ANPR LUMO

Developer guide

2023-11-0909 | v1.06

nedap

www.nedapidentification.com

Disclaimer
Nedap N.V. has made every effort to ensure the accuracy of the information contained in this document. However, Nedap N.V. makes no representations or warranties whatsoever whether express or implied as to the accuracy, correctness, completeness or fit-for-purpose or suitability for the purpose of this product. You use the products at your own risk. Nedap N.V. excludes any liability to the maximum extent permitted by applicable law for the damages caused by errors or failures made during the installation or improper use of this product or by not applying the instructions stated in this document.

Nedap N.V. reserves the right to make improvements or amendments to this document and/or the products described therein at any time without any notification. The latest version of this document can be found on our partner portal https://portal.nedapidentification.com. Please download the latest version of this document and keep a copy for your own records.

This document can be published in various languages but only the English language version will prevail. Nedap N.V. assumes no responsibility for any errors caused for the translations into another language.

Warranty and spare parts
Please consult the Nedap products dealer from whom you purchased this product, in regards to the applicable warranty conditions. This product cannot be used for any other purpose as described in this document. If the product is not installed according to this document; the warranty provided is not applicable. At the sole discretion of Nedap N.V., Nedap N.V. may decide to change the conditions of the warranty policy. You agree that Nedap N.V. is able to compensate you the pro-rata value of the warranty involved rather than replacing or repairing the product depending on the technical or economical value of the product.
Prior to applying the warranty, please verify if you comply with the warranty conditions of the warranty policy, whether you can successfully apply for the replacement or repair of a defective part. Parts can only be replaced with original Nedap parts, otherwise the warranty policy will not be applicable on the product. If the warranty is applicable, please contact the dealer or send the defective parts to the dealer.

Additional information
For any information or questions regarding the product, please contact your own dealer

Nedap N.V.
Identification Systems
Parallelweg 2
7141 DC Groenlo
The Netherlands

+31 (0)544 471 111
info@nedapidentification.com
www.nedapidentification.com

Contents

1 Introduction

The NEDAP ANPR LUMO License Plate Reader offers automatic number plate reading. The camera can be configured through the web based interface. For integration purposes there is a REST API available. This API can be used for different purposes, like integration with the available access list.

This document describes all available REST API calls including CURL examples for the latest LUMO version 4.7.0

REST API calls are available on ANPR LUMO cameras from version 4.3.1 and higher.

1.1 Login

Description

This API Endpoint is used to login with own credentials on the server. The login is required in order to perform any other action taken on the server.

URL	Method	URL Parameter	Data Parameter
/login	POST	None	login_string

Parameter

login_string

string with following format : "username=your_username&password=your_password"

Examples:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

1.2 Current Status

Description

This API Endpoint returns common properties of the camera such as library versions, uptime and so on.

URL	Method	URL Parameter	Data Parameter
/live/status	GET	None	none

Success response

HTTP Status code	Method	Sample Response

200	GET	{
		"addresses": "192.168.3.15",
		"LUMO": {
		"active": "eu_006.dat",
		"anpr": "2.6.0",
		"classifiers": [
		"au_003.dat",
		"eu_006.dat",
		"nz_002.dat",
		"usa_004.dat"
		],
		"lpr": "4.7.0"

```
  },
  "os": "Linux version 3.14.79-vc-z (root@vcbuilder2)
  (gcc version 4.8.4 (Ubuntu/Linaro 4.8.4-
2ubuntu1~14.04.1) )
  #2 SMP PREEMPT Wed Mar 21 11:12:01 CET 2018\n",
  "serialnumber": "4000702",
  "system-time": "2018-08-16T10:00:40.811701",
  "uptime": "1118641.87"
}
```


Examples:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # get list

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/live/status

2 System properties

2.1 GET and SET system date and time

Description

API Endpoints for getting and setting system properties of an ANPR LUMO camera.
Use the GET method to retrieve the property and the PATCH method for setting it.

URL	Method	URL Parameter	Data Parameter
/system/:property	GET	:property	None
/system/:property	PATCH	:property	:property=:value

Available properties

Property name	Value Format
Time	ISO 8601 formatted date and time

Success response

HTTP Status code	Method	Sample Response
200	GET	`{ :property: :value }`
200	PATCH	`{ "status": "success", "msg": "ok" }`

Examples:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # get system time

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/system/time

 # set system time

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/system/time -X PATCH --data-urlencode "time=2019-02-23T16:30:02+0100"

2.2 Digital IO

Description

On the camera the Digital IO pins can be used with the API Endpoint for pin status fetching and for activation/deactivation of available outputs. GET method can be used to fetch status of all pins and PATCH to set the status of an output pin.

URL	Method	URL Parameter	Data Parameter
/digitalio	GET	none	None
/digitalio	PATCH	none	:value

Parameter

Value that specifies output pin and the status to be set on given pin. Value is JSON struct with following format:

parameter	Value Format
:value	{ "index": pin number <integer>, "set": true to activate or false to deactivate selected pin }

Success response

HTTP Status code	Method	Sample Response
200	GET	{ "Inputs": [{ "#1": "active" }, { "#2": "active" }], "Outputs": [{ "#1": "passive" }, { "#2": "passive" }] }
200	PATCH	{ "msg": "Output #1 deactivated", "status": "success" }

Examples:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # get status of digital IO

curl --cookie cookies.log --cookie-jar cookies.log -X GET http://192.168.3.15/digitalio

activate status of the output pin #1

curl --cookie cookies.log --cookie-jar cookies.log -X PATCH http://192.168.3.15/digitalio --data '{"index":1, "set":true}'

2.3 Restart ANPR Server

Description

This API Endpoint restarts the ANPR server on the camera.

URL	Method	URL Parameter	Data Parameter
/camera/reset	POST	none	None

Success response

HTTP Status code	Method	Sample Response
200	GET	`{ "status": "ok", "msg": "Resetting..." }`

Example:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # restart camera

curl --cookie cookies.log --cookie-jar cookies.log -X POST http://192.168.3.15/camera/reset

2.4 Reboot Camera

Description

This API Endpoint reboots the camera.

URL	Method	URL Parameter	Data Parameter
/camera/reboot	POST	none	None

Success response

HTTP Status code	Method	Sample Response
200	GET	`{ "status": "ok", "msg": "Rebooting..." }`

Example:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # reboot

curl --cookie cookies.log --cookie-jar cookies.log -X POST http://192.168.3.15/camera/reboot

3 ANPR Properties

ANPR Properties are related to the ANPR server application. This endpoint allows the configuration of many properties and parameters related to the camera, trigger settings, ANPR Lumo engine and visualization.

This section additionally contains API endpoints used to get and set ROI properties

Note
New since the 4.5.0 release is the usage of section in the names of parameters. The names of parameters are constructed as following: section-parameter. As example parameter binning is since 4.5.0 renamed to camera-binning. In order to provide compatibility with previous releases old names are going to be supported in this release as well.

Property name	Value format	description
camera-binning	*true* or *false*	Camera low resolution on/off
camera-rotate_image	floating number [-180.0-180.0]	Rotation of the image before ANPR detection in degrees
camera-use_anpr_autolum	*true* or *false*	Activate/deactivate automatic Brightness Control
camera-min_shutter	integer number [10-]	Sensor minimum exposure time in micro sec
camera-min_gain	integer number [0-10]	Sensor minimum gain in dB
camera-max_shutter	integer number [10-50000]	Sensor maximum exposure time in micro sec
camera-max_gain	integer number [0-10]	Sensor maximum gain in dB
camera-accepted_brightness_min_amount	float number [0.0.-100.0]	Brightness Control minimum amount of well lighted pixels inside ROI (%)
camera-saturated_max_amount	float number [0.0-100.0]	Brightness Control maximum amount of accepted saturated pixels inside ROI
camera-lightenabled	*true* or *false*	Enable/Disable LED Flash
camera-lightduration	integer number [0-1000]	Duration of the flash in micro sec
camera-lightcurrent	integer number [0-1500]	Flash current in mA [0-1500]. Note: at VC cameras flash is off for values < 150 mA
camera-lightbrightness	integer number [0-2] 0 ... Dark 1 ... Middle 2 ... Bright	Set the brightness of the flash
trigger-triggertype	integer number 0 ... FreeRun 1 ... SingleLine 2 ... Start-Stop 3 ... Start-Time 4 ... Start-Sequence	Set trigger mode in combination with IO lines

trigger-stopdio	integer number	Input line for stop signal, used for trigger mode Start-Stop
trigger-startdio	integer number (depending on camera HW)	Input line for start signal, used for all trigger modes except free run
trigger-startdiopolarity	*true* or *false*	Define the trigger signal polarity: high->low when enabled, low->high when disabled
trigger-stopdiopolarity	*true* or *false*	Define the stop signal polarity: high->low when enabled, low->high when disabled
trigger-timemax	integer number [100-999999]	Max waiting time in millisec for ANPR reading after trigger signal, used in mode Start-Time
trigger-sequencelength	integer number [0-]	The number of frames read after trigger signal, used for trigger mode Start-Sequence
anpr_engine-mode	integer number 1 ... Fastest 2 ... Fast 3 ... Standard 4 ... High 5 ... Best	Engine performance modes
anpr_engine-minletterheight	integer number [6-1000]	Minimum height of a plate letter in pixels
anpr_engine-maxletterheight	integer number [0 - 1000]	Maximum height of a plate letter in pixels
anpr_engine-maxplateangle	integer number [0-90]	Maximum allowed license plate rotation angle in degrees
anpr_engine-minplateconfidence	integer number [0-100]	Minimum confidence for a plate to be accepted, plates below this confidence will be dismissed
anpr_engine-mincontrast	integer number [7-100]	Min contrast to differentiate characters from the license plate. Reducing this parameter can increase reading performance but also processing time
anpr_engine-maxplates	integer number [0-10, 0 = unlimited]	Maximum number of plates which can be detected in a single image
anpr_engine-scalewidth	integer number [1-200]	Scale the image width by this factor (in percent) before ANPR processing
anpr_engine-scaleheight	integer number [1-200]	Scale the image height by this factor (in percent) before ANPR processing
anpr_engine-searchinverted	integer number 0 ... disabled 1 ... search	Detection mode for white-on-black (inverted) License Plates (Disabled, Search only if no black-on-white

	2 ... always	plate found, or Always search). Search or Always will result in a lower processing frame rate
anpr_stream-streammode	integer number 0 ... Single 1 ... Parking 2 ... Offline 3 ... Freeflow	License plate sequence processing mode, Parking mode is recommended for access control
anpr_stream-minreadings	integer number [0-20]	Minimum detections of the same plate before accepting it-considering Maximum Characters Mismatch for comparison
anpr_stream-maxstringmatchingdistance	integer number [0-10]	Maximum difference between two license plates to be regarded as identical, recommended value: 1, 2
anpr_stream-noreadinterval	integer number [0-99]	Specifies the interval of images without plate reading after which the defined noread actions are triggered
anpr_stream-lowconflimit	integer number [0-100]	The threshold for which a confidence is considered low, this value must be >= Min Plate Confidence. Used in action type Low Confidence Read
hud_size	integer number 0 ... small 1 ... medium 2 ... large	Size of the HUD
application-hud_position	integer number 0 ... Upper Left 1 ... Upper Right 2 ... Lower Left 3 ... Lower Right	Position of the HUD
application-cam_prefix	string	Text shown as info overlay
application-draw_info	*true* or *false*	Turn on/off info overlay in the HUD
application-draw_time	*true* or *false*	Turn on/off time display in the HUD
application-draw_date	*true* or *false*	Turn on/off date display in the HUD
application-draw_shutter	*true* or *false*	Turn on/off shutter display in the HUD
application-draw_gain	*true* or *false*	Turn on/off gain display in the HUD
application-displaylprrate	*true* or *false*	Display the current reading frame rate of the camera
application-displayframerate	*true* or *false*	Turn on/off camera framerate display in the HUD
application-visualiseplate	*true* or *false*	Draw licence plate result into the result image

application-visualisestate	*true* or *false*	Draw licence plate result into the result image
application-visualisecharconf	*true* or *false*	Draw single character confidences into the result image buffer
application-jpeg_compression	integer number [1-100]	JPEG compression level for live images in the user interface
application-maxhistoryentries	integer number [0-10000]	The maximum number of ANPR results which are kept in the history
application-historyhours	integer number [0-10000]	The maximum time span in hours for which ANPR results are kept in the history buffer
application-sendutf	*true* or *false*	Send result string UTF-8 encoded

3.1 Get and set ANPR properties

Description

API Endpoints for getting or setting an ANPR property of a Carrida camera. Use GET method to retrieve the property and the PATCH method to set it.

URL	Method	URL Parameter	Data Parameter
/anpr/:property	GET	:property	None
/anpr/:property	PATCH	:property	Property=:value

Parameter

:property

Name of one of properties listed above.

:value

Value to be set for specified property. The format of the value is also specified in the property list from above.

Success response

HTTP Status code	Method	Sample Response
200	GET	`{ :property: :value }`
200	PATCH	`{ "status": "success", "msg": "ok" }`

Examples:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login
--data "username=admin&password=secret"

 # get an anpr property

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/anpr/mode

 # set an anpr property

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/anpr/mode -X PATCH --data-urlencode "mode=1"

3.2 Get and Set ROI properties

Description

API Endpoints from this section are used to get and set ROI properties. There are two ROI's used in the ANPR Server:
brightness ROI used by the Brightness Control ANPR ROI used by Nedap as region for plate detection
Use GET to fetch these properties from server and PATCH to set them.

URL	Method	URL Parameter	Data Parameter
/roi/:property	GET	Selected ROI	None
/roi/:property	PATCH	Selected ROI	:value

Parameter

:property

ROI name (either brightness or alpr)

value

JSON string with following format:

for brightness ROI:

```
{
 "points":
  [
    { % point #1
      "x": "x_value <integer>"
      "y": "y_value <integer>"
    },
    { % point #2
      "x": "x_value <integer>"
      "y": "y_value <integer>"
    },
    ...,
    { % point #n (>= 3)
      "x": "x_value <integer>"
      "y": "y_value <integer>"
    }
  ]
}
```

for anpr ROI:

```
[
  { % Area #1
    "angle": {
              "direction": "area_angle_direction<integer>",
              "range": "area_angle_range<direction>"
          },
    "name": "area_name",
    "points":
     [
       { % point #1
```

```
            "x": "x_value <integer>"
            "y": "y_value <integer>"
        },
        { % point #2
          "x": "x_value <integer>"
          "y": "y_value <integer>"
        },
        ...,
        { % point #n (>= 3)
          "x": "x_value <integer>"
          "y": "y_value <integer>"
        }
    ]
  },
  { % Area #2
   "angle": {
             "direction": "area_angle_direction<integer>",
             "range": "area_angle_range<direction>"
          },
   "name": "area_name",
   "points":
    [
      { % point #1
          "x": "x_value <integer>"
          "y": "y_value <integer>"
        },
        { % point #2
          "x": "x_value <integer>"
          "y": "y_value <integer>"
        },
        ...,
        { % point #n (>= 3)
          "x": "x_value <integer>"
          "y": "y_value <integer>"
        }
    ]
  },
  ...
  { % Area #n (>= 1)
   "angle": {
             "direction": "area_angle_direction<integer>",
             "range": "area_angle_range<direction>"
          },
   "name": "area_name",
   "points":
    [
      { % point #1
          "x": "x_value <integer>"
```

```
            "y": "y_value <integer>"
        },
        { % point #2
          "x": "x_value <integer>"
          "y": "y_value <integer>"
        },
        ...,
        { % point #n (>= 3)
          "x": "x_value <integer>"
          "y": "y_value <integer>"
        }
    ]
  }
]
```

Success response

HTTP Status code	Method	Sample Response

200	GET	<pre>{
		  "points": [
		  {
		  "x": 22,
		  "y": 320
		  },
		  {
		  "x": 1220,
		  "y": 150
		  },
		  {
		  "x": 1226,
		  "y": 734
		  },
		  {
		  "x": 22,
		  "y": 32
		  }
		  ]
		}</pre>

200	PATCH	<pre>{
		  "status": "success",
		  "msg": "ok"
		}</pre>

Examples:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # get ROI properties

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/roi/brightness

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/roi/alpr

 # set ROI properties

curl --cookie cookies.log --cookie-jar cookies.log -X PATCH http://192.168.3.15/roi/brightness --data '{"points": [{"x": 22,"y": 320},{"x": 1220,"y": 150},{"x": 1226,"y": 734},{"x": 22,"y": 32}]}'

curl --cookie cookies.log --cookie-jar cookies.log -X PATCH http://192.168.13.231/roi/alpr --data '[{"angle": {"direction": 0,"range": 128},"name": "area_1","points": [{"x": 1226,"y": 734},{"x": 374,"y": 778},

{"x": 338,"y": 70},{"x": 1226,"y": 112}]},{"angle": {"direction": 180,"range": 112},"name": "area_2","points": [{"x": 422,"y": 730},{"x": 486,"y": 84},{"x": 26,"y": 116},{"x": 30,"y": 736}]},{"angle": {"direction": 78,"range": 79},"name": "area_3","points": [{"x": 326,"y": 844},{"x": 368,"y": 60},{"x": 28,"y": 116},{"x": 42,"y": 736}]}]'

3.3 Classifier information

Description

This API Endpoint returns information regarding the currently used classifier.

URL	Method	URL Parameter	Data Parameter
/classifier/info	GET	None	None

Success response

HTTP Status code	Method	Sample Response

200	GET	{
		"classifier": "West Europe",
		"states": [
		"Austria",
		"Belgium",
		"Bulgaria",
		"Czech Republic",
		"France", "Germany",
		"Hungary", "Italy",
		"Latvia", "Lithuania"
		],
		"date": "2018-08-28",
		"name": "eu_001.dat"
		}

Examples:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # get info

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/classifier/info

4 ACCESS Properties

4.1 GET AND SET Access Lists

Description

API Endpoints for retrieving and adding the access lists of the ANPR LUMO. There are four lists:

- white list listname = 'white.txt'
- black list listname = 'black.txt'
- ignore list listname = 'ignore.txt'
- wiegand matchlist listname = ‘wiegandlist.txt‘

Use the GET method to retrieve the property and the PATCH method to set it.

URL	Method	URL Parameter	Data Parameter
/access/:listname	GET	:listname	:filter (optional, can be empty)

/access/:listname	PATCH	:listname	:entries
/access/:listname	DELETE	:listname	:specifier

Retrieving an access list

With a GET request a list of plates is returned. The plates can be filtered by sending a JSON document containing a pattern field that contains a regular expression (RegEx).

Add a plate to an access list

With a PATCH request a new entry for the access list can be added. The body of the PATCH request should contain a JSON document in same format as successful GET request response document.

Delete a plate from an access list

With a DELETE request, one or more entries can be deleted by either specifying one of the two fields in a JSON document inside the request body:

- index - delete a single item with the given index number
- pattern - delete every plate that matches the given pattern.

Parameter

:listname

Name of the list e.g. white.txt, black.txt, ignore.txt, wiegandlist.txt

:filter

JSON string of following format:

```
{
        "pattern":"your_pattern"
}
```

:entries

JSON string of following format:

```
 [
  {
   "always": true or false,  % if true specified plate will be always valid, otherwise it will be valid for defined schedules %
   "plate": "plate string",
   "schedules":  % if always set to false %
        [
         {
           "period":
           {
            "end": "yyyy-mm-dd",
            "start": "yyyy-mm-dd"
           },
           "time-span":
            {
              "from": "hh:mm",
              "to": "hh:mm"
            },
            "weekdays": ["Mo"] % list of days for which entry is valid ("Mo", "Tu", "We", "Th", "Fr", "Sa", "So") %
         } % this can be repeated 0 or more times %
        ]
  },
  {
```

} % there can be added multiple plates with a single call
]

:specifier

JSON string of one of following 2 formats:

```
{
  "index" : "entry_index<integer>"
}
{
  "pattern":"your_pattern"
}
```

Success response

HTTP Status code	Method	Sample Response

200	GET	<pre>[
		  {
		  "always":false,
		  "index":0,
		  "plate":"HHUL1114",
		  "schedules":[
		  {
		  "period":
		  {
		  "start":"2019-10-01",
		  "end":"2019-10-31"
		  },
		  "time-span":
		  {
		  "from":"08:00",
		  "to":"18:30"
		  },
		  "weekdays": ["Mo","Tu","We"]
		  }
		  ]
		  },
		  {
		  "always":true,
		  "index":1,
		  "plate":"80XHZ7"
		  },
		  {
		  "always":true,
		  "index":2,
		  "plate":"SBHR01"
		  }
		]</pre>

200	POST	<pre>{
		  "status": "success",
		  "msg": "Entry added"
		}</pre>

200	DELETE	<pre>{
		  "status": "success",
		  "msg": "Entry removed"
		}</pre>

Examples:

 # request whitelist

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/access/white.txt

 # add plate HHKF1114 to whitelist

```
curl --cookie cookies.log --cookie-jar cookies.log -X POST http://192.168.3.15/access/white.txt
  --data '[{"always":true,"plate":"HHKF1114"}]'

# add HHKF1115 to whitelist which is valid from 01-10-2019 until 31-10-2019 on Monday, Tuesday and
Wednesday from 08:00 until 18:30
curl --cookie cookies.log --cookie-jar cookies.log -X POST
  http://192.168.3.15/access/white.txt --data
'[{"always":false,"plate":"HHKF1115","schedules":[{"period":{"start":"2019-10-01", "end":"2019-10-31"},"time-
span":{"from":"08:00","to":"18:30"},"weekdays": ["Mo","Tu","We"]}]}]'

# request Wiegand matchlist
curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/access/wiegandlist.txt

# add HHKF1116 to Wiegand matchlist and match this to Wiegand ID 213
curl --cookie cookies.log --cookie-jar cookies.log -X POST http://192.168.3.15/access/wiegandlist.txt  --data
'[{"always":true,"plate":"HHKF1116", "id": 213}]'

# delete plate with index number 1 from whitelist
curl --cookie cookies.log --cookie-jar cookies.log -X DELETE http://192.168.3.15/access/white.txt
  --data '{"index":1}'

# delete plates with pattern HH from whitelist
curl --cookie cookies.log --cookie-jar cookies.log -X DELETE http://192.168.3.15/access/white.txt
  --data '{"pattern":"HH*"}'
```

4.2 Read and write to Access Lists files

Description

While the API endpoints from the previous section provide simplified interface to add/parse list entries, this Endpoint is used to download and/or to upload(replace) full list file. The format of those files is described in the installation guide. This endpoint is recommended to use in order to make a backup of the lists or to share list files across multiple devices. Use GET method to retrieve the a file and the PUT method to write to it.

URL	Method	URL Parameter	Data Parameter
/access/:listname/file	GET	:listname	None
/access/:listname/file	PUT	:listname	List=:filename

Parameters

listname

Key used to specify list from table from Available Lists (white, black, ignore, wiegandlist.txt).

filename

Filename of the local list file used for upload.

Examples:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # read access list file

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/access/white.txt/file

 # write to access list file

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/access/white.txt/file -X PUT --form list=@new-list.txt

5 ACTIONS

This Section contains the API Endpoints to retrieve and configure actions.

5.1 Retrieve list of available actions

Description

API Endpoint for retrieving the list of available action types with related parameters. These actions can be used to define actions for the given event types. The event types can be listed by fetching the action properties.

URL	Method	URL Parameter	Data Parameter
/available_actions	GET	None	None

Success response

HTTP Status code	Method	Sample Response
200	GET	{ "digitalout": ["enable", "areas", "output line", "invert signal", "output duration", "non-blocking"], "ftpdatabase": ["enable", "areas", "host", "user", "password", "filename", "entry pattern"], "ftpupload": ["enable", "areas", "host", "user", "password", "naming pattern", "image upload", "final width", "final height", "crop percentage", "jpeg quality", "upload overview", "overview width", "overview heigth", "overview crop", "overview jpeg"], "http": ["enable", "areas", "request type", "url", "user", "password", "message" }}

Examples:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # retrieve actions

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/ available_actions

5.2 Get and Set actions

Description

API Endpoints for manipulation of the actions for given event type. For each event type there can be defined one or multiple actions. As mentioned in previous section the action types together with their parameters can be listed using /available_actions endpoint.

It is possible to define multiple actions of the same type for any event type. List of the event types together with their values is obtainable by fetching the action properties. Use GET method to retrieve the actions defined for specified property, PATCH method to add action to it and DELETE to delete action at the given index.

Event types

From Release 4.3.2 there are following event types (action properties) available:

"blackmatch"
"blackmismatch"
"ignorematch"
"ignoremismatch"
"lcr"
"noread"
"read"
"whitematch"
"whitemismatch"
"wrongdir"

URL	Method	URL Parameter	Data Parameter
/action/:property	GET	Event type	None
/action/:property	PATCH	Event type	:action_struct
/action/:property	DELETE	Event type	:specifier

Success response

HTTP Status code	Method	Sample Response

200	GET	<pre>[
		  {
		  "name": "serial",
		  "params": {
		  "areas": "-1",
		  "baud rate": "9600",
		  "enable": "True",
		  "message": "%LP%"
		  }
		  }
		]</pre>

Examples:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # read access list file

curl --cookie cookies.log --cookie-jar cookies.log -GET http://192.168.13.231/action/read

 # add serial action to the read events

curl --cookie cookies.log --cookie-jar cookies.log -X PATCH http://192.168.3.15/action/read --data '{"name":"serial", "params":{"enable":true, "areas":-1, "baud rate":115200, "message":"%LP%%CR%%LF%"}}'

 # delete first action from the read action list

curl --cookie cookies.log --cookie-jar cookies.log -X DELETE http://192.168.3.15/action/read --data '{"index":1}'

6 Retrieve results

This section contains API Endpoints to retrieve results captured by the camera. There are two types of results available on the camera:
Live result is not saved to the camera's hard drive. It contains only the latest (if there is one) result that was captured by the camera. It cannot be deleted and therefore there is a single API Endpoint related to live result that allows to retrieve it as JSON from the camera.
Text results are saved to the camera's hard drive. The number of text results to be saved is limited by the anpr parameter MaxHistoryEntries. API Endpoints listed in this section offer to the user ability to fetch and to delete these results.
Live snapshot returns a snapshot of the current image from the ANPR LUMO.

6.1 Retrieve Live Result

Description
This API Endpoint returns the latest result that was captured by the ANPR LUMO camera.

URL	Method	URL Parameter	Data Parameter
/live/result	GET	None	None

Success response

HTTP Status code	Method	Sample Response

200	GET	{
		"image": {
		"data": BASE64 IMAGE",
		"height": 1,
		"step": 1,
		"type": -1,
		"width": 59951
		},
		"message": "",
		"plates": [
		{
		"charactercolor": 0,
		"confidence": 100,
		"platecolor": 2,
		"position": {
		"height": 0,
		"width": 0,
		"x": 0,
		"y": 0
		},
		"rows": 1,
		"state": "DE",
		"state_confidence": 99,
		"value": "HHKF1115",
		"value_with_spaces": "HH KF 1115"
		}
		],
		"timestamp": "20190228T084348.118500"
		}

Examples:
 # login
curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # retrieve and save the snapshot
curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/live/result

6.2 Retrieve Statistics

Description
This API Endpoint returns detection statistics of the engine running on the ANPR LUMO camera.

URL	Method	URL Parameter	Data Parameter
/live/result	GET	Statistics part	None

Success response

HTTP Status code	Method	Sample Response

200	GET	<pre>{
		"characters_height": {
		"avg": 43.30002212524414,
		"max":44.0,
		"min":42.0
		},
		"detections_count": {
		"negative": 0,
		"positive": 4673
		},
		"detections_heat_map": "..."
		}</pre>

Examples:
 # login
curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # retrieve whole statistics
curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/live/result/statistics/all

 # retrieve character height statistics of detected plates
curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/statistics/characters_height

6.3 Retrieve TEXT Result

Description
This API Endpoint returns a specified amount of results that were last captured by the ANPR LUMO camera.

Filter results
With a GET request, that has a JSON body containing a field "count" holding a number, a list of results is returned. The amount of results is smaller or equal to the specified number. The "count" field is mandatory on every GET request. With an additional "include-images" field in the JSON body, the image of the result is returned as a base64 encoded string. There are also some optional parameters for filtering:
"plate" - if a result's plate matches the given string in this field, it will be included in the output
"state" - if a result's state matches the given string, it will be included
"min-confidence" and "max-confidence" - only results that have at least or at most the specified confidence in percent will be included
"min-state-confidence" and "max-state-confidence" - same as the above parameter, but with the state confidence of the result

"min-date" and "max-date" - specify a certain date range for the results in the format of "YYY-MM-DD", e.g. "2019-02-02" for the 2nd February 2019
"min-time" and "max-time" - specify a certain daytime range for the results in the format of "HH:MM", e.g. "13:00" for 1:00 pm.

Delete results

Using a DELETE request with a JSON body containing either the field "all" or "pattern", results can be deleted. If "all" is set, all results will be deleted. If "pattern" is set and contains a string, every result that can be matched with the string pattern will be deleted.

URL	Method	URL Parameter	Data Parameter
/text/results	GET	None	JSON
/text/results	DELETE	None	JSON

Success response

HTTP Status code	Method	Sample Response
200	GET	`{ "results": [{ "plates": [{ "value": "LPR12345", "value-with-spaces": "LPR 123 45", "confidence": 90, "state": "NL", "state_confidence": 89, "position": { "x": 342, "y": 440, "width": 59, "height": 28 }, "rows": 1, "plate-color": "#FFFFFF", "character-color": "#000000", ... }], "timestamp": 1525690269, "message": "" }, ...] }`
200	DELETE	`{ "status": "success", "msg": "ok", "removed_results": 3, "deletion_errors": 0 }`

Examples:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # retrieve 5 results

```
curl --cookie cookies.log --cookie-jar cookies.log -X GET http:// 192.168.3.15/text/results
--data '{"count":5}'

# retrieve 5 results with images
curl --cookie cookies.log --cookie-jar cookies.log -X GET http:// 192.168.3.15/text/results
--data '{"count":5,"include-images":true}'

# retrieve 10 results with a minimum confidence of 60 percent
curl --cookie cookies.log --cookie-jar cookies.log -X GET http:// 192.168.3.15/text/results
--data '{"count":10,"min-confidence":60}'

# retrieve 10 results that were saved between 11:00 and 13:00
curl --cookie cookies.log --cookie-jar cookies.log -X GET http:// 192.168.3.15/text/results
--data '{"count":10,"min-time":"11:00","max-time":"13:00"}'

# retrieve 3 results that were saved between february and march
curl --cookie cookies.log --cookie-jar cookies.log -X GET http:// 192.168.3.15/text/results
--data '{"count":3,"min-date":"2019-02-01","max-date":"2019-03-31"}'

# retrieve 5 results which states contain the pattern 'NL'
curl --cookie cookies.log --cookie-jar cookies.log -X GET http:// 192.168.3.15/text/results
--data '{"count":5,"state": "*NL*"}'
```

6.4 Export all results as ZIP archive

Description

This API Endpoint returns a ZIP archive file containing all the gathered results by the ANPR LUMO as CSV file. The images are included.

URL	Method	URL Parameter	Data Parameter
/export/results	GET	None	none

Examples:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # Save ZIP archive

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/export/results > results.zip

Success response

HTTP Status code	Method	Sample Response
200	GET	`ZIP archive`

6.5 Retrieve a Live Image (snapshot)

Description

This API Endpoint returns a snapshot of the current image from the ANPR LUMO camera. The image is JPEG encoded.

URL	Method	URL Parameter	Data Parameter
/live/snapshot	GET	None	none

Success response

HTTP Status code	Method	Sample Response
200	GET	`Images encoded as JPEG`

Examples:

 # login

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/login --data "username=admin&password=secret"

 # retrieve and save the live image

curl --cookie cookies.log --cookie-jar cookies.log http://192.168.3.15/live/snapshot > image.jpg

7 General errors

The following HTTP errors responses may occur on every request:

HTTP Status code	Sample response

400	{
	"status": "error",
	"msg": "The browser (or proxy) sent a request that this server could not understand."
	}

403	{
	"msg": "Token invalid",
	"status": "error"
	}

500	{
	"status": "error",
	"msg": "Request failed"
	}

8 Deprecated API calls

Since version 4.6.0 the following calls are deprecated.

8.1 Retrieve previous Results (without images)

Description

This API Endpoint returns a specified amount of results that were last captured by the ANPR LUMO camera. The image data is not included. The URL parameter specifies the number of returned results.

Note

This API Endpoint parses the results saved in the camera. The maximum count of saved results is set by the parameter "HISTORY LIST LENGHT" and “HISTORY HOURS” and has a default value 0. In order to enable saving the results at the camera, this parameter should be set to a value >0. This can be done in the web interface at the menu “TEXT RESULTS” In the case that the specified count of results has a larger value than “History List Length” parameter, only the results found saved in the camera will be sent.

URL	Method	URL Parameter	Data Parameter
/text/results/:count	GET	:count - number of returned results	None

8.2 Retrieve previous Results (with images)

Description

This API Endpoint returns a specified amount of results that were last captured by the ANPR LUMO camera. The image data is included. The URL parameter specifies the number of returned results.

Note

This API Endpoint parses the results saved in the camera. The maximum count of saved results is set by the parameter "HISTORY LIST LENGHT" and “HISTORY HOURS” and has a default value 0. In order to enable saving the results at the camera, this parameter should be set to a value >0. This can be done in the web interface at the menu “TEXT RESULTS” In the case that the specified count of results has a larger value than “History List Length” parameter, only the results found saved in the camera will be sent.

URL	Method	URL Parameter	Data Parameter
/text/results-with-images/:count	GET	:count - number of returned results	None

8.3 Delete All Saved Results

Description

This API Endpoint is used to clear all text results saved on the camera. The return message contains count of deleted results and count of errors appeared during deletion.

URL	Method	URL Parameter	Data Parameter
/text/remove-all	PATCH	None	None

8.4 Delete Saved Results Using Pattern

Description

This API Endpoint is used to delete specified text results saved on the camera. The return message contains count of deleted results and count of errors appeared during deletion.

Note

When using a pattern that contains a character range, you must use either quotation marks with backslash in front of squared brackets, or -g flag in front of URL. (see examples) Otherwise an error will be returned similar to the following one: "`curl: (3) [globbing] bad range in column`".

URL	Method	URL Parameter	Data Parameter
/text/remove-pattern/:pattern	PATCH	:pattern	:pattern=:pattern used for matching

A Document revision

Version	Date	Comment
1.06	2023-11-09	Minor improvements
1.05	2022-08-29	new features for version 4.7.0
1.03	2020-01-18	new features for version 4.5.0
1.02	2019-09-18	Added new features for version 4.3.2
1.01	2019-02-28	HR reviewed
1.00	2019-02-27	Created