

MOBOTIX SYNC Server



API definition Guide

Version 3.2.0

(Last update: 23/8/25)

About this manual

This manual is intended for developers of MOBOTIX SYNC Server API application and is applicable to its version 1.0. Covers access to configuration, operation, audit and SignalR messages of MOBOTIX SYNC Server. The most recent version of this document is available at www.vaxtor.com.

Before using MOBOTIX SYNC ServerAPI it is very important to read the whole this document and MOBOTIX SYNC user and administration manual to understand concepts and configurations.

Important!

This **MOBOTIX SYNC ServerAPI definition Guide** applies to contemporary or later SYNC server (or plugin).

For previous MOBOTIX SYNC Server (or plugin) distribution, review **MOBOTIX SYNC ServerAPI definition Guide Version 1.8.0** or previous.

The software described in this document is provided under license. The software may only be used or copied in the cases described in the license agreement. Whole or part reproduction of this document, in any form, electronic or another, and for any purpose whatsoever, without written authorization of Vaxtor is prohibited

Table of Contents

HOW TO USE THIS GUIDE	9
REST API	9
REST API VERBS.....	9
REQUEST & RESPONSE.....	9
HTTP REST URI	9
HOW TO BUILD MOBOTIX SYNC SERVER API URIS	10
QUERY STRING	10
FILTERS.....	11
BODIES: REQUEST & RESPONSE	11
PAGED RESPONSES	12
MOBOTIX SYNC SERVER API	13
COMMON PARAMETERS AND RESPONSE CODES	13
API KEY PARAMETER	13
PAGING PARAMETERS	13
DOWNLOAD CSV - FILTERING AND COLUMNS	13
PAGING AND FILTERING PARAMETERS	14
UPLOAD FILE PARAMETER.....	14
RESPONSE CODES.....	14
URL.....	15
ACTUATORS	16
ACTUATORS	16
ACTUATOR LINK	17
ACTUATORS VALID TYPES.....	18
AUDIT	19
AUTHORIZATION	20
CAMERAS	22
AFORO (CAPACITY)	24
CATEGORIES	25
CONFIGURATIONS	28
ALERT CONFIG.....	28
LDAP CONFIG.....	28
PLATE MANAGER CONFIG.....	29
SERVER CONFIG.....	29
SMTP CONFIG	30
SMTP TEST	30
TELEGRAM CONFIG.....	31
EXTERNAL	31
GRAFANA	33

HEARTBEAT	35
GDRP	36
IMAGES	36
LINKED CAMERAS	37
PEOPLE	40
PLATES	41
PLATES.....	41
ASSOCIATES.....	42
PLATE LISTS	44
PLATE LIST ITEMS	45
REGULAR EXPRESSIONS (SCHEDULE)	47
ROLLCALL	48
SCHEDULES	49
SECTIONS	51
TIME FRAMES (SCHEDULE)	53
USERS	54
ZONES	56
PLUGINS API	58
SANCTIONS	58
SANCTION CONFIG	58
CANCELLATIONS.....	58
SANCTIONS.....	61
CAMERA SANCTIONS	62
PLATE SANCTIONS	64
FILE DOWNLOADS	66
AUDIT RECORD	66
AUTHORIZATION	66
CATEGORY	67
GDPR	67
OCR PLATE IMAGES	67
OCR AND ENVIRONMENT IMAGES	67
PEOPLE	68
PLATES	68
REGISTERS.....	68
IMAGES.....	69
PLATE LIST ITEMS	69
FILE UPLOAD	70

AUTHORIZATION	70
CATEGORY	70
IMAGE UPLOAD	70
PEOPLE	71
PERSON IMAGE	71
PLATE LIST ITEMS	72
SIGNALR MESSAGING	74
COMMON	74
COMMON SCHEMA.....	74
MESSAGES BODIES	74
AFORO MESSAGE.....	74
LISTITEM MESSAGE.....	74
PLATE MESSAGE.....	74
ALERT MESSAGE	75
ROWCALL MESSAGE	76
DATA MODELS	77
COMMON ELEMENTS	77
PAGING RESPONSE FIELDS.....	77
NOTES.....	77
ACTUATORS	78
ACTUATOR	78
VALIDTYPES	78
ACTUATORLINK	79
AUDIT	79
PAGE AUDIT	79
AUDIT	79
AFORO	80
AFORO (CAPACITY)	80
AFORO FIELDS VALIDATORS	81
AUTHORIZATION	81
PAGE AUTHORIZATION	81
AUTHORIZATION	81
AUTHORIZATION CSV	82
ZONESCHEDULE	82
CAMERAS	84
PAGE CAMERA	84
CAMERA	84
CATEGORY	86
PAGE CATEGORY	86

CATEGORY	86
CATEGORY CSV	88
CONFIGURATION	88
ALERTCONFIG	88
LDAPCONFIG	92
PLATEMANAGERCONFIG	92
REASONCONFIG	93
SERVERCONFIG	93
SMTPCONFIG	95
SMTPTTEST	96
TELEGRAMCONFIG	96
EXTERNAL DATA	97
PAGEEXTERNAL	97
EXTRENAL	97
GRAFANA	97
GRAFANA	97
HEARTBEATS	97
HEARTBEAT	97
HEARTBEATRESPONSE	98
IMAGES	99
PAGEIMAGE	99
IMAGE	100
LINKED CAMERAS	101
PAGELINKEDCAMERA	101
LINKEDCAMERA	101
LISTITEM	101
PAGELISTITEM	101
LISTITEM	101
LISTITEM CSV	102
PERSON	104
PAGEPERSON	104
PERSON	104
PERSON CSV	105
PLATES	106
PAGEPLATE	106
PLATE	106
PLATE LISTS	109
PAGEPLATELIST	109
PLATELIST	109
REGULAR EXPRESSION (AUTHEXPRESSION)	111
PAGEAUTHEXPRESSION	111
AUTHEXPRESSION	111
ROLLCALL	111
PAGEROLLCALL	111

ROLLCALL	112
SCHEDULE	113
PAGE SCHEDULE	113
SCHEDULE	113
SECTION	113
PAGE SECTION	113
SECTION	114
TIMEFRAME	114
PAGE TIMEFRAME	114
TIMEFRAME	114
USER	116
PAGE USER	116
USER	116
ZONE	118
PAGE ZONE	118
ZONE	118
ZONE FIELDS VALIDATORS	118
PLUGINS DATA MODELS	120
SANCTIONS PLUGIN	120
SANCTIONCONFIG	120
CAMERASANCTIONS	120
CAMERASANCTIONS FIELDS VALIDATORS	120
PAGE CANCELLATION	120
CANCELLATION	120
CANCELLATION FIELDS VALIDATORS	121
PAGE PLATESANCTIONS	121
PLATESANCTIONS	121
PLATESANCTIONS FIELDS VALIDATORS	122
PAGE SANCTION	122
SANCTION	122
SANCTION FIELDS VALIDATORS	123
VALID TYPES	123
EXPORT SANCTIONS	123
EXPORTSANCTIONCONFIG	123
EXPORTSANCTIONCONFIG FIELDS VALIDATORS	124
VALID TYPE	124
FILTER KEYS	125
NOTES	125
AUDITS	125

AUTHORIZATIONS	125
CATEGORY	126
PERSON	126
PLATES.....	126
PLATE LIST ITEMS	128
ROLLCALL	128
CLAIMS	129
ACCESSES ALLOWED	129
ANNEXES	132
HOW TO USE POSTMAN	132
HOW TO USE SWAGGER	133
API DOCUMENTATION.....	133
CODE GENERATION.....	134
CLIENT CODE EXAMPLE (C#).....	134

How to use this guide

REST API

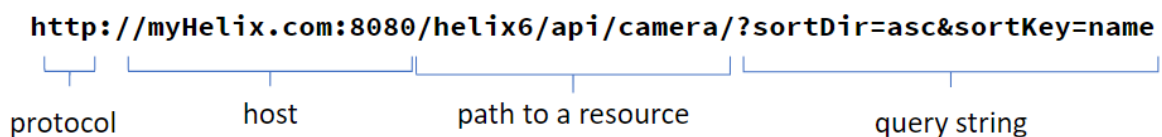
REST API Verbs

Verb	/camera	/camera/1234	/camera/all
POST (Create)	Create a new Camera whose data are included in body	Error 405	Error 405
GET (Read)	Paged list of Cameras	Camera with ID = 1234	List of all Cameras
PUT (Update)	Update Camera whose data are included in body (It must have ID)	Update Camera with ID = 1234 and whose data are included in body NOTE: If body data include ID, MOBOTIX SYNC will ignore it.	Error 405
DELETE	Error 405	Delete Camera ID = 1234	Error 405

Request & Response

Operation	Verb	Recommendation	Response
Update a resource	PUT	<ul style="list-style-type: none"> Need ID in order to submit the request Idempotent and not safe (has side effect) operation Not necessary to return a link via a Location header 	<ul style="list-style-type: none"> 201 (OK), with data 404 (Not found)
Creation of new resource	POST	<ul style="list-style-type: none"> Not idempotent and not safe 	<ul style="list-style-type: none"> 201 Request accepted
Read	GET	<ul style="list-style-type: none"> Idempotent and safe operation 	<ul style="list-style-type: none"> 200 (OK), with data 404 (Not found)
Deleting a resource	DELETE	<ul style="list-style-type: none"> Idempotent and not safe operation 	<ul style="list-style-type: none"> 200 (OK), without data 404 (Not found)

HTTP REST URI



- The protocol identifies the transport scheme that will be used to process and respond to the request.
- The host name identifies the server address of the resource.
- The path and query string can be used to identify and customize the accessed resource.
- The body (not showed) transports transaction data to server.
- HTTP is built in with a set of status codes for various types of scenarios:
 - **2xx** Success (200 OK, 201 Created...)
 - **3xx** Redirection (303 See other)
 - **4xx** Client error (404 Not Found)
 - **5xx** Server error (500 Internal Server Error)

How to build MOBOTIX SYNC Server API URIs

Query string

Plates

GET	{baseUrl}/api/plate	
Description	Get Plates by pages.	
Filter	Filter keys (Plates)	
Include	Paging and Filtering parameters (optional)	
Response		
Code	Description	Body response
200	Filtered and paged Plate records	PagePlate

Plates

Concept	filterKey	filterValue
CAMERA	camera	Camera ID or camera name
COUNTRY	country	Three letters Country code
DIRECTION	direction	unknown, gettingCloser or gettingFarther
PLATE	plate	p.e. M4618WZ
STATUS	status	ok, pending, confirmed, cancelled, exported, prescribed, noAuth or sanctionPending
ZONE	zone	Zone ID or Zone name
CAPTURER ID	captureId	Capturer ID
CAPTURERTYPE	captureType	Capturer Type
ALARM	alarm	true or false
MAX LATITUDE	maxlatitude	Max latitude (double)
MIN LATITUDE	minlatitude	Min latitude (double)
MAX LONGITUDE	maxlongitude	Max longitude (double)
MIN LONGITUDE	minlongitude	Min longitude (double)
DATE FROM	dateFrom	DateTime format dd/MM/yyyy hh:mm:ss example: 08/01/2018 16:22:29
DATE TO	dateTo	DateTime format dd/MM/yyyy hh:mm:ss example: 08/01/2018 16:22:29
SPEED	speed	Speed (integer)
VEHICLEMAKER	vehicleMaker	Vehicle maker (text)
VEHICLEMODEL	vehicleModel	Vehicle model (text)
VEHICLECOLOR	vehicleColor	Vehicle color (text)

Parameter	Description	Where	Required	Type	Format
sortKey	Column name to sort Records	Query	false	string	string
sortDir	Sort direction: ASC (ascendent) DESC (descendent)	Query	false	string	string
page	Page number	Query	false	integer	int32
pageSize	Records per page	Query	false	integer	int32
filterKey	List of search criteria names See Filter Keys	Query	false	string	string
filterValue	List of search criteria values See Filter Keys	Query	false	string	string

<http://myHelix.com:8080/helix6/api/camera/?sortDir=asc&sortKey=name>

Filters

Paging and Filtering parameters					
Parameter	Description	Where	Required	Type	Format
sortKey	Column name to sort Records	Query	false	string	string
sortDir	Sort direction: ASC (ascendent) DESC (descendent)	Query	false	string	string
page	Page number	Query	false	integer	int32
pageSize	Records per page	Query	false	integer	int32
filterKey	List of search criteria names See Filter Keys	Query	false	string	string
filterValue	List of search criteria values See Filter Keys	Query	false	string	string

Plates

Concept	filterKey	filterValue
CAMERA	camera	Camera ID or camera name
COUNTRY	country	Three letters Country code
DIRECTION	direction	unknown, gettingCloser or gettingFarther
PLATE	plate	p.e. M456WZ
STATUS	status	ok, pending, confirmed, cancelled, exported, prescribed, noAuth or sanctionPending
ZONE	zone	Zone ID or Zone name
CAPTURER ID	captureId	Capturer ID
CAPTURERTYPE	captureType	Capturer Type
ALARM	alarm	true or false
MAX LATITUDE	maxlatitude	Max latitude (double)
MIN LATITUDE	minlatitude	Min latitude (double)
MAXLONGITUDE	maxlongitude	Max longitude (double)
MIN LONGITUDE	minlongitude	Min longitude (double)
DATE FROM	dateFrom	Date/Time format dd/MM/yyyy hh:mm:ss example 08/01/2018 16:22:29
DATE TO	dateTo	Date/Time format dd/MM/yyyy hh:mm:ss example 08/01/2018 16:22:29
SPEED	speed	Speed (integer)
VEHICLEMAKER	vehicleMaker	Vehicle marker (text)
VEHICLEMODEL	vehicleModel	Vehicle model (text)
VEHICLECOLOR	vehicleColor	Vehicle color (text)

<http://myHelix.com:8080/helix6/api/plates?>

filterKey=["country","plate","status"]&filterValue=["\"SGP\"","\"AA0032\"","\"sanctionPending\""]&page=1&pageSize=20&sortDir=desc&sortKey=takenOn&totalRecords=0

Bodies: Request & Response

PUT {baseUrl}/api/camera					
Description Update Camera					
Parameter	Description	Where	Required	Type	Format
camera	Camera record.	Body	true	object	JSON

Response		
Code	Description	Body response
202	Updated Camera record	Camera

Camera		
Field	Type	Example
name	String	"Test camera"
plateReaderId	Integer (int32)	2
type	"reader" "environment"	"reader"
zoneId	Integer (int32)	2
reportType	"both" "freeflow" "signaled"	"both"
captureId	String	12
expectedDirection	"unknown" "gettingCloser" "gettingFarther"	"unknown"
graceHour	Integer (int32)	8
address	String	"Po Castellana, 123 - Madrid"
latitude	Double	12.1234
longitude	Double	12.1234
id	Integer (int32)	7

```
{
  "name": "Camera 1",
  "plateReaderId": 0,
  "type": "reader",
  "address": "",
  "zoneId": 7,
  "reportType": "both",
  "captureId": "Set1",
  "expectedDirection": "unknown",
  "graceHours": 0,
  "id": 13
}
```

Paged responses

Cameras

GET	{baseUrl}/api/camera
Description	Get Camera by pages.
Include	Paging parameters (optional)
Response	
Code	Body response
200	Paged Camera records PageCamera

```

{
  "page": 1,
  "pageSize": 10,
  "sortKey": "ID",
  "sortDirection": "descending",
  "totalRecords": 9,
  "records": [
    {
      "name": "Camera 1",
      "plateReaderId": 0,
      "type": "reader",
      "address": "",
      "zoneId": 7,
      "reportType": "both",
      "captureId": "Set1",
      "expectedDirection": "unknown",
      "graceHours": 0,
      "id": 13
    },
    ...
  ]
}
    
```

Cameras

PageCamera		
Include Paging response fields		
Field	Type	
records	Camera	See Camera
Camera		
Field	Type	Example
name	String	"Test camera"
plateReaderId	Integer Sint32	2
type	"reader" "environment"	"reader"
zoneId	Integer Sint32	2
reportType	"both" "freeflow" "signaled"	"both"
captureId	String	12
expectedDirection	"unknown" "gettingCloser" "gettingFarther"	"unknown"
graceHour	Integer Sint32	6
address	String	"Pe Castellana, 113 - Madrid"
latitude	Double	12.1234
longitude	Double	12.1234
id	Integer Sint32	7

MOBOTIX SYNC Server API

Common Parameters and Response codes

API Key parameter

Important!:

- All API calls must send Api Key to server using ONE of following alternatives

Parameter	Description	Where	Required	Type	Format
x-api-key	API key	Header	true		
ApiKey	API Key	Query	true	string	string

Paging parameters

Parameter	Description	Where	Required	Type	Format
sortKey	Column name to sort records	Query	false	string	string
sortDir	Sort direction: ASC (ascendent) , DESC (descendent)	Query	false	string	string
page	Page number	Query	false	integer	int32
pageSize	Records per page	Query	false	integer	int32

Download CSV - Filtering and Columns

Parameter	Description	Where	Required	Type	Format
columns	Comma (,) separated names of columns	Query	false	string	string
sortKey	Column name to sort records	Query	false	string	string
sortDir	Sort direction: ASC (ascendent) , DESC (descendent)	Query	false	string	string
filterKey	List of search criteria names	Query	false	string	string

	See Filter Keys				
filterValue	List of search criteria values See Filter Keys .	Query	false	string	string

Paging and Filtering parameters

<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
sortKey	Column name to sort Records	Query	false	string	string
sortDir	Sort direction: ASC (ascendent) , DESC (descendent)	Query	false	string	string
page	Page number	Query	false	integer	int32
pageSize	Records per page	Query	false	integer	int32
filterKey	List of search criteria names See Filter Keys	Query	false	string	string
filterValue	List of search criteria values See Filter Keys .	Query	false	string	string

Upload file parameter

<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
Content-Type	multipart/form-data	Header	true		

Response codes

Note:

- The server responses to API calls could have one of these codes.
- Following pages just document the correct responses.

<i>Code</i>	<i>Description</i>	<i>Body response</i>
200	Ok (GET DELETE)	Relative to Route/Endpoint and Method.
201	Ok (POST)	Relative to Route/Endpoint

Response codes

202	Ok (PUT)	Relative to Route/Endpoint
400	Bad Request	String with error. Example: "Unexpected character encountered while parsing value: l. Path ", line 0, position 0."
401	Unauthorized	JSON with error. Example: { "error": "Unauthorized" }
403	Forbidden	Empty
404	Not found	JSON with error. Example: { "statusCode": "notFound", "message": "The resource you have requested cannot be found.", "details": "Error details are currently disabled. Please set <code>StaticConfiguration.DisableErrorTraces = false;</code> to enable." }
405	Method not allowed	
500	Bad request	JSON with error. Example: { "message": "Error message", "description": "Error description" }

URL

<i>URL</i>	<i>Description</i>	<i>Example</i>
baseURL	Application URL	https://192.168.0.88:3000/SYNC

Actuators

Actuators

GET	{baseUrl}/api/actuator/{id}				
<i>Description</i>	Get actuator by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Actuator ID	Path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	actuator record	Actuator			
GET	{baseUrl}/api/actuator/all				
<i>Description</i>	Get all Actuators				
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	All Actuators	List of Actuator			
GET	{baseUrl}/api/actuator/{id}/test				
<i>Description</i>	Test actuator connection by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Actuator ID	Path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Connection OK				
POST	{baseUrl}/api/actuator				
<i>Description</i>	Create a new Actuator				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
Actuator	New Actuator record, excluded ID	Body	true	Actuator	JSON
Response					

Code	Description	Body response
201	Actuator record, included new ID	Actuator

PUT		{baseUrl}/api/actuator				
<i>Description</i>		Update Actuator				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	
Actuator	Actuator record	Body	true	Actuator	JSON	

Response		
Code	Description	Body response
202	Update Actuator record,	Actuator

DELETE		{baseUrl}/api/actuator/{id}				
<i>Description</i>		Delete Actuator by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	
id	Actuator ID	path	true	integer	int32	

Response		
Code	Description	Body response
200	Ok	Empty

Actuator Link

GET		{baseUrl}/api/actuatorlink/{id}				
<i>Description</i>		Get Actuator link by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	
id	Actuator link ID	Path	true	integer	int32	

Response		
Code	Description	Body response
200	Actuator link record	ActuatorLink

GET		{baseUrl}/api/actuatorlink/all				
<i>Description</i>		Get all Actuator links				

Response		
Code	Description	Body response
200	All Actuator links	List of ActuatorLink

POST	{baseUrl}/api/actuatorlink				
<i>Description</i>	Create a new Actuator link				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
Actuator link	New Actuator link recor excluded ID	Body	true	ActuatorLink	JSON

Response		
Code	Description	Body response
201	Actuator Actuator link, included new ID	ActuatorLink

PUT	{baseUrl}/api/actuatorlink				
<i>Description</i>	Update Actuator link				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
Actuator link	Actuator link record	Body	true	ActuatorLink	JSON

Response		
Code	Description	Body response
202	Update Actuator link record,	ActuatorLink

DELETE	{baseUrl}/api/actuatorlink/{id}				
<i>Description</i>	Delete Actuator link by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Actuator link ID	path	true	integer	int32

Response		
Code	Description	Body response
200	Ok	Empty

Actuators Valid types

GET	{baseUrl}/api/actuator/validtypes				
------------	--	--	--	--	--

<i>Description</i>	Get all valid types of Actuators	
Response		
<i>Code</i>	<i>Description</i>	<i>Body response</i>
200	All valid types Actuators	List of String

Audit

GET	{baseUrl}/api/audit	
<i>Description</i>	Get Audit records by pages - Only for admin users	
<i>Filter</i>	Filter keys (Audits)	
<i>Include</i>	Paging and Filtering parameters (optionals)	

Response		
<i>Code</i>	<i>Description</i>	<i>Body response</i>
200	Filtered and paged Audit records	PageAudit

GET	{baseUrl}/api/audit/{id}				
<i>Description</i>	Get Audit records by ID - Only for admin users				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Audit ID	path	true	integer	int32

Response		
<i>Code</i>	<i>Description</i>	<i>Body response</i>
200	Audit Record	Audit

GET	{baseUrl}/api/audit/all	
<i>Description</i>	Get all Audit records. Only for admin users Be careful! Could be thousand of registers	
<i>Include</i>	API Key parameter (required)	

Response		
Important!: Audit could have thousand of records		
<i>Code</i>	<i>Description</i>	<i>Body response</i>
200	All audit records.	List of Audit .

Authorization

GET		{baseUrl}/api/authorization			
<i>Description</i>		Get Authorizations by pages.			
<i>Include</i>		Paging parameters (optional)			
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Paged Authorization records	PageAuthorization			
GET		{baseUrl}/api/authorization/{id}			
<i>Description</i>		Get authorization by ID			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Authorization ID	Path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Authorization record	Authorization			
GET		{baseUrl}/api/authorization/all			
<i>Description</i>		Get all Authorizations			
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	All Authorizations	List of Authorization.			
POST		{baseUrl}/api/authorization			
<i>Description</i>		Create a new Authorization			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
authorization	New authorization record, excluded ID	Body	true	Authorization	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
201	Created authorization record, Included new ID	Authorization			

PUT	{baseUrl}/api/authorization				
<i>Description</i>	Update Authorization				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
authorization	Authorization record	Body	true	Authorization	JSON

Response					
Code	Description	Body response			
201	Updated authorization record	Authorization			
DELETE		{baseUrl}/api/authorization/{id}			
<i>Description</i>		Delete Authorization by ID			
<i>Parameter</i>		<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>
id	authorization ID	path	true	integer	int32
Response					
Code	Description	Body response			
200	Ok	Empty			

Cameras

GET		{baseUrl}/api/camera			
<i>Description</i>		Get Camera by pages.			
<i>Include</i>		Paging parameters (optional)			
Response					
Code	Description	Body response			
200	Paged Camera records	PageCamera			
GET		{baseUrl}/api/camera/{id}			
<i>Description</i>		Get camera by ID			
<i>Parameter</i>		<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>
id	Camera ID	Path	true	integer	int32
Response					
Code	Description	Body response			
200	Camera record	Camera			

GET	{baseUrl}/api/camera/all					
<i>Description</i>	Get all Cameras					
Response						
<i>Code</i>	<i>Description</i>	<i>Body response</i>				
200	All Cameras	List of Camera .				
POST	{baseUrl}/api/camera					
<i>Description</i>	Create a new Camera					
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	
camera	New camera record, excluded ID	Body	true	Camera	JSON	
Response						
<i>Code</i>	<i>Description</i>	<i>Body response</i>				
201	Camera record, included new ID	Camera				
PUT	{baseUrl}/api/camera					
<i>Description</i>	Update Camera					
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	
camera	Camera record	Body	true	Camera	JSON	
Response						
<i>Code</i>	<i>Description</i>	<i>Body response</i>				
202	Updated Camera record	Camera				
DELETE	{baseUrl}/api/camera/{id}					
<i>Description</i>	Delete camera by ID					
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	
id	Camera ID	path	true	integer	int32	
Response						
<i>Code</i>	<i>Description</i>	<i>Body response</i>				
200	Ok	Empty				

Aforo (Capacity)

GET		{baseUrl}/api/aforo/{id}			
<i>Description</i>		Get capacity (aforo) by ID			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Aforo ID	Path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Aforo record	Aforo			
GET		{baseUrl}/api/aforo/all			
<i>Description</i>		Get all Capacities (Aforos)			
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	All Aforos	List of Aforo			
POST		{baseUrl}/api/aforo			
<i>Description</i>		Create a new Capacity (Aforo)			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
aforo	New Aforo record, excluded ID	Body	true	Aforo	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
201	Aforo record, included new ID	Aforo			
PUT		{baseUrl}/api/aforo			
<i>Description</i>		Update Capacity (Aforo)			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
aforo	Aforo record	Body	true	Aforo	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			

202	Updated Aforo record		Aforo		
DELETE	{baseUrl}/api/aforo/{id}				
<i>Description</i>	Delete Capacity (Aforo) by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Aforo ID	path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Ok	Empty			

Categories

GET	{baseUrl}/api/category				
<i>Description</i>	Get Categories by pages.				
<i>Include</i>	Paging parameters (optional)				
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Paged Category records	PageCategory			
GET	{baseUrl}/api/category/{id}				
<i>Description</i>	Get Category by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Category ID	Path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Category record	Category			
GET	{baseUrl}/api/category/all				
<i>Description</i>	Get all Categories				
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			

200	All Categories	List of Category
-----	----------------	----------------------------------

POST	{baseUrl}/api/category				
<i>Description</i>	Create a new Category				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
aforo	New Category record, excluded ID	Body	true	Category	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
201	Category record, included new ID	Category			
PUT	{baseUrl}/api/category				
<i>Description</i>	Update Category				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
aforo	Category record	Body	true	Category	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
202	Updated Category record	Category			
DELETE	{baseUrl}/api/category/{id}				
<i>Description</i>	Delete Category by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Category ID	path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Ok	Empty			

Configurations

Alert Config

GET		{baseUrl}/api/config/alert			
<i>Description</i>		Get Alert config			
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Alert config record	AlertConfig			
PUT		{baseUrl}/api/config/alert			
<i>Description</i>		Update alert configuration			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
plate	Alert config record	Body	true	AlertConfig	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
202	Updated Alert config record	AlertConfig			

LDAP Config

GET		{baseUrl}/api/config/ldap			
<i>Description</i>		Get LDAP config			
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	LDAP config record	LDAPConfig			
PUT		{baseUrl}/api/config/ldap			
<i>Description</i>		Update LDAP configuration			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
plate	Alert config record	Body	true	LDAPConfig	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			

202	Updated LDAP config record	LDAPConfig
-----	----------------------------	----------------------------

Plate Manager Config

GET	{baseUrl}/api/config/plateManager	
<i>Description</i>	Get Plate manager config	

Response		
Code	Description	Body response
200	Plate manager config record	PlateMngConfig

PUT	{baseUrl}/api/config/plateManager				
<i>Description</i>	Update Plate manager configuration				
Parameter	Description	Where	Required	Type	Format
plateMngConfig	Plate Mngr config record	Body	true	PlateMngConfig	JSON

Response		
Code	Description	Body response
202	Plate manager config record	PlateMngConfig

Server Config

GET	{baseUrl}/api/config/server	
<i>Description</i>	Get Server config	

Response		
Code	Description	Body response
200	Server config record	ServerConfig

PUT	{baseUrl}/api/config/server				
<i>Description</i>	Update Server configuration				
Parameter	Description	Where	Required	Type	Format
serverConfig	Server config record	Body	true	ServerConfig	JSON

Response		
Code	Description	Body response

202	Updates Server Config record	ServerConfig
-----	------------------------------	------------------------------

SMTP Config

GET		{baseUrl}/api/config/SMTP			
<i>Description</i>		Get SMTP config			
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	SMTP config record	SMTPConfig			
PUT		{baseUrl}/api/config/SMTP			
<i>Description</i>		Update SMTP configuration			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
SMTPConfig	SMTP config record	Body	true	SMTPConfig	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
202	Updated SMTP config record	SMTPConfig			

SMTP Test

GET		{baseUrl}/api/config/smtptest			
<i>Description</i>		Get SMTP config			
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	SMTP config record	SMTPTest			
PUT		{baseUrl}/api/config/smtptest			
<i>Description</i>		Update SMTP configuration			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
SMTPConfig	SMTP config record	Body	true	SMTPTest	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			

202	Updated SMTP config record	SMTPTest
------------	----------------------------	--------------------------

Telegram config

GET	{baseUrl}/api/config/telegram				
<i>Description</i>	Get Telegram config				

Response

<i>Code</i>	<i>Description</i>	<i>Body response</i>
200	Telegram config record	Telegram Config

PUT	{baseUrl}/api/config/telegram				
<i>Description</i>	Update Telegram configuration				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
TelegramConfig	Telegram config record	Body	true	Telegram Config	JSON

Response

<i>Code</i>	<i>Description</i>	<i>Body response</i>
202	Updated Telegram config record	Telegram Config

External

GET	{baseUrl}/api/external				
<i>Description</i>	Get External data by pages.				
<i>Include</i>	Paging parameters (optional)				

Response

<i>Code</i>	<i>Description</i>	<i>Body response</i>
200	Paged External data records	See Page External

GET	{baseUrl}/api/external/{id}				
<i>Description</i>	Get External data by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	External data ID	Path	true	integer	int32

Response						
Code	Description	Body response				
200	External data record	See External				
GET	{baseUrl}/api/external/all					
Description	Get all External data					
Response						
Code	Description	Body response				
200	All External data	List of External				
POST	{baseUrl}/api/category					
Description	Create a new External data					
Parameter	Description	Where	Required	Type	Format	
aforo	New External data record, excluded ID	Body	true	External	JSON	
Response						
Code	Description	Body response				
201	Category record, included new ID	External				
PUT	{baseUrl}/api/category					
Description	Update External data					
Parameter	Description	Where	Required	Type	Format	
aforo	External data record	Body	true	External	JSON	
Response						
Code	Description	Body response				
202	Updated External data record	External				
DELETE	{baseUrl}/api/category/{id}					
Description	Delete External data by ID					
Parameter	Description	Where	Required	Type	Format	
id	External data ID	path	true	integer	int32	

Response		
Code	Description	Body response
200	Ok	Empty

Grafana

GET	{baseUrl}/api/grafana/{id}				
Description	Get grafana configuration by User. With ID equal to the user ID				
Parameter	Description	Where	Required	Type	Format
id	user ID	Path	true	integer	int32

Response		
Code	Description	Body response
200	Image record	Grafana

POST	{baseUrl}/api/grafana				
Description	Create a new Grafana configuration				
Parameter	Description	Where	Required	Type	Format
grafana	New grafana record, excluded ID	Body	true	Grafana	JSON

Response		
Code	Description	Body response
201	Grafana record, included new ID	Grafana

PUT	{baseUrl}/api/grafana				
Description	Update Grafana configuration				
Parameter	Description	Where	Required	Type	Format
grafana	Grafana record	Body	true	Grafana	JSON

Response		
Code	Description	Body response
202	Updated Grafana record	Grafana

Heartbeat

NOTE: These methods are included **for information only**, since they are for the **exclusive use of VaxALPR applications** embedded in cameras.

POST	{baseUrl}/api/heartbeat/				
<i>Description</i>	New Camera Heartbeat (update camera info)				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
heartbeat	Camera Heartbeat record, including all heartbeat fields, also Request.UserHostAddress	Body	true	Heartbeat	JSON

Response

<i>Code</i>	<i>Description</i>	<i>Body response</i>
200	Heartbeat response	Heartbeat Response

POST	{baseUrl}/api/heartbeat/				
<i>Description</i>	New Camera Heartbeat (update camera info)				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
heartbeat	Camera Heartbeat record, including the fields Config and ConfigSHA	Body	true	Heartbeat	JSON

Response

<i>Code</i>	<i>Description</i>	<i>Body response</i>
200	Response	

POST	{baseUrl}/api/heartbeat/				
<i>Description</i>	New Camera Heartbeat (update camera info)				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
heartbeat	Camera Heartbeat record, including the field Frame	Body	true	Heartbeat	JSON

Response

<i>Code</i>	<i>Description</i>	<i>Body response</i>
200	Response	

GDRP

DELETE	{baseUrl}/api/gdpr/{plateNumber}				
<i>Description</i>	Delete all Plate register with this plateNumber				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	plateNumber	path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Ok	Empty			

Images

GET	{baseUrl}/api/image				
<i>Description</i>	Get Images by pages.				
<i>Include</i>	Paging parameters (optional)				
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Paged Image records	PageImage			
GET	{baseUrl}/api/image/{id}				
<i>Description</i>	Get image by ID. With ID equal to: <ul style="list-style-type: none"> • Plate - ocrlmageId or • Plate - envImageIds [i] 				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	image ID	Path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Image record	Image			
POST	{baseUrl}/api/image				
<i>Description</i>	Create a new Image				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>

image	New image record, excluded ID	Body	true	Image	JSON
Response					
Code	Description	Body response			
201	Image record, included new ID	Image			
PUT	{baseUrl}/api/image				
Description	Update Image				
Parameter	Description	Where	Required	Type	Format
image	Image record	Body	true	Image	JSON
Response					
Code	Description	Body response			
202	Updated Image record	Image			

Linked Cameras

GET	{baseUrl}/api/linkedCamera				
Description	Get People by pages.				
Include	Paging parameters (optional)				
Response					
Code	Description	Body response			
200	Paged Linked Camera records	PagedLinkedCameras			
GET	{baseUrl}/api/linkedCamera/{id}				
Description	Get Linked Camera by ID				
Parameter	Description	Where	Required	Type	Format
id	Linked Camera ID	Path	true	integer	int32
Response					
Code	Description	Body response			
200	Linked Camera record	LinkedCamera			

GET	{baseUrl}/api/linkedCamera/all
<i>Description</i>	Get all Linked Cameras

Response					
Code	Description	Body response			
200	All Linked Cameras	List of LinkedCamera			
POST	{baseUrl}/api/linkedCamera				
<i>Description</i>	Create a new Linked Camera				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
aforo	New Linked Camera record, excluded ID	Body	true	LinkedCamera	JSON
Response					
Code	Description	Body response			
201	Linked Camera record, included new ID	LinkedCamera			
PUT	{baseUrl}/api/linkedCamera				
<i>Description</i>	Update Linked Camera				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
aforo	Linked Camera record	Body	true	LinkedCamera	JSON
Response					
Code	Description	Body response			
202	Updated Linked Camera record	LinkedCamera			
DELETE	{baseUrl}/api/linkedCamera/{id}				
<i>Description</i>	Delete Linked Camera by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Linked Camera ID	path	true	integer	int32
Response					
Code	Description	Body response			
200	Ok	Empty			

People

GET	{baseUrl}/api/person				
<i>Description</i>	Get People by pages.				
<i>Include</i>	Paging parameters (optional)				
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Paged Person records	PagePerson			
GET	{baseUrl}/api/person/{id}				
<i>Description</i>	Get Person by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Person ID	Path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Person record	Person			
GET	{baseUrl}/api/person/all				
<i>Description</i>	Get all People				
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	All People	List of Person			
POST	{baseUrl}/api/person				
<i>Description</i>	Create a new Person				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
aforo	New Person record, excluded ID	Body	true	Person	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
201	Person record, included new ID	Person			

PUT		{baseUrl}/api/person			
<i>Description</i>		Update Person			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
aforo	Person record	Body	true	Person	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
202	Updated Person record	Person			
DELETE		{baseUrl}/api/person/{id}			
<i>Description</i>		Delete Person by ID			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Person ID	path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Ok	Empty			

Plates

Plates

GET		{baseUrl}/api/plate			
<i>Description</i>		Get Plates by pages.			
<i>Filter</i>		Filter keys (Plates)			
<i>Include</i>		Paging and Filtering parameters (optionals)			
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Filtered and paged Plate records	PagePlate			
GET		{baseUrl}/api/plate/{id}			
<i>Description</i>		Get plate by ID			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>

id	plate ID	path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Plate record	Plate			

PUT	{baseUrl}/api/plate				
<i>Description</i>	Update plate				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
plate	Plate record	Body	true	Plate	JSON

Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
202	Updated Plate record	Plate			

POST	{baseUrl}/api/plate				
<i>Description</i>	Create a new plate				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
plate	Plate record	Body	true	Plate	JSON

<i>Notes</i>					
1.	To create just the plate record, does not include				

Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
202	Updated Plate record	Plate			

Associates

GET	{baseUrl}/api/plate/associates				
<i>Description</i>	Get associated Plates lists - Under license				
<i>Filter</i>	Filter keys (Plates)				
<i>Include</i>	Paging and Filtering parameters				

Response		
Code	Description	<i>Body response</i>
200	List of associates Plate	List of Plate

Plate lists

GET		{baseUrl}/api/platelist			
<i>Description</i>		Get Platelist by pages.			
<i>Include</i>		Paging parameters (optional)			
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Paged Platelist records	PagePlatelist			
GET		{baseUrl}/api/platelist/{id}			
<i>Description</i>		Get Platelist by ID			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Platelist ID	Path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Platelist record	Platelist			
GET		{baseUrl}/api/platelist/all			
<i>Description</i>		Get all Platelists			
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	All recorded Platelists	List of Platelist			
POST		{baseUrl}/api/platelist			
<i>Description</i>		Create a new Platelist			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
platelist	New Platelist record, excluded ID	Body	true	Platelist	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
201	Platelist record, included new ID	Platelist			

PUT		{baseUrl}/api/platelist				
<i>Description</i>		Update Platelist				
<i>Parameter</i>		<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
platelist		Platelist record	Body	true	Platelist	JSON
Response						
<i>Code</i>		<i>Description</i>	<i>Body response</i>			
202		Updated platelist record	Platelist			
DELETE		{baseUrl}/api/platelist/{id}				
<i>Description</i>		Delete Platelist by ID				
<i>Parameter</i>		<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id		Platelist ID	path	true	integer	int32
Response						
<i>Code</i>		<i>Description</i>	<i>Body response</i>			
200		Ok	Empty			

Plate list items

GET		{baseUrl}/api/listitem				
<i>Description</i>		Get Listitems by pages.				
<i>Filter</i>		List Items filter keys				
<i>Include</i>		Paging and Filtering parameters (optionals)				
Response						
<i>Code</i>		<i>Description</i>	<i>Body response</i>			
200		Filtered and paged Listitem records	PageListitem			
GET		{baseUrl}/api/listitem/{id}				
<i>Description</i>		Get Listitem by ID				
<i>Parameter</i>		<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id		Listitem ID	Path	true	integer	int32

Response						
Code	Description	Body response				
200	Listitem record	Listitem				
GET		{baseUrl}/api/listitem/all				
<i>Description</i>		Get all Listitems				
Response						
Code	Description	Body response				
200	All Listitems	List of Listitem				
POST		{baseUrl}/api/listitem				
<i>Description</i>		Create a new Listitem				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	
listitem	New listitem record, excluded ID	Body	true	Listitem	JSON	
Response						
Code	Description	Body response				
201	Listitem record, included new ID	Listitem				
PUT		{baseUrl}/api/listitem				
<i>Description</i>		Update Listitem				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	
listitem	Listitem record	Body	true	Listitem	JSON	
Response						
Code	Description	Body response				
202	Updated Listitem record	Listitem				
DELETE		{baseUrl}/api/listitem/{id}				
<i>Description</i>		Delete Listitem by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	
id	Listitem ID	path	true	integer	int32	

Response		
Code	Description	Body response
200	Ok	Empty

Regular expressions (Schedule)

GET	{baseURL}/api/authExpression
Description	Get the Regular expressions by pages.
Include	Paging parameters (optional)

Response		
Code	Description	Body response
200	Paged the Regular expression records	PageAuthExpression

GET	{baseURL}/api/authExpression/{id}				
Description	Get Timeframe by ID				
Parameter	Description	Where	Required	Type	Format
id	authEpression ID	Path	true	integer	int32

Response		
Code	Description	Body response
200	the Regular expressions record	AuthExpression

GET	{baseURL}/api/authExpression/all
Description	Get all Regular expressions

Response		
Code	Description	Body response
200	All Regular expressions	List of AuthExpression

POST	{baseURL}/api/authExpression				
Description	Create a new Regular expression				
Parameter	Description	Where	Required	Type	Format
AuthExpres	New Regular expressions record,	Body	true	AuthExpress	JSON

sion	excluded ID			ion	
Response					
Code	Description	Body response			
201	the Regular expressions record, included new ID	AuthExpression			

PUT	{baseUrl}/api/authExpression				
Description	Update Timeframe				
Parameter	Description	Where	Required	Type	Format
AuthExpression	the Regular expressions record	Body	true	AuthExpressi on	JSON

Response					
Code	Description	Body response			
202	Updated Timeframe record	Timeframe			

DELETE	{baseUrl}/api/authExpression/{id}				
Description	Delete a Regular expression by ID				
Parameter	Description	Where	Required	Type	Format
id	the Regular expressions ID	path	true	integer	int32

Response					
Code	Description	Body response			
200	Ok	Empty			

Rollcall

GET	{baseUrl}/api/rollcall				
Description	Get Rollcalls by pages.				
Include	Paging parameters (optional)				

Response					
Code	Description	Body response			

200	Paged Rollcal records	PageRollcall			
GET	{baseUrl}/api/rollcall/all				
<i>Description</i>	Get all Rollcal				
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	All Rollcal	List of Rollcall			
DELETE	{baseUrl}/api/rollcall/{id}				
<i>Description</i>	Delete Person by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Rollcal ID	path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Ok	Empty			

Schedules

GET	{baseUrl}/api/schedule				
<i>Description</i>	Get Schedules by pages.				
<i>Include</i>	Paging parameters (optional)				
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Paged Schedule records	PageSchedule			
GET	{baseUrl}/api/schedule/{id}				
<i>Description</i>	Get Schedule by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	schedule ID	Path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			

200	Schedule record	Schedule			
GET	{baseUrl}/api/schedule/all				
<i>Description</i>	Get all Schedules				
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	All Schedules	List of Schedule			
POST	{baseUrl}/api/schedule				
<i>Description</i>	Create a new Schedule				
<i>Include</i>	API Key parameter (required)				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
schedule	New Schedule record, excluded ID	Body	true	Schedule	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
201	Schedule record, included new ID	Schedule			
PUT	{baseUrl}/api/schedule				
<i>Description</i>	Update Schedule				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
schedule	Schedule record	Body	true	Schedule	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
202	Updated Schedule record	Schedule			
DELETE	{baseUrl}/api/schedule/{id}				
<i>Description</i>	Delete schedule by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Schedule ID	path	true	integer	int32
Response					

Code	Description	Body response
200	Ok	Empty

Sections

GET	{baseUrl}/api/section
Description	Get Sections by pages.
Include	Paging parameters (optional)

Response

Code	Description	Body response
200	Paged Section records	PageSection

GET	{baseUrl}/api/section/{id}				
Description	Get Section by ID				
Parameter	Description	Where	Required	Type	Format
id	Section ID	Path	true	integer	int32

Response

Code	Description	Body response
200	Section record	Section

GET	{baseUrl}/api/section/all
Description	Get all Sections

Response						
Code	Description	Body response				
200	All Sections	List of Section				
POST	{baseUrl}/api/section					
<i>Description</i>	Create a new Section					
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	
section	New section record, excluded ID	Body	true	Section	JSON	
Response						
Code	Description	Body response				
201	Section record, included new ID	Section				
PUT	{baseUrl}/api/section					
<i>Description</i>	Update Section					
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	
section	Section record	Body	true	Section	JSON	
Response						
Code	Description	Body response				
202	Updated Section record	Section				
DELETE	{baseUrl}/api/section/{id}					
<i>Description</i>	Delete Section by ID					
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	
id	Section ID	path	true	integer	int32	
Response						
Code	Description	Body response				
200	Ok	Empty				

Time frames (Schedule)

GET	{baseUrl}/api/timeframe				
<i>Description</i>	Get Timeframe by pages.				
<i>Include</i>	Paging parameters (optional)				
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Paged Timeframe records	PageTimeframe			
GET	{baseUrl}/api/timeframe/{id}				
<i>Description</i>	Get Timeframe by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Timeframe ID	Path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Timeframe record	Timeframe			
GET	{baseUrl}/api/timeframe/all				
<i>Description</i>	Get all Timeframes				
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	All Timeframes	List of Timeframe			
POST	{baseUrl}/api/timeframe				
<i>Description</i>	Create a new Timeframe				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
timeframe	New Timeframe record, excluded ID	Body	true	Timeframe	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
201	Timeframe record, included new ID	Timeframe			

PUT		{baseUrl}/api/timeframe				
<i>Description</i>		Update Timeframe				
<i>Parameter</i>		<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
timeframe		Timeframe record	Body	true	Timeframe	JSON
Response						
<i>Code</i>	<i>Description</i>		<i>Body response</i>			
202	Updated Timeframe record		Timeframe			
DELETE		{baseUrl}/api/timeframe/{id}				
<i>Description</i>		Delete Timeframe by ID				
<i>Parameter</i>		<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id		Timeframe ID	path	true	integer	int32
Response						
<i>Code</i>	<i>Description</i>		<i>Body response</i>			
200	Ok		Empty			

Users

GET		{baseUrl}/api/user				
<i>Description</i>		Get Users by pages.				
<i>Include</i>		Paging parameters (optional)				
Response						
<i>Code</i>	<i>Description</i>		<i>Body response</i>			
200	Paged User records		PageUser			
GET		{baseUrl}/api/user/{id}				
<i>Description</i>		Get user by ID				
<i>Parameter</i>		<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id		User ID	Path	true	integer	int32

Response						
Code	Description	Body response				
200	User record	User				
GET	{baseUrl}/api/user/all					
Description	Get all Users					
Response						
Code	Description	Body response				
200	All Users	List of User				
POST	{baseUrl}/api/user					
Description	Create a new User					
Parameter	Description	Where	Required	Type	Format	
user	New user record, excluded ID	Body	true	User	JSON	
Response						
Code	Description	Body response				
201	User record, included new ID	User				
PUT	{baseUrl}/api/user					
Description	Update User					
Parameter	Description	Where	Required	Type	Format	
user	User record	Body	true	User	JSON	
Response						
Code	Description	Body response				
202	Updated user record	User				
DELETE	{baseUrl}/api/user/{id}					
Description	Delete user by ID					
Parameter	Description	Where	Required	Type	Format	
id	User ID	path	true	integer	int32	

Response		
Code	Description	Body response
200	Ok	Empty

Zones

GET	{baseUrl}/api/zone
Description	Get Zones by pages.
Include	Paging parameters (optional)

Response		
Code	Description	Body response
200	Paged Zone records	PageZone

GET	{baseUrl}/api/zone/{id}				
Description	Get Zone by ID				
Parameter	Description	Where	Required	Type	Format
id	Zone ID	Path	true	integer	int32

Response		
Code	Description	Body response
200	Zone record	Zone

GET	{baseUrl}/api/zone/all
Description	Get all Zones

Response		
Code	Description	Body response
200	All Zones	List of Zone

POST	{baseUrl}/api/zone				
Description	Create a new Zone				
Parameter	Description	Where	Required	Type	Format
zone	New Zone record, excluded ID	Body	true	Zone	JSON

Response						
Code	Description	Body response				
201	Zone record, included new ID	Zone				
PUT		{baseUrl}/api/zone				
<i>Description</i>		Update Zone				
<i>Parameter</i>		<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
zone		Zone record	Body	true	Zone	JSON
Response						
Code	Description	Body response				
202	Updated zone record	Zone				
DELETE		{baseUrl}/api/zone/{id}				
<i>Description</i>		Delete Zone by ID				
<i>Parameter</i>		<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id		Zone ID	path	true	integer	int32
Response						
Code	Description	Body response				
200	Ok	Empty				

Plugins API

Sanctions

Sanction config

GET	{baseUrl}/api/config/sanction				
<i>Description</i>	Get Sanctions configuration				
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Sanction config record	SanctionConfig			
PUT	{baseUrl}/api/config/sanction				
<i>Description</i>	Update Sanctions configuration				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
plate	Sanction config record	Body	true	SanctionConfig	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
202	Sanction config record	SanctionConfig			

Cancellations

GET	{baseUrl}/api/cancellation				
<i>Description</i>	Get cancellation by pages				
<i>Include</i>	Paging parameters (optional)				
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Paged Cancellation records	PageCancellation			
GET	{baseUrl}/api/cancellation/{id}				
<i>Description</i>	Get Cancellation by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>

id	Cancellation ID	Path	true	integer	int32
-----------	-----------------	------	------	---------	-------

Response

Code	Description	Body response
200	Cancellation record	Cancellation

GET	{baseUrl}/api/cancellation/all				
<i>Description</i>	Get all Cancellations				

Response

Code	Description	Body response
200	All Cancellations	List of Cancellation .

POST	{baseUrl}/api/cancellation				
<i>Description</i>	Create a new Cancellation				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
Cancellation	New Cancellation record, excluded ID	Body	true	Cancellation	JSON

Response

Code	Description	Body response
201	Cancellation record, included new ID	Cancellation .

PUT	{baseUrl}/api/cancellation				
<i>Description</i>	Update Cancellation				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
Cancellation	Cancellation record	Body	true	Cancellation	JSON

Response

Code	Description	Body response
202	Update Cancellation record,	Cancellation

DELETE	{baseUrl}/api/cancellation/{id}				
<i>Description</i>	Delete Cancellation by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>

id	Cancellation ID	path	true	integer	int32
-----------	-----------------	------	------	---------	-------

Response		
Code	Description	Body response
200	Ok	Empty

Sanctions

GET	{baseUrl}/api/sanction
Description	Get Sanctions by pages
Include	Paging parameters (optional)

Response		
Code	Description	Body response
200	Paged Sanction records	PageSanction

GET	{baseUrl}/api/sanction/{id}				
Description	Get sanction by ID				
Parameter	Description	Where	Required	Type	Format
id	Sanction ID	Path	true	integer	int32

Response		
Code	Description	Body response
200	Sanction record	Sanction

GET	{baseUrl}/api/sanction/all
Description	Get all Sanctions

Response		
Code	Description	Body response
200	All Sanctions	List of Sanction

POST	{baseUrl}/api/sanction				
Description	Create a new Sanction				
Parameter	Description	Where	Required	Type	Format
sanction	New Sanction record, excluded ID	Body	true	Sanction	JSON

Response		
Code	Description	Body response
201	Sanction record, included new ID	Sanction

PUT	{baseUrl}/api/sanction				
<i>Description</i>	Update Sanction				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
sanction	Sanction record	Body	true	Sanction	JSON

Response		
Code	Description	Body response
202	Update Sanction record,	Sanction

DELETE	{baseUrl}/api/sanction/{id}				
<i>Description</i>	Delete Sanction by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Sanction ID	path	true	integer	int32

Response		
Code	Description	Body response
200	Ok	Empty

Camera Sanctions

GET	{baseUrl}/api/camerasanction/{id}				
<i>Description</i>	Get Camera-sanctions by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Camera-sanctions ID	Path	true	integer	int32

Response		
Code	Description	Body response
200	Camera-sanctions record	CameraSanctions

GET	{baseUrl}/api/camerasanction/all				
------------	---	--	--	--	--

<i>Description</i>	Get all Camera-sanctions				
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	All Camera-sanctions records	List of CameraSanctions			
POST	{baseUrl}/api/camerasanction				
<i>Description</i>	Create a new Camera-sanctions				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
sanction	New Camera-sanctions record, excluded ID	Body	true	CameraSanctions	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
201	Cameraanction record, included new ID	CameraSanctions			
PUT	{baseUrl}/api/camerasanction				
<i>Description</i>	Update Camera-sanction				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
sanction	Updated Camera-sanctions record,	Body	true	CameraSanctions	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
202	Update Camera-sanction record,	CameraSanctions			
DELETE	{baseUrl}/api/camerasanction/{id}				
<i>Description</i>	Delete Camera-sanctions by ID				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Camera-sanctions ID	path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Ok	Empty			

Plate Sanctions

GET		{baseUrl}/api/platesanction			
<i>Description</i>		Get Plate-sanctions by pages.			
<i>Include</i>		Paging parameters (optional)			
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Paged Plate-sanctions records	PagePlateSanctions			
GET		{baseUrl}/api/platesanction/{id}			
<i>Description</i>		Get Plate-sanctions by ID			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Plate-sanctions ID	Path	true	integer	int32
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
200	Plate-sanctions record	PlateSanctions			
POST		{baseUrl}/api/platesanction			
<i>Description</i>		Create a new Plate-sanction			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
plateSanction	New Plate-sanction record, excluded ID	Body	true	PlateSanctions	JSON
Response					
<i>Code</i>	<i>Description</i>	<i>Body response</i>			
201	Plate-sanction record, included new ID	PlateSanctions			
PUT		{baseUrl}/api/platesanction			
<i>Description</i>		Update Camera-sanction			
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
sanction	Updated PLate-sanctions record,	Body	true	PlateSanctions	JSON

Response		
Code	Description	Body response
202	Updated Plate-sanction record,	PlateSanctions

File downloads¹

Audit record

GET	{baseURL}/api/audit/export.csv
<i>Description</i>	Get audit records - Only for admin users
<i>Filter</i>	Filter keys (Audits)
<i>Include</i>	Download CSV - Filtering and Columns See columns names in Audit
Response	
Important!: Audit could have thousand of records	
<i>Code</i>	<i>Description</i>
200	CSV file with header including selected columns of filtered Audit records.

Authorization

GET	{baseURL}/api/authorization/export.csv
<i>Description</i>	Get all or filtered items from Authorization in CSV format
<i>Filter</i>	Authorizations filter keys
<i>Include</i>	Download CSV - Filtering and Columns. See columns names in Authorization
Response	
<i>Code</i>	<i>Description</i>
200	CSV file with header including selected columns of filtered Authorization records.

¹ Must include [API Key parameter](#).

Category

GET	{baseUrl}/api/category/export.csv	
<i>Description</i>	Get all or filtered items from Category in CSV format	
<i>Filter</i>	Filter keys (Category)	
<i>Include</i>	Download CSV - Filtering and Columns See columns names in Category	
Important!: Audit could have thousand of records		
<i>Code</i>	<i>Description</i>	
200	CSV file with header including selected columns of filtered Category records.	

GDPR

GET	{baseUrl}/api/gdpr/{plateNumber}				
<i>Description</i>	Get all items from Plates with Plate Number = plateNumber				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	plateNumber	path	true	integer	int32
<i>Code</i>	<i>Description</i>				
200	PDF file including selected records of Plate table.				

OCR plate images

GET	{baseUrl}/PlateImages/{id}.jpg	
<i>Description</i>	Get JPG image clipping with plate., ID = Plate - ocrImageld.	
Response		
<i>Code</i>	<i>Description</i>	<i>Body response</i>
200	JPG image file	binary JPG

OCR and environment images

GET	{baseUrl}/Images/{id}.jpg
------------	----------------------------------

<i>Description</i>		Get a JPG image clipping with a plate. With ID equal to: <ul style="list-style-type: none"> • Plate - <code>ocrImageId</code> or • Plate - <code>envImageIds[i]</code>
Response		
<i>Code</i>	<i>Description</i>	<i>Body response</i>
200	JPG image file	binary JPG

People

GET	{baseUrl}/api/person/export.csv	
<i>Description</i>	Get all or filtered items from Person in CSV format	
<i>Filter</i>	People filter keys	
<i>Include</i>	Download CSV - Filtering and Columns. See columns names in People	
Response		
<i>Code</i>	<i>Description</i>	
200	CSV file with header including selected columns of filtered People records.	

Plates

Registers

GET	{baseUrl}/api/plate/export.csv	
<i>Description</i>	Get all or filtered plates records in CSV format Important!: Plates could have thousand of records	
<i>Filter</i>	Filter keys (Plates)	
<i>Include</i>	Download CSV Filtering and Columns. See columns names in Plate	
Response		
<i>Code</i>	<i>Description</i>	
200	CSV file with header including selected columns of filtered Plate records.	

Images

GET	{baseUrl}/api/plate/images.zip	
<i>Description</i>	Get all or filtered images plates records in ZIP format	
<i>Filter</i>	Filter keys (Plates)	
<i>Include</i>	Paging and Filtering parameters (optionals)	
Response		
Important!: Plates could have thousand of records		
<i>Code</i>	<i>Description</i>	<i>Body response</i>
200	ZIP file, with Plate images records.	zip file named export.zip with images associated to export.csv

Plate list items

GET	{baseUrl}/api/listitem/export.csv	
<i>Description</i>	Get all or filtered items from plate list (black or white) in CSV format	
<i>Filter</i>	List Items filter keys	
<i>Include</i>	Download CSV - Filtering and Columns. See columns names in Listitem	
Response		
<i>Code</i>	<i>Description</i>	
200	CSV file with header including selected columns of filtered Listitem records.	

File upload

NOTES:

1. Must include API Key parameter (see [API Key parameters](#))
2. Must include Upload file parameter (see [Upload file parameter](#)).
3. It must use a semicolon (;) as a field separator in CSV files.

Authorization

POST		{baseUrl}/api/authupload				
<i>Description</i>		Upload items from Authorization in CSV format				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	
CSV file	CSV file with authorized plates	body	true	Authorization CSV	form-data	
Response						
<i>Code</i>	<i>Description</i>		<i>Body response</i>			
200	All new Authorization records.		List of new Authorizations			

Category

POST		{baseUrl}/api/categoryupload				
<i>Description</i>		Upload items from Authorization in CSV format				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	
CSV file	CSV file with categories	body	true	Category CSV	form-data	
Response						
<i>Code</i>	<i>Description</i>		<i>Body response</i>			
200	All new Categories records.		List of new Category			

Image upload

POST		{baseUrl}/api/imageupload/{id}				
<i>Description</i>		Upload image JPG format associated with Image ID (see Images)				
<i>Parameter</i>	<i>Value</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>	

id	Image ID	path	true	integer	int32
JPG image	Image file	body	true	JPG	form-data
Response					
<i>Code</i>	<i>Description</i>			<i>Body response</i>	
200	Image record associated with uploaded image			Image	

People

POST	{baseUrl}/api/personupload/{id}				
<i>Description</i>	Upload image JPG format associated with Image ID (see Images)				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
CSV file	CSV file with People	body	true	PersonCSV	CSV file
Response					
<i>Code</i>	<i>Description</i>			<i>Body response</i>	
200	All new Person records.			List of new People	

Person Image

POST	{baseUrl}/api/personimageupload/{id}				
<i>Description</i>	Upload image JPG format associated with person ID (see Person)				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Image ID	path	true	integer	int32
JPG image	Image file	body	true	JPG	binary
Response					
<i>Code</i>	<i>Description</i>			<i>Body response</i>	
200	Image record associated with uploaded image			Person	

Plate list items

POST	{baseUrl}/api/listupload/{id}				
<i>Description</i>	Upload items from plate list (black or white) in CSV format				
<i>Parameter</i>	<i>Description</i>	<i>Where</i>	<i>Required</i>	<i>Type</i>	<i>Format</i>
id	Plate list ID	path	true	integer	int32
CSV file	CSV file with Listitems	body	true	ListItemCSV	CSV file

Response		
<i>Code</i>	<i>Description</i>	<i>Body response</i>
200	All new Listitem records.	List of Listitem

SignalR messaging

Common

Common schema		
Field	Type	Example
name	aforo listitem plate alert rollcall	
body	JSON	See message bodies

Messages bodies

Aforo message		
Description	Message with name aforo	
Field	Type	Example
action	created updated deleted	
resource	Aforo	See Aforo

Listitem message		
Description	Message with name listitem (Blacklist event)	
Field	Type	Example
action	updated	“updated”
resource	Listitem	See Listitem

Plate message		
Description	Message with name plate	
Field	Type	Example
action	created updated	
resource	Plate	See Plate

Alert message		
<i>Description</i>	Message with name alert	
<i>Field</i>	<i>Type</i>	<i>Example</i>
plateId	Integer(\$int32)	1
itemId	Integer(\$int32)	1
type	Blacklist Speed Prowling Kamikaze Timer Adr NoAuthExceed CameraConnection CameraTimeSync CameraReset CameraQueue RoiCrossing ForbiddenTurn	
plateNumber	String	“0933BFF”
description	String	“0933BFF”
cameraId	Integer(\$int32)	1
Description field format		
<i>Type</i>	<i>Format</i>	
Blacklist	{ listItem.Description }	
Speed	{ plate.Speed }	
Prowling	{ Count of detection in Prowling zone }	
Kamikaze	{ plate.Direction }	
Timer	{ entrance_Plate.TakenOn }	
Adr	<i>Not include info</i>	
NoAuthExceed	{ person.Title } { person.Forename } { person.Surname } { person.Company } { person.IsVIP ? ", VIP" : "" }	
CameraConnection	{ camera.Name } { zone.Name }	
CameraTimeSync		
CameraReset		
CameraQueue		
RoiCrossing	<i>Not include info</i>	
ForbiddenTurn	{ source_plate.id }	

Rowcall message		
<i>Description</i>	Message with name rollcall	
<i>Field</i>	<i>Type</i>	<i>Example</i>
action	created deleted	
resource	Rowcall	See Rollcall

Data models

Common elements

Paging response fields		
<i>Field</i>	<i>Type</i>	<i>Example</i>
page	Integer(\$int32)	1
pageSize	Integer(\$int32)	20
sortKey	String	"id"
sortDirection	String	"ascending"
totalRecords	Integer(\$int32)	1268

Notes	
DateTime	DateTime fields are UTC time in ISO-8601 format encapsulated in String format: "YYYY-MM-DDTHH:MM:SS.SSSSSZ" with HH in 24h format.
Null attributes	When the Entity attribute is null in the database, MOBOTIX SYNC doesn't send this field in response.

Actuators

Actuator		
Field	Type	Example
id	Integer(\$int32)	
description	String	“Entrance control”
type	String	See Valid types .
iP	String	“192.168.0.3”
port	Integer(\$int32)	700
user	String	“admin”
password	String	“password”
channel	Integer(\$int32)	2
state	Integer(\$int32)	
period	Integer(\$int32)	

Actuator fields validators

Field	Validation
type	It must be a valid type (See Valid types)
description	Not blank
iP	Not blank

ValidTypes

Field	Type	Example
validTypes	List of <i>Valid Types</i>	List of valid types
Valid Types	Adam6050Controller, Adam6052Controller, Adam6060Controller, Adam6266Controller, AxisController, SkydataController, VaxtorPanelController, ...	

ActuatorLink

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	
camerald	Integer(\$int32)	
actuatorsId	List of Integer(\$int32)	
eventType	EventType	
externalAuth	Boolean	false

ActuatorLink fields validators

<i>Field</i>	<i>Validation</i>
camerald	Camera must exist
actuatorsId	Actuators must exist
EventType	authorized blacklist

Audit**PageAudit**

Include: [Paging response fields](#).

<i>Field</i>	<i>Type</i>	
records	Audit	See Audit

Audit

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	5
timestamp	DateTime as String	"2017-12-18T15:31:19.501154Z"
userId	Integer(\$int32)	1
sourceAddress	String	"152.72.151.145",
action	String	"GET /api/user/all"
details	String	"{\r\n \"page\": 1,\r\n \"pageSize\": 10,\r\n

Audit		
Field	Type	Example
		<code>\\"sortKey\\": \\"id\\",\r\n \\"sortDirection\\": \\"ascending\\",\r\n \\"totalRecords\\": 0\r\n}"</code>
failed	Boolean	false

Aforo

Aforo (Capacity)		
Field	Type	Example
id	Integer(\$int32)	5
name	String	“Parking capacity”
entranceCameraIds	List of Integer(\$int32)	[1]
exitCameraIds	List of Integer(\$int32)	[2, 3]
count	Integer(\$int32)	0
resetTime	Date Time as String	“2017-12-31T23:59:59.00Z”
maxCount	Integer(\$int32)	100
ignoreCarDirection	Boolean	false
maxMinutesInZone	Integer(\$int32)	30
timeInZoneAlarm	Boolean	false
crosstime	Boolean	false
maxSearchTime	Integer(\$int32)	86400
resetAforo	Boolean	
denyIfFull	Boolean	
ignoreIfNotAuthorized	Boolean	
categoryId	Integer(\$int32)	
isVirtual	Boolean	

Aforo fields validators

Field	Validation
name	Not blank
entranceCameralds	Not empty collection
exitCameralds	Not empty collection
resetTime	Not null
maxMinutesInZone	>= 1 && <= (24 * 60)
maxSearchTime	>= 1 && <= (24 * 60 * 60 * 30)

Authorization**PageAuthorization**

Include: [Paging response fields](#).

Field	Type	
records	Authorization	See Authorization

Authorization

Field	Type	Example
id	Integer(\$int32)	5
plate	String	“0933BFF”
activeFrom	DateTime as String	“2017-01-01T00:00:00.00Z”
activeUntil	DateTime as String	“2017-12-31T23:59:59.00Z”
zoneSchedules	List of ZoneSchedule	See ZoneSchedule
zonelds	List of Integer(\$int32)	[5]
description	String	“Visit”
comments	String	
personId	Integer(\$int32)	
lastSeen	DateTime as String	
lastSeenId	Integer(\$int32)	

Authorization

<i>Field</i>	<i>Type</i>	<i>Example</i>
tokens	Integer(\$int32)	
make	String	
model	String	
color	String	
isDisabled	Boolean	

Authorization CSV

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	5
plate (+)	String	“0933BFF”
activeFrom (+)	DateTime as String	“2017-01-01T00:00:00.00Z”
activeUntil (+)	DateTime as String	“2017-12-31T23:59:59.00Z”
zoneSchedules	List of ZoneSchedule	See ZoneSchedule
description	String	“Visit”
comments	String	“”
isVip	Boolean	
personId	Integer(\$int32)	
tokens	Integer(\$int32)	
make	String	
model	String	
color	String	
isDisabled	Boolean	

(+) required**ZoneSchedule**

<i>Field</i>	<i>Type</i>	<i>Example</i>
--------------	-------------	----------------

id	Integer(\$int32)	
zoneld	Integer(\$int32)	5
schedulesIds	List of Integer(\$int32)	[12, 15]

Authorization fields validators

<i>Field</i>	<i>Validation</i>
plate	Not blank
activeFrom	Not null
activeUntil	Null or activeFrom >= activeUntil
zoneSchedules	Not null
personId	Valid and existing person
make	Not null
model	Not null
color	Not null

Cameras**PageCamera**

Include: [Paging response fields](#).

<i>Field</i>	<i>Type</i>	
records	Camera	See Camera

Camera

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	7
name	String	“Test camera”
plateReaderId	Integer(\$int32)	2
type	CameraType	
address	String	“Po Castellana, 123 - Madrid”
zoneId	Integer(\$int32)	2
latitude	Double	12.1234
longitude	Double	12.1234

Camera		
<i>Field</i>	<i>Type</i>	<i>Example</i>
reportType	ReportType	
captureId	String	12
expectedDirection	ExpectedDirection	
graceHour	Integer(\$int32)	0
roId	Integer(\$int32)	7
overlayInfo	String	
serialNumber	String	
brand	String	
model	String	
remarks	String	
onError	Boolean	
firmwareVersion	String	
softwareVersion	String	
timeSync	Boolean	
pending	Integer(\$int32)	
reset	Integer(\$int32)	
config	String	
configSHA	String	
lastPing	DateTime	
lastFrame	DateTime	
lastHost	String	
downloadCameraConfig	Boolean	
UpdateFrameTimeout	Integer(\$int32)	
updateFrameQuality	Integer(\$int32)	
cameraFrame	Integer(\$int32)	

Camera

Field	Type	Example
enableMaintenanceAlarms	Boolean	

Camera fields validators

Field	Validation
name	Not blank
zoneId	Valid and existing zone
roId	$0 \leq x \leq 9$
graceHour	$0 \leq x$

Valid types

Field	Validation
CameraType	reader environment
ReportType	both freeflow signaled
ExpectedDirection	Unknown GettingCloser GettingFarther Stopped Left Right

Category

PageCategory

Include: [Paging response fields](#).

Field	Type	
records	Category	See Category

Category

Field	Type	Example
id	Integer(\$int32)	
name	String	

description	String	
zoneSchedules	List of ZoneSchedule	See ZoneSchedule
userIds	List of Integer(\$int32)	[5, 2]
color	String	

Category fields validators

Field	Validation
name	Not blank
zoneSchedules	Not null
Description	Not blank
userIds	Valid list of existing users

Category CSV

Field	Type	Example
id	Integer(\$int32)	5
name (+)	String	“0933BFF”
Description (+)	String	
zoneSchedules	List of ZoneSchedule	“2017-12-31T23:59:59.00Z”
userIds	List of Integer(\$int32)	See ZoneSchedule
color	String	[5]

Configuration**AlertConfig**

Field	Type	Example
id	Integer(\$int32)	1
secondsBetweenAlerts	Integer(\$int32)	0
sendEmailAlerts	Boolean	false
alertEmailFromAddress	String	“username@server.com”
alertEmailFromName	String	“MOBOTIX SYNC-Server”
blacklistAlertEmailSubject	String	
blacklistAlertEmailBody	String	
speedAlertEmailSubject	String	

AlertConfig		
<i>Field</i>	<i>Type</i>	<i>Example</i>
speedAlertEmailBody	String	
prowlingAlertEmailSubject	String	
prowlingAlertEmailBody	String	
kamikazeAlertEmailSubject	String	
kamikazeAlertEmailBody	String	
timeInZoneAlertEmailSubject	String	
timeInZoneAlertEmailBody	String	
timeInZonePlateDistance	Integer(\$int32)	
adrAlertEmailSubject	String	
adrAlertEmailBody	String	
roiCrossingEmailAlertSubject	String	
roiCrossingEmailAlertBody	String	
noAuthVehicleAlertEnable	String	
noAuthVehicleEmailAlertSubject	String	
noAuthVehicleEmailAlertBody	String	
alertEmailHasOcrImage	Boolean	
alertEmailHasEnvImages	Boolean	
blacklistLevenshteinDistance	Integer(\$int32)	
whitelistLevenshteinDistance	Integer(\$int32)	
authorizedVehicleMessage	String	
noAuthVehicleMessage	String	
noAuthExceedVehicleMessage	String	
statusPanelDirectory	String	
pendingQueueThreshold	Integer(\$int32)	
missedPingThreshold	Integer(\$int32)	

AlertConfig

<i>Field</i>	<i>Type</i>	<i>Example</i>
cameraConnectionSubject	String	
cameraConnectionBody	String	
cameraConnectionSolvedSubject	String	
cameraConnectionSolvedBody	String	
cameraPendingQueueSubject	String	
cameraPendingQueueBody	String	
cameraPendingQueueSolvedSubject	String	
cameraPendingQueueSolvedBody	String	
cameraResetSubject	String	
cameraResetBody	String	
cameraResetSolvedSubject	String	
cameraResetSolvedBody	String	
cameraTimeSyncSubject	String	
cameraTimeSyncSolvedSubject	String	
cameraTimeSyncSolvedBody	String	
ForbiddenTurnEmailSubject	String	
ForbiddenTurnEmailBody	String	

AlertConfig fields validators

<i>Field</i>	<i>Validation</i>
secondsBetweenAlerts	$0 \leq x$
AlertEmailFromAddress	Not empty
BlacklistLevenshteinDistance	$0 \leq x \leq 3$
whitelistLevenshteinDistance	$0 \leq x \leq 3$
timeInZonePlateDistance	$0 \leq x \leq 3$
pendingQueueThreshold	$0 \leq x \leq 1000$

missedPingThreshold

$0 \leq x \leq 10$

LDAPConfig

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	1
IDAPServer	String	
IDAPPort	Integer(\$int32)	
IDAPConnection	String	
IDAPSearch	String	
useLDAP	Boolean	

LDAPConfig fields validators

<i>Field</i>	<i>Validation</i>
IDAPServer	Not blank
IDAPPort	1 <= x <= 65535
IDAPConnection	Not blank
IDAPSearch	Not blank

PlateManagerConfig

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	1
imagePath	String	“c:\\images”
savePlateImages	Boolean	true
imageDaysLimit	Integer(\$int32)	30
processLocalPlateRead	Boolean	true
plateDaysLimit	Integer(\$int32)	0
aforoDaysLimit	Integer(\$int32)	10

PlateManagerConfig fields validators

<i>Field</i>	<i>Validation</i>
imagePath	Valid and existing Path
imageDaysLimit	0 <= x
plateDaysLimit	0 <= x
aforoDaysLimit	0 <= x

ReasonConfig

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	1
enforceReason	Boolean	
allowOtherReason	Boolean	
configuredReasons	String	
enforceReasonDetails	Boolean	
reasonLabel	String	
descriptionLabel	String	

ReasonConfig fields validators

<i>Field</i>	<i>Validation</i>
configuredReason	Not null or white space
reasonLabel	Not blank
descriptionLabel	NotBlank

ServerConfig

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	1
serverApiKey	String	
resetApiKey	Boolean	true
enableSsl	Boolean	

ServerConfig		
Field	Type	Example
redirectToHttps	Boolean	
sslCertHash	String	
helix6Port	Integer(\$int32)	
sslPort	Integer(\$int32)	
grafanaPort	Integer(\$int32)	
maximumConcurrentPetition	Integer(\$int32)	
maximumLoginTries	Integer(\$int32)	3
disableTime	Integer(\$int32)	5
auditDaysLimit	Integer(\$int32)	7
auditAllUsers	Boolean	true
autoLogoutTimer	Integer(\$int32)	0
encryptLists	Boolean	false
passwordExpirationDays	Integer(\$int32)	30
timeInPlateList	Integer(\$int32)	100
timeInAuthorizationList	Integer(\$int32)	
defaultZoneSchedules	List of ZoneSchedule	
deleteExpiredAuths	Boolean	
deleteIfNotSeenAuths	Integer(\$int32)	
deleteOutOfTokenAuths	Boolean	
deleteExpiredListitems	Boolean	
daysExpiredListitems	Integer(\$int32)	
ignoreSamePlateSeconds	Integer(\$int32)	
antipassbackMode	Antipassback mode	
antipassbackMode_enum	List of String including all Antipassback modes	
logLevel	Log level	

ServerConfig		
Field	Type	Example
logLevel_enum	List of String including all Log levels	
virtualListFileFolder	String	
virtualListUpdatePeriod	Integer(\$int32)	
roiCrossingEnable	Boolean	
addWhitelistListId	Integer(\$int32)	
addWhitelistHowOld	Integer(\$int32)	

ServerConfig fields validators	
Field	Validation
virtualListFileFolder	Must be a valid Path and exist
disableTime	0 <= x
maximumConcurrentPetition	0 <= x
maximumLoginTries	0 <= x <= 10
auditDaysLimit	0 <= x <= 3600
autoLogoutTimer	0 <= x <= 1440

Valid types	
Field	Validation
Antipassback mode	Disabled, Aforo, Category, Global
Log level	Info, Debug, Trace

SMTPConfig		
Field	Type	Example
id	Integer(\$int32)	1
smtpEnableSsl	Boolean	false
smtpPort	Integer(\$int32)	25
smtpHost	String	“localhost”

SMTPConfig

<i>Field</i>	<i>Type</i>	<i>Example</i>
smtpTimeout	Integer(\$int32)	1000
smtpUseDefaultCredentials	Boolean	false
smtpPassword	String	“1234”
smtpUsername	String	“admin”

SMTPConfig fields validators

<i>Field</i>	<i>Validation</i>
smtpPort	0 <= x <= 65535
smtpTimeout	0 <= x

SMTPTest

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	1
smtpTestEmail	String	“user@server.com”

SMTPTest fields validators

<i>Field</i>	<i>Validation</i>
smtpTestEmail	Valid email address

TelegramConfig

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	1
enableTelegramBot	Boolean	false
telegramBotToken	String	
telegramLoginRedirectUrl	String	

External data

PageExternal		
Include: Paging response fields .		
Field	Type	
records	External	See External
Extrenal		
Field	Type	Example
id	Integer(\$int32)	
timestamp	DateTime as String	“2017-01-01T00:00:00.00Z”
camerald	Integer(\$int32)	
value	double	
type	string	

Grafana

Grafana		
Field	Type	Example
id	Integer(\$int32)	
userId	Integer(\$int32)	
total	Integer(\$int32)	
urls	List of Strings	

Heartbeats

Heartbeat		
Field	Type	Example
id	Integer(\$int32)	
CameraId	Integer(\$int32)	

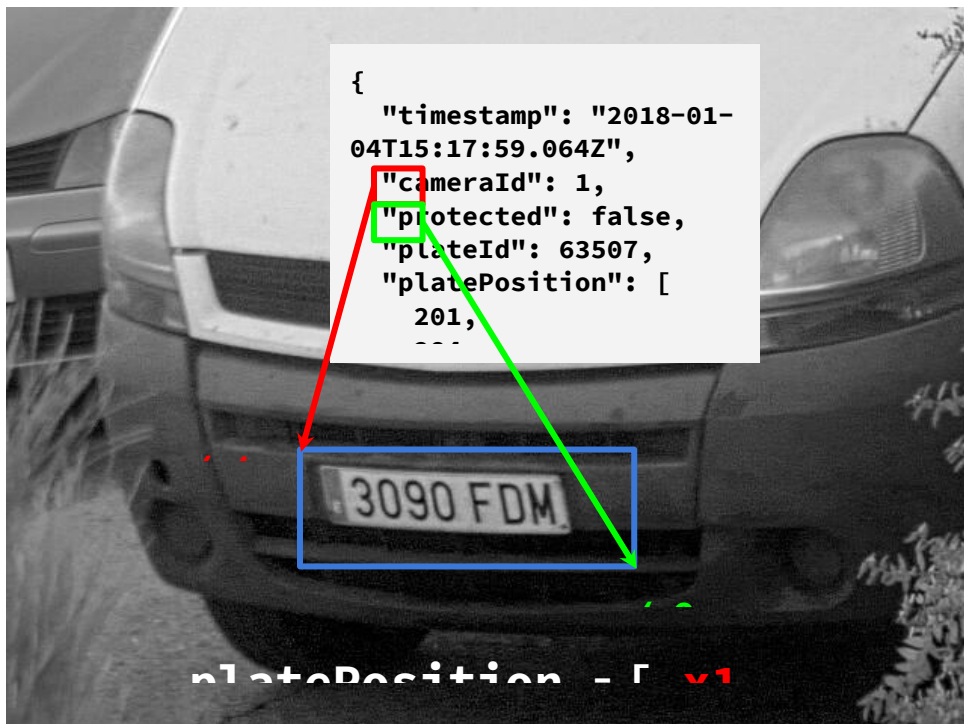
Heartbeat		
<i>Field</i>	<i>Type</i>	<i>Example</i>
ReaderId	Integer(\$int32)	
Pending	Integer(\$int32)	
SoftwareVersion	Strings	
FirmwareVersion	Strings	
BuildNumber	Strings	
Manufacturer	Strings	
SerialNumber	Strings	
Model	Strings	
Date	DateTime as String	“2017-12-19T11:38:33.12Z”
PID	Strings	
PingTimer	Integer(\$int32)	
Config	Strings	
Frame	Strings	
ConfigSHA	Strings	
HeartbeatResponse		
<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	
requestFrame	Integer(\$int32)	
requestConfig	Boolean	
setConfig	Strings	

Images

PageImage		
Include: Paging response fields .		
<i>Field</i>	<i>Type</i>	
records	Image	See Image

Image		
Field	Type	Example
id	Integer(\$int32)	7
timestamp	DateTime as String	“2017-01-01T00:00:00.00Z”
cameraId	Integer(\$int32)	2
protected	Boolean	false
plateId	Integer(\$int32)	2
platePosition	List of Integer(\$int32)	[362,273,545,360,0]
hasFile	Boolean	true
sHA	String	

Image fields validators	
Field	Validation
cameraId	Must be a valid camera
plateId	Must be a valid plate
cameraId & PlateId	Both have the same plateReaderId



Linked Cameras

PageLinkedCamera		
Include: Paging response fields .		
Field	Type	
records	Listitem	(See LinkedCamera)
LinkedCamera		
Field	Type	Example
id	Integer(\$int32)	5
cameralid	Integer(\$int32)	5
linkedCameralid	Integer(\$int32)	5
TimeLapse	Integer(\$int32)	5
TimeUnit	TimeUnit	Milliseconds Seconds Minutes
ForbiddenTurn	Boolean	false
AddCrossLink	Boolean	false

Listitem

PageListitem		
Include: Paging response fields .		
Field	Type	
records	Listitem	(See Listitem)
Listitem		
Field	Type	Example
id	Integer(\$int32)	5
plate	String	“0933BFF”
listId	Integer(\$int32)	5

Listitem

Field	Type	Example
activeFrom	DateTime as String	“2017-01-01T00:00:00.00Z”
activeUntil	DateTime as String	“2017-12-31T23:59:59.00Z”
comments	String	“”
description	String	“”
hitCount	Integer(\$int32)	5
lastDetection	DateTime as String	“2017-12-19T11:38:33.12Z”

Listitem fields validators

Field	Validation
listId	Valid and existing platelist
plate	Not blank
description	Not blank
activeFrom	Not null
activeTo	Null or activeFrom >= activeTo
hitCount	Must be 0 in POST action
LastDetection	Must be null in POST action

Listitem CSV

Field	Type	Example
plate (+)	String	“0933BFF”
activefrom (+)	DateTime as String	“2017-01-01T00:00:00.00Z”
activeuntil (+)	DateTime as String	“2017-12-31T23:59:59.00Z”
comments	String	“”
description (+)	String	“”
hitcount	Integer(\$int32)	5
lastDetection	DateTime as String	“2017-12-19T11:38:33.12Z”

(+) required

Person

PagePerson

Include: [Paging response fields](#).

Field	Type	
records	Authorization	See Person

Person

Field	Type	Example
id	Integer(\$int32)	
title	String	
forename	String	
surname	String	
position	String	[5, 2]
company	String	
personalId	Integer(\$int32)	
hasImage	Boolean	
categoryId	Integer(\$int32)	
isVIP	Boolean	
userDef1	String	
userDef2	String	
userDef3	String	
userDef4	String	
userDef5	String	
userDef6	String	
isGeneric	Boolean	
isDisabled	Boolean	

Person fields validators

<i>Field</i>	<i>Validation</i>
surename	Not blank
forename	Not blank
categoryld	Must be a valid category

Person CSV

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	5
Title	String	“0933BFF”
forename (+)	String	“2017-01-01T00:00:00.00Z”
surname (+)	String	“2017-12-31T23:59:59.00Z”
position	String	
company	String	[5]
personalld	Integer(\$int32)	
hasImage	Boolean	
categoryld	Integer(\$int32)	
isVIP	Boolean	
userDef1	String	
userDef2	String	
userDef3	String	
userDef4	String	
userDef5	String	
userDef6	String	
isGeneric	Boolean	
isDisabled	Boolean	

(+) required

Plates

PagePlate		
Include: Paging response fields .		
Field	Type	
records	Plate	(See Plate)
Plate		
Field	Type	Example
id	Integer(\$int32)	1
camerald	Integer(\$int32)	1
zoneld	Integer(\$int32)	1
cameraName	String	“Camera #1”
latitude	Number(\$double)	12.1234
longitude	Number(\$double)	12.1234
plateNumber	String	“M4616WZ”
originalPlateNumber	String	“M4616WZ”
country	String	“ESP”
originalCountry	String	“ESP”
takenOn	DateTime as String	“2017-12-19T11:38:33.12Z”
status	Status	
direction	VehicleDirection	
speed	Integer(\$int32)	0
confidence	Integer(\$int32)	89
envImageIds	List of Integer(\$int32)	[12300, 12302, 12303]
ocrImageId	Integer(\$int32)	12301
signaled	Boolean	false
avgDigitsHeight	Integer(\$int32)	16

Plate		
Field	Type	Example
id	Integer(\$int32)	1
camerald	Integer(\$int32)	1
zoneld	Integer(\$int32)	1
cameraName	String	“Camera #1”
latitude	Number(\$double)	12.1234
longitude	Number(\$double)	12.1234
plateNumber	String	“M4616WZ”
multiplateRate	Integer(\$int32)	2
processingTimeOCR	Number(\$double)	72.817330121994019
capturerId	Integer(\$int32)	1
capturerType	String	
isAlarm	Boolean	false
kamikazeAlarm	Boolean	false
timeInZoneAlarm	Boolean	false
blacklistAlarm	Boolean	true
alarmListIds	List of Integer(\$int32)	[20, 21]
speedAlarm	Boolean	false
prowlingAlarm	Boolean	false
vehicleMaker	String	“BMW”
vehicleModel	String	“M5”
vehicleColor	String	“UNKNOWN”
crosstime	Number(\$double)	0
hasExit	Boolean	false
signalId	String	
ocrImage	Image	See Image (for PUT action)

Plate		
Field	Type	Example
id	Integer(\$int32)	1
camerald	Integer(\$int32)	1
zoneld	Integer(\$int32)	1
cameraName	String	“Camera #1”
latitude	Number(\$double)	12.1234
longitude	Number(\$double)	12.1234
plateNumber	String	“M4616WZ”
envImages	List of Image	See Image (for PUT action)
authorized	Boolean	
category	String	
vehicleClass	String	
region	String	

SignalId field format

<i>Example</i>	{ main: [1, 5, 9], cross: [] }
<i>Field</i>	<i>Validation</i>
main	Main field includes the plate.id's of linked captures (this field is setted when the camera is main camera)
cross	Cross field includes all main camera plate.id's with this capture is linked (this field is setted when the camera is a linked camera).
NOTE: a camera could be a main and also a liked camera.	

Plate fields validators

<i>Field</i>	<i>Validation</i>
plateNumber	Not blank
originalPlateNumber	Not blank
camerald	Valid and existing camera
confidence	0 <= x <= 100

ocrImageId	Valid and existing image
envImageIds	Empty or a Valid list of existing images
Valid types	
<i>Field</i>	<i>Validation</i>
Status	ok pending confirmed cancelled exported prescribed noAuth sanctionPending ignored
VehicleDirection	Unknown GettingCloser GettingFarther Stopped Left Right

Plate lists

PagePlatelist		
Include: Paging response fields .		
<i>Field</i>	<i>Type</i>	
records	Platelist	(See Platelist)
Platelist		
<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	5
name	String	"Employees"
type	ListType	
color	Hex color as String	"#0077FF"
description	String	"White list of employees"
userIds	List of Integer(\$int32)	[1, 3]
ownerId	Integer(\$int32)	1
zoneId	Integer(\$int32)	5
version	String	
guid	String	
renewItemsExpiration	Boolean	false

IsVirtual	Boolean	false
------------------	---------	-------

Platelist fields validators

<i>Field</i>	<i>Validation</i>
name	Not blank
color	Valid color
UserIds	Valid list of existing users
zonelid	Valid and existing zone
ListType	blacklist, whitelist

Valid type

<i>Field</i>	<i>Type</i>
ListType	blacklist, whitelist

Regular expression (AuthExpression)Page**AuthExpression**

Include: [Paging response fields](#).

Field	Type	
records	Timeframe	(See AuthExpression)

AuthExpression

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	1
description	String	Plate number is odd
regExpression	String	^[a-zA-Z]*)(\d*)(0 1)\$

RollcallPage**Rollcall**

Include: [Paging response fields](#).

<i>Field</i>	<i>Type</i>	
records	Rollcall	Rollcall

Rollcall

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	
aforold	Integer(\$int32)	
person	Person	See Person
plate	Plate	See Plate

Schedule

PageSchedule

Include: [Paging response fields](#).

Field	Type	
records	Schedule	See Schedule

Schedule

Field	Type	Example
id	Integer(\$int32)	12
description	String	“Working time”
activeFrom	UTC as String	"2016-11-06T08:40:00Z"
activeUntil	UTC as String	"2027-11-25T08:40:00Z"
global	Boolean	false
timeFrameIds	List of Integer(\$int32)	[2, 5]
authExpressionIds	List of Integer(\$int32)	[]

Schedule fields validators

Field	Validation
description	Not blank
timeFrameIds	Valid list of existing timeframes
activeFrom && activeUntil	activeFrom >= activeUntil

Section

PageSection

Include: [Paging response fields](#).

Field	Type	
records	Section	(See Section)

Section		
<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	12
name	String	“Security section”
distance	Integer(\$int32)	200 // Meters
maximumSeconds	Integer(\$int32)	12
entranceCameraId	Integer(\$int32)	5
exitCameraId	Integer(\$int32)	3

Section fields validators

<i>Field</i>	<i>Validation</i>
name	Not blank
distance	1 <= x
maximumSeconds	1 <= x
entranceCameraId	Valid and existing camera
exitCameraId	Valid and existing camera

Timeframe

PageTimeframe

Include: [Paging response fields](#).

Field	Type	
records	Timeframe	(See Timeframe)

Timeframe

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	12
description	String	“2 days week, odd months”
ignoreYear	Boolean	false

Timeframe		
Field	Type	Example
startYear	Integer(\$int32)	2015
stopYear	Integer(\$int32)	9999
startHour	Integer(\$int32)	0
stopHour	Integer(\$int32)	23
startMinute	Integer(\$int32)	0
stopMinute	Integer(\$int32)	59
startDay	Integer(\$int32)	1
stopDay	Integer(\$int32)	7
daysOfWeek	List of Day	["monday", "tuesday"]
months	array of Mont	[1, 3, 5, 6, 9, 11]

Timeframe fields validators	
Field	Validation
daysOfWeek	Not empty valid day list
months	Not empty valid month list
description	Not blank
startDay	1 <= x <= 31
stopDay	1 <= x <= 31
startHour	1 <= x <= 23
stopHour	1 <= x <= 23
startMinute	1 <= x <= 59
stopMinute	1 <= x <= 59
startHour, stopHour, startMinute, stopMinute, startDay, stopDay, startYear and stopYear	(startHour < stopHour) or (startHour == stopHour and startMinute <= stopMinute)
	startDay <= stopDay

	startYear <= stopYear
--	-----------------------

Valid types

Field	Type
Day	sunday monday tuesday wednesday thursday friday saturday
Moth	1: January, 2: February, 3: March, 4: April, 5: May, 6: June, 7: July, 8: August, 9: September, 10: October, 11: November, 12: December

User

Page**User**

Include: [Paging response fields](#).

Field	Type	
records	User	(See User)

User

Field	Type	Example
id	Integer(\$int32)	12
username	String	“admin”
password	String	“”
rememberMe	Boolean	true
claims	List of Claims	[“Admin”]
email	String	“admin@vaxtor.com”
firstName	String	“System”
lastName	String	“Admin”
audited	Boolean	false
language	Language	“en”

User		
<i>Field</i>	<i>Type</i>	<i>Example</i>
apiKey	String	"d7a3ceb36b254085a064fc20182e70fd"
resetApiKey	Boolean	false
userId	String	"12"
pluginIds	List of Integer(\$int32)	[]
passwordExpirationDate	DateTime as String	"2017-01-01T00:00:00.00Z"
pushbulletApiKey	String	
availablePushbulletDevices	List of PushbulletDevice	
selectedPushbulletDevices	List of PushbulletDevice	
setEmptyPassword	Boolean	

User fields validators

<i>Field</i>	<i>Validation</i>
username	Not blank and not repeated
userId	added for sanctions, where corresponding to Agent ID
pushbullet	Valid Pushbullet configuration.

Valid types

<i>Field</i>	<i>Types</i>
Claims	Admin Actuators Basic Authorizations AdminLists Cameras Lists Plates Schedules Statistics Users Zones
Language	en es ca zh th vn de fr

PushbulletDevice

<i>Field</i>	<i>Type</i>	<i>Example</i>
iden	String	
nickname	String	

Zone

PageZone

Include: [Paging response fields](#).

Field	Type	
records	Zone	(See Zone)

Zone

Field	Type	Example
id	Integer(\$int32)	12
name	String	“Main zone”
scheduleIds	List of Integer(\$int32)	[2, 5]
parentZoneId	Integer(\$int32)	
inheritSchedules	Boolean	false
userIds	List of Integer(\$int32)	[2, 5]
speedAlarm	Boolean	true
speedLimit	Integer(\$int32)	50
prowlingAlarm	Boolean	false
prowlingTimesDetected	Integer(\$int32)	0
prowlingMinutes	Integer(\$int32)	0
kamikazeAlarm	Boolean	false
adrAlarm	Boolean	
townCode	String	“28000”
UseAuthorizationRules	Boolean	

Zone fields validators

Field	Validation
name	Not blank
scheduleIds	Not null and valid list of existing schedules

Zone fields validators

<i>Field</i>	<i>Validation</i>
parentZoneId	if x != 0, it must be a valid and existing zone.
userId	Valid and existing user
speedLimit	$0 \leq x$
prowlingMinutes	$0 \leq x$
prowlingTimesDetected	$0 \leq x$

Plugins data models

Sanctions Plugin

SanctionConfig		
Field	Type	Example
id	Integer(\$int32)	1
caseFileFormat	String	“\$year\$-\$id\$”
prescribedDays	Integer(\$int32)	25
lastExpedient	Integer(\$int32)	100
maxCountPathLength	Integer(\$int32)	100

CameraSanctions		
Field	Type	Example
id	Integer(\$int32)	7
camerald	Integer(\$int32)	2
sanctionIds	List of Integer(\$int32)	[2, 5]

CameraSanctions fields validators	
Field	Validation
camerald	Not null and valid and existing camera
sanctionIds	Not null and valid list of existing sanctions

PageCancellation		
Include: Paging response fields .		
Field	Type	
records	Cancellation	(See Cancellation)

Cancellation		
Field	Type	Example
id	Integer(\$int32)	7

code	Integer(\$int32)	2
description	String	“By error”

Cancellation fields validators

<i>Field</i>	<i>Validation</i>
description	Not blank
code	Not blank

PagePlateSanctions

Include: [Paging response fields](#).

<i>Field</i>	<i>Type</i>	
records	PlateSanctions	(See PlateSanctions)

PlateSanctions

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	5
userId	Integer(\$int32)	2
plateId	Integer(\$int32)	2
imageIds	List of Integer(\$int32)	[255, 256]
sanctionId	Integer(\$int32)	5
cancellationId	Integer(\$int32)	7
exportedOn	DateTime as String	“2017-01-01T00:00:00.00Z”
createdOn	DateTime as String	“2017-01-01T00:00:00.00Z”
country	String	“es”
caseFile	String	format:<yyyy>-<PlateSanction.id>
plateNumber	String	“0933BFF”
cameraId	Integer(\$int32)	5
takenOn	DateTime as String	“2017-01-01T00:00:00.00Z”

PlateSanctions fields validators

<i>Field</i>	<i>Validation</i>
plateId	Existing plate for POST and PUT
sanctionId	0 <= x
cancellationId	0 <= x
sanctionId and cancellationId	sanctionId != 0 or cancellationId != 0

PageSanction

Include: [Paging response fields](#).

<i>Field</i>	<i>Type</i>	
records	Sanction	(See Sanction)

Sanction

<i>Field</i>	<i>Type</i>	<i>Example</i>
id	Integer(\$int32)	12
shortDescription	String	“Red Light”
type	SanctionType	
trigger	SanctionTrigger	
penaltyMoney	Double	100.00
description	String	“Red Light violation”
rule	String	“Rule”
article	String	“Article xx”
section	String	“Section yy”
subsection	String	“Subsection zz”
behaviorNumber	String	“NN”
penaltyPoints	Integer(\$int32)	6
legalTex	String	“Legal text...”
licenceWithdrawal	Boolean	false

Sanction		
Field	Type	Example
complaintOrigin	String	Note: Gespol plugin
nonNotificationCause	String	Note: Gespol plugin

Sanction fields validators	
Field	Validation
description	Not blank
article	Not blank
penaltyMoney	0 <= x
role	Not blank

Valid types	
Field	Types
SanctionType	zoneNotAuthorized redLigh wrongDirection forbiddenTurn
SanctionTrigger	plateNotAuthorized timeInZone

Export sanctions

ExportSanctionConfig		
Field	Type	Example
id	Integer(\$int32)	1
eXTransferDir	String	“\$year\$-\$id\$”
eXNotificationEmail	String	25
eXExportSanctionType	Export Sanction Type	
eXExportSanctionType_enum	List of String including all Export Sanction Types	
eXRowTemplate	String	
eXIncludeImages	Boolean	

ExportSanctionConfig fields validators

<i>Field</i>	<i>Validation</i>
eXTransferDir	Must be a valid Path and exist
eXNotificationEmail	Not null, not empty

Valid type

<i>Field</i>	<i>Validation</i>
Export Sanction Type	csv, xml, json

Filter keys

Notes

1. Date fields must be ISO-8601 format (ex. Javascript Date)
2. Converted all values to string (ex using JSON.stringify)

Examples (Plates)

```
filterKey:["camera","dateFrom","alarm"]
filterValue:["1","2018-01-03T23:00:00.000Z","true"]
```

Audits

Token	filterKey	filterValue
ACTION	action	Action Example: ACTION:GET /api/user/all
DETAILS	details	Text included in Details attribute.
DATE FROM	dateFrom	DateTime format dd/MM/yyyy hh:mm:ss, example: 08/01/2018 10:22:29
DATE TO	dateTo	DateTime format dd/MM/yyyy hh:mm:ss, example: 08/01/2018 10:22:29
FAILED	failed	true or false
USER ID	userId	User ID or user username
SOURCE ADDRESS	sourceAddress	IP address Example: SOURCE:192.168.0.131

Authorizations

Token	filterKey	filterValue
PLATE	plate	p.e. M4616WZ
DESCRIPTION DES	description	Description
COMMENTS	comments	Comments

ELASTIC	elastic	free text to search in any text fields of Authorization
PERSON OWNER PERSONID	personId	Person ID

Category

Token	filterKey	filterValue
NAME	name	Category name
DESCRIPTION	description	Category description

Person

Token	filterKey	filterValue
SURNAME	surname	free text
FORENAME	forename	free text
COMPANY	company	free text
PERSONAL ID	personalId	free text
CATEGORY ID	categoryId	Category ID (from Category entity)
POSITION	position	free text
ELASTIC	elastic	free text to search in any text fields of Person

Plates

Token	filterKey	filterValue
CAMERA	camera	Camera ID or Camera name
COUNTRY	country	Three letters Country code
DIRECTION DIR	direction	Unknown GettingCloser GettingFarther Stopped Left Right
PLATE	plate	p.e. M4616WZ
STATUS	status	OK, Pending, Confirmed, Canceled, Exported, Prescribed, NoAuth, SanctionPending, NoAuthExceed, Ignored

Token	filterKey	filterValue
ZONE	zone	Zone ID or Zone name
CAPTURERID CAPID	captureId	Capturer ID
CAPTURERTYPE CAPTYPE	captureType	Capturer Type
MAXLATITUDE MAXLAT	maxlatitude	Max latitude (double)
MINLATITUDE MINLAT	minlatitude	Min latitude (double)
MAXLONGITUDE MAXLON	maxlongitude	Max longitude (double)
MINLONGITUDE MINLON	minlongitude	Min longitude (double)
DATEFROM FROM	dateFrom	DateTime format dd/MM/yyyy hh:mm:ss, example: 08/01/2018 10:22:29
DATETO TO	dateTo	DateTime format dd/MM/yyyy hh:mm:ss, example: 08/01/2018 10:22:29
SPEED	speed	Speed (integer)
SIGNALID	signalId	Signal ID (integer)
VEHICLEMAKER	vehicleMaker	Vehicle marker (text)
VEHICLEMODEL	vehicleModel	Vehicle model (text)
VEHICLECOLOR	vehicleColor	Vehicle color (text)
VEHICLECLASS	vehicleClass	Vehicle class
CATEGORY	category	Category
REGION	region	Region
ALARM	alarm	If it is an alarm (boolean)
EXIT	exit	If it is an exit (boolean)

Token	filterKey	filterValue
CROSSTIME	crosstime	If crosstime is >=
ELASTIC	elastic	
N/A	list	If it is in a list with ID == value.
N/A	plateId	Plate with id == plateId

Plate list items

Token	filterKey	filterValue
LISTIDS LIST	listIds	List ID or List name, separated with comma (,)
TYPE	type	blacklist or whitelist
ACTIVE	active	true or false
PLATE	plate	p.e. M4616WZ
LOCATED	located	true or false
DATEFROM FROM	lastDetectionFrom	DateTime format dd/MM/yyyy hh:mm:ss, example: 08/01/2018 10:22:29
DATETO TO	lastDetectionTo	DateTime format dd/MM/yyyy hh:mm:ss, example: 08/01/2018 10:22:29
DESCRIPTION DESC	description	Description
COMMENTS	comments	Comments

Rollcall

Token	filterKey	filterValue
AFOROS	aforosIds	p.e. [2, 5, 9]
PERSON	personId	

Claims

Accesses allowed

CLAIM	Resource	Method
Admin	<i>All</i>	<i>Full access</i>
Actuators	/api/actuator	GET, PUT, POST, DELETE
AdminLists	/api/user	GET *
	/api/zone	GET
	/api/platelist	GET, PUT, POST, DELETE, Upload Listitems
	/api/listitem	GET, PUT, POST, DELETE
AdminAuthorizations	/api/authorization	GET, PUT, POST, DELETE
	/api/category	GET, PUT, POST, DELETE
	/api/person	GET, PUT, POST, DELETE
	/api/rollcall	GET, DELETE
	/api/schedule	GET
	/api/zone	GET
Authorizations	/api/authorization	GET, PUT, POST, DELETE
	/api/category	GET, PUT, POST, DELETE
	/api/person	GET, PUT, POST, DELETE
	/api/rollcall	GET, DELETE
	/api/schedule	GET
	/api/zone	GET

CLAIM	Resource	Method
Basic	/api/aforo	GET
	/api/camera	GET
	/api/external	GET
	/api/linked	GET
	/api/image	GET
	/api/plate	GET
	/api/authExpression	GET
	/api/schedule	GET
	/api/timeframe	GET
	/api/zone	GET
Cameras	/api/camera	GET, PUT, POST, DELETE
	/api/linkedCamera	GET, PUT, POST, DELETE
	/api/zone	GET
Lists	/api/user	GET *
	/api/zone	GET
	/api/platelist	GET, PUT, POST, DELETE, Upload Listitems
	/api/listitem	GET, PUT, POST, DELETE
Plates	/api/camera	GET
	/api/linkedCamera	GET
	/api/image	GET, PUT, POST

CLAIM	Resource	Method
	/api/imageupload	POST
	/api/plate	GET, PUT, POST
	/api/zone	GET
Schedules	/api/authExpression	GET, PUT, POST, DELETE
	/api/schedle	GET, PUT, POST, DELETE
	/api/timeframe	GET, PUT, POST, DELETE
Statistics	/api/grafana	GET, PUT, POST
Users	/api/user	GET, PUT, POST, DELETE
Zones	/api/aforo	GET, PUT, POST, DELETE
	/api/camera	GET **
	/api/linkedCamera	GET **
	/api/plateList	GET
	/api/schedule	GET
	/api/section	GET
	/api/user	GET *
	/api/zone	GET, PUT, POST, DELETE

Note:

(*) Can see other users (of the same list or zone.)

(**) If combine Zones and Camera claims, can see all records.

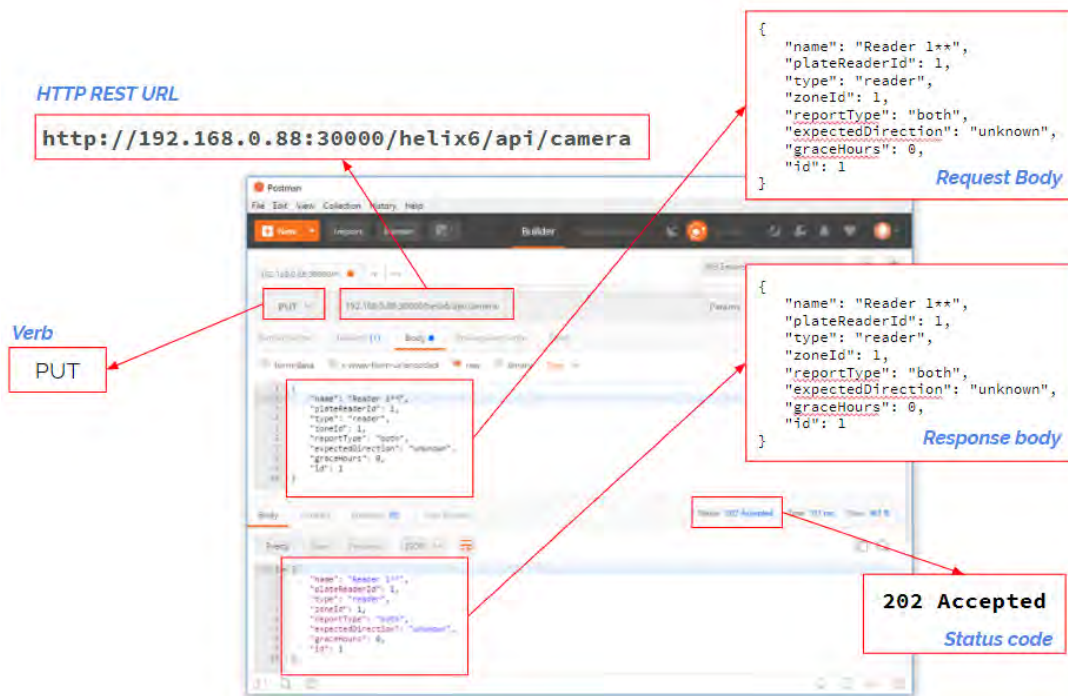
Annexes

How to use Postman²

In order to work with MOBOTIX SYNC Server through its REST API, It must to following next steps



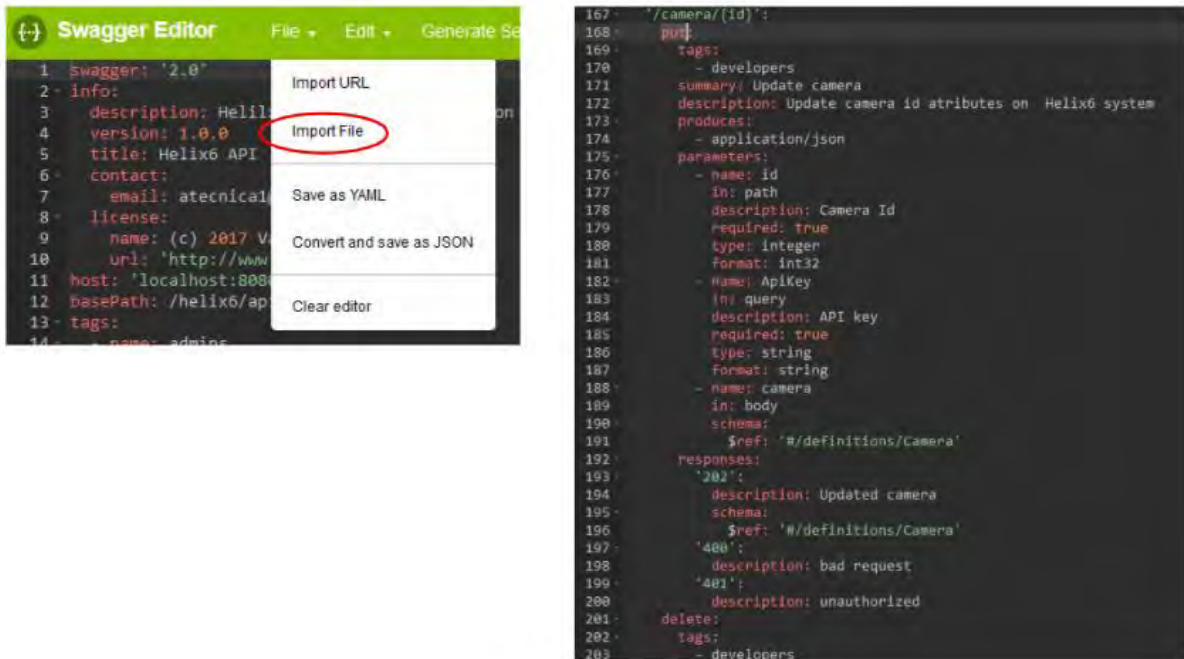
1. Set Verb
2. Set HTTP REST URL
3. Set Parameters (if it has any)
4. Set **x-api-key** in header (or **ApiKey** in query string)
5. Click [Send] button and get response..



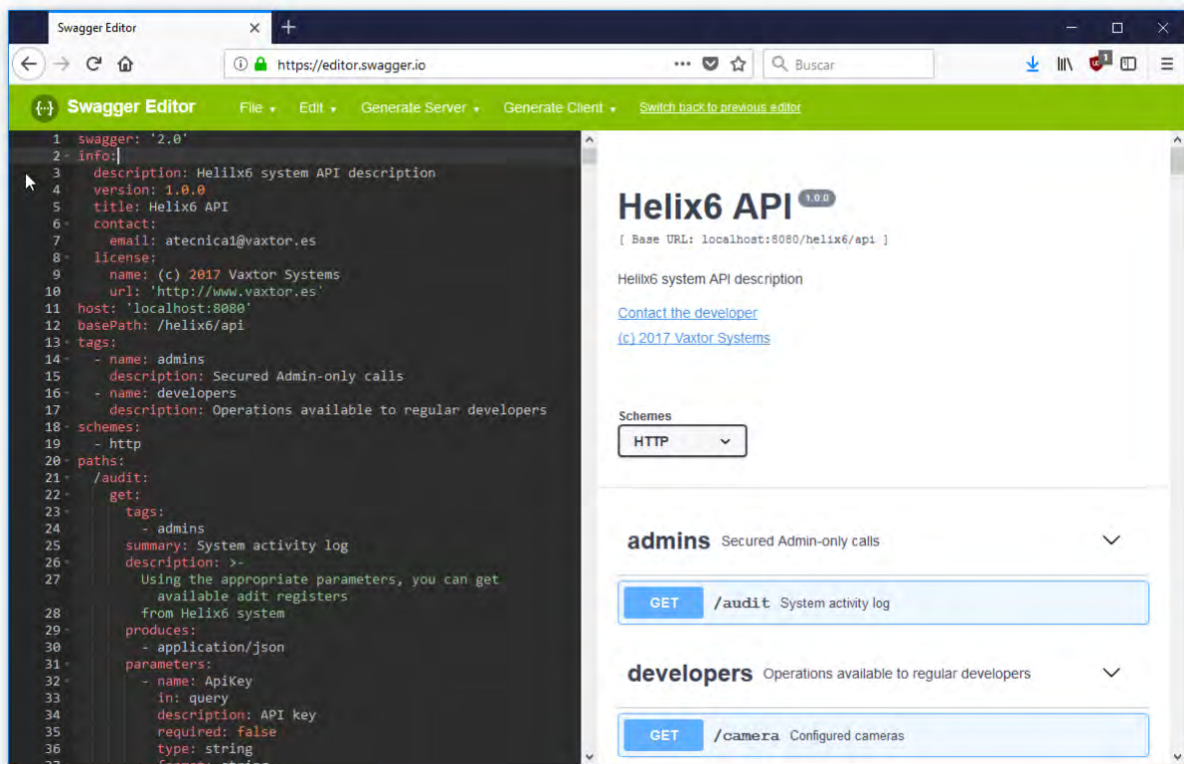
² www.getpostman.com

How to use Swagger³

Load **MOBOTIX SYNC Server API-x.x.x.json** file

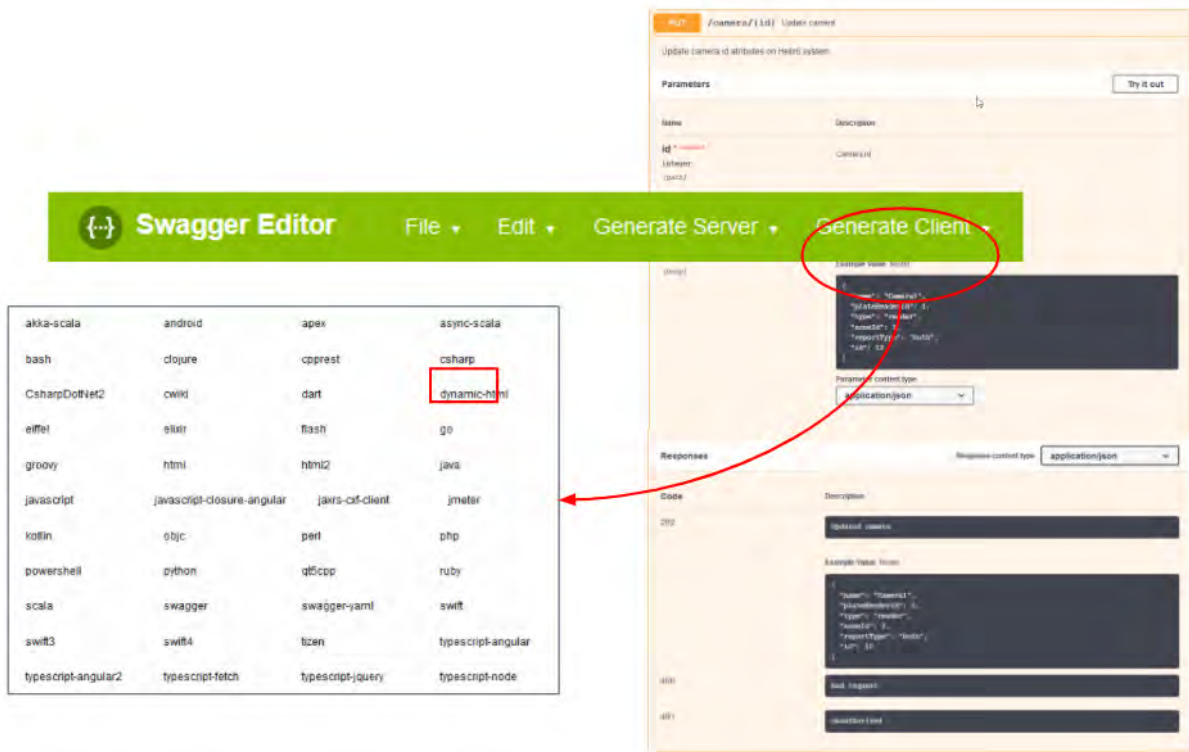


API Documentation



³ Documentation: <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md>

Code Generation



Client code example (C#)

```

...
// RestSharp: https://www.nuget.org/packages/RestSharp/105.1.0
using RestSharp;
...

RestClient = new RestClient( "http://myMOBOTIX.com:8080/SYNC/api" );
...

public Object CallApi
(
    String path,
    RestSharp.Method method,
    Dictionary<String, String> queryParams,
    Object postBody,
    Dictionary<String, String> headerParams,
    Dictionary<String, String> formParams,
    Dictionary<String, FileParameter> fileParams,
    Dictionary<String, String> pathParams,
    String contentType)
{
    var request = PrepareRequest(
        path, method,
        queryParams,
        postBody,
        headerParams,
        formParams,
        fileParams,
        pathParams,
        contentType );
}

```

```

// set timeout
RestClient.Timeout = Configuration.Timeout;
// set user agent
RestClient.UserAgent = Configuration.UserAgent;

InterceptRequest(request);
var response = RestClient.Execute(request);
InterceptResponse(request, response);

return (Object) response;
}

...
var _Path = "/camera/{id}";
var _PathParams = new Dictionary<String, String>();
var _QueryParams = new Dictionary<String, String>();
var _FormParams = new Dictionary<String, String>();
var _Method = Method.PUT;
var _FileParams = new Dictionary<String, FileParameter>();
var _HeaderParams = new Dictionary<String,
String>(Configuration.DefaultHeader);
...

String[] _HttpContentTypes = new String[] {};
String _HttpContentType =

Configuration.ApiClient.SelectHeaderContentType(_HttpContentTypes);
...
String[] _HttpHeaderAccepts = new String[] { "application/json" };
_HeaderParams.Add("Accept", _HttpHeaderAccept);
...
_PathParams.Add("id", Configuration.ApiClient.ParameterToString(id));
_QueryParams.Add("ApiKey", Configuration.ApiClient.ParameterToString(apiKey));
..
_PostBody = Configuration.ApiClient.Serialize(camera);
...
IRestResponse _Response = (IRestResponse) CallApi( _Path,

int _StatusCode = (int) _Response.StatusCode;
...

```

Important!
Based on Swagger C#

CallApi()

Summary
Makes the HTTP request (Sync).

Params
path = URL path
method = HTTP method.
queryParams = Query parameters.
postBody = HTTP body (POST request).
headerParams = Header parameters.
formParams = Form parameters.

```

_Method,
_QueryParams,
_PostBody,
_HeaderParams,
_FormParams,
_FileParams,
_PathParams,
_HttpContentType);

```