, at most 1 month after `valid_from`. Default: 7 days after `valid_from`.
truck_type	TRUCK TYPE or ARRAY(TRUCK TYPE) **required**	The required truck types, one or more, separated by comma. E.g. `box,tilt`
weight	DECIMAL **required**	The load weight, in tonnes.
length **width** **height**	DECIMAL optional	The load length/width/height, in meters.
volume	DECIMAL optional	The load volume, in m³. If omitted, it is calculated as length × width × height, when all three are given.
groupage	STRING (FTL|LTL) optional	`FTL` = full truck, `LTL` = groupage. Omitted = no preference.
description	TEXT optional	The ad description.
palletized	BOOLEAN optional	1 if the freight is palletized. Default: 0
pallets	INTEGER optional	The number of pallets. Required if `palletized` is 1.

PARAMETER	TYPE	DESCRIPTION
pallet_type	STRING(3) optional	`EUR` = European, `IND` = Industrial, `AME` = American, `ASI` = Asian, `AUS` = Australian, `OTH` = other type. Required if `palletized` and `pallets` are given.
payment	INTEGER optional	Payment term: `0` = unspecified; `1` = payment order at the receipt of documents (after unloading); `21` = payment order at unloading; `20` = cash at unloading; `7/15/30/45/60` = payment order at 7/15/30/45/60 days. Default: 0
price	DECIMAL optional	The offered price.
price_unit	STRING(3) **required with price**	The price currency, as a 3-letter ISO 4217 code (EUR, RON, PLN etc.).
VAT	BOOLEAN optional	1 if the price includes VAT. Default: 0
ADR	BOOLEAN optional	1 if you transport dangerous goods. Default: 0
reefer	BOOLEAN optional	1 if you need a truck with cooling system. Can be combined with `truck_type=box`. Default: 0
lift	BOOLEAN optional	1 if you need a truck equipped with lift. Default: 0
walking_floor	BOOLEAN optional	1 if you need a truck equipped with walking floor. Default: 0
sandbox	BOOLEAN optional	1 if you are just testing and don't want the load to be visible to the other users. While your account is not yet approved for publishing via API, sandbox is always 1 (see Sandbox and approval). Default: 0
id	INTEGER optional	The Cargopedia id of the load, to edit it (see Editing a load).
oid	STRING optional	Your own id of the load (see Own IDs).

Multiple loading/unloading places

The order of the places is kept. On the ad page and on the map, the route is shown as: departure → other loadings → other unloadings → destination, and the distance of the ad is computed along this route.

Every place must be recognized. If one of them can't be resolved, the load is not published and you get an error which tells you which place is the problem:

```
{
  "ok": false,
  "code": "unresolved_place",
  "message": "Cannot resolve \"from\" field!",
  "field": "from",
  "unresolved": "Rasnovv,Romania"
}
```

The dates (`valid_from`, `valid_to`) apply to the whole ad: there are no separate dates for each place.

Example

Request (GET)

```
https://www.cargopedia.net/api/v1/loads/publish?
key=<key>&user_id=<user_id>&from=Frankfurt,Germany&to=Warsaw,Poland&weight=5&truck_type=box,tilt
&palletized=1&pallets=6&pallet_type=EUR
```

The same request, by POST, with the key in a header (recommended)

```
curl https://www.cargopedia.net/api/v1/loads/publish \
  -H "X-Api-Key: <key>" \
  -d user_id=<user_id> \
  -d from=Frankfurt,Germany -d to=Warsaw,Poland \
  -d weight=5 -d truck_type=box,tilt \
  -d palletized=1 -d pallets=6 -d pallet_type=EUR
```

Response

```
{
  "ok": true,
  "message": "The load #710717 has been successfully added.",
  "id": 710717,
  "url": "https://www.cargopedia.net/load-to-transport-from-frankfurt-de-to-warsaw-pl-l710717"
}
```

Notice the returned `id`: this is the number Cargopedia assigns to your ad. You can use it to edit, up or delete the ad – or you can use your own ids instead (see Own IDs).

Cargopedia is a multi-domain, multi-language freight exchange (www.cargopedia.net, www.cargopedia.ro, www.cargopedia.de etc.), but all these sites share the same database: an ad is one and the same on all of them. So the `url` is just a display convention: the link to the ad in the language given by `lang` – for example, with `lang=ro` you get the link from www.cargopedia.ro. Without `lang`, the link is in English.

Editing a load

Send the same request, with all the parameters, plus `id=710717` (or the `oid` of the ad). The ad is replaced with the data you send. Editing also ups the ad, if allowed (see Upping loads).

```
{
  "ok": true,
  "message": "The load #710717 has been successfully edited.",
  "id": 710717,
  "url": "https://www.cargopedia.net/load-to-transport-from-frankfurt-de-to-warsaw-pl-l710717"
}
```

Deleted ads

An ad that was deleted from the site can no longer be edited by `id` (you get the `not_found` error). If you publish it again with the same `oid`, the ad is restored (with the same id), unless a similar ad is active in the meantime (see Duplicates).

Duplicates

If you publish a new load that matches an active load of your company (see below when two loads match), a second ad is not created. You get a response like this, with the `id`, `user_id` and `url` of the existing ad:

```
{
  "ok": false,
  "code": "duplicate",
  "message": "Duplicate detected. User #50003 added a similar load.",
  "duplicate": true,
  "id": 710702,
  "user_id": 50003,
  "url": "https://www.cargopedia.net/load-to-transport-from-frankfurt-de-to-warsaw-pl-l710702"
}
```

What happens to the existing ad:

- If it is yours (the same `user_id`), it is updated with the data you sent and upped, if allowed (see Upping loads). So you can update a load just by publishing it again.
- If it belongs to another user of your company, it is not changed.

The same goes when you publish again, by `oid`, an ad that was deleted: it is not restored if a similar ad is active. And an edit (by `id` or `oid`) that would make your ad match another active ad of your company is refused with the same error: both ads stay as they were.

Two loads match when they have the same departure (the first loading place) and destination (the last unloading place), the same weight, the same dimensions (`length`, `width`, `height`, `volume` – only those you send), the same `ADR` and `reefer`, at least one truck type in common and overlapping periods (`valid_from`–`valid_to`). The price, the description and the payment terms are not compared. Sandbox ads are compared only with sandbox ads (see Sandbox and approval).

Deleting loads

GET / POST `https://www.cargopedia.net/api/v1/loads/delete`

PARAMETER	TYPE	DESCRIPTION
key	STRING(15) **required**	Your Cargopedia API key – or send it in the `X-Api-Key` header.
user_id	INTEGER **required**	Your Cargopedia user id.
id	INTEGER or ARRAY(INT) **required (or oid)**	The id(s) returned by Cargopedia when you published the ad(s).
oid	STRING or ARRAY(STRING) **required (or id)**	Your own id(s) of the ad(s) (see Own IDs).

Request (GET)

```
https://www.cargopedia.net/api/v1/loads/delete?key=<key>&user_id=<user_id>&id=710717,710718
```

Response

```
{
  "ok": true,
  "message": "2 load(s) successfully deleted.",
  "ids": [710717, 710718]
}
```

Note If an ad doesn't exist anymore (for example, it was already deleted from the site) or doesn't belong to the given `user_id`, you get the `not_found` error. When deleting, you can treat it as a success.

Upping loads

Upping (or “refreshing”) a load brings it to the front page and on top of the list, in the same way as if it had been published as new.

> **GET / POST** `https://www.cargopedia.net/api/v1/loads/up`

PARAMETER	TYPE	DESCRIPTION
key	STRING(15) **required**	Your Cargopedia API key – or send it in the `X-Api-Key` header.
user_id	INTEGER **required**	Your Cargopedia user id.
id	INTEGER or ARRAY(INT) **required (or oid)**	The id(s) of the ad(s).
oid	STRING or ARRAY(STRING) **required (or id)**	Your own id(s) of the ad(s).

Request (GET)

```
https://www.cargopedia.net/api/v1/loads/up?key=<key>&user_id=<user_id>&id=710705,710706
```

Response

```
{
  "ok": true,
  "message": "1 load(s) successfully upped.",
  "upped": [710705],
  "not_upped": [710706]
}
```

> **Rules** An ad published via API can be upped **at most 5 times**: the first 3 times once an hour (60 min), then once every 3 hours. Expired ads can’t be upped.
>
> An ad that can’t be upped yet is returned in the `not_upped` array.

Listing your loads

Returns your loads and their status (active, expired or deleted). Useful for keeping your TMS/ERP in sync with Cargopedia, or for checking the result of a request that timed out.

GET / POST `https://www.cargopedia.net/api/v1/loads/list`

PARAMETER	TYPE	DESCRIPTION
key	STRING(15) **required**	Your Cargopedia API key – or send it in the `X-Api-Key` header.
user_id	INTEGER optional	Only the loads of this user. If omitted, the loads of all the company users are returned.
id	INTEGER or ARRAY(INT) optional	Only these loads (Cargopedia ids).
oid	STRING or ARRAY(STRING) optional	Only these loads (own ids).
status	STRING optional	`active`, `expired`, `deleted` or `all`. Default: `active` (`all`, if `id` or `oid` is given)
lang	STRING(2) optional	The language of the place names and of the ad links (`url`) in the results. Default: en
page	INTEGER optional	The page number. Default: 1
limit	INTEGER optional	Results per page, maximum 500. Default: 100

Request (GET)

```
https://www.cargopedia.net/api/v1/loads/list?key=<key>&oid=5021
```

Response


```
{
  "ok": true,
  "total": 1,
  "page": 1,
  "pages_total": 1,
  "results": [
    {
      "id": 710717,
      "oid": "5021",
      "user_id": 50003,
      "status": "active",
      "sandbox": false,
      "from": {
        "label": "Frankfurt am Main, Hesse, Germany",
        "city": "Frankfurt am Main",
        "region": "Hesse",
        "country": "Germany",
        "countryCode": "DE",
        "lat": "50.110920",
        "lng": "8.682130"
      },
      "to": { ... },
      "loadings": ["Frankfurt am Main, Hesse, Germany"],
      "unloadings": ["Warsaw, Masovia, Poland"],
      "requires": ["box", "tilt"],
      "palletized": true,
      "pallets": 6,
      "pallet_type": "EUR",
      "weight": 5,
      ...
      "valid_from": "2026-10-27",
      "valid_to": "2026-11-03",
      "created_at": "2026-10-27T09:59:31+00:00",
      "upped_at": "2026-10-27T11:02:10+00:00",
      "deleted_at": null,
      "url": "https://www.cargopedia.net/load-to-transport-from-frankfurt-de-to-warsaw-pl-
l710717"
    }
  ]
}
```


FIELD	DESCRIPTION
`status`	`active`, `expired` (the `valid_to` date has passed) or `deleted` (deleted from the site or via API).
`sandbox`	true if the ad is not visible to the other users (see Sandbox and approval).
`loadings, unloadings`	All the loading/unloading places, in order.
`created_at, upped_at, deleted_at`	DATETIME. `deleted_at` is null if the ad wasn't deleted.
`...`	The other fields (dimensions, reefer, ADR, lift, walking_floor etc.) have the same meaning as the loads/publish parameters.

Adding/updating a truck

Truck ads work in the same way as loads: the same `key`, `user_id`, `id`/`oid` and `sandbox` parameters, and the same kind of responses. Publishing trucks via API must be approved for your account too (see Sandbox and approval).

> **GET / POST** `https://www.cargopedia.net/api/v1/trucks/publish`

PARAMETER	TYPE	DESCRIPTION
key	STRING(15) **required**	Your Cargopedia API key – or send it in the `X-Api-Key` header.
user_id	INTEGER **required**	Your Cargopedia user id.
from	PLACE **required**	The place where the truck is available. Can be a locality, region or country.
to	PLACE **required**	The destination. Can be a locality, region or country, or: `*` = anywhere; `EU` = anywhere in the European Union. A too broad route (e.g. Romania → anywhere) is not accepted: you get the `route_too_broad` error. In this case, give a locality or region as `from`.
lang	STRING(2) optional	The language of the place names in `from` and `to`. Default: en
valid_from	DATE optional	The date from which the truck is available. Default: today.
valid_to	DATE optional	The end date of the ad, at most 12 months after `valid_from`. Default: 12 months after `valid_from`.
truck_type	TRUCK TYPE **required**	The truck type. If you give several types, only the first one is used.
weight	DECIMAL **required**	The truck capacity, in tonnes.
length **width** **height**	DECIMAL optional	The cargo space length/width/height, in meters.
volume	DECIMAL optional	The cargo space volume, in m³. If omitted, it is calculated as length × width × height, when all three are given.
description	TEXT optional	The ad description.
price	DECIMAL optional	The price per kilometer.
price_unit	STRING(3) **required with price**	The price currency, as a 3-letter ISO 4217 code (EUR, RON, PLN etc.).
VAT	BOOLEAN optional	1 if the price includes VAT. Default: 0

PARAMETER	TYPE	DESCRIPTION
reefer **ADR** **lift** **walking_floor**	BOOLEAN optional	1 if the truck has a cooling system / can transport dangerous goods / has a lift / has a walking floor. Default: 0
gps	BOOLEAN optional	1 if the truck has GPS tracking. Default: 0
CMR	INTEGER optional	The CMR insurance amount, in EUR.
email_notify	BOOLEAN optional	Notify the user by e-mail about new matching loads. Default: 1
SMS_notify	BOOLEAN optional	Notify the user by SMS about new matching loads. Default: 0
PA_notify	BOOLEAN optional	Notify the user also about matching loads published by individuals (not companies). Default: 1
sandbox	BOOLEAN optional	The same as for loads. Default: 0
id	INTEGER optional	The Cargopedia id of the truck, to edit it.
oid	STRING optional	Your own id of the truck (see Own IDs).

Request (GET)

```
https://www.cargopedia.net/api/v1/trucks/publish?
key=<key>&user_id=<user_id>&from=Brasov,Romania&to=*&truck_type=tilt&weight=24&oid=T-77
```

Response

```
{
  "ok": true,
  "message": "The truck #95210 has been successfully added.",
  "id": 95210,
  "url": "https://www.cargopedia.net/tilt-truck-from-brasov-ro-t95210"
}
```

Duplicates If you add a truck that matches an active truck of your company, a second ad is not created: you get the `duplicate` error, with the `id`, `user_id` and `url` of the existing truck. If that truck is yours, it is upped, if allowed; a truck of another user is not changed. Edits are refused in the same way as for loads (see Duplicates).

Deleting, upping and listing trucks

GET / POST `https://www.cargopedia.net/api/v1/trucks/delete`
GET / POST `https://www.cargopedia.net/api/v1/trucks/up`

Both work exactly like `loads/delete` and `loads/up`: the same parameters (`key`, `user_id`, and `id` or `oid` – one or more, separated by comma), the same rules and the same responses, with "truck(s)" instead of "load(s)" in the message.

GET / POST `https://www.cargopedia.net/api/v1/trucks/list`

Works like `loads/list`, with the same parameters (`user_id`, `id`, `oid`, `status`, `lang`, `page`, `limit`). Each result contains the truck fields, plus `oid`, `user_id`, `status`, `sandbox`, `created_at`, `upped_at` and `deleted_at`:

```
{
  "ok": true,
  "total": 1,
  "page": 1,
  "pages_total": 1,
  "results": [
    {
      "id": 95210,
      "oid": "T-77",
      "user_id": 50003,
      "status": "active",
      "sandbox": false,
      "from": "Brasov, Brasov County, Romania",
      "to": "Anywhere",
      "destination_type": "*",
      "truck_type": "tilt",
      "weight": 24,
      ...
      "valid_from": "2026-10-27",
      "valid_to": "2027-10-27",
      "created_at": "2026-10-27T09:59:31+00:00",
      "upped_at": "2026-10-27T09:59:31+00:00",
      "deleted_at": null,
      "url": "https://www.cargopedia.net/tilt-truck-from-brasov-ro-t95210"
    }
  ]
}
```

`destination_type` is `exact` (a locality or region), a country code (anywhere in that country), `*` (anywhere) or `EU` (anywhere in the European Union).

Own IDs

When you add an ad, Cargopedia assigns it an id and returns it; you can use this id later to edit, up or delete the ad. But if your TMS/ERP already has its own ids for loads/trucks, you can use them instead – no need to store the Cargopedia ids. Just send your id as `oid` when publishing:

```
https://www.cargopedia.net/api/v1/loads/publish?
key=<key>&user_id=<user_id>&from=Frankfurt,Germany&to=Warsaw,Poland&weight=5&truck_type=box,tilt
&oid=5021
```

Later, to delete this load:

```
https://www.cargopedia.net/api/v1/loads/delete?key=<key>&user_id=<user_id>&oid=5021
```

You can use `oid` instead of `id` everywhere: add, edit, up, delete, list.

> **Good to know** The `oid` is unique per user (`user_id`), not per company: the same `oid` sent with another `user_id` creates a new ad (or gets the `duplicate` error, if it matches an active ad of that user).
>
> Publishing with an `oid` that already exists (for that user) edits the ad, it doesn't duplicate it. So, if a publish request times out and you don't know whether the ad was created, just send the same request again: it is safe. You can also check it with `loads/list`.

Listing company users

Returns the list of the company's active users, with their ids – useful for an easier TMS/ERP integration.

> **GET / POST** `https://www.cargopedia.net/api/v1/users/list`

PARAMETER	TYPE	DESCRIPTION
key	STRING(15) **required**	Your Cargopedia API key – or send it in the `X-Api-Key` header.

Request (GET)

```
https://www.cargopedia.net/api/v1/users/list?key=<key>
```

Response


```
{
  "ok": true,
  "users": [
    {
      "id": 50003,
      "name": "John Smith",
      "first_name": "John",
      "last_name": "Smith",
      "email": "john.smith@mycompany.com"
    },
    {
      "id": 50102,
      "name": "Jane Williams",
      "first_name": "Jane",
      "last_name": "Williams",
      "email": "jane.williams@mycompany.com"
    }
  ]
}
```

Errors

Every response is a JSON object, with HTTP status 200: check the `ok` field. When `ok` is false, `code` is a stable error code, meant to be tested by your program, and `message` is a human-readable description, which may change in time. The only exceptions are HTTP status 429 (too many requests, see Request limits) and, rarely, HTTP status 5xx (server error).

Validation errors also contain the `errors` object, with the invalid fields and the problems found (the `error` key is kept for backward compatibility):

```
{
  "ok": false,
  "code": "validation_error",
  "message": "Validation error!",
  "error": "Validation error!",
  "errors": {
    "weight": ["The field is required"]
  }
}
```

CODE	MEANING
`missing_key`	The API key is missing (neither the `X-Api-Key` header, nor the `key` parameter).
`invalid_key`	The API key is not valid.
`missing_user_id`	The `user_id` parameter is missing.
`invalid_user_id`	The `user_id` doesn't belong to your company.
`banned`	The company or the user account is banned. See also `banned` and `ban_reason`.
`validation_error`	One or more parameters are invalid. See `errors`.
`unresolved_place`	A place could not be recognized. See `field` (from or to) and `unresolved` (the place).
`not_a_locality`	The place must be a locality (`from` and `to`, for loads).
`route_too_broad`	The truck route is too broad (`trucks/publish`).
`duplicate`	A similar ad already exists. See `id`, `user_id` and `url` (the existing ad), and Duplicates.
`missing_id`	The `id` or `oid` parameter is missing.
`not_found`	The ad doesn't exist (anymore) or doesn't belong to the given `user_id`.
`no_active_users`	The company has no active users (`users/list`).
`too_many_requests`	Too many requests (HTTP status 429). Wait the number of seconds from the `Retry-After` header, then try again.
`internal_error`	Unexpected error on our side. Please write us at api@cargopedia.net.
`api_disabled`	The API is temporarily disabled.