

Cargopedia API documentation, v1.20231105

SEVERAL NOTES REGARDING THE API

1. In order to use the Cargopedia API, you – the user – MUST have a Cargopedia account and use a valid **API KEY / USER ID** combination. You can get your API details from your Cargopedia account, by accessing Settings → Company settings → API access data.
2. By default, your (any) user account is allowed to use the API, but load publishing is sandboxed, so the ads are not visible on the site for the other users, only for you. This allows you to safely test your API integration, without any risks. Once you are ready to integrate our API, send us an e-mail to api@cargopedia.net in order to lift that limitation. Be sure at that stage that you MASTER the API (i.e. thoroughly test every feature that you'll use, like add/edit/delete operations etc.).
3. Each API request can be made either by GET (parameters provided in URL) or by POST. Your choice.
4. Each response is a JSON object, like this:

```
{
  "ok": true,
  "message": "THE LOAD #710717 HAS BEEN SUCCESSFULLY ADDED.",
  "id": 710717
}
```

If the request is unsuccessful, then “ok” will be false. Moreover, if the account is banned, the keys “banned” (set to true) and “ban_reason” will be present in the response, like this:

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

5. Currently there are no API limits. If we impose certain limits in the future we will notify you.
6. The API shall be used responsibly, in good faith. Do not ABUSE it to spam the site with fake ads or other malicious activity, otherwise – besides banning your account – we reserve the right to take legal action against you and your company.
7. If you have questions or suggestions to improve the API, write us at please send an e-mail to api@cargopedia.net.

Subject	Page
TYPE GLOSSARY	3
ADDING/UPDATING A LOAD	4
DELETING A LOAD	7
UPPING/REFRESHING A LOAD	8
ADVANCED FUNCTIONALITY	
USING YOUR OWN IDS	9
LISTING COMPANY USERS	10

PLACE Can be a locality, region or country, if not specified otherwise.

The system is very flexible regarding place names. For example, all these are accepted:

1. Bucharest	city (simplest, but not recommended)
2. Bucharest, Romania	city, country name
3. Bucharest, RO	city, country code
4. Curtea de Arges, Arges, Romania	city, region, country
5. Curtea de Arges, Arges, RO	city, region, country code
6. RO-115300	postal code only
7. RO-115300, Curtea de Arges, RO	postal code, city, country code
8. RO-115300, Curtea de Arges, Arges, RO	postal code, city, region, country code

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

- For best results (and to be failsafe), we recommend that **if you have the postal code, to provide it, but ideally together with city name too, like in example (7)**. We do not recommend to only provide postal code, as it is not always reliable. If you don't have the postal code, we recommend using <city, region, country code> variant, if you know the region. If the user did not provide it, then <city, country code>. Even there is a risk of being two localities with the same name, the system is smart and will match first the most big/important city, usually that's the right/wanted one.
- Even if you can use any language/naming variant when providing a city - for example **Bucharest** (English name) or **București** (Romanian name) or **Bucarest** (Spanish name) and so on – it is best to provide the names in the same language as the **lang** parameter. For example, if you use English names (Bucharest etc.), set lang=en. If you prefer using Romanian names (București etc), then set lang=ro. For Spanish, lang=es, for French, lang=fr, and so on.

TRUCK TYPE Can have the following values: tilt (or tent), box, cartransporter, flatbed, openbody, tipper, tractorunit, container, animaltransporter, crane, tank, liquidfoodtank.

DATE Must be provided in YYYY-MM-DD format. 2023-10-30 means, for example, 30 October 2023.

INTEGER Must be an integer number.

DECIMAL Can be an integer or decimal number. Both decimal separators – comma and dot – are supported.
Examples: 5, 10.5, 20,7

BOOLEAN Must be 0 (false) or 1 (true).

TEXT A text block. Usually can be of any size, if not specified otherwise.

STRING(LENGTH) A string. Usually can be of any size, if not specified otherwise.

ARRAY(TYPE) Multiple values, separated by comma. For example: **100,101** or **blue,white**. Basically, this is a string.

ADDING/UPDATING A LOAD

[GET/POST]

<https://www.cargopedia.net/api/v1/loads/publish>

Parameters:

key STRING(15) *required*

Your Cargopedia API key

user_id INTEGER *required*

Your Cargopedia user id

from PLACE *required*

Loading/Departing place. Must be a locality.

to PLACE *required*

Unloading /Arrival place. Must be a locality. If you have multiple unloadings, in different cities, you can separate them with „|“. For example: Wien,Austria|Schwechat,Austria.

lang STRING(2) *optional*

The language in which you specify the from and to fields. It is used to parse locality/region/country names. For example:

- ...&from=București,RO&to=Varșovia,PL&lang=ro...
- ...&from=Bucharest,RO&to=Warsaw,PL&lang=en...

Default: en

valid_from DATE *optional*

The start date of the ad. If omitted, will be considered as „today“.

valid_to DATE *optional*

The end date of the ad. If omitted, will be considered as 7 days after valid_from

truck_type TRUCK TYPE | ARRAY(TRUCK TYPE) *required*

The required truck types. Can be one or more truck types, separated by comma.

For example: 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 cubic meters (m³). If omitted, it is automatically calculated by “length x width x height” formula, given all these parameters are provided.

groupage STRING (FTL|LTL) *optional*

Omitted = No preference; FTL = Full truck; LTL = Grouped transport

description TEXT *optional*

The ad description.

palletized BOOLEAN *optional*

Set it to 1 if the freight is palletized.

Default: 0 (false)

pallets INTEGER *optional*

The number of pallets. Required if palletized is set to 1.

pallet_type STRING(3) *optional*

The pallet type. Can have one of these values:

EUR = European pallet

IND = Industrial pallet

AME = American pallet

ASI = Asian pallet

AUS = Australian pallet

OTH = Other type of pallet

payment INTEGER *optional*

Payment time. Can have one of these values:

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 (unspecified)

price DECIMAL *optional*

The offered price.

price_unit DECIMAL *optional, required with: price*

The offered price currency. Required if price is given.

VAT BOOLEAN *optional*

Does the price include VAT?

Default: 0 (false)

ADR BOOLEAN *optional*

Set it to 1 if you transport dangerous goods.

Default: 0 (false)

reefer BOOLEAN *optional*

Set it to 1 if you need a truck with cooling system. Can be set in combination with truck_type=box.

Default: 0 (false)

lift BOOLEAN *optional*

Set it to 1 if you require a truck equipped with lift.

Default: 0 (false)

walking_floor BOOLEAN *optional*

Set it to 1 if you require a truck equipped with walking floor.

Default: 0 (false)

sandbox BOOLEAN *optional*

Set it to 1 if you are just testing the API and don't want the load to be visible on the site, when published. **Note that while your account is not yet approved for load publishing via API, sandbox is enforced to 1 (true), so this parameter has no effect in that case.**

Default: 0 (false)

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
```

A successful response will look like:

```
{
  "ok": true,
  "message": "THE LOAD #710717 HAS BEEN SUCCESSFULLY ADDED.",
  "id": 710717
}
```

Notice the returned id parameter. This is the id number Cargopedia assigns to your ad. You can use this number to edit or delete your ad.

For example, if you want to edit your ad, add **&id=710717** to your request, like:

```
https://www.cargopedia.net/api/v1/loads/publish?key=<key>&user_id=<user_id>&from=Berlin,Germany&to=Warsaw,Poland&weight=5&truck_type=box,tilt&palletized=1&pallets=6&pallet_type=EUR&id=710717
```

```
{
  "ok": true,
  "message": "THE LOAD #710717 HAS BEEN SUCCESSFULLY EDITED.",
  "id": 710717
}
```

Note: Cargopedia allows you to use your own ids when adding ads. This is especially useful if you already use an TMS/ERP to manage your loads/trucks and you want to insert/track your Cargopedia ads by the same IDs you use in your application. For more information, see [Advanced Functionality chapter](#).

DELETING A LOAD (OR MORE)

[GET/POST]

<https://www.cargopedia.net/api/v1/loads/delete>

Parameters:

key STRING(15) *required*

Your Cargopedia API key

user_id INTEGER *required*

Your Cargopedia user id

id INTEGER|ARRAY(INT) *required*

The ID that Cargopedia returned when you previously published the ad.

OR

oid STRING|ARRAY(STRING) *required*

Own ID (if you gave one). For more information, see Advanced Functionality chapter.

Example request (GET):

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

A successful response will look like:

```
{
  "ok": true,
  "message": "1 load(s) successfully deleted.",
  "ids": [
    710717
  ]
}
```

Note: You can also submit multiple ids in order to delete more than a load at once. For example:

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

will produce something like:

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

UPPING / REFRESHING A LOAD (OR MORE)

Upping (or "refreshing") a load means bringing it to the front page or 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>

Parameters:

key STRING(15) *required*

Your Cargopedia API key

user_id INTEGER *required*

Your Cargopedia user id

id INTEGER|ARRAY(INT) *required*

The ID that Cargopedia returned when you previously published the ad.

OR

oid STRING|ARRAY(STRING) *required*

Own ID (if you gave one). For more information, see Advanced Functionality chapter.

Example request (GET):

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

A successful response will look like:

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

Note: You can also submit multiple ids in order to delete more than a load at once. For example:

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

will produce something like:

```
{
  "ok": true,
  "message": "2 load(s) successfully upped.",
  "upped": [
    710705,
    710706
  ],
  "not_upped": []
}
```


Note: A load can be upped once in an hour (60 min). If you try to up a load more recently posted, it will not be upped, instead, it will be returned in the not_upped array.

ADVANCED FUNCTIONALITY

OWN IDS

When adding ads, Cargopedia automatically assigns and returns ID numbers.

Take this example:

```
{
  "ok": true,
  "message": "THE LOAD #1841 HAS BEEN SUCCESSFULLY ADDED.",
  "id": 1841
}
```

So, now our load has the id **1841**. You can use later this id to delete the load via API. Something like:

https://www.cargopedia.net/api/v1/loads/delete?key=<key>&user_id=<user_id>&id=1841

BUT, Cargopedia allows you to use specific IDs for your load/truck ads. This is especially useful if you already use an TMS/ERP to manage your loads/trucks and you want to insert/track your Cargopedia ads by the same IDs you use in your application.

Concrete example:

Suppose your load has id 5021 in your database. When you delete load #5021 from your program you would prefer to also delete de load #5021 from Cargopedia. Something like:

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

and NOT

https://www.cargopedia.net/api/v1/loads/delete?key=<key>&user_id=<user_id>&id=1841 (less nice, would actually require storing the id number Cargopedia gave you in your database).

Fortunately, Cargopedia allows you to specify a **oid** (own id), and you very well can add loads or trucks like this:

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

Now, supposing you want to delete this load, make a request to:

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

To sum it up, you can use **oid** (own id) instead of **id** (Cargopedia assigned id) wherever you want: add, update, delete etc.

LISTING COMPANY USERS

Returns a list of company's active users. This is useful for an easier TMS/ERP integration.

[GET/POST]

<https://www.cargopedia.net/api/v1/users/list>

Parameters:

key STRING(15) *required*

Your Cargopedia API key

Example request (GET):

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

A successful response will look like:

```
{
  "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"
    }
  ]
}
```