ADAPTIVE RECOGNITION

AUTOMATIC NUMBER PLATE RECOGNITION (ANPR/ALPR) SOFTWARE

Reference Manual

Carmen®Go REST API

Carmen GO REST API
Reference Manual

Document version: 2021.09.03.

Table of Contents

Adaptive Recognition **America**
Adaptive Recognition **Hungary**
Adaptive Recognition **Nordic**
Adaptive Recognition **Singapore**
SENDINFO@ARH.HU
WWW.ADAPTIVERECOGNITION.COM

1. REST API OVERVIEW

Rest API root path: /api

1.1. GENERAL API CALL PARAMETERS

Format

JSON output format. Default value is minimal.

Values:

- pretty - Human readable JSON format.
- minimal - Minimal JSON format

Example:

[host]/api/[resource]?format=pretty

1.2. ERROR RESPONSE

Codes:

- 400 (BAD_REQUEST) - If your request is not acceptable
- 405 (METHOD_NOT_ALLOWED) - If your request method is not supported on resource
- 403 (FORBIDDEN) - If you have not permission for the resource
- 500 (INTERNAL_SERVER_ERROR) - If server does not work properly

Structure

```
{
  "error" : <error message>,
  "type": <error type>
}
```

Example

```
{
  "error" : "Invalid operation",
  "type": "client"
}
```

2. VERSION RESOURCE

Version resource is available in all service API. Return with the service version and metadata for actual instance.

URL: /api/version
Method: GET

2.1. RESPONSE

Success response

Status code: 200 (OK)

```
{
  "name" : "Carmen GO",             // Service name
  "major" : 1,                      // Service major version
  "minor" : 1,                      // Service minor version
  "build" : 0,                      // Build number
  "build_spec" : "official release" // Build spec
  "build_time" : 645343             // Specified build timestamp
  "uuid" : "3B44CE61C6E3E9F1A2F6C4005A7A2BD97D0751D862E4A32CDF9E57B08630A68B" //
Build ID
  "common_version" : "1.2.365"      // ASL Core version
}
```

Build spec

The build field specified type of build. (eg.: developer build, beta build, branch name, ...)

General build spec values:

- official release
- beta release
- developer build
- nightly build
- Branch name (eg.: 23-anpr-bug-fix)

UUID

Service unique ID. Specified service instance.

2.2. VERSIONING

Version information parts

- Major
- Minor
- Build
- Version hash
- Build type
- Core version

Details:

Major version

No compatibility between different version

Minor version

Partial compatibility between different version

Build version

Guarantee compatibility between different version

Version hash

GIT Commit hash

Define source code state

Build type

Define build type. Build types included test processes.

- Official release
- Beta relese
- Nightly build
- Developer build
- Branch name (eg.: 18-bugfix)

Core version

ASL Library version. Low level function implementation.

Adaptive Recognition **America**
Adaptive Recognition **Hungary**
Adaptive Recognition **Nordic**
Adaptive Recognition **Singapore**
SENDINFO@ARH.HU
WWW.ADAPTIVERECOGNITION.COM

3. WEB SERVER LOAD

URL: /api/server/load

Success response:

Code: 200 (OK)

Content

Name	Type	Description
pending_request	Integer	Pending http request number

Example

```
{
  "pending_request" : 1
}
```

4. MICROSERVICE ARCHITECTURE

Gateway

The gateway is CAMRN GO server. Every REST API call forwarded to the other microservices.

Adaptive Recognition **America**
Adaptive Recognition **Hungary**
Adaptive Recognition **Nordic**
Adaptive Recognition **Singapore**
SENDINFO@ARH.HU
WWW.ADAPTIVERECOGNITION.COM

5. AUTHENTICATION SERVER

Executable name: auth-server
Service name: CARMEN Authentication Server
Display name: CARMEN Authentication Server
API Default port: 9860

Authentication Server version

Requests

Request	Operation	Description	Remarks
GET /api/version	Authentication Server Version		It is not redirected through carmen-go-server's proxy! It has its own version request with the same path.
POST /api/access/login	Login	Login with username and password and get an access token	
GET /api/access/key	Trusted Key Request By Key Id	Get the public key and its expiration data for a specified key id	
GET/api/access/users	User List	Get list of users	
GET /api/access/users/[username]	Get User Info		
POST /api/access/users/[username]	Create New User		
PATCH /api/access/users/[username]	Modify User	Modify an existing user	Username cannot be changed. Password change is optional, only changed if present in request.
GET /api/access/roles	Get Role List	List of possible user roles	default in V1.4: Administrator, Maintainer, Viewer
GET /api/access/renew	Renew Access Token	Renews a currently valid access token	
PATCH /api/access/currentuser	Modify Current User	Modifies the current user	Needs less privilege than the general "Modify User" request.

Login

POST /api/access/login

Request content JSON:

```
{
  "username":"...",
  "password":"..."
}
```

Response:

```
{
        "access_token": "...",
        "expires_in": 3600
}
```

Trusted Public Key Request By Key Id

GET /api/access/key

Get key info for a trusted public key that is used by the Access Server for signing access tokens.

example: GET /api/access/key?keyId=1

Query string params:

- "keyId": "kid" key id from Json Web Token (access token)

Response:

```
{
        "keyId": 1,
        "publicKey": "",
        "createdAt": 1622717881,
        "expiresAt": 1623927481,
        "disabled": false
}
```

List Of Users

GET /api/access/users

Response:


```
[
        {
                "username": "admin",
                "firstName": "",
                "lastName": "",
                "email": "",
                "roles": [
                        "Administrator"
                ],
                "permissions": [
                        "DATABASE_SETUP",
                        "DATABASE_VIEW",
                        "LICENCE_INFO",
                        "LOGGED_IN",
                        "OUTPUT_SETUP",
                        "RESULT_STREAM_VIEW",
                        "STATISTICS_VIEW",
                        "STREAM_SETUP",
                        "STREAM_START_STOP",
                        "STREAM_VIEW",
                        "USER_CREATE",
                        "USER_DELETE",
                        "USER_LIST",
                        "USER_UPDATE"
                ]
        }
]
```


Get User Info

GET /api/access/users/[username]

Create New User

POST /api/access/users

Request content JSON:


```
{
  "username": "user00",
  "firstName": "you",
  "lastName": "ser",
  "email": "e@mail.com",
  "password": "12345678",
  "roles": ["Maintainer"]
}
```

Modify User

PATCH /api/access/users/[username]

Request JSON:
Same format as used at user creation.
You cannot change the "username" since users are identified by it.

Optional fields:

- If "roles" is not present roles will not be changed.
- If "firstName", "lastName" or "email" is not present the missing field will be changed to empty string.
- If "password" is not present the password will not be changed.

Delete User

DELETE /api/access/users/[username]

Get Role List

GET /api/access/roles

Response:

```
["Administrator","Maintainer","Viewer"]
```

Renew Access Token

GET /api/access/renew

Response:

```
{
        "access_token": "...",
        "expires_in": 3600
}
```

Modify Current User

PATCH /api/access/currenttuser

- Request JSON: Same format as used at PATCH /api/access/users except roles should not be present. You cannot change the user roles with this request.
- You cannot change your "username"
- Optional fields: same applies as for /api/access/users optional fields.
- You don't need any permission for this request. If you are logged in / have a valid access token then you can change your own user info settings.
- If "password" is not present the password will not be changed.

If you want your password then the old password must be provided too:

```
{
  "password": "my_new_password"
  "oldPassword": "my_old_password"
}
```

6. MMR SERVER

6.1. SERVICE DATA

- Executable: mmr-server
- Display name: CARMEN MMR Server
- Default API port: 9850

Requests

Request	Operation	Description	Remarks
GET /api/version	Get service version		
GET /api/licence	List MMR engines	List MMR engines and whether there's suitable license available for them	
GET /api/license		alias for "/api/licence"	
GET /api/countries			
GET /api/countries/.+			
POST /api/countries			
POST /api/mmr	Run MMR on a single image		

Run MMR on a single image

URL: /api/mmr

MMR relies on the information about the plate that was determined by an ANPR call.
The request must send a multipart content with two parts, one with the image and one with the information about the plate.

Image part

Supported image formats / "Content-Type"s:

- image/jpeg
- image/bmp
- image/png
-

Data part

Content-Type: application/json

- country: country profile name
- platesize: plate width in millimeter
- frame: frame coordinates of the recognized plate on the image

Adaptive Recognition **America**
Adaptive Recognition **Hungary**
Adaptive Recognition **Nordic**
Adaptive Recognition **Singapore**
SENDINFO@ARH.HU
WWW.ADAPTIVERECOGNITION.COM

7. LOG SERVER

Executable name: log-server
Display name: CARMEN Log Server
Description: Adaptive Recognition Log collector solution

7.1. LOG LEVELS

- DEBUG
- INFO
- NOTIFICATION
- WARNING
- ERROR
- CRITICAL
- ALERT
- EMERGENCY

7.2. FORMAT

[datatime][loglevel][host/process name:PID] message
Date/time format – ISO 6801

Adaptive Recognition **America**
Adaptive Recognition **Hungary**
Adaptive Recognition **Nordic**
Adaptive Recognition **Singapore**
SENDINFO@ARH.HU
WWW.ADAPTIVERECOGNITION.COM

7.3. OUTPUT

File

Example: carmen_dls.log

UDP

Example: udp://:

Standard Output (for DEBUG)

Example: std://

Config

Config file: log.cfg

Config	Description	Default
log.udp.port	UDP Log system port	9801
web.port	REST API Port	9800
log.remote	Enable remote message receive (false) for internal usage	false
log.level	Log level (INFO) or debug	INFO
log.target	Message output („carmen_dls.log") for debug	carmengo.log

Rest log

Log message.

URL: /api/log

Method: POST

Data parameters:

Parameters	Description	Type	
facility	Facility description	Integer	Not available on REST
level	Log level	Integer	
host	Host name	String	
app	Application name	String	
pid	Process ID	Integer	
scope	Scope	String	Not available on REST
message	Log message	String	

Example

```
{
  "level" : 0,
  "host" : "example.com",
  "app" : "ANPR Server",
  "pid" : 6581,
  "message" : "Hello world",
}
```

Success response

Code: 204 (NO_CONTENT)

Websocket channel (Coming soon)

8. EVENT SERVER

Executable name: event-server
Service name: cmevent-server
Display name: CARMEN Event Server
API Default port: 9830

8.1. DATA STRUCTURES

Number plate structure

Parameter	Type	Description
text	String	Number plate text
type	String	Country
state	String	State (Optional)
frame	Array	Number plate frame points

Example

```
{
  "text" : "ARH-001"
  "type" : "HUN"
  "state" : ""
  "frame": [
{
"x": 1019,
"y": 919
},
{
"x": 1203,
"y": 923
},
{
"x": 1202,
"y": 957
},
{
"x": 1018,
"y": 953
}
]
```

Vehicle structure

Parameter	Type	Description
class	String	Vehicle class (Values: BUS/CAR/.... TODO)
color	Integer	Vehicle color code (Black: #000000 -> 0, Red #FF0000 -> 16711680)
plate	Number plate structure	Plate data

Adaptive Recognition **America**
Adaptive Recognition **Hungary**
Adaptive Recognition **Nordic**
Adaptive Recognition **Singapore**
SENDINFO@ARH.HU
WWW.ADAPTIVERECOGNITION.COM

Example

```
{
  "class" : "BUS",
  "color" : 0,
  "plate" : {
    "text" : "ARH-001"
    "type" : "HUN"
    "state" : ""
    "frame" : [
       { "x" : 10, "y" : 10 },
       { "x" : 10, "y" : 100 },
       { "x" : 100, "y" : 100 },
       { "x" : 100, "y" : 10 }
    ]
  }
}
```

Event structure

Parameter	Type	Description
id	Unsigned integer	Event id
UUID	String	Event unique ID RFC 4122
timestamp	Unsigned integer	Event timestamp
source	String	Event source
vehicle	Vehicle structure	Vehicle structure

Example

```
{
  "id" : 313,
  "UUID" : "81c797ba-d586-4883-97d0-6229e52dc23c",
  "timestamp" : 26485565,
  "source" : "Parking lane #1",
  "vehicle" : {
    "class" : "BUS",
    "color" : 0,
    "plate" : {
      "text" : "ARH-001"
      "type" : "HUN"
      "state" : ""
      "frame" : [
         { "x" : 10, "y" : 10 },
         { "x" : 10, "y" : 100 },
         { "x" : 100, "y" : 100 },
         { "x" : 100, "y" : 10 }
      ]
    }
  }
}
```

Legacy event structure

Parameter	Type	Description	Values
UUID	String	Event unique ID RFC 4122	
id	Unsigned integer	Event id	
sourceId	Integer	Source ID	
plate	String	Number plate text	
country	String	Country	
state	String	State (Optional)	
timestamp	Unsigned integer	Event timestamp	
make	String	Vehicle manufacturer	
model	String	Vehicle model	
color	String	Vehicle color	Black, Brown, Blue, Purple, Red, Green, Orange, Yellow, Gray, White
color_code	Integer	Vehicle color code in BGR	-1 is undefined color
category	String	Vehicle class eg.: BUS	Car, Motorbike, Bus, Light truck, Van, Heavy truck, Unknown
frame	Array	Number plate frame point pair	
details	Object	Computed localization information ISO-3166	

Adaptive Recognition **America**
Adaptive Recognition **Hungary**
Adaptive Recognition **Nordic**
Adaptive Recognition **Singapore**
SENDINFO@ARH.HU
WWW.ADAPTIVERECOGNITION.COM

Example


```
{
  "UUID": "e8d0a500-8d41-1b16-0280-78d00421fb6d",
  "id": 93,
  "sourceId": 1,
  "plate": "ABC123",
  "country": "DNK",
  "state": "DNK",
  "timestamp": 1592938965878,
  "make": "",
  "model": "",
  "color": "",
  "color_code": -1,
  "category": "",
  "frame": [
      {
        "x": 230,
        "y": 186
      },
      {
        "x": 296,
        "y": 175
      },
      {
        "x": 295,
        "y": 192
      },
      {
        "x": 229,
        "y": 203
      }
    ],
    "details": {
      "country": {
         "numeric": 208,
         "iso2": "DK",
         "iso3": "DNK",
         "code": "DK",
         "name": "Denmark"
      }
    }
  }
```

8.2. DATABASE VERSION

Url: /api/database/version
Method: GET

Success response

Code: 200 (OK)
Return with database structure version identifier.

Example

```
{
    "major": 1
    "minor" : 0
}
```

9. EVENT SERVER API

Event CURD API
URL: /api/events
/api/database (Deprecated)

9.1. SEND EVENT

Save event to database.

Method: POST

Data parameters

Multipart "Legacy" event JSON Structure and images (image/jpeg, image/jpg, image/png, image/bmp)

Success response:

Code: 201 (CREATED)

Name	Type	Value
id	int	New event internal ID

Example

```
{ "id": 7545 }
```


Adaptive Recognition America
Adaptive Recognition Hungary
Adaptive Recognition Nordic
Adaptive Recognition Singapore
SENDINFO@ARH.HU
WWW.ADAPTIVERECOGNITION.COM

9.2. EVENT LIST

List of stored events.
Method: GET
Success response:
Code: 200 (OK)

Structure

Example

```
[
	{
		"UUID": "e8d0a500-8d41-1b16-0280-78d00421fb6d",
		"id": 93,
		"sourceId": 1,
		"plate": "ABC123",
		"country": "DNK",
		"state": "DNK",
		"timestamp": 1592938965878,
		"make": "",
		"model": "",
		"color": "",
		"color_code": -1,
		"category": "",
		"frame": [
			{
				"x": 230,
				"y": 186
			},
			{
				"x": 296,
				"y": 175
			},
			{
				"x": 295,
				"y": 192
			},
			{
				"x": 229,
				"y": 203
			}
		],
		"details": {
			"country": {
				"numeric": 208,
				"iso2": "DK",
				"iso3": "DNK",
				"code": "DK",
				"name": "Denmark"
			}
		}
	},
//...
]
```

Filter

Parameter	Type	Description
plate	String(Regex)	Plate text
country	String	Country code
state	String	State code
from	Integer	First timestamp
to	Integer	Last timestamp
sourceId	Integer	Source ID
limit	Integer	Result number limit
offset	Integer	Result offset
orderby	String(Column name)	Order by column. (id, uuid, sourceId, timestamp, plate, country, state, make, model, color, category)
order	String	Order direction (asc/desc)
make	String	Vehicle make
model	String	Vehicle model
color	String	Vehicle color
category	String	Vehicle category

Example

http://192.168.3.162/api/events?country=HUN&limit=4&format=pretty

9.3. DELETE ALL EVENTS

Method: DELETE
Success response:
Code: 202 (ACCEPTED)

10. SINGLE EVENT API

Single event data or image.

URL: /api/events/[ID or UUID]
Method: GET

Name	Required	Type	Description	Default	Value
id	ID or UUID	Integer	Internal ID		
uuid	ID or UUID	String	Event UUID		
type	Optional	String	Image or Meta-date	image	image, data
index	Optional	Integer	Image index	0	

Example by ID

http://192.168.3.162/api/events/57

Example by UUID

http://192.168.3.162/api/events/e8bdf440-43ca-1916-8b81-78d00421fb6d

Example with parameter

http://192.168.3.162/api/events/57?type=data

Success response:
Code: 200 (OK)

11. EVENT CHANNELS API

Output channel CURD Resource.

URL: /api/outputs

Output channel list

Event output list.

Method: GET

Success response:

Code: 200 (OK)

Content

Name	Type	Description
name	String	Channel name
url	String (URL)	Output channel target url

Example

```
[
  {
    "name": "myCsv",
    "url": "csv://"
  },
  {
    "name": "myFtp",
    "url": "ftp://127.0.0.1:22/results"
  },
  //...
]
```

Remove channel

Disable output channel.

Method: DELETE

If data parameter is empty remove all channel.
If parameter is an object with channel name remove the channel.
If parameter is a list of object with channels remove channels.

Request example

Remove single channel

```
{
  "name" : "myFtp"
}
```

Remove multi channel

```
[
  {
    "name": "myFtp"
  },
  {
    "name": "myCsv"
  }
]
```

Success response:

Code: 204 (NO CONTENT)

Create or change output

Method: POST, PATCH

Change output channel parameter. If method POST create new channel else change existing channel.

Name	Type	Required	Description
name	String	Required	Channel name
url	String (URL)	Required	Output channel target url

Single channel

```
{
   "name" : "myFtp",
   "url" : "ftp://127.0.0.1:21/results"
}
```

Multi channel

```
[
  {
     "name" : "myFtp",
     "url" : "ftp://127.0.0.1:21/results"
  },
  {
     "name" : "myFtp2",
     "url" : "ftp://192.168.1.2:21/results"
  },
  //...
]
```

Success response

Code:

- 201 (CREATED) If create new channel
- 202 (ACCEPTED) If modified existing channel

Return with new channel parameters.

Example

```
[
   {
       "name": "myFtp"
       "url" : "ftp://127.0.0.1:21/results"
   }
]
```


Adaptive Recognition **America**
Adaptive Recognition **Hungary**
Adaptive Recognition **Nordic**
Adaptive Recognition **Singapore**
SENDINFO@ARH.HU
WWW.ADAPTIVERECOGNITION.COM

12. EVENT WEBSOCKET

URL: /api/websocket

Method: GET
It will switch Websocket protocol. Binary and text packets will arrive. The text will contain the json meta data. See here. The binary will contain the image in JPEG compression.

Adaptive Recognition **America**
Adaptive Recognition **Hungary**
Adaptive Recognition **Nordic**
Adaptive Recognition **Singapore**
SENDINFO@ARH.HU
WWW.ADAPTIVERECOGNITION.COM

13. DEVICE DISCOVERY, ONVIF

The primary purpose of the Carmen Go Discovery API is to detect and display ONVIF cameras on the network (their address) and to ease camera selection, i.e., no need to manually enter the stream URL.
Stream URLs are defined in two steps:

1. WS Discovery: list camera URLs
2. ONVIF query to determine the STREAM URL for the camera selected from the list.
    - this requires ONVIF username-password pair (our cameras accept anything from default)

ONVIF username / password storage within Carmen Go:

For ease of use, ONVIF username / password pairs are automatically saved when the onvif_feature query is referred. The key for the entry is the host (not the stream URL). If you have a stored password for a particular host, you do not need to enter it again.

13.1. DEVICE (WS) DISCOVERY

URL: /api/discovery
Method: GET

Returns the URLs found by device discovery.

Example:

```
[
  { "host": "...", "port": 80, "storedCredentials": true, "credentialsRequired":
true, "secured": false },
  { "host": "...", "port": 80, "storedCredentials": true, "credentialsRequired":
true, "secured": false }
]
```

13.2. ONVIF FEATURES QUERY

URL: /api/discovery/onvif_feature
Method: POST

Queries the properties of the specified ONVIF camera.

HTTP POST JSON parameters:

```
{
  "host": "...",
  "port": 80, /* optional */
  "secured": false, /* optional, in case of https should be true */
  "username": "...",
  "password": "..."
}
```

If there is a password stored for the host, the username, password parameters are optional.
If no username is specified or blank, Carmen Go will check to see if there is a stored password for the host.
If not, there is an error: in this case the user / password must be entered.
If there is, it will use it.
If a user / password is specified and stored, it will use the entered one and update the stored one.

Response example:

```
{
      "IP_address" : "192.168.3.21",
      "URI_path" : "/onvif/services/media",
      "fw_version" : "V3.6r1.3808",
      "manufacturer" : "ARH",
      "model" : "SMARTCAM",
      "streamURIs" : [
                  "rtsp://192.168.3.21/stream/jpeg",
                  "rtsp://192.168.3.21/stream/jpegsen2src1n2"
            ]
}
```

Authentication error or in case of invalid input: Bad Request (400) error.

In case of other server side error: Internal Server error (500).

14. LICENCES

URL: /api/licence Method: GET

Success response

Code: 200 (OK)

Name	Type	Description
device	Integer	Number of licence device
stream	Integer	Number of available stream
core	Integer	Number of ANPR core
anpr	List	List of ANPR engines and licence info
mmr	List	List of MMR engines and licence info

Example


```
{
	"device": 1,
	"stream": 4,
	"core": 4,
	"anpr": [
		{
			"name": "cmanpr-7.3.9.202 : latin_vq_go",
			"licence": true
		},
		{
			"name": "cmanpr-7.3.11.153 : eur_go",
			"licence": true
		},
		{
			"name": "cmanpr-7.3.9.227 : asia_vq_go",
			"licence": true
		},
		{
			"name": "cmanpr-7.3.11.188 : sas_go",
			"licence": true
		},
		{
			"name": "cmanpr-7.3.11.216 : vq_go",
			"licence": true
		},
		{
			"name": "cmanpr-7.3.11.69 : gen_go",
			"licence": true
		}
	],
	"mmr": [
		{
			"name": "mmr-7.3.1.7 : mmr-mys",
			"licence": false
		}
	]
}
```


Adaptive Recognition America
Adaptive Recognition Hungary
Adaptive Recognition Nordic
Adaptive Recognition Singapore
SENDINFO@ARH.HU
WWW.ADAPTIVERECOGNITION.COM

15. PROCESSES

15.1. VIDEO PROCESSING METHODS

List of installed video analytics.

URL: /api/process/methods
Method: GET

Name	Type	Description
id	String	Method identifier
name	String	Pretty method name
description	String	Method description

```
[
	{
		"id": "generalprocess",
		"name": "General video processor",
		"description": "CARMEN GO General algorithm. Recommended for city,
highway or faster parking use-case."
	},
	{
		"id": "parkingprocess",
		"name": "Parking",
		"description": "Recommended for parking situation."
	},
	{
		"id": "platefinderprocess",
		"name": "Plate Finder",
		"description": "CARMEN GO Plate Tracker algorithm based on CARMEN
Plate Finder."
	},
	{
		"id": "stopandgoprocess",
		"name": "Stop and Go",
		"description": "This method is recommended for very slow traffic
(e.g.: Access control)"
	}
]
```

15.2. PROCESS INFORMATION

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

Name	Type	Description	Read Only
name	String	Stream name	
url	String	Stream URL	
format	String	Live view stream format (only "mjpeg" is supported for now)	
decode	String	Video decoder mode (Software/Hardware/Auto)	
encode	String	Video encoder mode (Software/Hardware/Auto)	
stream_resolution	String	Live view stream resolution (320p/480p/1080p)	x
status	String	Processing status (Running/Stopped)	x
statusText	String	Processing status description	x
run	Boolean	Expected status	x
country	String	Video analytics country	
method	String	Selected analytics	
stream	String	Stream port	x
roi	List of points	Region of interest	
sensitivity	Integer	Analytics scale factor	
type	String	Video source type (video/camera)	
mmr	Boolean	Make and model recognition	
mmr_color_recognition	Boolean	Enable vehicle color recognition.	
anpr_color_recognition	Boolean	Enable plate color recognition. Might improve performance when turned off	
event_duplication_timeout	Integer	Minimum time between duplicated results	

Adaptive Recognition **America**
Adaptive Recognition **Hungary**
Adaptive Recognition **Nordic**
Adaptive Recognition **Singapore**
SENDINFO@ARH.HU
WWW.ADAPTIVERECOGNITION.COM

```
{
	"name": "firstStream",
	"url": "url",
	"format": "mjpeg",
	"decode": "software",
	"encode": "software",
	"stream_resolution": "480p",
	"status": "Stopped",
	"statusText": "Stopped",
	"run": false,
	"country": "",
	"method": "generalprocess",
	"stream": "9841",
	"roi": [
		{
			"x": 0,
			"y": 0
		},
		{
			"x": 0,
			"y": 100
		},
		{
			"x": 100,
			"y": 100
		},
		{
			"x": 100,
			"y": 0
		}
	],
	"sensitivity": 50,
	"type": "video",
	"mmr": false
}
```


Change stream settings

Method: PATCH
URL: /api/process/[0-9]+

The non-readonly properties described above can be changed.
If the processing method was running at the moment of the modification, then the modified parameters will not take effect until the processing method is restarted.

Adaptive Recognition **America**
Adaptive Recognition **Hungary**
Adaptive Recognition **Nordic**
Adaptive Recognition **Singapore**
SENDINFO@ARH.HU
WWW.ADAPTIVERECOGNITION.COM

15.3. PROPERTIES OF STREAM SOURCE

Relevant properties:

- type (camera or video)
- url
- manufacturer (only for camera type)
- other properties for non-generic camera types

Non-generic camera types:

The stream source will always be defined by an URL but there is a non-generic interface implemented on the API for convenience that generates these URLs for some known camera models. That means that there's always an equivalent generic URL for non-generic camera types. The URL templates are defined in "data/camera.json" file in the www directory of CARMEN GO's install folder.

Generic camera example:

```
{
  "...": "...",
  "type": "camera",
  "manufacturer": "Generic",
  "url": "rtsp://192.168.0.2:554/stream/h264"
}
```

Non-Generic (ARH/Freeway) camera example:

```
{
  "...": "...",
  "type": "camera",
  "manufacturer": "ARH",
  "model": "Freeway",
  "source": "rtsp",
  "host": "192.168.6.91",
  "other": {
    "port": "554",
    "username": "",
    "password": ""
  }
}
```

Start/Stop stream / processing method

Method: POST
URL: /api/process/[0-9]+

```
{"status":"Start"}

{"status":"Stop"}
```

Contact Information

Headquarters:
Adaptive Recognition Hungary Inc.
Alkotás utca 41 HU-
1123 Budapest Hungary
Phone: +36 1 201 9650
Fax: +36 1 201 9651
Web: www.adaptiverecognition.com

Service Address:
Adaptive Recognition Hungary Inc.
Ipari Park HRSZ1113/1 HU
2074 Perbál Hungary
Phone: +36 1 2019650
E-mail: rmarequest@adaptiverecognition.com

Adaptive Recognition Hungary Technical Support System (ATSS) is designed to provide you the fastest and most proficient assistance, so you can quickly get back to business.
For further technical information about our products, please visit our official website.

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@adaptiverecognition.com for help.