ARH

# CARMEN® GO

## REST API

# REFERENCE MANUAL

WWW.ARH.HU

# CARMEN® GO

## REST API REFERENCE MANUAL

Document version: V.1.1.0.

### Table of Contents

# INTRODUCTION

A RESTful (**RE**presentational **S**tate **T**ransfer) API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.

A RESTful API -- also referred to as a RESTful web service -- is based on representational state transfer (REST) technology, an architectural style and approach to communications often used in web services development.

With CARMEN® GO rest API, it is possible to set the different inputs, ANPR settings and get event data from the internal database. The output interface returns data in JSON format.

# REST API OVERVIEW

## 1. GENERAL

**Root path:** /api

**Global URL parameters:**

format – JSON Output format (pretty | minimal, Default: minimal)

**Error response:**

**Codes:**

- 400 (BAD_REQUEST)
- 405 (METHOD_NOT_ALLOWED)
- 403 (FORBIDDEN)
- 500 (INTERNAL_SERVER_ERROR)

**Content template:**

{"error" : [Error message], "type": [Error type]}

**Error types:**

- Client – User side error (e.g.: Invalid query or method)
- Security – Security problem (e.g.: Access denied)
- Server – Server-side problem (e.g.: Not enough memory)

## 2. VERSION

Software version and metadata.

**URL:** /api/version

**Method:** GET

**Success response:**

- **Code:** 200 (OK)

- **Content:**


```
{
    „name“ : „Carmen GO“,
    „major“ : 1,
    „minor“ : 1,
    „build“ : „945“,
    „uuid“ : “8487e2c6-7556-b415-0280-1831bf521b75“,
}
```


**Error response:**

- **Code:** 400 (BAD_REQUEST)
- **Content:** {"error" : [Error message], "type": [Error type]}

# LICENSE HANDLING

## 1. GET LICENSES

It gives information about the software licenses.

**URL:** /api/license
**Method:** GET

**Success response:**

- **Code:** 200 (OK)
- **Content:**

{
„device" : 1, (*number of hardware key*)
„stream" : 8, (*number of streams*)
„core" : 4 (*number of ANPR cores*)
}

**Error response:**

- **Code:** 400 (BAD_REQUEST)
- **Content:** {"error" : [Error message], "type": [Error type]}

# COUNTRIES

## 1. GET COUNTRIES

Proxy call. Lists the installed ANPR profile (country) list – dependent on regional engines installed. If the ANPR server is not available, it returns internal server error.

**URL:** /api/countries
**Method:** GET

**Success response:**

- **Code:** 200 (OK)
- **Content:**

```
[
        {
                „name" : „USA",                 (name of the country)
                „region" : „North America",     (region of the country)
                „code" : „usa"                  (identification code of the country)
                „available" : true              (all conditions are given to use – engine, license)
        },
        ...
]
```

**Error response:**

- **Code:** 400 (BAD_REQUEST)
- **Content:** {"error" : [Error message], "type": [Error type]}

# PROCESSES (STREAMS)

## 1. GET PROCESS STATUS

All the information related to a stream, including name, url, format, trigger method, etc.. is called a "process". "Get process status" results this parameter list.

**URL:** /api/process/"id"
**Method:** GET
**URL parameter:**
id – Event id (*Required*)
**Success response:**

- **Code:** 200 (OK)
- **Content:**


```
{
	„name" : „Front",
	„url" : „http://192.0.2.3/video.mp4",
	„format" : „mjpeg",
	„source" : "http",
	„decode" : „software",
	„status" : „Running",
	„country" : „hungary",
	„method" : „generalprocess",
	„stream" : 9848,
	„roi" : [
		{"x" : 12.3456, "y" : 10.0000},
		{"x" : 22.3456, "y" : 100.0000},
		{"x" : 42.3456, "y" : 70.2345},
		{"x" : 12.2340, "y" : 30.0000}
		],
	„sensitivity" : „50",
	„host" : "192.0.2.3",
	„manufacturer" : „ARH",
	„model" : "Freeway"
	„other" : ""
	„type" : „camera",
}
```

**Where:**

- **name**: *name of the stream*
- **url**: *source of the stream*
- **format**: *live view stream format*
- **source**: *source type of the stream (http or rtsp)*
- **decode**: *mode of video decoder, (only "software" available currently)*
- **status**: *status of the process*
- **country**: *country set for the actual stream or video*
- **method**: *image processing / trigger algorithm*
- **stream**: *stream port, for external streaming*
- **roi**: *selected area of interest in the stream or video*
- **sensitivity**: *sensitivity of trigger algorithm, (only for "Stop and Go" processing method)*
- **host:** *ip address of the camera*
- **manufacturer**: *manufacturer of the camera*
- **model:** *camera model*
- **other:** *camera dependent data (if available)*
- **type** : *type of source (camera or video)*

**Error response:**

- **Code:** 400 (BAD_REQUEST)
- **Content:** {"error" : [Error message], "type": [Error type]}

## 2. START/STOP PROCESS

Start or stop video processing of a given stream.

**URL:** /api/process/"id"
**Method:** POST
**Data parameter:**
status – [Start|Stop] (*Required*)

**Success response:**

- **Code:** 204 (NO_CONTENT)

**Error response:**

- **Code:** 400 (BAD_REQUEST)
- **Content:** {"error" : [Error message], "type": [Error type]}

## 3. MODIFY PROCESS

Change video/stream processing parameters.

**URL:** /api/process/"id"
**Method:** PATCH

**Data parameter:**

- **name** – Name of stream (*Optional*)
- **url** – Stream url (*Optional*)
- **format** – stream format (mjpeg or H.264) (*Optional*)
- **source** – source type of the stream (http or rtsp) (*Optional*)
- **country** – Country (*Optional*)
- **method** – Method (*Optional*)
- **stream** – stream port, for external streaming (*Optional*)
- **host:** Camera Network address (Optional)
- **sensitivity**: sensitivity of trigger algorithm, (only for "Stop and Go" processing method) (Optional)
- **manufacture** – Device manufacturer (*Optional*)
- **model** – Device model (Optional)
- **roi** – Region of interest (Optional)

**Success response:**

- **Code:** 204 (NO_CONTENT)

**Error response:**

- **Code:** 400 (BAD_REQUEST)
- **Content:** {"error" : [Error message], "type": [Error type]}

# 4. AVAILABLE METHODS (EVENT TRIGGERS)

Image preselection for ANPR is done by software triggers called "methods". Installed method list and descriptions are available through "methods".

**URL:** /api/process/methods
**Method:** GET

**Success response:**

- **Code:** 200 (OK)
- **Content:**

```
[
        {
                „id" : „generalprocess"
                „name" : "General video processor"
                „description" : „CARMEN GO General algorithm. Recommended for..."
        },
        ...
]
```

# VIDEOS

## 1. UPLOAD VIDEO

Upload video to the local video storage of Carmen GO.

**URL:** /api/videos
**Method:** POST

**Data parameter:**

Video file content

**File size limit:** 5GB

**Supported MIME types:**

- mp4 - video/mp4
- mp4 - application/octet-stream
- mpg - video/mpeg
- mpg – application/octet-stream
- mpeg – video/mpeg
- mpeg – application/octet-stream
- mov – video/quicktime
- avi – video/avi
- mkv – video/mkv
- webm – video/webm
- ogg – video/ogg
- mjpg – application/octet-stream
- mjpeg - application/octet-stream

**Success response:**

- **Code:** 204 (NO_CONTENT)

**Error response:**

- **Code**: 400 (BAD_REQUEST)
- **Content:** {"error" : [Error message], "type": [Error type]}

## 2. VIDEO LIST

List of the uploaded video files in the local storage.

**URL:** /api/videos
**Method:** GET

**Success response:**

- **Code:** 200 (OK)
- **Content:**

```
[
        {
                „name“ : „test.mp4“
        },
        ...
]
```

**Error response:**

- **Code:** 400 (BAD_REQUEST)
- **Content:** {"error" : [Error message], "type": [Error type]}

# 3. DELETE VIDEO

Remove video from the storage.

**URL:** /api/videos/"name"
**Method:** DELETE

**URL parameter:**

name – Name of video

**Success response:**

- **Code:** 202 (ACCEPTED)

**Error response:**

- **Code:** 400 (BAD_REQUEST)
- **Content:** {"error" : [Error message], "type": [Error type]}

# DATABASE

## 1. DATABASE / EVENTS

Get event result data from the internal database. Filtering of the results available through parameters below.

**URL:** /api/database

- **Method:** GET
- **Content:**


```
[
{
        "UUID": "944ef733-3676-b215-9582-8cec4be8c227",
        "id": 722,
        "sourceId": 8,
        "plate": "MEG374",
        "country": "HUN",
        "state": "",
        "timestamp": 1563441996159,
        "frame":
[
                {
                        "x": 1412,
                        "y": 453
                },
                {
                        "x": 1551,
                        "y": 445
                },
                {
                        "x": 1550,
                        "y": 471
                },
                {
                        "x": 1412,
                        "y": 478
                }
```

```
            ]
        },
]
```

**Where:**

- **UUID**: RFC 4122 - Universally Unique identifier
- **id**: unique id for an event on the server in the database
- **sourceID**: stream id
- **plate**: number plate text
- **country**: 3 letter country code
- **state**: the name of the state or province – where available
- **timestamp**: event time in Epoch time format

**Parameters:**

- **plate**

  License plate text filter.

  If specified, returns all event data for all occurrences **of the specified** license plate.

  If not specified, returns all event data for all license plates.

  (Example: http://127.0.0.1/api/database?plate=LSL178 )

  *(default: unspecified)*

- **country**

  3 letter country code filter. If not specified, returns all event data for all types.

  (Example: http://127.0.0.1/api/database?country=GRC )

  *(default: unspecified)*

- **from**

  Filter from timestamp in second

  (Epoch time. Zero = 1970 January 1st. )

  *(default: 0)*

- **to**

  Filter to timestamp in second

  (Epoch time)

  *(default: current timestamp)*

- **sourceId**

Filter to Stream ID. If zero or omitted, returns all event data for all stream's events.
*(default: 0)*

- **limit**
  Count of maximum returned events. If zero or omitted, returns all event data for all events.
  *(default: 0)*

- **offset**
  Offset of event list. Defines the 1st event to return from the saved database.
  If zero or omitted, returns all event data for all events.
  *(default: 0)*

- **orderby**
  Sort by specified column. Possible values: [ text | type | timestamp ]
  *(default: timestamp)*

- **order**
  Sort order of returned events. Possible values: [ asc | desc ]
  *(default: desc)*

**Example for combined use:**
http://127.0.0.1/api/database?sourceId=1&limit=10&order=asc
Get maximum 10 event data from Stream #1, in ascending order.

**Success response:**

- **Code:** 200 (OK)

**Error response:**

- **Code**: 400 (BAD_REQUEST)
- **Content**: {"error" : [Error message], "type": [Error type]}

## 2. GET EVENT

Get a single event data or image.

**URL:** /api/events/"id"
**Method:** GET

**URL parameter:**

- id – Event id (*Required*)
- type – Data or image (*Optional default: image*)
- index – Image index (*Optional default: 0*)

**Success response:**

- **Code**: 200 (OK)
- **Content**: Image data or JSON (image/jpeg, image/jpg, image/png, image/bmp)

**Error response:**

- **Code**: 400 (BAD_REQUEST)
- **Content**: {"error" : [Error message], "type": [Error type]

# 3. EMPTY DATABASE

Delete the whole database from server. Equals to the „Delete Database" button on the web interface.

**Method:** DELETE
**URL:** /api/database

# 4. IMAGES

All events have a unique id on the server in the database. Images can be retrieved based on this id number.

**Method:** GET
**URL:** /api/database/[0-9]+

**Example:**
http://127.0.0.1/api/database/146
Get the image for the specified event with the id „146"

# CONTACT INFORMATION


Headquarters:
ARH Inc.
Alkotás utca 41 HU-
1123 Budapest Hungary
Phone: +36 1 201 9650
Fax: +36 1 201 9651
Web: www.arh.hu

ARH America:
ARH America Corp.
28059 US Highway 19 North Suite
Clearwater, FL 33761
Phone: (727) 724-4219
Fax: (727) 724-4290
Web: www.adaptiverecognition.com

Service Address:
ARH Inc.
Ipari Park HRSZ1113/1 HU
2074 Perbál Hungary
Phone: +36 1 2019650
E-mail: rmarequest@arh.hu


ARH Technical Support System (ATSS) is designed to provide you the fastest and most proficient assistance, so you can quickly get back to business.

Information regarding hardware, software, manuals and FAQ are easily accessible for customers who previously registered to enter the dedicated ATSS site. Besides offering assistance, the site is also designed to provide maximum protection while managing your business information and technical solutions utilized.

New User
If this is your first online support request, please create an account by clicking on this link.

Returning User
All registered ATSS customers receive a personal access link via e-mail. If you previously received a confirmation message from ATSS, it contains the embedded link that allows you to securely enter the support site.

If you need assistance with login or registration, please contact atsshelp@arh.hu for help.