

---

# CargoSnap API documentation

## Contents

Overview.....	3
Version Information .....	3
Get Started .....	3
API Endpoints .....	3
Authentication.....	3
Supported HTTP Methods .....	3
Error Codes.....	3
Resource Information.....	4
Resource: files (GET).....	5
Parameters .....	5
Response values .....	7
Example Request .....	9
Resource: files (POST).....	10
Parameters .....	10
Response values .....	10
Example Request .....	10
Resource: fields (POST) .....	12
Route .....	12
Parameter.....	12
Body Parameters .....	12
Response values .....	13
Example Request .....	13
Resource: reports (POST) .....	14
Parameters .....	14
Response values .....	15
Example Request .....	15
Resource: shares (GET).....	16
Parameters .....	16
Response values .....	17



Example Request .....	17
Resource: forms/{id} (GET).....	19
Parameters .....	19
Response values .....	20
Example Request .....	21
Webhook: snaps.....	22
POST values .....	22
Webhook: forms.....	29
POST values .....	29

## Overview

The CargoSnap RESTful API gives you programmatic access to all CargoSnap functionality, for any given file. The API will allow you to pull all information as well as provide you a link to the images. The information is yours, so may be used to populate fields in an own database, download the information or create reports, to name a few applications.

## Version Information

This page describes version 2.12 of the CargoSnap RESTful API.

## Get Started

Before you get started, you'll need to upgrade your license to an Enterprise version, or purchase the API functionality (which is totally worth it). Then, as a "Super Manager", or "License Admin", you may [register](#) your app. This will give you a unique token which is needed for using the API. Fill in the API request form, and this token will be generated for you.

## API Endpoints

All API resources are accessed from the <https://api.cargosnap.com/api/v2> endpoint, which supports **SSL only**. Accessing any API resource via vanilla HTTP will return an HTTP status of **400 Bad Request** in the response headers.

## Authentication

All calls to the CargoSnap API must be authenticated by a valid API token; this can be specified either as a URL query parameter or as an HTTP request header. If an invalid API token is passed or if the API token is missing, an HTTP status of **401 Unauthorized** will be returned in both the response headers and in the response status field.

A valid API token can be passed by the **token** query parameter, in the form **token=api-token**. If no token is provided as a query parameter, the token must be passed as a request header in the form **token: [API-TOKEN]**.

**So, for each API call, ensure to include the following field in the header of your call (be it GET or POST).**

### **token** required (string)

A valid API token; if not supplied as a parameter, a token must be supplied in the request header

**Example value:** **token=[API-TOKEN]**

## Supported HTTP Methods

All resources on the CargoSnap API support HTTP **GET** and **POST** only; if an attempt is made to access the API using any other method, an HTTP status of **405 Not allowed** will be returned in both the response headers and in the response status field.

## Error Codes

Wherever possible, the CargoSnap API tries to use the appropriate HTTP status code to indicate problems or errors when invoking an API resource; the status code and reason message are additionally returned in the **status** section of the response body as **status** and **reason** respectively.

Both the HTTP status and the API status should be checked; it is entirely possible to have an HTTP status of **200** returned because the API request was well formed but to have an API error code returned because the request generated an error or contained an invalid parameter. Specifically, attempting to request an invalid file would return an HTTP status of **200** and an API error code of **300**.

- 200** The API resource call was well formed and successfully handled; additional information may also be available in the **status** section of the response body.
- 400** An attempt was made to access an SSL-only API endpoint via vanilla HTTP.
- 400** A required parameter was missing or contained an invalid value.
- 401** An invalid CargoSnap API token was supplied or the CargoSnap API token was missing.
- 404** An invalid or unknown API resource was invoked.
- 405** An invalid HTTP request method was used to access an API resource.
- 422** A required parameter was missing or contained an invalid value.
- 500** A problem has occurred with the CargoSnap API.
- 503** A problem has occurred with the CargoSnap API.

Additional CargoSnap specific error details are also provided in the **status** section of the response body as follows.

## Authentication Failures

- 1** Authentication failed; missing required **token** parameter or header.
- 2** Authentication failed; invalid **token** parameter value or header value.
- 3** Authentication failed, the specified token is inactive.

## Optional Parameter Validation Failures

- 100** The **format** parameter is invalid or missing a display type.
- 101** The **file** parameter is invalid or missing a string value.
- 102** The **lang** parameter is invalid or missing a language code.

## Mandatory Parameter Validation Failures

- 200** Missing required **file** parameter.
- 201** Missing required **date** parameter.
- 202** Missing required **lang** parameter.
- 203** Missing required **format** parameter.

## File Failures

- 300** Invalid or non-existent file.
- 301** Invalid or non-existent file.

## Resource Information

Response format:  
JSON

Requires authentication?  
Yes

Timestamps are always in UTC and date/time formats are in ISO  
Example: 2020-07-05T20:40:00 would be a valid format to query start/end date fields by a timestamp

## Resource: files (GET)

The CargoSnap RESTful API gives you programmatic access to the "CargoSnaps" generated with the CargoSnap app, for any given file (or "barcode" as scanned in the app). The API will allow you to pull all information as well as provide you a link to the image. The information is yours, so may be used to populate fields in an own database, download the information or create reports, to name a few applications.

Call the Files with the /files route.

### Parameters

**reference** optional (string)

The used file reference, as a string. This represents the barcode (scan code) value used in the barcode scanner, as well as the value used on the platform.

**Example value:** `reference=76543234567`

**find** optional (string)

Finds the used file based on a partial match of a file reference. For example.

**Example value:** `find=765432`

Would match 123**765432**123 for there is a partial match.

**closed** optional (string)

Finds the used file based on a partial match of a file reference. For example.

**Example values:**

`closed=true` will return only files that are closed.

`closed=false` will return only files that are open.

`Omitting this parameter` will return all files, regardless if open or closed.

**startdate** optional (string)

A date/time field of the file. The API will return the files that were active at the time of the filedate.

The date/time format must be ISO-8601 (YYYY-MM-DD), and the time zone should be UTC. The time field is optional; we will assume the start of the day in the UTC time zone.

**Example value:** `startdate=2017-03-01`

**enddate** optional (string)

A date/time field of the file. The API will return the file that was active at the time of the filedate.

The date/time format must be ISO-8601 (YYYY-MM-DD), and the time zone should be UTC. The time field is optional; we will assume the end of the day in the UTC time zone.

**Example value:** `enddate=2017-03-01`

**updated\_start** optional (string)

A date/time field of the file. The API will return the file that was have been updated after the said date.

The date/time format must be ISO-8601 (YYYY-MM-DD), and the time zone should be UTC. The time field is optional; we will assume the start of the day in the UTC time zone.

**Example value:** `startdate=2017-03-01`

**updated\_end** optional (string)

A date/time field of the file. The API will return the file that was active at the time of the filedate. If filedate is not supplied and multiple files exist, each instance of the file will be supplied as separate object in the response. Separate instances of the file may be recognised by having a different "file\_id" value.

The date/time format must be ISO-8601 (YYYY-MM-DD), and the time zone should be UTC. The time field is optional; we will assume the end of the day in the UTC time zone.

**Example value:** `startdate=2017-03-01`

**planned language** optional (string)

A supported language as an [ISO 639-1](#) 2 letter code. Defaults to `en`

**Example value:** `language=fr`

**Limit** optional, default = 50, max = 250 (integer)

Limits the response to a certain number of fields. Note: if there are more records available, the response will be paginated. The response code will contain data about the number of available pages and the "next\_page\_url" will indicate how to obtain the next set of data.

**include[]** optional (string)

Possible values:

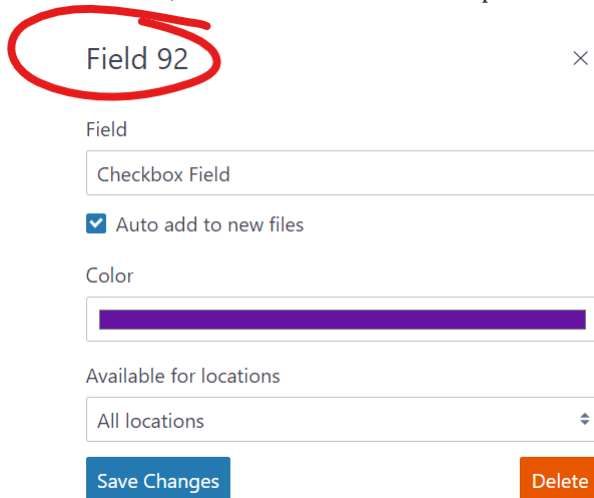
- `uploads` -> includes individual uploads (will increase filesize!)
- `geocoding` -> if uploads are included, will include geocoding data
- `forms` -> includes submitted forms
- `fields` -> includes custom fields (AKA tags)
- `locations` -> includes information about the location settings of the device(s) used to create the file

The include switch may be repeated multiple times: for example, if you need uploads and forms in the response, include "`include[]=uploads&include[]=forms`" in the get statement.

**Warning:** including these switches will lead to slower response times and higher data traffic. Please only include items you know you will need!

**field\_id[ id ]** optional (integer)

Filter to show only files containing Fields in the CargoSnap platform. You can obtain the id from the Field, in the edit screen of the specific field:



## Response values

All response values, except Boolean and Integers, are provided as string values and are appropriately escaped.

The content of the response will depend on the query parameters, the implemented features on CargoSnap as well as the actual available data. The most common data fields are:

### data

#### id

Unique ID of the CargoSnap file.

#### scan\_code

String value, of the file number which is queried: your reference code.

#### scan\_code\_format

In case a barcode was scanned to provide the file number, this value displays the type of barcode that was scanned.

#### created\_at

Date/timestamp of the time the first CargoSnap or Form was uploaded and registered in the platform, in the time zone UTC.

### data.uploads:

#### id

Unique ID of the CargoSnap image/upload.

#### device\_id

Unique ID of the CargoSnap device, used to take the image.

#### created\_at

Date/timestamp of the time the CargoSnap was uploaded and registered in the platform, in the time zone UTC.

#### scan\_date\_time

Date/timestamp of the time the CargoSnap was taken on the app, in the time zone UTC.

#### upload\_type

String representation of the type of data. Possible values include "snap", "manual" and "document".

**damage\_type\_desc**

Indicates if the CargoSnap app user indicated if any damage is to be seen on the image. The value will be displayed in text, in the language provided in the lang parameter (if available and if supplied).

**has\_damage**

Indicates if the CargoSnap app user indicated if any damage is to be seen on the image. Value=1 indicates such a case.

**document\_type\_desc**

Indicates the selected type of document, in case the user selected to create a document scan.

**latitude**

Latitude of the user, when creating the CargoSnap.

**longitude**

Longitude of the user, when creating the CargoSnap.

**image\_name**

Filename of the Snap

**image\_mime**

Snap Image type. Typically the image/jpeg

**comment**

If available, provides the comment the CargoSnap app user provided with the CargoSnap.

**image\_url**

URL path of the full, uncompressed, original image.

**image\_thumb**

URL path of 'thumbnail' of the image. The thumbnail contains the full content of the file, but is significantly reduced in filesize, making the presentation to the end-user significantly more responsive on most internet connections.

**workflow\_id**

If the image was taken as a part of a workflow, a unique identifier of the workflow is indicated here.

**workflow\_description**

If the image was taken as a part of a workflow, the description -as given in the platform- of the workflow is indicated here.

**workflow\_step\_id**

If the image was taken as a part of a workflow, a unique identifier of the specific step in the workflow is indicated here.

**workflow\_step\_description**

If the image was taken as a part of a workflow, the description of the specific step -as given in the platform- of the workflow is indicated here.

**data.form\_submits:**

**id**

Unique reference for the data to the submitted form.

**form.id**

Unique number of the specific form that was filled in, at the specific submit

**form.title**

Title of the form, as given.

**fields.label & id**

Label of the specific question, for the specific ID

**answers.value & id**



Title of the form, as filled in, for the specific ID

**data.fields:**

**name**

Tag (field) name, as given

**value**

Value, as submitted

### Example Request

GET [https://api.cargosnap.com/api/v2/files?format=json&token=\[MY-API-TOKEN\]&limit=10&include\[\]=uploads&include\[\]=forms&include\[\]=fields&field\\_id\[1\]=Apple&startdate=2019-01-15&enddate=2019-01-23](https://api.cargosnap.com/api/v2/files?format=json&token=[MY-API-TOKEN]&limit=10&include[]=uploads&include[]=forms&include[]=fields&field_id[1]=Apple&startdate=2019-01-15&enddate=2019-01-23)

This query would:

- list up to 10 results in JSON format
- for files created between Jan 15, 2019 and Jan 23<sup>rd</sup>, 2019
- Include details of uploads, forms and fields
- Select the specific fields (tags) where the tag id 1 has the value “Apple”

## Resource: files (POST)

The API will allow you to generate a new file.

Call the Files with the /files route.

### Parameters

**reference** required (max 255 char)

The used file reference, as a string. This represents the barcode (scan code) value used in the barcode scanner, as well as the value used on the platform. Note: make sure to URL encode any special characters!

**Example value:** `reference=76543234567`

**token** required

A valid API token; if not supplied as a parameter, a token must be supplied in the request header

**Example value:** `token=[API-TOKEN]`

**close** optional

A Boolean value which will indicate to close a file should a file with the existing ; if not supplied value is assumed to be "false"

**Example value:** `close=true`

**location** optional

A location reference. If not supplied, the file will be created in the first available known location. The location is a unique number and can be found in the platform, under "locations" and select the location: then the number will show in the header information.

### Response values

The content of the response will depend on the query parameters, the implemented features on CargoSnap as well as the actual available data. The typical response will be:

If file does not exist: "success"

If file already exists: "failed", including the ID of the file that is currently open under that reference.

If file already exists and parameter "close=true" is set: "success" (so the existing open file will be closed and a new file will be returned).

### Example Request

POST [https://api.cargosnap.com/api/v2/files?token=\[MY-API-TOKEN\]&reference=ABC12345&closed=false&location=1](https://api.cargosnap.com/api/v2/files?token=[MY-API-TOKEN]&reference=ABC12345&closed=false&location=1)

This query would:

Return a 200 response code, with the URL to the report in the content body:



**CARGOSNAP**

---

```
{  
  "response": "success",  
  "status": "File was successfully created",  
  "id": 183  
}
```

## Resource: fields (POST)

The API will allow you to add on or more (predefined) fields with values to an existing file.

### Route

Call the Files with the `/fields/{reference}` route.

### reference required (max 255)

The used file reference, as a string. This represents the barcode (scan code) value used in the barcode scanner, as well as the value used on the platform. Make sure it is properly escaped, should it contain any non URL style parameters.

Important: the file should exist and be open (otherwise an error will be returned)

**Example value:** `reference=76543234567`

### Parameter

#### token required

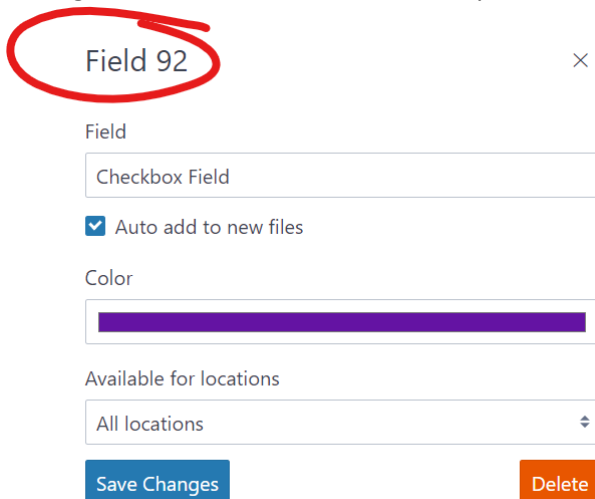
A valid API token; if not supplied as a parameter, a token must be supplied in the request header

**Example value:** `token=[API-TOKEN]`

### Body Parameters

In the Body of the POST, you can include one or more fields, with the value you would like to set, in an array (JSON). Make sure you properly escape the full body value.

The fields have an ID / Value combination. The ID can be obtained from the platform, under Global Settings -> Fields -> Select the field and you will be able to identify the id:



The screenshot shows a modal window for editing a field. The title is "Field 92" with a close button (X) on the right. The modal contains the following elements:

- A "Field" label above a text input field containing "Checkbox Field".
- A checked checkbox labeled "Auto add to new files".
- A "Color" label above a color picker showing a purple bar.
- An "Available for locations" label above a dropdown menu showing "All locations".
- Two buttons at the bottom: "Save Changes" (blue) and "Delete" (orange).

For values, please make sure the value types correspond to the field type (do not put text into a numerical field); as obviously this will lead to errors. The max field length is 255 characters.

## Response values

The content of the response will depend on the query parameters, the implemented features on CargoSnap as well as the actual available data. The typical response will be:

If file does not exist: "success"

If file already exists: "failed", including the ID of the file that is currently open under that reference.

If file already exists and parameter "close=true" is set: "success" (so the existing open file will be closed and a new file will be returned).

## Example Request

POST [https://api.cargosnap.com/api/v2/fields/ABC-123-abc?token=\[MY-API-TOKEN\]](https://api.cargosnap.com/api/v2/fields/ABC-123-abc?token=[MY-API-TOKEN])

With Body:

```
[
  {
    "id": "1", "value": "string for example"
  },
  {
    "id": "3", "value": "87654321654"
  }
]
```

This query would:

Add a string value to the Field with the ID 1 and a numerical value to the Field with ID 3.

Return a 200 response code, with the URL to the report in the content body:

```
{
  "response": "success",
}
```

## Resource: reports (POST)

The API will allow you to generate a new report and retrieve the download link for that report.

Call the Reports with the /reports route.

### Parameters

#### **files** required

files: array of files to be included in the new report

This represents the unique file reference as used on the platform. This value can be obtained from the platform, but more typically from the CargoSnap “Files-API”.

Note: maximum 100 unique files may be added into 1 report (using more will generate an error response).

#### **token** required

A valid API token; if not supplied as a parameter, a token must be supplied in the request header

**Example value:** token=[API-TOKEN]

#### **template** optional (default = 2)

Template id, may be obtained from the CargoSnap platform.

**Example value:** template: “12392”,

#### **filename** optional (by default, CargoSnap will generate a timestamp filename)

**Example value:** MyDocumentFromCargoSnap.docx

#### **settings** optional

Settings object has the following optional settings:

- *locale*: possible values: . Default en-US.
- *format*: possible values docx, pdf. Default docx
- *dateTime.zone*: possible values see: [https://en.wikipedia.org/wiki/List\\_of\\_tz\\_database\\_time\\_zones](https://en.wikipedia.org/wiki/List_of_tz_database_time_zones) (column “TZ database name”). Default UTC
- *dateTime.format*: for example “Y-m-d H:i:s”. Default Y-m-d H:i:s

#### **Example value:**

files: [

12345,  
134543,  
65433

```
],
settings: {
  locale: "en-US",
  format: "docx",
  dateTime: {
    zone: "Europe/Amsterdam",
    format: "Y-m-d H:i:s"
  }
}
```

### Response values

The content of the response will depend on the query parameters, the implemented features on CargoSnap as well as the actual available data. The typical response will be:

url: <https://cargosnapstorage.blob.windows.net/123/2345/gfbd24scdn5465hgre76fwscgfgfedsbcbggrf/MyDocumentFromCargoSnap.docx>

### Example Request

POST [https://api.cargosnap.com/api/v2/reports?token=\[MY-API-TOKEN\]](https://api.cargosnap.com/api/v2/reports?token=[MY-API-TOKEN])

POST DATA:

```
{
  "files": [
    12345,
    134543,
    65433
  ],
  "template": 12392,
  "filename": "HASU1234576-20190219",
  "settings": {
    "locale": "en-US",
    "format": "docx",
    "dateTime": {
      "zone": "Europe/Amsterdam",
      "format": "Y-m-d H:i:s T"
    }
  }
}
```

This query would:

- Return a 201 response code, with the URL to the report in the content body:

```
{ url:      "https://cargosnapstorage.blob.windows.net/123/2345/
            gfbd24scdn5465hgre76fwscgfgfedsbcbggrf/MyDocumentFromCargoSnap.docx" }
```

## Resource: share (GET)

The API will allow you to generate a new share and retrieve the share link for that file.

Call the Files with the /share route, i.e. <https://api.cargosnap.com/api/v2/share>

### Parameters

**reference** required (max 255)

The used file reference, as a string. This represents the barcode (scan code) value used in the barcode scanner, as well as the value used on the platform.

**Example value:** `reference=76543234567`

**token** required

A valid API token; if not supplied as a parameter, a token must be supplied in the request header

**Example value:** `token=[API-TOKEN]`

**expires** optional

A date/time field indicating when the share should expire. Upon expiration, the link will no longer be valid. If not supplied, the share will never expire.

The date/time format must be ISO-8601, and the time zone should be UTC. The **time field** is optional; we will assume the start of the day in the UTC time zone.

**Example value:** `expires=2017-03-01`

**language** optional

A supported language as an [ISO 639-1](#) 2 letter code. Sets the language of the shared file view. If not supplied, or invalid, the API defaults to the language setting of the location the reference belongs to.

**Example value:** `language=fr`

**dl** optional

The DL reference, as defined in CargoSnap. This allows a share being created for multiple recipients in 1 call. This is best used for sharing between companies, where we suggest the used dl reference corresponds to the customer reference id in the host system (WMS/TMS).

**Example value:** `dl=1234abc`

**email** optional

The email address of the recipient of the share. If not supplied, or invalid, the API defaults to the language setting of the location the reference belongs to.

**Example value:** `email=example@cargosnap.com`

**send\_email** optional (default = true)

If set to “false”, there will be no email send to the email recipient. This is handy if you want to send the sharing mail yourself.



*planned* **username** optional

Adds the user name of the person creating the share to the shared file and user interface.  
Defaults to the name given when generating the **token** (i.e. the name of the root system)

**Example value:** `username=example@cargosnap.com`

## Response values

All response values are provided as string values, and are appropriately escaped. The API will respond with a "Cache-Control=no-cache" header always.

### **response**

String, with value either "success" or "error"

### **status**

String, describing detailed (error) information.

### **share\_url**

Unique URL of the CargoSnap device, used to take the image.

## Example Request

```
GET /api/v2/share
?token=NjJPV25wcXZwV3pjRkFr*****
&reference=351606436
&expires=28-09-2017
&timezone=UTC
&language=en
&username=CargoSnap (optional, def=system api token)
```

URL: <https://api.cargosnap.com/api/v2/share>

```
GET https://api.cargosnap.com/api/v2/share?reference=123456789&token=MY-API-TOKEN&lang=en
```

```
curl --request GET \
--url 'https://api.cargosnap.com/api/v2/share?reference=123456789&token=MY-API-TOKEN&lang=en'
```

Example Result:

[https://api.cargosnap.com/api/v2/share?token=\\*\\*\\*\\*\\*=&reference=testgg](https://api.cargosnap.com/api/v2/share?token=*****=&reference=testgg)

Header: 200 SUCCESS

```
{
  "response": "success",
  "status": "SHARE CREATED",
```



```
"share_url":"https://platform.cargosnap.com/shared_files?token=fB4djl9x5nYZ3W*****&email=api_11294&l=en"
```

Header: 401 NOT AUTHORIZED

```
{  
  "response":"error",  
  "status":"Unauthorized"  
}
```

Header: 404 FILE NOT FOUND

```
{  
  "response":"error",  
  "status":"NO FILE FOUND"  
}
```

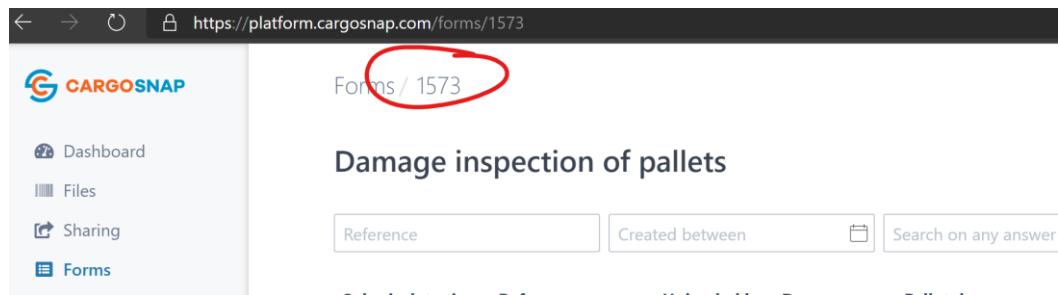
Header: 422

```
{  
  "response":"error",  
  "status":"INVALID TIMEZONE"  
}
```

## Resource: forms/{id} (GET)

The CargoSnap RESTful API gives you programmatic access to the "forms" submitted with the CargoSnap app or platform. The information is yours, so may be used to populate fields in an own database, download the information or create repots, to name a few applications.

Call the forms with the /forms/{id} route, where the {id} represents the ID of the specific form. This ID is an integer, which you may obtain from the platform on <https://platform.cargosnap.com/forms>, and then selecting the specific form you wish to get the data from, the specific ID will show up as a breadcrumb in the header of the data:



### Parameters

**reference** optional (max 255)

The used file reference, as a string. This represents the barcode (scan code) value used in the barcode scanner, as well as the value used on the platform.

**Example value:** `reference=76543234567`

**token** required

A valid API token; if not supplied as a parameter, a token must be supplied in the request header

**Example value:** `token=[API-TOKEN]`

**startdate** optional

A date/time field of the file. The API will return the files that were active at the time of the filedate.

The date/time format must be ISO-8601 (YYYY-MM-DD), and the time zone should be UTC. The time field is optional; we will assume the start of the day in the UTC time zone.

**Example value:** `startdate=2017-03-01`

**enddate** optional

A date/time field of the file. The API will return the file that was active at the time of the filedate.

The date/time format must be ISO-8601 (YYYY-MM-DD), and the time zone should be UTC. The time field is optional; we will assume the end of the day in the UTC time zone.

**Example value:** `enddate=2017-03-01`

**updated\_start** optional

A date/time field of the file. The API will return the file that was have been updated after the said date.

The date/time format must be ISO-8601 (YYYY-MM-DD), and the time zone should be UTC. The time field is optional; we will assume the start of the day in the UTC time zone.

**Example value:** `startdate=2017-03-01`

**updated\_end** optional

A date/time field of the file. The API will return the file that was active at the time of the filedate. If filedate is not supplied and multiple files exist, each instance of the file will be supplied as separate object in the response. Separate instances of the file may be recognised by having a different "file\_id" value.

The date/time format must be ISO-8601 (YYYY-MM-DD), and the time zone should be UTC. The time field is optional; we will assume the end of the day in the UTC time zone.

**Example value:** `startdate=2017-03-01`

**planned language** optional

A supported language as an [ISO 639-1](#) 2 letter code. Defaults to `en`

**Example value:** `language=fr`

**Limit** optional, default = 50, max = 250

Limits the response to a certain number of fields. Note: if there are more records available, the response will be paginated. The response code will contain data about the number of available pages and the "next\_page\_url" will indicate how to obtain the next set of data.

## Response values

All response values, except Boolean and Integers, are provided as string values and are appropriately escaped.

The content of the response will depend on the query parameters, the implemented features on CargoSnap as well as the actual available data. The most common data fields are:

**data**

**id**

Unique ID of the CargoSnap file.

**scan\_code**

String value, of the file number which is queried: your reference code.

**scan\_code\_format**

In case a barcode was scanned to provide the file number, this value displays the type of barcode that was scanned.

**created\_at**

Date/timestamp of the time the first CargoSnap or Form was uploaded and registered in the platform, in the time zone UTC.

**data.form\_submits:**

**id**

Unique reference for the data to the submitted form.

**form.id**

Unique number of the specific form that was filled in, at the specific submit

**form.title**

Title of the form, as given.

**fields.label & id**

Label of the specific question, for the specific ID

**answers.value & id**

Title of the form, as filled in, for the specific ID

## Example Request

GET [https://api.cargosnap.com/api/v2/forms/1234?format=json&token=\[MY-API-TOKEN\]&limit=10&startdate=2019-01-15&enddate=2019-01-23](https://api.cargosnap.com/api/v2/forms/1234?format=json&token=[MY-API-TOKEN]&limit=10&startdate=2019-01-15&enddate=2019-01-23)

This query would:

- Query submits on the form with the id = 1234
- list up to 10 results in JSON format
- for forms submitted between Jan 15, 2019 and Jan 23<sup>rd</sup>, 2019

## Webhook: snaps

Configure a webhook to be informed when a new snap/document is uploaded from a device onto the platform. The webhook will POST all information regarding the image to the endpoint you configure in the platform.

### POST values

Note: there may be a lot more data fields in the Webhook; depending on the data collected by the app and options you may have enabled (or not). Here follow some of the most relevant fields. Should you miss something, please check the actual POST from the webhook, or contact us.

All response values are provided as string values, and are appropriately escaped.

#### **id**

Unique reference for each Snap/Document

#### **scan\_code**

Scanned reference (e.g. the file barcode)

#### **device\_nick**

Given name of the device (app), used to take the image.

#### **scan\_date\_time**

Date/timestamp of the picture taken

#### **has\_damage**

Boolean to describe if the user indicated there is a problem with this snap.

#### **comment**

Comment entered by app user.

#### **image\_url**

URL to the full image

#### **image\_thumb**

URL to the thumbnail of the image

The full schema of the response is:

```
{
  "type": "object",
  "properties": {
    "has_damage": {
      "type": "boolean"
    },
    "upload_type": {
      "type": "string"
    },
    "include_in_share": {
```



```
    "type": "boolean"
  },
  "file_id": {
    "type": "integer"
  },
  "device_id": {
    "type": "integer"
  },
  "tenant_id": {
    "type": "integer"
  },
  "device_nick": {
    "type": "string"
  },
  "location_id": {
    "type": "integer"
  },
  "scan_code": {
    "type": "string"
  },
  "scan_date_time": {
    "type": "string"
  },
  "scan_code_format": {
    "type": "string"
  },
  "image_name": {
    "type": "string"
  },
  "image_size": {
    "type": "integer"
  },
  "image_mime": {
    "type": "string"
  },
  "image_path": {
    "type": "string"
  },
  "latitude": {
    "type": "string"
  },
  "longitude": {
    "type": "string"
  },
  "speed": {
    "type": "string"
  }
```



```
},
"accuracy": {
  "type": "string"
},
"comment": {},
"geocoding": {
  "type": "object",
  "properties": {
    "summary": {
      "type": "object",
      "properties": {
        "queryTime": {
          "type": "integer"
        },
        "numResults": {
          "type": "integer"
        }
      }
    }
  },
  "addresses": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "address": {
          "type": "object",
          "properties": {
            "buildingNumber": {
              "type": "string"
            },
            "streetNumber": {
              "type": "string"
            },
            "routeNumbers": {
              "type": "array"
            },
            "street": {
              "type": "string"
            },
            "streetName": {
              "type": "string"
            },
            "streetNameAndNumber": {
              "type": "string"
            },
            "countryCode": {
```





```
        "type": "string"
    },
    "countrySubdivision": {
        "type": "string"
    },
    "municipality": {
        "type": "string"
    },
    "postalCode": {
        "type": "string"
    },
    "country": {
        "type": "string"
    },
    "countryCodeISO3": {
        "type": "string"
    },
    "freeformAddress": {
        "type": "string"
    },
    "boundingBox": {
        "type": "object",
        "properties": {
            "northEast": {
                "type": "string"
            },
            "southWest": {
                "type": "string"
            },
            "entity": {
                "type": "string"
            }
        }
    },
    "extendedPostalCode": {
        "type": "string"
    },
    "localName": {
        "type": "string"
    }
}
},
"position": {
    "type": "string"
}
},
```



```
        "required": [
            "address",
            "position"
        ]
    }
}
},
"properties": {
    "type": "object",
    "properties": {
        "weather": {
            "type": "object",
            "properties": {
                "weather_code": {
                    "type": "integer"
                },
                "wind_speed": {
                    "type": "integer"
                },
                "wind_degree": {
                    "type": "integer"
                },
                "wind_dir": {
                    "type": "string"
                },
                "humidity": {
                    "type": "integer"
                },
                "temperature": {
                    "type": "integer"
                },
                "dewpoint": {
                    "type": "integer"
                },
                "precip": {
                    "type": "integer"
                },
                "pressure": {
                    "type": "integer"
                },
                "visibility": {
                    "type": "integer"
                },
                "cloudcover": {
                    "type": "integer"
                }
            }
        }
    }
}
```



```
    }
  }
}
},
"updated_at": {
  "type": "string"
},
"created_at": {
  "type": "string"
},
"id": {
  "type": "integer"
},
"image_url": {
  "type": "string"
},
"image_thumb": {
  "type": "string"
},
"image_thumb_sm": {
  "type": "string"
}
}
}
```

Example response:

```
{
  "id": 4316126,
  "tenant_id": 1,
  "device_id": 1292,
  "file_id": 632873,
  "scan_code": "2020-05-12T07:58:51Z",
  "scan_code_format": "timestamp",
  "device_nick": "mmxS9",
  "image_path": "uploads/20200512/[...]_af44_fe9f5aca56d8.jpg",
  "image_name": "c336f111-1df5-46a0-af44-fe9f5aca56d8.jpg",
  "image_mime": "image/jpeg",
  "image_size": 215452,
  "scan_date_time": "2020-05-12T08:00:21+00:00",
  "latitude": "52.0891014",
  "longitude": "5.1270544",
  "speed": "0",
  "accuracy": "12.125",
  "has_damage": false,
  "damage_type_desc": null,
  "damage_type_id": null,
}
```



```
"comment": null,  
"created_at": "2020-05-12T08:00:23+00:00",  
"updated_at": "2020-05-12T08:00:23+00:00",  
"deleted_at": null,  
"deleted_reason_id": null,  
"include_in_share": true,  
"updated_by": null,  
"upload_type": "snap",  
"document_type_desc": null,  
"document_type_id": null,  
"location_id": 947,  
"workflow_id": 1120,  
"workflow_step_id": 5881,  
"workflow_description": "3 forms",  
"workflow_step_description": "Take photos upclose",  
"image_url": "https://cargosnapstorage.[...]46a0_af44_fe9f5aca56d8.jpg",  
"image_thumb": "https://images.cargosnap.net/thumbnails/md/[...]1_df5_aca56d8.jpg",  
"image_thumb_sm": "https://images.cargosnap.net/thumbnails/[...]44_fe9f5aca56d8.jpg",  
}
```

## Webhook: forms

Configure a webhook to be informed when a new form is uploaded from a device or from the browser into the platform. The webhook will POST all information regarding the form to the endpoint you configure in the platform.

### POST values

Note: there may be a lot more data fields in the Webhook; depending on the data collected by the app and options you may have enabled (or not). Here follow some of the most relevant fields. Should you miss something, please check the actual POST from the webhook, or contact us.

All response values are provided as string values and are appropriately escaped.

#### **id**

Unique reference for each submitted form

#### **scan\_code**

Scanned reference (e.g. the file barcode)

#### **device\_nick / nick**

Given name of the device (app), used to fill in the form.

#### **submit\_date\_time**

Date/timestamp of the form being saved

#### **Form.title**

The given title of the form

Form.fields is an array (depending on the form length), with the following important key values:

#### **Form.fields[].name**

The given name (question) of the form field

#### **Form.fields[].value**

The given output of the form field

The full schema of the response is:

```
{
  "type": "object",
  "properties": {
    "tenant_id": {
      "type": "integer"
    },
    "form_id": {
      "type": "integer"
    },
    "device_id": {
      "type": "integer"
    }
  }
}
```



```
},
"file_id": {
  "type": "integer"
},
"submit_date_time": {
  "type": "string"
},
},
"nick": {
  "type": "string"
},
},
"location_id": {
  "type": "integer"
},
},
"scan_code": {
  "type": "string"
},
},
"shared": {
  "type": "boolean"
},
},
"latitude": {
  "type": "string"
},
},
"longitude": {
  "type": "string"
},
},
"gps_timestamp": {
  "type": "string"
},
},
"workflow_id": {
  "type": "integer"
},
},
"workflow_description": {
  "type": "string"
},
},
"workflow_step_id": {
  "type": "integer"
},
},
"workflow_step_description": {
  "type": "string"
},
},
"geocoding": {
  "type": "object",
  "properties": {
    "summary": {
      "type": "object",
      "properties": {
```



```
    "queryTime": {
      "type": "integer"
    },
    "numResults": {
      "type": "integer"
    }
  }
},
"addresses": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "address": {
        "type": "object",
        "properties": {
          "buildingNumber": {
            "type": "string"
          },
          "streetNumber": {
            "type": "string"
          },
          "routeNumbers": {
            "type": "array"
          },
          "street": {
            "type": "string"
          },
          "streetName": {
            "type": "string"
          },
          "streetNameAndNumber": {
            "type": "string"
          },
          "countryCode": {
            "type": "string"
          },
          "countrySubdivision": {
            "type": "string"
          },
          "municipality": {
            "type": "string"
          },
          "postalCode": {
            "type": "string"
          }
        }
      }
    }
  }
}
```



```
    "country": {
      "type": "string"
    },
    "countryCodeISO3": {
      "type": "string"
    },
    "freeformAddress": {
      "type": "string"
    },
    "boundingBox": {
      "type": "object",
      "properties": {
        "northEast": {
          "type": "string"
        },
        "southWest": {
          "type": "string"
        },
        "entity": {
          "type": "string"
        }
      }
    },
    "extendedPostalCode": {
      "type": "string"
    },
    "localName": {
      "type": "string"
    }
  },
  "position": {
    "type": "string"
  }
},
"required": [
  "address",
  "position"
]
}
}
},
"updated_at": {
  "type": "string"
},
```





```
"created_at": {
  "type": "string"
},
"id": {
  "type": "integer"
},
"upload_type": {
  "type": "string"
},
"scan_date_time": {
  "type": "string"
},
"form": {
  "type": "object",
  "properties": {
    "id": {
      "type": "integer"
    },
    "title": {
      "type": "string"
    },
    "tenant_id": {
      "type": "integer"
    },
    "created_at": {
      "type": "string"
    },
    "updated_at": {
      "type": "string"
    },
    "deleted_at": {},
    "default_shared": {
      "type": "boolean"
    },
    "location_id": {
      "type": "integer"
    },
    "display_empty_values": {
      "type": "boolean"
    },
    "order": {
      "type": "integer"
    },
    "language": {},
    "location_filter": {},
    "fields": {
```



```
"type": "array",
"items": {
  "type": "object",
  "properties": {
    "form_id": {
      "type": "integer"
    },
    "value": {
      "type": "string"
    },
    "created_at": {
      "type": "string"
    },
    "updated_at": {
      "type": "string"
    },
    "field_name": {
      "type": "string"
    },
    "type": {
      "type": "string"
    },
    "label": {
      "type": "string"
    },
    "id": {
      "type": "integer"
    }
  },
  "required": [
    "form_id",
    "value",
    "created_at",
    "updated_at",
    "field_name",
    "type",
    "label",
    "id"
  ]
}
}
```

Example response:

```
{
  "id": 228900,
  "form_id": 1573,
  "device_id": 2927,
  "file_id": 903926,
  "submit_date_time": "2020-09-18T07:58:17+00:00",
  "created_at": "2020-09-18T07:58:17+00:00",
  "updated_at": "2020-09-18T08:01:53+00:00",
  "deleted_at": null,
  "location_id": 9,
  "scan_code": "9780984380213",
  "nick": "DK",
  "user_id": null,
  "shared": true,
  "latitude": "51.9923665",
  "longitude": "4.5037559",
  "gps_timestamp": "1600415844057",
  "workflow_id": 1771,
  "workflow_step_id": 10758,
  "workflow_description": "Damage Inspection",
  "workflow_step_description": "Fill in damage form",
  "upload_type": "form",
  "scan_date_time": "2020-09-18T07:58:17.827000Z",
  "form": {
    "id": 1573,
    "title": "Damage inspection of pallets",
    "tenant_id": 1,
    "created_at": "2020-07-22T08:31:59+00:00",
    "updated_at": "2020-07-22T08:31:59+00:00",
    "deleted_at": null,
    "default_shared": true,
    "location_id": 1,
    "display_empty_values": true,
    "order": null,
    "language": null,
    "location_filter": null,
    "fields": [
      {
        "form_id": 1573,
        "value": "black",
        "created_at": "2020-09-18T07:58:18+00:00",
        "updated_at": "2020-09-18T08:00:14+00:00",
        "field_name": "",
        "type": "single_select",
        "label": "color of the pallet",
        "id": 17072
      },
      {
        "form_id": 1573,
        "value": "Right",
```



```
"created_at": "2020-09-18T07:58:18+00:00",
"updated_at": "2020-09-18T07:58:18+00:00",
"field_name": "",
"type": "single_select",
"label": "Damage area",
"id": 15974
},
{
  "form_id": 1573,
  "value": "Top",
  "created_at": "2020-09-18T07:58:18+00:00",
  "updated_at": "2020-09-18T07:58:18+00:00",
  "field_name": "",
  "type": "single_select",
  "label": "Pallet damage area",
  "id": 15975
}
{
  "form_id": 1573,
  "value":
  "https:\\\\cargosnapstorage.blob.core.windows.net\\tenant1\\drawings\\20200918\\84YL1eXGhIJUExW
G8t38sD\\drawing.svg",
  "created_at": "2020-09-18T07:58:18+00:00",
  "updated_at": "2020-09-18T07:58:18+00:00",
  "field_name": "",
  "type": "drawing",
  "label": "sign",
  "id": 16129
}
]
}
}
```

## Webhook: workflow\_run\_complete

Configure a webhook to be informed when a workflow has been completed by a device. The webhook will POST all information regarding the workflow to the endpoint you configure in the platform.

**Important:** note that because of the offline nature of devices (sometimes), it may occur that a workflow has been completed on/by a device, before uploading all data pertaining to that workflow. In that case such information will thus be missing. This is an exceptional situation, but when a complete payload is required, your logic may need to include a (delayed) API call to collect the full data set.

### POST values

Note: there may be a lot more data fields in the Webhook; depending on the data collected by the app and options you may have enabled (or not). Here follow some of the most relevant fields. Should you miss something, please check the actual POST from the webhook, or contact us.

All response values are provided as string values and are appropriately escaped.