



REST API

for Eocortex

Table of content

Introduction	6
Resource description	7
Using REST API	11
Description of response codes	12
Logging requests on server side	13
configure/channels	14
Brief information regarding all cameras	14
Adding new cameras	15
Modification of existing cameras	22
configure/channels/{channel_id}	24
Detailed camera information	24
Deletion of existing camera	26
configure/channels/{channel_id}/scenarios	27
Description of camera scenarios	27
Modification of camera scenarios	30
configure/channels/{channel_id}/facescomplete	35
Receiving Face Recognition Complete module's settings	35
Modifying settings of the Face Recognition Complete module	36
configure/channels/{channel_id}/peoplecounter	41
Acquiring settings of visitor counting module.....	41
Modifying settings of visitor counting module	41
configure/channels/{channel_id}/visitors	44
Receiving Unique Visitor Counting module's settings	44
Modifying settings of the Unique Visitor Counting module	45

configure/channels/{channel_id}/queuecounter	50
Receiving People Counting in Queue module's setting	50
Modifying settings of the People Counting in Queue module	51
configure/channels/{channel_id}/platescomplete	54
Receiving License Plate Recognition Complete module's settings	54
Modifying settings of the License Plate Recognition Complete module	54
configure/secobjects	56
Root security object	56
Adding new security objects	57
Modifying existing security objects	59
configure/secobjects/{secobject_id}	62
Selected security objects	62
Deleting existing security object	63
configure/groups	64
Adding new groups	64
Modifying existing groups	66
configure/groups/{group_id}	69
Deleting an existing group	74
configure/groups/{group_id}/users	75
configure/users	76
Adding new users	76
Modifying existing users	78
configure/users/{user_id}	80
Deleting existing user	80
configure/viewprofiles	81
Adding new profiles	82

Modifying existing profiles.....	84
configure/viewprofiles/{viewprofile_id}	87
Deleting an existing profile	87
configure/servers	88
configure/servers/{server_id}.....	90
configure/servers/{server_id}/channels	91
configure/devicemanufacturers	92
configure/monitoring	95
Modifying monitoring settings.....	95
api/channels	98
api/channels/{channel_id}	100
api/channels/{channel_id}/status	101
REST API for face recognition.....	103
Requirements.....	103
Requirements for the photos to be uploaded	104
GET /api/faceconfig.....	105
GET /api/faces.....	106
POST /api/faces.....	108
GET /api/faces/<id>	111
PUT /api/faces/<id>	113
DELETE /api/faces/<id>	116
GET /api/faces-groups.....	117
POST /api/faces-groups.....	119
GET /api/faces-groups/<id>	121
PUT /api/faces-groups/<id>	122
DELETE /api/faces-groups/<id>	124

REST API interaction with License Plate database	125
Use cases	126
Requirements, limitations, recommendations	126
Requests	127
GET /api/carconfig	127
GET /api/cars	128
POST /api/cars	132
GET /api/cars/<id>	134
PUT /api/cars/<id>	136
DELETE /api/cars/	138
GET /api/cars-groups	139
POST /api/cars-groups	142
GET /api/cars-groups/<id>	144
PUT /api/cars-groups/<id>	145
DELETE /api/cars-groups/	147
REST API for getting data from the Object Classification and Counting module	148
GET /api/object_counting/current_counters	149
GET /api/object_counting/report.....	151

Introduction

Eocortex REST API permits to:

- Configure cameras, users, groups, screen profiles, certain automated scenarios, people counting module.
- Review information regarding the status of servers and cameras and supported camera modules.
- Work with face databases: receive, add and modify entries regarding faces and groups.

Eocortex REST API can be used for configuring the system without using Eocortex Configurator application.

The **configure/** resource of the function is only available for Eocortex with **Enterprise** and **Ultra** licenses. Configuring with REST API can only be performed by a user with full rights, i.e. by the user belonging to the **Senior Administrators** group.

api/ resource is available for all users and in any **Eocortex** product.

Resource description

REST API allows to receive, add, modify and delete various resources.

All the resources will be listed in the tables below (GET: reading of objects, POST: adding objects, PUT: modifying objects, DELETE: deleting objects).

Description of the resources used for configuring the system (such requests are to be sent strictly to the main video surveillance server; the request string starts with the word **configure**):

Version: indicates the version with which the resource appears in the REST API.

Resource	Description	Version	GET	POST	PUT	DELETE
configure/channels	Brief information regarding all cameras, adding/modifying cameras	2.5	+	+	+	n/a
configure/channels/{channel_id}	Detailed information regarding a camera, deleting a camera	2.5	+	n/a	n/a	+
configure/channels/{channel_id}/scenarios	Information regarding a camera's scenarios, modification of a camera's scenarios	2.5	+	n/a	+	n/a
configure/channels/{channel_id}/facescomplete	Face Recognition Complete module settings, modification of Face Recognition Complete module's settings	3.4	+	n/a	+	n/a
configure/channels/{channel_id}/peoplecounter	People Counting module settings, changing People Counting module settings	2.5	+	n/a	+	n/a
configure/channels/{channel_id}/visitors	Unique Visitor Counting module settings, modification of Unique Visitor Counting module's settings	3.4	+	n/a	+	n/a

configure/channels/{channel_id}/queuecounter	People Counting in Queue module settings, modification of People Counting in Queue module's settings	3.4	+	n/a	+	n/a
configure/channels/{channel_id}/platescomplete	License Plate Recognition Complete module settings, modification of License Plate Recognition Complete module's settings	3.4	+	n/a	+	n/a
configure/secobjects	Root safety object, adding/modifying safety objects	2.5	+	+	+	n/a
configure/secobjects/{secobject_id}	Selected safety object, deleting safety object	2.5	+	n/a	n/a	+
configure/groups	Brief description of all groups, adding/modifying groups	2.5	+	+	+	n/a
configure/groups/{group_id}	Detailed description of a group, deleting a group	2.5	+	n/a	n/a	+
configure/groups/{group_id}/users	Brief description of all users of the group	2.5	+	n/a	n/a	n/a
configure/users	Brief description of all users, adding/modifying users	2.5	+	+	+	n/a
configure/users/{user_id}	Detailed description of a user, deleting a user	2.5	+	n/a	n/a	+/-
configure/viewprofiles	Brief description of all profiles, adding/modifying profiles	2.5	+	+	+	n/a
configure/viewprofiles/{viewprofile_id}	Description of a profile, deleting a profile	2.5	+	n/a	n/a	+

configure/servers	Brief information regarding all servers	2.5	+	n/a	n/a	n/a
configure/servers/{server_id}	Detailed information regarding a server	2.5	+	n/a	n/a	n/a
configure/servers/{server_id}/channels	Brief information regarding the server's cameras	2.5	+	n/a	n/a	n/a
configure/devicemanufacturers	Supported camera models	2.5	+	n/a	n/a	n/a
configure/monitoring	Monitoring settings	3.5	+	n/a	+	n/a

Description of the resources used for obtaining information regarding a server (such requests can be sent to any video surveillance server; the request string starts with the word "api"):

Resource	Description	Version	GET	POST	PUT	DELETE
api/channels	Brief information regarding all cameras assigned to a server	2.5	+	n/a	n/a	n/a
api/channels/{channel_id}	Brief information regarding a camera	2.5	+	n/a	n/a	n/a
api/channels/{channel_id}/status	Camera performance status (reception bitrate, recording bitrate, issues detected)	2.5	+	n/a	n/a	n/a
api/faceconfig	Information regarding available face recognition modules	2.8	+	n/a	n/a	n/a
api/faces	Description of all faces in database with possibility to add one new face	2.8	+	+	n/a	n/a
api/faces/<id>	Detailed description of a face with images, updating/deleting/modifying a face	2.8	+	n/a	+	+

api/faces-groups	List of groups of faces with description, adding a group of faces	2.8	+	+	n/a	n/a
api/faces-groups/<id>	Description of a group with list of faces in it	2.8	+	n/a	+	+

Using REST API

All the requests are to be made on behalf of the Senior Administrator group.

All the requests for configuring the system (configure) are to be sent to the system's main server. The requests for information about the server (api) can be sent to any video surveillance server.

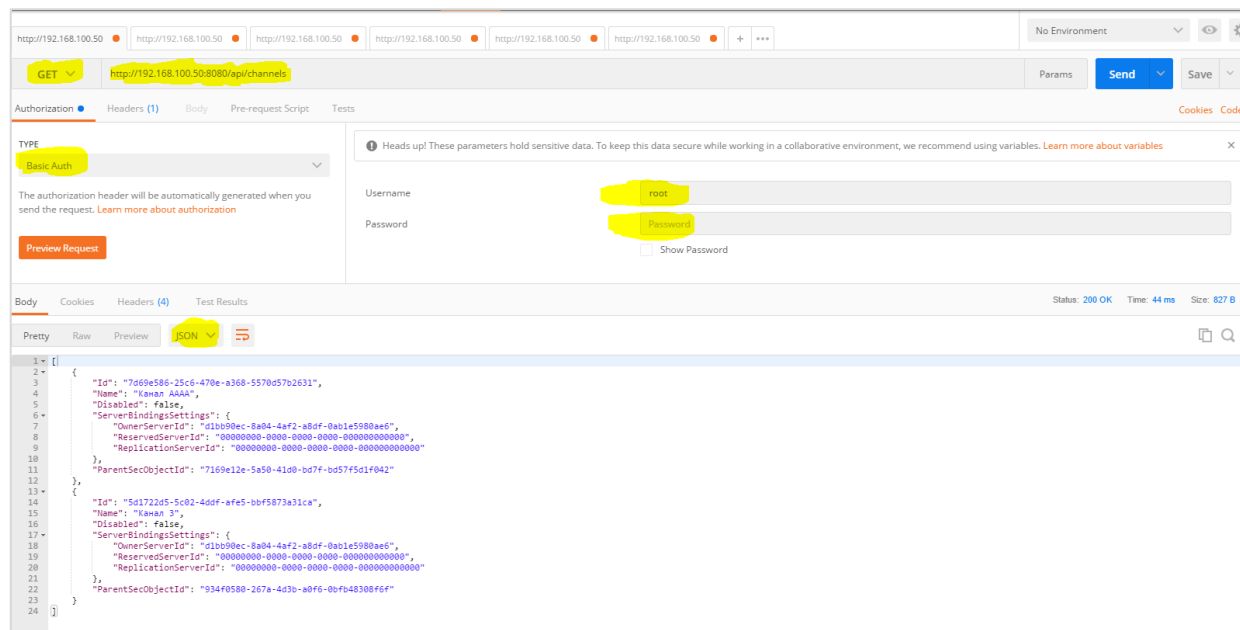
To obtain access to resources, it is required to use HTTP Basic Authentication (https://en.wikipedia.org/wiki/Basic_access_authentication). Base64 is to be taken from the view string: "login:password".

At that, the password shall be transferred as an md5 hash (for example, it can be generated here: <https://www.md5hashgenerator.com/>).

Please note that the configuration feature is available for Eocortex Ultra only and can be executed exclusively by a user with full rights, i.e. by the user belonging to the Senior Administrators group.

To use REST API, a browser only will not be enough, since many requests are POST, PUT, and DELETE ones; that is why we recommend to use applications like Postman (<https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop?hl=ru>)

Below is a screenshot of the Postman application (to use the program, it is required to enter the path to the resource and activate the Basic Authorization):



Description of response codes

HTTP code	Description
200 (OK)	Request completed successfully.
400 (BadRequest)	Request not completed. Request created wrongly, or an unexpected error occurred during fulfilling the request.
403 (Forbidden)	Request not completed. Resource requested forbidden for current user; make sure that user has configuration rights.
404 (NotFound)	Request not completed. Requested resource not found.
409 (Conflict)	Request not completed. Configuration modification detected in the course of making the request. Repeat request later.

At that, in case of an error, the response is usually accompanied by a textual description of the error, e.g. the GET request with a misprint ("channel" instead of "channels") `http://192.168.100.50:8080/configure/channel` will be responded with the HTTP code 404 (NotFound), and the response will also contain the following description of the error:

```
{
  "ErrorMessage": "Resource is not found (/configure/channel | GET)"
}
```

Logging requests on server side

All the requests are logged on the server side in the standard folder with the Eocortex Server logs. The requests for configuring the system are logged in the log entitled RestConfigure.log; the requests for server information are logged in the log entitled RestApi.log.

Here is an example of logging the misprinted request <http://192.168.100.50:8080/configure/channel>:

```
[2018-01-28 16:33:42,961 Thread=Server Worker: 1D0AFD5]
Start (/configure/channel | GET)

[2018-01-28 16:33:42,962 Thread=Server Worker: 1D0AFD5]
ERROR
Resource is not found (/configure/channel | GET)
ConfigStorage.Rest.Exceptions.RestNotFoundException: Application error.
  ConfigStorage.Rest.RestResourcesFactory.Create(String rawUrl) in
Q:\AlarusRootGit_rest\Alarus\Core\ConfigStorage\Rest\RestResourcesFactory.cs:string 86
  ConfigStorage.Rest.RestRequestProcessor.ProcessRequest() in
Q:\AlarusRootGit_rest\Alarus\Core\ConfigStorage\Rest\RestRequestProcessor.cs:string 63

[2018-01-28 16:33:42,964 Thread=Server Worker: 1D0AFD5]
Generating response (/configure/channel | GET)

[2018-01-28 16:33:42,966 Thread=Server Worker: 1D0AFD5]
Finish (/configure/channel | GET)
```

configure/channels

Brief information regarding all cameras

Example of a request

```
GET http://192.168.100.50:8080/configure/channels/
```

Example of a response

```
[
  {
    "Id": "428d7aff-2e4a-46df-acff-0550cd827cd3",
    "Name": "Channel 1",
    "Disabled": false,
    "ServerBindingsSettings": {
      "OwnerServerId": "cd81195c-9c0e-42f0-8535-c55af9c35161",
      "ReservedServerId": "00000000-0000-0000-0000-000000000000",
      "ReplicationServerId": "00000000-0000-0000-0000-000000000000"
    },
    "ParentSecObjectId": "1f2c7319-9e87-46a2-be33-61bf2872b7df"
  },
  {
    "Id": "44afcd50-7b7c-4f91-89f5-5385acd66c96",
    "Name": "Channel 2",
    "Disabled": false,
    "ServerBindingsSettings": {
      "OwnerServerId": "9162138b-046a-48ed-82ed-6c6c7f6a52ba",
      "ReservedServerId": "00000000-0000-0000-0000-000000000000",
      "ReplicationServerId": "00000000-0000-0000-0000-000000000000"
    },
    "ParentSecObjectId": "25f867d0-92d5-4571-86da-403ea9458501"
  },
  {
    "Id": "56eeec10-c6ac-4257-8821-907dd85013ff",
    "Name": "Channel 3",
    "Disabled": false,
    "ServerBindingsSettings": {
```

```

    "OwnerServerId": "cd81195c-9c0e-42f0-8535-c55af9c35161",
    "ReservedServerId": "00000000-0000-0000-0000-000000000000",
    "ReplicationServerId": "00000000-0000-0000-0000-000000000000"
  },
  "ParentSecObjectId": "1f2c7319-9e87-46a2-be33-61bf2872b7df"
},
{
  "Id": "f6ffec63-3255-47de-85d6-99692ebace41",
  "Name": "Channel 4",
  "Disabled": false,
  "ServerBindingsSettings": {
    "OwnerServerId": "9162138b-046a-48ed-82ed-6c6c7f6a52ba",
    "ReservedServerId": "00000000-0000-0000-0000-000000000000",
    "ReplicationServerId": "00000000-0000-0000-0000-000000000000"
  },
  "ParentSecObjectId": "25f867d0-92d5-4571-86da-403ea9458501"
}
]

```

Adding new cameras

Example of a request

```
POST http://192.168.100.50:8080/configure/channels
```

The body of a POST request shall contain the array of JSON objects corresponding with the cameras being added (see the `configure/channels/{channel_id}` resource).

The description of the fields of a JSON object of the camera being added is as follows:

Field	Description
Name	Name of camera
Disabled	Indicates that camera is off

* ServerBindingsSettings	Camera binding to server parameters
* ServerBindingsSettings.OwnerServerId	ID of server where the camera archive is recorded. It is considered that the camera is bound to this server.
ServerBindingsSettings.ReservedServerId	ID of server to which the camera performs backup
ServerBindingsSettings.ReplicationServerId	ID of server to which the camera performs replication
ParentSecObjectId	ID of security object where the camera is located
Description	Camera description
* ConnectionSettings	Camera connection parameters
* ConnectionSettings.ModelId	Camera model. The value can be found in the configure/devicemanufacturers resource
ConnectionSettings.Hostname	Camera connection address
ConnectionSettings.Login	Camera connection login
ConnectionSettings.Password	Camera connection password
ConnectionSettings.MainVideoStreamFormat	Main stream video format: MJPEG, MPEG4_Part2, H264, Hvc
ConnectionSettings.AltVideoStreamEnabled	Indicates that alternative stream is enabled
ConnectionSettings.AltVideoStreamFormat	Alternative video stream format: MJPEG, MPEG4_Part2, H264, Hvc
ConnectionSettings.SoundReceivingEnabled	Indicates that sound reception is enabled

ConnectionSettings.PtzEnabled	Indicates that PTZ camera control is enabled
ConnectionSettings.Ipv6Enabled	Indicates that IPV6 is enabled
ConnectionSettings.PtzEnabled	Indicates that PTZ camera control is enabled
ConnectionSettings.MultiChannelServerEnabled	Indicates that the camera is a channel of a video server
ConnectionSettings.MultiChannelServerChannelNum	Video server channel number
ConnectionSettings.IoProcessingEnabled	Indicates that camera I/O processing is enabled
ConnectionSettings.SoundTransmittingEnabled	Indicates that transmission of sound to camera is enabled
ConnectionSettings.ExternalNetworkPorts	External network ports of the camera. Integrated camera port (DefaultPortValue) and external port (PortValue) are indicated. It is not allowed to modify the integrated port. When making a modification, it is required to indicate the integrated port and the modified external port.
ConnectionSettings.SoundVolume	Sound level. Value lies in the range of 0 to 100
ConnectionSettings.MainVideoAdditionalParams	Additional main stream connection parameters
ConnectionSettings.AltVideoAdditionalParams	Additional alternative stream connection parameters
ConnectionSettings.MainStreamRotationMode	Rotation of main stream when displaying in client: None, By90ClockwiseDegree, By90AntiClockwiseDegree, By180Degree. Note: it is not recommended to modify this field because it can cause malfunctions of motion detector and video analysis modules when rotating. Perform rotation in IP camera settings.

ConnectionSettings.AltStreamRotationMode	Rotation of alternative stream when displaying in client: None, By90ClockwiseDegree, By90AntiClockwiseDegree, By180Degree. Note: it is not recommended to modify this field because it can cause malfunctions of motion detector and video analysis modules when rotating. Perform rotation in IP camera settings.
ConnectionSettings.DirectNetworkConnectionToCameraEnabled	Indicates that the client application will connect to camera directly. Note: it is not recommended to modify this field because it can cause serious issues.
ConnectionSettings.GpuClientVideoDecodingEnabled	Indicates that decoding on video card is enabled in client application. Note: for decoding to operate on video card, it is also required to enable the corresponding option in the client application settings.
ConnectionSettings.AlternativeVideoDecoderEnabled	Indicates that alternative decoder is enabled for video stream decoding. Note: it is not recommended to modify this field because it can cause serious issues.
ConnectionSettings.StreamSettings.MainStreamSettings.VideoStreamEnabled	Whether or not the main stream is enabled. Read-only field, main stream cannot be disabled
ConnectionSettings.StreamSettings.MainStreamSettings.VideoFormat	Main stream video format: MJPEG, MPEG4_Part2, H264, Hvc
ConnectionSettings.StreamSettings.MainStreamSettings.VideoAdditionalParams	Additional connection options for the main stream

ConnectionSettings.StreamSettings.MainStreamSettings.RotationMode	Rotation of the main stream when displayed in the client: None, By90ClockwiseDegree, By90AntiClockwiseDegree, By180Degree. It is not recommended to change this field, because motion detection and video analytics modules may not work correctly when turning. Set the rotation of the stream in the settings of the IP camera.
ConnectionSettings.StreamSettings.MainStreamSettings.Roles	Roles of the main stream main stream roles: Archive stream is used for recording i archive, VideoAnalysis stream is used for video analysis.
ConnectionSettings.StreamSettings.AlternativeStreamSettings.[...]	Additional stream setting 1
ConnectionSettings.StreamSettings.SecondAlternativeStreamSettings.[...]	Additional stream setting 2
ConnectionSettings.StreamSettings.ThirdAlternativeStreamSettings.[...]	Additional stream setting 3
ArchiveSettings	Parameters of recording to archive
ArchiveSettings.ArchivingDisabled	Indicates that archiving is enabled
ArchiveSettings.ArchivingMode	Archiving mode: MDandManual – by detector, AlwaysOn - continuous, OnlyManual - manual, BySchedule – according to schedule
ArchiveSettings.ArchivingVideoStreamType	Video stream to be saved in the archive: Main, Alternative
ArchiveSettings.ArchivingMinDeepnessInDays	Store the archive during the specified time, no less than, days
ArchiveSettings.ArchivingMaxDeepnessInDays	Store the archive during the specified time, not exceeding, days
ArchiveSettings.ArchivingSoundDisabled	Indicates that the recording of sound to the archive is disabled

ArchiveSettings.PreMotionDetectionRecSecs	Number of seconds of recording to archive before motion (when recording to archive on motion is enabled). Value in seconds from 1 to 6
ArchiveSettings.PostMotionDetectionRecSecs	Number of seconds of recording to archive after the motion start (when recording to archive on motion start is enabled). Value is in seconds from 1 to 6
ArchiveSettings.LimitRecordFpsAndOnlyIndepFramesEnabled	Indicates that recording to archive frequency is limited
ArchiveSettings.LimitedRecordFps	Frames per second value for recording to archive when recording to archive frequency is enabled
ArchiveSettings.DeviceArchiveEnabled	Indicates that the access to archive of device (camera SD card) is enabled
ArchiveSettings.DeviceArchiveAutoSynchronizationEnabled	Indicates that the automatic synchronization of main archive and device archive (camera SD card) is enabled
AnalyzeSettings	Analysis parameters
AnalyzeSettings.MotionDetectorEnabled	Indicates that the motion detector is enabled
AnalyzeSettings.MotionDetectorSettings	Settings of the motion detector
[...].BuildInCameraDetectorEnabled	Indicates that the motion detector built into the camera is enabled
[...].GenerationOfEventMotionStartAndEndEnabled	Indicates that the generation of an event upon starting and ending of movement is enabled
[...].DetectOnlyIndependentFrames	Detection of key frames only. Note: this mode may significantly decrease the CPU load

[...].CustomFpsEnabled	Indication of motion detection with specified frequency. Note: this mode allows to detect motion with higher fps, but its activation may significantly increase CPU load
[...].CustomFpsValue	Frequency of detection of frames. Note: the value will only be used if CustomFpsEnabled field == true
[...].Zones	Detection zones
AnalyzeSettings.AnalyzeStreamType	Video stream for analysis: Main, Alternative. Note: enabling Main mode will lead to high CPU load
AnalyzeSettings.FaceDetectorEnabled	Indicates that the face recognition module is enabled. Note: enabling this mode will lead to high CPU load
AnalyzeSettings.FaceDetectorMode	Face recognition module quality settings: Basic, OptimizedQuality. Note: OptimizedQuality consumes more CPU resources
AnalyzeSettings.InteractiveSearchEnabled	Indicates that the interactive search is enabled. Note: enabling this mode will lead to high CPU load
AnalyzeSettings.InteractiveSearchSettings	Interactive search settings

* Mandatory field

Example of a POST request body for adding 2 new cameras bound to the cd81195c-9c0e-42f0-8535-c55af9c35161 server:

```
[
  {
    "Name": "New Camera 1",
    "ServerBindingsSettings": {
      "OwnerServerId": "cd81195c-9c0e-42f0-8535-c55af9c35161"
```

```

    },
    "ConnectionSettings": {
      "ModelId": "b499b4c5-4109-475d-8f08-341a5a5e70bd"
    }
  },
  {
    "Name": "New Camera 2",
    "ServerBindingsSettings": {
      "OwnerServerId": "cd81195c-9c0e-42f0-8535-c55af9c35161"
    },
    "ConnectionSettings": {
      "ModelId": "b499b4c5-4109-475d-8f08-341a5a5e70bd"
    }
  }
]

```

Example of a response

```

{
  "CreatedObjectsIds": [
    "aafb80e1-56a0-488d-9e99-8683d34fd37a",
    "7e2fc411-000b-4c12-b62e-1cd17f2f291e"
  ]
}

```

Note: the response contains a JSON object with the array of the IDs of the added cameras.

Modification of existing cameras

Example of a request

```
PUT http://192.168.100.50:8080/configure/channels
```

PUT request body must contain an array of JSON objects corresponding to the cameras being modified (see channels/{channel_id} resource).

Description of the JSON object of the camera being modified (see other fields in the Adding New Cameras section):

Field	Description
* Id	Camera ID

* Mandatory field; a request will not be successfully fulfilled without it.

Example of a body of PUT request for modification of the names of 2 cameras:

```
[
  {
    "Id" : "aafb80e1-56a0-488d-9e99-8683d34fd37a",
    "Name": "Renamed Camera 1",
  },
  {
    "Id" : "cd81195c-9c0e-42f0-8535-c55af9c35161",
    "Name": "Renamed Camera 2",
  }
]
```

Note: PUT request is used for modifying several cameras, but it modifies not the whole array of cameras but only the cameras for which the identifiers were specified (i.e. not all the existing cameras are modified). It is a deviation from REST, but it allows to conveniently modify several cameras at the same time.

configure/channels/{channel_id}

Detailed camera information

Example of a request

```
GET http://192.168.100.50:8080/configure/channels/7d69e586-25c6-470e-a368-5570d57b2631
```

Example of a response

```
{
  "Id": "b63d7868-9325-4b9e-9475-7a7668fa0719",
  "Name": "Camera Axis",
  "Disabled": false,
  "ServerBindingsSettings": {
    "OwnerServerId": "d1bb90ec-8a04-4af2-a8df-0ab1e5980ae6",
    "ReservedServerId": "00000000-0000-0000-0000-000000000000",
    "ReplicationServerId": "00000000-0000-0000-0000-000000000000"
  },
  "ParentSecObjectId": "934f0580-267a-4d3b-a0f6-0bfb48308f6f",
  "Description": "",
  "ConnectionSettings": {
    "ModelId": "b499b4c5-4109-475d-8f08-341a5a5e70bd",
    "Hostname": "127.0.0.1",
    "Login": null,
    "Password": null,
    "MainVideoStreamFormat": "H264",
    "AltVideoStreamEnabled": false,
    "AltVideoStreamFormat": "MJPEG",
    "SoundReceivingEnabled": false,
    "PtzEnabled": false,
    "Ipv6Enabled": false,
    "MultiChannelServerEnabled": false,
    "MultiChannelServerChannelNum": 1,
    "IoProcessingEnabled": false,
    "SoundTransmittingEnabled": false,
    "ExternalNetworkPorts": [
      {
```



```
        "DefaultPortValue": 554,
        "PortValue": 554,
        "IsExternallyDefined": false
    }
],
"SoundVolume": 100,
"MainVideoAdditionalParams": null,
"AltVideoAdditionalParams": null,
"MainStreamRotationMode": "None",
"AltStreamRotationMode": "None",
"DirectNetworkConnectionToCameraEnabled": false,
"GpuClientVideoDecodingEnabled": false,
"AlternativeVideoDecoderEnabled": false
},
"ArchiveSettings": {
    "ArchivingDisabled": true,
    "ArchivingMode": "AlwaysOn",
    "ArchivingVideoStreamType": "Main",
    "ArchivingMinDeepnessInDays": 7,
    "ArchivingMaxDeepnessInDays": 365,
    "ArchivingSoundDisabled": false,
    "PreMotionDetectionRecSecs": 1,
    "PostMotionDetectionRecSecs": 1,
    "LimitRecordFpsAndOnlyIndepFramesEnabled": false,
    "LimitedRecordFps": 30,
    "DeviceArchiveEnabled": false,
    "DeviceArchiveAutoSynchronizationEnabled": false
},
"AnalyzeSettings": {
    "MotionDetectorEnabled": true,
    "MotionDetectorSettings": {
        "BuildInCameraDetectorEnabled": false,
        "GenerationOfEventMotionStartAndEndEnabled": true,
        "DetectOnlyIndependentFrames": false,
        "CustomFpsEnabled": false,
        "CustomFpsValue": 3,
        "Zones": [
            {
                "Id": "3bc01441-41d5-4948-97ff-cb04f826f291",
```

```
        "Name": "Zone 5",
        "MinObjWidth": 0.08,
        "MinObjHeight": 0.08
    }
]
},
"AnalyzeStreamType": "Main",
"FaceDetectorEnabled": false,
"FaceDetectorMode": "Basic",
"InteractiveSearchEnabled": false,
"InteractiveSearchSettings": {
    "IndexatorEnabled": false,
    "IndexatorProportionsEnabled": false,
    "IndexatorMinObjWidth": 0.05,
    "IndexatorMinObjHeight": 0.05,
    "IndexatorMaxObjWidth": 1,
    "IndexatorMaxObjHeight": 1
}
}
```

Deletion of existing camera

Example of a request

```
DELETE http://192.168.100.50:8080/configure/channels/7e2fc411-000b-4c12-b62e-1cd17f2f291e
```

configure/channels/{channel_id}/scenarios

Description of camera scenarios

Example of a request

```
GET http://192.168.100.50:8080/configure/channels/7d69e586-25c6-470e-a368-5570d57b2631/scenarios
```

Example of a response

```
[
  {
    "EventType": "MotionFact",
    "Actions": [
      {
        "ActionType": "AlarmGeneration",
        "ActionBodyJson": "{\"RepeatIntervalMs\":7000}"
      }
    ]
  },
  {
    "EventType": "MotionBegin",
    "Actions": [
      {
        "ActionType": "ArchiveRecordingOn",
        "ActionBodyJson": "{\"RecordingDurationMs\":300000,\"RepeatIntervalMs\":5000}"
      }
    ]
  },
  {
    "EventType": "MotionEnd",
    "Actions": [
      {
        "ActionType": "ArchiveRecordingOn",
        "ActionBodyJson": "{\"RecordingDurationMs\":300000,\"RepeatIntervalMs\":5000}"
      }
    ]
  }
],
```

```
{
  "EventType": "CameraLostConnection",
  "Actions": [
    {
      "ActionType": "AlarmGeneration",
      "ActionBodyJson": "{\"RepeatIntervalMs\":5000}"
    }
  ]
},
{
  "EventType": "CameraEstablishConnection",
  "Actions": [
    {
      "ActionType": "AlarmGeneration",
      "ActionBodyJson": "{\"RepeatIntervalMs\":5000}"
    }
  ]
},
{
  "EventType": "CameraNoConnection",
  "Actions": [
    {
      "ActionType": "AlarmGeneration",
      "ActionBodyJson": "{\"RepeatIntervalMs\":5000}"
    },
    {
      "ActionType": "AlarmGeneration",
      "ActionBodyJson": "{\"RepeatIntervalMs\":5000}"
    }
  ]
},
{
  "EventType": "CameraInput",
  "Actions": [
    {
      "ActionType": "AlarmGeneration",
      "ActionBodyJson": "{\"RepeatIntervalMs\":5000}"
    }
  ]
}
```

```

    },
    {
      "EventType": "UserAlarm",
      "Actions": [
        {
          "ActionType": "AlarmGeneration",
          "ActionBodyJson": "{\"RepeatIntervalMs\":5000}"
        }
      ]
    }
  ]
}

```

Response consists of an array of JSON objects (scenarios) constituting an event (EventType) and the actions (Actions).

Description of events (EventType):

Parameter	Description
MotionFact	Movement detected in frame (event will be continuously generated until there is no movement in the frame)
MotionBegin	Commencement of movement in frame (no repeated generation of event will be performed until movement stops). Do not forget to enable the corresponding option responsible for generating this event in the motion detector settings.
MotionEnd	End of movement in frame (no repeated generation of event will be performed until movement stops). Do not forget to enable the corresponding option responsible for generating this event in the motion detector settings.
CameraLostConnection	Connection with camera is lost
CameraEstablishConnection	Connection with camera is restored
CameraNoConnection	Lasting absence of connection with camera
CameraInput	Signal on camera input detected

UserAlarm	User alarm detected (generated from client application)
------------------	---------------------------------------------------------

Description of actions (Actions):

Parameter	Description
ArchiveRecordingOn	Enabling manual recording to archive
ArchiveRecordingOff	Disabling manual recording to archive
SendingEmail	Sending email message
ExecutingCameraOutput	Setting value on camera output
AlarmGeneration	Alarm generation

Modification of camera scenarios

Example of a request

```
PUT http://192.168.100.50:8080/configure/channels/7d69e586-25c6-470e-a368-5570d57b2631/scenarios
```

PUT response must contain an array of JSON objects (scenarios, configure/channels/{channel_id}/scenarios) constituting an event (EventType) and the actions (Actions).

Description of events (EventType):

Parameter	Description
MotionFact	Movement is detected in frame (event will be generated continuously until there is movement in the frame)
MotionBegin	Commencement of movement in frame (no repeated generation of event will be performed until movement stops).
MotionEnd	End of movement in frame (no repeated generation of event will be performed until movement stops).
CameraLostConnection	Connection with camera lost
CameraEstablishConnection	Connection with camera restored
CameraNoConnection	Lasting absence of connection with camera
CameraInput	Signal on camera input detected
UserAlarm	User alarm detected (generated from client application)

Description of actions (Actions):

Parameter	Description
ArchiveRecordingOn	Enabling manual recording to archive
ArchiveRecordingOff	Disabling manual recording to archive
SendingEmail	Sending email message
ExecutingCameraOutput	Setting value on camera output
AlarmGeneration	Alarm generation

Example of a PUT request body that modifies the repeat interval of Alarm Generation event for Movement event type to 30000 milliseconds:

```
[
  {
    "EventType": "MotionFact",
    "Actions": [
      {
        "ActionType": "AlarmGeneration",
        "ActionBodyJson": "{\"RepeatIntervalMs\":30000}"
      }
    ]
  },
  {
    "EventType": "MotionBegin",
    "Actions": [
      {
        "ActionType": "ArchiveRecordingOn",
        "ActionBodyJson": "{\"RecordingDurationMs\":300000,\"RepeatIntervalMs\":5000}"
      }
    ]
  },
  {
    "EventType": "MotionEnd",
    "Actions": [
      {
        "ActionType": "ArchiveRecordingOn",
        "ActionBodyJson": "{\"RecordingDurationMs\":300000,\"RepeatIntervalMs\":5000}"
      }
    ]
  },
  {
    "EventType": "CameraLostConnection",
    "Actions": [
      {
        "ActionType": "AlarmGeneration",
        "ActionBodyJson": "{\"RepeatIntervalMs\":5000}"
      }
    ]
  },
  {

```



```
"EventType": "CameraEstablishConnection",
"Actions": [
  {
    "ActionType": "AlarmGeneration",
    "ActionBodyJson": "{\"RepeatIntervalMs\":5000}"
  }
],
},
{
  "EventType": "CameraNoConnection",
  "Actions": [
    {
      "ActionType": "AlarmGeneration",
      "ActionBodyJson": "{\"RepeatIntervalMs\":5000}"
    },
    {
      "ActionType": "AlarmGeneration",
      "ActionBodyJson": "{\"RepeatIntervalMs\":5000}"
    }
  ]
},
{
  "EventType": "CameraInput",
  "Actions": [
    {
      "ActionType": "AlarmGeneration",
      "ActionBodyJson": "{\"RepeatIntervalMs\":5000}"
    }
  ]
},
{
  "EventType": "UserAlarm",
  "Actions": [
    {
      "ActionType": "AlarmGeneration",
      "ActionBodyJson": "{\"RepeatIntervalMs\":5000}"
    }
  ]
}
```

Note: the PUT request is used for modifying multiple scenarios, and it modifies the whole array of scenarios (i.e. all the existing scenarios are modified). This behavior is different from modifying the `configure/channels/{channel_id}` resource.

configure/channels/{channel_id}/facescomplete

Appeared in version 3.4.

Receiving Face Recognition Complete module's settings

Example of request

```
GET http://192.168.100.50:8080/configure/channels/7d69e586-25c6-470e-a368-5570d57b2631/facescomplete
```

Example of response

```
{
  "GeneralSettings": {
    "DatabaseSettings": {
      "IsDedicated": false,
      "IpAddress": "127.0.0.1",
      "Port": 3050,
      "User": "SYSDBA",
      "Password": "masterkey",
      "OsType": "Windows"
    }
  },
  "ChannelSettings": {
    "DelayBetweenSameDetectionsSeconds": 5,
    "MinObjectSize": "0.1,0.1,0.06,0.06",
    "MaxObjectSize": "0.3,0.3,0.5,0.5",
    "UnknownFaceThreshold": 85.0,
    "FaceDetectorSensitivity": "Medium",
  }
}
```


Parameter	Description
Enabled	On or off state of the module True: the module is on False: the module is off
GeneralSettings	Common settings of the face recognition module that apply to all cameras with the module enabled
GeneralSettings.DatabaseSettings	Common settings of the module's database that apply to all the cameras with the module enabled
GeneralSettings.DatabaseSettings.IsDedicated	Remote or local database True: a remote database is used False: a local database is used Values of fields IPAddress; Port; User; Password; OsType must be entered if GeneralSettings.DatabaseSettings.IsDedicated is True
GeneralSettings.DatabaseSettings.IpAddress	Remote database server address, string
GeneralSettings.DatabaseSettings.Port	port, number
GeneralSettings.DatabaseSettings.User	User name, string
GeneralSettings.DatabaseSettings.Password	User password, string

GeneralSettings.DatabaseSettings.Os Type	Type of operating system, Windows or Linux string
ChannelSettings	Settings of the face recognition module specific to a specific camera
ChannelSettings.DelayBetweenSameDetectionsSeconds	Minimum time in seconds before the repeated recognition of a person (0, 5, 30, 60)
ChannelSettings.MinObjectSize	<p>Minimum size of a face in the frame, the rectangle defined by 4 digits in the interval of 0 to 1, "x, y, w, h"</p> <p>X: X axis position</p> <p>Y: Y axis position</p> <p>W: width</p> <p>H: height</p>
ChannelSettings.MaxObjectSize	<p>Maximum size of a face in the frame, the rectangle defined by 4 digits in the interval of 0 to 1, "x, y, w, h"</p> <p>X: X axis position</p> <p>Y: Y axis position</p> <p>W: width</p> <p>H: height</p>
ChannelSettings.UnknownFaceThreshold	Minimal likeness with the recognition sample in percentage points, the value in the range of 60 to 99
ChannelSettings.FaceDetectorSensitivity	Sensitivity of searching for faces in a frame, string, Minimum, Low, Medium, High, Maximum
ChannelSettings.DetectionMap	<p>Detection zone, mask array [40x40], 0 and 1 2-D array</p> <p>1: a pixel is used in detection</p>

	0: a pixel is not used in detection
ChannelSettings.DbId	<p>Database identifier, string</p> <p>Empty string: common database usage</p> <p>Non-empty string: unique database usage (it is possible to specify the same identifier for several cameras; in this case, they will interact with the same database)</p> <p>The identifier must be no more than 32-symbol long, contain Latin characters, digits and underscore (_), otherwise no unique database will be set</p>
ChannelSettings.RecognizeCoveredFaces	<p>Necessity to attempt to recognize faces covered by face masks or kerchiefs/scarfs.</p> <p>True: covered faces will be detected and recognized in case of matching</p> <p>False: covered faces will be detected but will never be recognized</p>
ChannelSettings.RecognizeRotatedFaces	<p>Necessity to attempt to recognize faces turned to more than ~30 from the face-forward position</p> <p>True: turned faces will be detected and recognized in case of matching</p> <p>False: turned faces will be detected but will never be recognized</p>

Example of PUT request body for disabling the module:

```
{
  "GeneralSettings": {
    "DatabaseSettings": {
      "IsDedicated": false,
      "IpAddress": "127.0.0.1",
      "Port": 3050,
      "User": "SYSDBA",
      "Password": "masterkey",
      "OsType": "Windows"
    }
  },
  "ChannelSettings": {
    "DelayBetweenSameDetectionsSeconds": 5,
```

[illegible]

configure/channels/{channel_id}/peoplecounter

Since version 3.4, new fields have been added.

Acquiring settings of visitor counting module

Example of a request

```
GET http://192.168.100.50:8080/configure/channels/7d69e586-25c6-470e-a368-5570d57b2631/peoplecounter
```

Example of a response

```
{
  "Enabled": true,
  "CountingMethod": "MultiplePeople",
  "ObjWidth": 0.3,
  "ObjHeight": 0.3,
  "LineBeginX": 0.05,
  "LineBeginY": 0.5,
  "LineEndX": 0.95,
  "LineEndY": 0.5,
  "CountingInversedDirection": false,
  "UseChannelResetCounterSettings": false,
  "ZeroCountingTimeUTC": "2021-04-28T11:24:30.8956469Z",
  "IntervalInHours": 0,
  "PeopleCountingVariant": "InOut",
  "Sensitivity": "Medium",
  "ProcessFullFrame": false,
}
```

Modifying settings of visitor counting module

Example of a request

```
PUT http://192.168.100.50:8080/configure/channels/7d69e586-25c6-470e-a368-5570d57b2631/peoplecounter
```

Description of resource parameters

Parameter	Description	Version
Enabled	Indicates that the module is enabled. Note: make sure that the motion detector is on. It is required for the correct operation of the module.	2.5
CountingMethod	Counting method: SinglePeople – for counting individual people, MultiplePeople – for counting people in streams. Note: for the MultiplePeople method, the camera must be placed vertically to face the floor.	2.5
ObjWidth	Width of object in frame. Value is in the range of 0 to 1	2.5
ObjHeight	Height of object in frame. Value is in the range of 0 to 1	2.5
LineBeginX	Initial point of visitor counting line of the X coordinate. Value is in the range of 0 to 1	2.5
LineBeginY	Initial point of visitor counting line of the Y coordinate. Value is in the range of 0 to 1	2.5
LineEndX	End of visitor counting line of the X coordinate. Value in the range of 0 to 1	2.5
LineEndY	End of visitor counting line of the Y coordinate. Value in the range of 0 to 1	2.5
CountingInversedDirection	Inverting counting direction	2.5
UseChannelResetCounterSettings	set unique counter reset settings true - unique settings false - does not change camera reset settings	3.4
ZeroCountingTimeUTC	initial reset time, date in UTC format, changed if UseChannelResetCounterSettings is passed true	3.4
IntervalInHours	reset period in hours, number, 0, 12, 24, 168, changed if UseChannelResetCounterSettings is passed true	3.4
PeopleCountingVariant	visitor counting option, string	3.4

	InOut - incoming and outgoing InOnly - incoming OutOnly - outgoing	
Sensitivity	algorithm sensitivity, string, Minimum, Low, Medium, High, Maximum	3.4
ProcessFullFrame	process full frame	3.4

Example of the PUT request body for disabling the module:

```
{
  "Enabled": false,
  "CountingMethod": "MultiplePeople",
  "ObjWidth": 0.3,
  "ObjHeight": 0.3,
  "LineBeginX": 0.05,
  "LineBeginY": 0.5,
  "LineEndX": 0.95,
  "LineEndY": 0.5,
  "CountingInversedDirection": false,
  "UseChannelResetCounterSettings": false,
  "ZeroCountingTimeUTC": "2021-04-02T10:34:59.6615578Z",
  "IntervalInHours": 0,
  "PeopleCountingVariant": "InOut",
  "Sensitivity": "Medium",
  "ProcessFullFrame": false,
}
```

configure/channels/{channel_id}/visitors

Appeared in version 3.4.

Receiving Unique Visitor Counting module's settings

Example of request

```
GET http://192.168.100.50:8080/configure/channels/7d69e586-25c6-470e-a368-5570d57b2631/visitors
```

Example of response

```
{
  "GeneralSettings": {
    "DatabaseSettings": {
      "IsDedicated": false,
      "IpAddress": "127.0.0.1",
      "Port": 3050,
      "User": "SYSDBA",
      "Password": "masterkey",
      "OsType": "Windows"
    }
  },
  "ChannelSettings": {
    "DelayBetweenSameDetectionsSeconds": 5,
    "MinObjectSize": "0.1,0.1,0.06,0.06",
    "MaxObjectSize": "0.3,0.3,0.5,0.5",
    "UnknownFaceThreshold": 85.0,
    "FaceDetectorSensitivity": "Medium",
  }
}
```


Parameter	Description
Enabled	On or off state of the module True: the module is on False: the module is off
GeneralSettings	Common settings of the Unique Visitor Counting module, applied to all cameras with the module enabled
GeneralSettings.DatabaseSettings	Common settings of the module's database applied to all cameras with the module enabled
GeneralSettings.DatabaseSettings.IsDedicated	Remote or local database used True: a remote database is used False: a local database is used Values of the fields IPAddress; Port; User; Password; OsType are to be entered if GeneralSettings.DatabaseSettings.IsDedicated is True
GeneralSettings.DatabaseSettings.IPAddress	Remote database server address, string
GeneralSettings.DatabaseSettings.Port	port, number
GeneralSettings.DatabaseSettings.User	User name, string
GeneralSettings.DatabaseSettings.Password	User password, string
GeneralSettings.DatabaseSettings.OsType	Type of operating system, Windows or Linux string

ChannelSettings	Settings of the Unique Visitor Counting module specific to a specific camera
ChannelSettings.DelayBetweenSameDetection sSeconds	Minimum time in seconds before the repeated recognition of a person: 0, 5, 30, 60
ChannelSettings.MinObjectSize	Minimum size of a face in the frame, the rectangle defined by 4 digits x, y, w, h X: X axis position Y: Y axis position W: width H: height
ChannelSettings.MaxObjectSize	Minimum size of a face in the frame, the rectangle defined by 4 digits x, y, w, h X: X axis position Y: Y axis position W: width H: height
ChannelSettings.UnknownTreshold	Minimal likeness with the recognition sample in percentage points, the value in the range of 0 to 100
ChannelSettings.FaceDetectorSensitivity	Sensitivity of searching for faces in a frame, string, Minimum, Low, Medium, High, Maximum
ChannelSettings.DetectionMap	Detection zone, mask array [40x40], 0 and 1 2-D array 1: a pixel is used in detection 0: a pixel is not used in detection
ChannelSettings.DbId	Database identifier, string Empty string: common database usage

	<p>Non-empty string: unique database usage (it is possible to specify the same identifier for several cameras; in this case, they will interact with the same database)</p> <p>The identifier must be no more than 32-symbol long, contain Latin characters, digits and underscore (_), otherwise no unique database will be set</p>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Example of PUT request body for disabling the module:

```
{
  "GeneralSettings": {
    "DatabaseSettings": {
      "IsDedicated": false,
      "IpAddress": "127.0.0.1",
      "Port": 3050,
      "User": "SYSDBA",
      "Password": "masterkey",
      "OsType": "Windows"
    }
  },
  "ChannelSettings": {
    "DelayBetweenSameDetectionsSeconds": 5,
    "MinObjectSize": "0.1,0.1,0.06,0.06",
    "MaxObjectSize": "0.3,0.3,0.5,0.5",
    "UnknownFaceThreshold": 85.0,
    "FaceDetectorSensitivity": "Medium",
  }
}
```


configure/channels/{channel_id}/queuecounter

Appeared in version 3.4.

Receiving People Counting in Queue module's setting

Example of request

```
GET http://192.168.100.50:8080/configure/channels/7d69e586-25c6-470e-a368-5570d57b2631/queuecounter
```

Example of response

```
{
  "ChannelSettings": {
    "DetectionZones": [
      {
        "Contour": {
          "Name": "Area 1",
          "Id": "55f42d44-c772-4d81-8937-a76bfa1be739",
          "Points": [
            "0.25,0.25",
            "0.75,0.25",
            "0.75,0.75",
            "0.25,0.75"
          ]
        },
        "ShouldAlertOnMaxCount": false,
        "MaxAllowedCount": 0,
        "Sensitivity": "Medium",
        "SmoothingMode": "Interval10Sec",
        "Id": "55f42d44-c772-4d81-8937-a76bfa1be739",
        "Name": "Area 1"
      },
      {
        "Contour": {
          "Name": "Area 2",
          "Id": "2663d439-6713-4a0e-a168-7c3c573c14a1",
          "Points": [
```

```

        "0.25,0.25",
        "0.75,0.25",
        "0.75,0.75",
        "0.25,0.75"
    ],
    },
    "ShouldAlertOnMaxCount": false,
    "MaxAllowedCount": 0,
    "Sensitivity": "Medium",
    "SmoothingMode": "Interval10Sec",
    "Id": "2663d439-6713-4a0e-a168-7c3c573c14a1",
    "Name": "Area 2"
}
    ]
},
"Enabled": true
}

```

Modifying settings of the People Counting in Queue module

Example of request

```
PUT http://192.168.100.50:8080/configure/channels/7d69e586-25c6-470e-a368-5570d57b2631/queuecounter
```

Description of resource parameters

Parameter	Description
Enabled	<p>On or off state of the module</p> <p>True: the module is on</p> <p>False: the module is off</p>

ChannelSettings.DetectionZones	List of detections zones with settings of each zone Settings of each zone included into ChannelSettings.DetectionZones[i] are indicated further (see example of response to a GET query)
Contour.Name	Name of zone, string
Contour.Id	Zone identifier, string
Contour.Points	Coordinates of rectangle defining the zone in the following format: ["x1, y1", "x2, y2", "x3, y3", "x4, y4"], where x and y are numbers in the interval of 0 to 1
ShouldAlertOnMaxCount	Generation of an alarm upon exceedance of the preset number of people in the queue True: generate False: not generate
MaxAllowedCount	Maximum allowable number of people in the zone, number
Sensitivity	Algorithm sensitivity, string, Minimum, Low, Medium, High, Maximum
SmoothingMode	Averaging of results, string, None, Interval10Sec, Interval15Sec, Interval30Sec, Interval60Sec None: no averaging Interval10Sec: averaging of results for 10 seconds Interval15Sec: averaging of results for 15 seconds Interval30Sec: averaging of results for 30 seconds Interval60Sec: averaging of results for 60 seconds

Example of the PUT request body for disabling the module:

```
{
  "ChannelSettings": {
    "DetectionZones": [
      {
```

```
    "Contour": {
      "Name": "Area 1",
      "Id": "55f42d44-c772-4d81-8937-a76bfa1be739",
      "Points": [
        "0.25,0.25",
        "0.75,0.25",
        "0.75,0.75",
        "0.25,0.75"
      ]
    },
    "ShouldAlertOnMaxCount": false,
    "MaxAllowedCount": 0,
    "Sensitivity": "Medium",
    "SmoothingMode": "Interval10Sec",
    "Id": "55f42d44-c772-4d81-8937-a76bfa1be739",
    "Name": "Area 1"
  }
]
},
"Enabled": false
}
```

configure/channels/{channel_id}/platescomplete

Appeared in version 3.4.

Receiving License Plate Recognition Complete module's settings

Example of request

```
GET http://192.168.100.50:8080/configure/channels/7d69e586-25c6-470e-a368-5570d57b2631/platescomplete
```

Example of response

```
{  
  "ChannelSettings": {  
    "DbId": "123"  
  },  
  "Enabled": false  
}
```

Modifying settings of the License Plate Recognition Complete module

Example of request

```
PUT http://192.168.100.50:8080/configure/channels/7d69e586-25c6-470e-a368-5570d57b2631/platescomplete
```

Description of resource parameters

Parameter	Description
Enabled	On or off state of the module True: the module is on False: the module is off
ChannelSettings.DbId	Database identifier, string Empty string: common database usage Non-empty string: unique database usage (it is possible to specify the same identifier for several cameras; in this case, they will interact with the same database) The identifier must be no more than 32-symbol long, contain Latin characters, digits and underscore (_), otherwise no unique database will be set

Example of the PUT request body for disabling the module:

```
{
  "ChannelSettings": {
    "DbId": "123"
  },
  "Enabled": false
}
```

configure/secobjects

Root security object

Returns the description of a root security object and all the inner security objects. Also, the representation of the objects will include a brief description of the associated cameras.

Example of a request

```
GET http://192.168.100.50:8080/configure/secobjects/
```

Example of a response

```
{
  "Id": "934f0580-267a-4d3b-a0f6-0bfb48308f6f",
  "ParentId": "00000000-0000-0000-0000-000000000000",
  "Name": null,
  "ChildObjects": [
    {
      "Id": "601c8ff9-8a3a-4fd8-a19c-dfaa1729cac4",
      "ParentId": "934f0580-267a-4d3b-a0f6-0bfb48308f6f",
      "Name": "Object 1",
      "ChildObjects": [
        {
          "Id": "7169e12e-5a50-41d0-bd7f-bd57f5d1f042",
          "ParentId": "601c8ff9-8a3a-4fd8-a19c-dfaa1729cac4",
          "Name": "Object 6",
          "ChildObjects": [],
          "ChildChannels": [
            {
              "Id": "7d69e586-25c6-470e-a368-5570d57b2631",
              "Name": "Channel 1"
            }
          ]
        }
      ]
    }
  ],
  "ChildChannels": []
},
```



```
{
  {
    "Id": "cc586375-1f7e-49ca-845e-ebc1c2d95f78",
    "ParentId": "934f0580-267a-4d3b-a0f6-0bfb48308f6f",
    "Name": "Object 3",
    "ChildObjects": [],
    "ChildChannels": []
  }
},
"ChildChannels": [
  {
    "Id": "5d1722d5-5c02-4ddf-afe5-bbf5873a31ca",
    "Name": "Channel 3"
  }
]
}
```

Note: the root (allocated, the first) security object has the ParentId value: "00000000-0000-0000-0000-000000000000" and "Name": null, and these values cannot be modified.

Adding new security objects

Example of a response

```
POST http://192.168.100.50:8080/configure/secobjects
```

POST request body must contain JSON objects corresponding with the security objects (see `configure/secobjects/{secobject_id}` resource).

Description of a JSON object:

Parameter	Description
ParentId	ID of parent security object. If the field is left empty, the security object will be added to the root security object.
Name	Name of security object
ChildObjects	Inner security objects. Specify the existing security objects here. If the security object is embedded into another security object at the moment of fulfilling the request, it will be transferred to the object being created. The objects contained within the ChildObjects array are similar to the JSON object under consideration.
ChildChannels	Embedded cameras. Specify the existing cameras here. If the camera is embedded into another security object, it will be transferred to the object being created. See the description of JSON objects contained within the ChildChannels array below.

Description of the objects contained in the ChildChannels array:

Parameter	Description
* Id	ID of camera contained in the related security object

* Mandatory parameter

Example of the POST request that adds 2 security objects; at that, the second security object will contain 1 embedded security object and 1 embedded camera:

```
[
  {
    "Name": "New object 1"
  },
  {
    "Name": "New object 2",
    "ChildObjects": [
      {
        "Id": "7169e12e-5a50-41d0-bd7f-bd57f5d1f042",
```

```
        },
      ],
      "ChildChannels": [
        {
          "Id": "7d69e586-25c6-470e-a368-5570d57b2631",
        }
      ]
    }
  ]
}
```

Example of a response

```
{
  "CreatedObjectsIds": [
    "ec1baec4-88f4-4704-b8e6-7327a3b3ffaf",
    "8a1851cc-a66a-483d-9170-3893a808be3a"
  ]
}
```

Modifying existing security objects

Example of a request

```
PUT http://192.168.100.50:8080/configure/secobjects
```

PUT request body must contain JSON objects that correspond to the security objects being modified (see `configure/secobjects/{secobject_id}` resource).

JSON object description:

Parameter	Description
ParentId	ID of parent security object. If this field is left blank, the parent security object will not be modified.
Name	Name of security object
ChildObjects	Embedded security objects. Specify the existing security objects here. If the security object is embedded into another security object at the moment of fulfilling the request, it will be transferred. The objects contained within the ChildObjects array are similar to the JSON object under consideration.
ChildChannels	Embedded cameras. Specify the existing cameras here. If the camera is embedded into another security object, it will be transferred. See the description of JSON objects contained within the ChildChannels array below.

Description of objects contained in the ChildChannels array:

Parameter	Description
* Id	ID of camera contained within the corresponding security object

* Mandatory parameter

Example of the PUT request modifying 2 security objects:

```
[
  {
    "Name": "Renamed object 1"
  },
  {
    "Name": "Renamed object 2",
    "ChildObjects": [
      {
        "Id": "7169e12e-5a50-41d0-bd7f-bd57f5d1f042",
      }
    ],
  },
]
```

```
    "ChildChannels": [  
        {  
            "Id": "7d69e586-25c6-470e-a368-5570d57b2631",  
        }  
    ]  
}
```

Note: PUT request is used for modifying several security objects, but it modifies not the whole array of security objects but only those for which the identifiers were specified (i.e. not all the existing security objects are modified). It is a deviation from REST, but it allows to conveniently modify several security objects at the same time.

configure/secobjects/{secobject_id}

Selected security objects

Returns a description of the selected and all the inner security objects. Also, the representation of the objects will include a brief description of the associated cameras.

Example of a request

```
GET http://192.168.100.50:8080/configure/secobjects/a78db714-69f3-4fec-9792-a708cb88c3ac
```

Example of a response

```
{
  "Id": "a78db714-69f3-4fec-9792-a708cb88c3ac",
  "ParentId": "934f0580-267a-4d3b-a0f6-0bfb48308f6f",
  "Name": null,
  "ChildObjects": [
    {
      "Id": "7169e12e-5a50-41d0-bd7f-bd57f5d1f042",
      "ParentId": "a78db714-69f3-4fec-9792-a708cb88c3ac",
      "Name": "Object 6",
      "ChildObjects": [
        {
          "Id": "ea9d5762-0ebb-4e22-bf73-62ec8089a0a6",
          "ParentId": "7169e12e-5a50-41d0-bd7f-bd57f5d1f042",
          "Name": "allyourbasearebelongtous",
          "ChildObjects": [],
          "ChildChannels": []
        }
      ],
      "ChildChannels": []
    }
  ],
  "ChildChannels": [
    {
      "Id": "7d69e586-25c6-470e-a368-5570d57b2631",
      "Name": "Channel 1"
    }
  ]
}
```

```
}  
  1  
}
```

Deleting existing security object

Example of a request

```
DELETE http://192.168.100.50:8080/configure/secobjects/0f8dd2e1-ffcc-4764-b008-23e494ff5adf
```

Note: deleting a security object will also delete all the embedded cameras and security objects.

configure/groups

Brief description of all groups

Example of a request

```
GET http://192.168.100.50:8080/configure/groups
```

Example of a response

```
[
  {
    "Id": "a03f6f5b-7f43-4537-89ea-77eee1e2d15a",
    "Name": "Senior Administrators ",
    "ConfiguringType": "SeniorAdmin"
  },
  {
    "Id": "4a553dc0-660f-4153-8d7b-6a99e6492b0d",
    "Name": "Junior Administrators",
    "ConfiguringType": "JuniorAdmin"
  },
  {
    "Id": "f0254188-79b1-4b87-a670-6c95dbd24f96",
    "Name": "Operators",
    "ConfiguringType": "Operator"
  }
]
```

Adding new groups

Example of a request

```
POST http://192.168.100.50:8080/configure/groups
```


POST request body must contain the array of JSON objects corresponding to the groups being added (see [configure/groups](#) resource).

Description of a JSON object of the group being added:

Parameter	Description
* Name	Name of group
* ConfiguringType	Group type: SeniorAdmin – senior administrator (this group is embedded, it holds all rights, and it is impossible to delete it or create a second one); JuniorAdmin – junior administrator (holds rights to configure parts of the system), Operator – a standard group with no rights to configure the system.
ConfiguringJuniorAdminOptions	Settings of the rights to configure the system. It is required to specify this field for junior administrators only (ConfiguringType == JuniorAdmin)
WorkplaceOptions	Client application settings. It is required to specify this field for junior administrators and operators. See the object description in the configure/groups/{group_id} resource.
ChannelsAccessOptions	Settings of access to cameras. It is required to specify this field only for junior administrators and operators. See the object description in the configure/groups/{group_id} resource.
ChannelsAccessOptions.AllowedOnePermissionForRealtimeAndArchiveEnabled	If the value of this field is “true”, the archive access rights will be the same as the rights for real time access. The cameras will not be considered in the list of the archive access rights. The value of the field is “true” by default.
ChannelsAccessOptions.AccessToNewCamerasEnabled	The rights of access to the newly added cameras. It is set for operators only. For junior administrators, this value is always “false”.
WebAndMobileOptions	Settings of access to mobile applications and web client. It is required to specify this field for junior administrators and operators only. See object description on the configure/groups/{group_id} resource.
Comment	Comment regarding a group

* Mandatory field

Example of a POST request body that adds Admins and Users groups:

```
[
  {
    "Name": "Admins",
    "ConfiguringType": "JuniorAdmin"
  },
  {
    "Name": "Users",
    "ConfiguringType": "Operator"
  }
]
```

Example of a response

```
{
  "CreatedObjectsIds": [
    "0369a2ce-9f18-4666-9160-dc3f8223a5ab",
    "6cba6b02-ca33-4e22-8bbd-072493d82c86"
  ]
}
```

Note: the Senior Administrators group has full and unalterable access. It is impossible to add another Senior Administrators type group.

Modifying existing groups

Example of a request

```
PUT http://192.168.100.50:8080/configure/groups
```

PUT request body must contain an array of JSON objects corresponding with the groups being modified (see the `configure/groups` resource).

Description of a JSON object of the group being added:

Parameter	Description
* Id	Unique identifier of the group
Name	Name of group
ConfiguringType	Type of group: SeniorAdmin – senior administrator (this group is embedded, it has all rights and it is impossible to delete it or create a second one); JuniorAdmin – junior administrator (it has the rights for configuring parts of the system); Operator is a standard group that cannot perform configuring of the system.
ConfiguringJuniorAdminOptions	Settings of system configuring rights. It is required to specify this field for junior administrators only (ConfiguringType == JuniorAdmin)
WorkplaceOptions	Settings of client application. It is required to specify this field for junior administrators and operators only. See configure/groups/{group_id} resource for object description.
ChannelsAccessOptions	Camera access settings. It is required to specify this field for junior administrators and operators only. See the configure/groups/{group_id} resource for object description.
ChannelsAccessOptions.AllowedOnePermissionForRealtimeAndArchiveEnabled	If the value of this field is “true”, the archive access rights will be the same as the rights for real time access. The cameras will not be considered in the list of the archive access rights. The value of the field is “true” by default.
ChannelsAccessOptions.AccessToNewCameraEnabled	Rights of access to the newly added cameras. It is set for operators only. This value is always “false” for junior administrators.
WebAndMobileOptions	Settings of access to mobile applications and web client. This value is to be specified for junior administrators and operators only. See configure/groups/{group_id} resource for object description.
Comment	Comment for group

* Mandatory field

Example of a PUT request body modifying Admins and Users groups into NonAdmins and NonUsers groups:

```
[
  {
    "Id": "0369a2ce-9f18-4666-9160-dc3f8223a5ab",
    "Name": "NonAdmins",
  },
  {
    "Id": "6cba6b02-ca33-4e22-8bbd-072493d82c86"
    "Name": "NonUsers",
  }
]
```

Note: a PUT request is used for modifying multiple groups, but it modifies not the whole array of groups but only those for which the identifiers were specified. It is a deviation from REST, but it allows to conveniently modify several groups at the same time.

Note: the Senior Administrators group has full and unalterable access, that is why it is impossible to modify the settings of this group.

configure/groups/{group_id}

Detailed group description

Example of a request

```
GET http://192.168.100.50:8080/configure/groups/a03f6f5b-7f43-4537-89ea-77eee1e2d15a
```

Example of a response for senior administrator:

```
{
  "Id": "a03f6f5b-7f43-4537-89ea-77eee1e2d15a",
  "Name": "Senior Administrators ",
  "ConfiguringType": "SeniorAdmin",
  "ConfiguringJuniorAdminOptions": null,
  "WorkplaceOptions": null,
  "ChannelsAccessOptions": null,
  "WebAndMobileOptions": null,
  "Comment": null
}
```

Example of a response for junior administrator:

```
{
  "Id": "4a553dc0-660f-4153-8d7b-6a99e6492b0d",
  "Name": "JunAmin",
  "ConfiguringType": "JuniorAdmin",
  "ConfiguringJuniorAdminOptions": {
    "ConfigureChannelsEnabled": true,
    "ConfigureAddChannelsEnabled": true,
    "ConfigureModifyChannelsEnabled": true,
    "ConfigureServersEnabled": true,
    "ConfigureRemoveChannelsEnabled": true,
    "ConfigureChannelsAllowed": [
      "7d69e586-25c6-470e-a368-5570d57b2631",
      "5d1722d5-5c02-4ddf-afe5-bbf5873a31ca"
    ]
  },
}
```

```
    "ConfigureVideowallEnabled": false
  },
  "WorkplaceOptions": {
    "ConfigurationEnabled": true,
    "ShutdownEnabled": true,
    "ChangeChannelModeEnabledEnabled": true,
    "ManageRecEnabled": true,
    "AccessExpertModeEnabled": true,
    "PtzEnabled": true,
    "ReceiveSoundEnabled": true,
    "TransmitSoundEnabled": true,
    "ExportVideoEnabled": true,
    "ExportVideoToAviEnabled": true,
    "ReceiveMainStreamEnabled": true,
    "UnifiedLogEnabled": true,
    "UnifiedLogAccessToAllUsersEnabled": true,
    "UnifiedLogForbiddenEventTypes": "None",
    "ArchiveMarksEnabled": true,
    "InteractiveSearchEnabled": true,
    "ReportsEnabled": true,
    "AnalystPluginsEditingEnabled": true,
    "PlansEnabled": true,
    "VideowallBrowsingEnabled": false,
    "ChatReceiveMessagesEnabled": false,
    "ChatSendMessageEnabled": false,
    "ChangePasswordEnabled": false,
    "DelayedReloadClientEnabled": false,
    "IsWatermarkEnabled": false,
    "WatermarkDisplayMode": 1,
    "PtzPriority": 0,
    "GridTypesAllowed": [
      "GridType1",
      "GridType2",
      "GridType4",
      "GridType1x4",
      "GridType6",
      "GridType3x2",
      "GridType2X4",
      "GridType7",
```

```
        "GridType8",
        "GridType4x2",
        "GridType9",
        "GridType10",
        "GridType12",
        "GridType13",
        "GridType16",
        "GridType17",
        "GridType20",
        "GridType24",
        "GridType25",
        "GridType30",
        "GridType36",
        "GridType64",
        "GridType100",
        "GridType110",
        "GridType12X11",
        "GridType256",
        "GridType262"
    ],
    "AnaliticModulesForbidden": [],
    "PlansForbidden": []
},
"ChannelsAccessOptions": {
    "AllowedOnePermissionForRealtimeAndArchiveEnabled": true,
    "ChannelsRealtimeAllowed": [
        "7d69e586-25c6-470e-a368-5570d57b2631",
        "5d1722d5-5c02-4ddf-afe5-bbf5873a31ca"
    ],
    "ChannelsArchiveAllowed": [
        "7d69e586-25c6-470e-a368-5570d57b2631",
        "5d1722d5-5c02-4ddf-afe5-bbf5873a31ca"
    ],
    "AccessToNewCameraEnabled": false,
},
"WebAndMobileOptions": {
    "MobilePushReceivingEnabled": true,
    "TranscodedVideoFromMobileServerEnabled": true,
    "VideoViaWebEnabled": true
}
```

```
    },  
    "Comment": null  
}
```

Example of a response for operator:

```
{  
  "Id": "f0254188-79b1-4b87-a670-6c95dbd24f96",  
  "Name": "User",  
  "ConfiguringType": "Operator",  
  "ConfiguringJuniorAdminOptions": null,  
  "WorkplaceOptions": {  
    "ConfigurationEnabled": true,  
    "ShutdownEnabled": true,  
    "ChangeChannelModeEnabled": true,  
    "ManageRecEnabled": true,  
    "AccessExpertModeEnabled": true,  
    "PtzEnabled": true,  
    "ReceiveSoundEnabled": true,  
    "TransmitSoundEnabled": true,  
    "ExportVideoEnabled": true,  
    "ExportVideoToAviEnabled": true,  
    "ReceiveMainStreamEnabled": true,  
    "UnifiedLogEnabled": true,  
    "UnifiedLogAccessToAllUsersEnabled": true,  
    "UnifiedLogForbiddenEventTypes": "None",  
    "ArchiveMarksEnabled": true,  
    "InteractiveSearchEnabled": true,  
    "ReportsEnabled": true,  
    "AnalystPluginsEditingEnabled": true,  
    "PlansEnabled": true,  
    "VideowallBrowsingEnabled": false,  
    "ChatReceiveMessagesEnabled": false,  
    "ChatSendMessageEnabled": false,  
    "ChangePasswordEnabled": false,  
    "DelayedReloadClientEnabled": false,  
    "IsWatermarkEnabled": true,  
    "WatermarkDisplayMode": 2,  
    "PtzPriority": 0,  
    "GridTypesAllowed": [  

```



```
    "GridType1",
    "GridType2",
    "GridType4",
    "GridType1x4",
    "GridType6",
    "GridType3x2",
    "GridType2X4",
    "GridType7",
    "GridType8",
    "GridType4x2",
    "GridType9",
    "GridType10",
    "GridType12",
    "GridType13",
    "GridType16",
    "GridType17",
    "GridType20",
    "GridType24",
    "GridType25",
    "GridType30",
    "GridType36",
    "GridType64",
    "GridType100",
    "GridType110",
    "GridType12X11",
    "GridType256",
    "GridType262"
  ],
  "AnaliticModulesForbidden": [],
  "PlansForbidden": []
},
"ChannelsAccessOptions": {
  "AllowedOnePermissionForRealtimeAndArchiveEnabled": true,
  "ChannelsRealtimeAllowed": [
    "7d69e586-25c6-470e-a368-5570d57b2631",
    "5d1722d5-5c02-4ddf-afe5-bbf5873a31ca"
  ],
  "ChannelsArchiveAllowed": [
    "7d69e586-25c6-470e-a368-5570d57b2631",
```

```
        "5d1722d5-5c02-4ddf-afe5-bbf5873a31ca"
    ],
    "AccessToNewCameraEnabled": true,
  },
  "WebAndMobileOptions": {
    "MobilePushReceivingEnabled": true,
    "TranscodedVideoFromMobileServerEnabled": true,
    "VideoViaWebEnabled": true
  },
  "Comment": null
}
```

Deleting an existing group

Example of a request

```
DELETE http://192.168.100.50:8080/configure/groups/8eeab264-274a-43f3-b2ee-1ea041ef5e98
```

Note: deleting a group deletes all the users of this group.

configure/groups/{group_id}/users

Brief description of the group's users

Example of a request

```
GET http://192.168.100.50:8080/configure/groups/a03f6f5b-7f43-4537-89ea-77eee1e2d15a/users
```

Example of a response

```
[
  {
    "Id": "4dbb3d27-51a4-4a94-92e3-fbe3a522431b",
    "GroupId": "a03f6f5b-7f43-4537-89ea-77eee1e2d15a",
    "Login": "root"
  }
]
```

configure/users

Brief description of all users

Example of a request

```
GET http://192.168.100.50:8080/configure/users
```

Example of a response

```
[
  {
    "Id": "4dbb3d27-51a4-4a94-92e3-fbe3a522431b",
    "GroupId": "a03f6f5b-7f43-4537-89ea-77eee1e2d15a",
    "Login": "root"
  },
  {
    "Id": "a7357e79-32d7-449c-b0e6-b563a45398c4",
    "GroupId": "4a553dc0-660f-4153-8d7b-6a99e6492b0d",
    "Login": "usr"
  }
]
```

Adding new users

Note: ActiveDirectory of users cannot be added using REST API.

Example of a request

```
POST http://192.168.100.50:8080/configure/users
```

POST request body must contain an array of JSON objects corresponding to the users being added (see `configure/users/{user_id}` resource).

Description of a JSON object of the user being added:

Parameter	Description
* GroupId	User group ID
* Login	User name
NewPassword	User password. If this field is not specified, the default password will be the empty one.
UserBlocked	Indicates that the account is blocked
Comment	Comment to user

* Mandatory field

Example of a POST request body for adding a user called Petrov:

```
[
  {
    "GroupId": "beaa06e0-45a2-4fff-97fa-ac120f284368",
    "Login": "Petrov"
  }
]
```

Example of a response

```
{
  "CreatedObjectsIds": [
    "96c4c285-5111-4789-a342-12985deb47d6"
  ]
}
```

Modifying existing users

Example of a request

PUT <http://192.168.100.50:8080/configure/users>

PUT response body must contain JSON objects corresponding to the `configure/users/{user_id}` resource and obligatorily containing user ID and other information being modified.

PUT request body must contain an array of JSON objects corresponding to the users being modified (see `configure/users/{user_id}` resource).

Description of a JSON object of the user being modified:

Parameter	Description
* Id	User ID
GroupId	User group ID. If this field is not filled, the group will not be modified.
Login	User name. Note: it is impossible to modify names for ActiveDirectory users.
NewPassword	New user password
UserBlocked	Indicates that the account is blocked
Comment	Comment to user

* Mandatory field

Example of a PUT request body for renaming user called Petrov into PetrovJunior:

```
[
  {
    "Id": "96c4c285-5111-4789-a342-12985deb47d6",
    "Login": "PetrovJunior"
  }
]
```

Note: a PUT request is used for modifying multiple users, but it modifies not the whole array of users but only those for which the identifiers were specified. It is a deviation from REST, but it allows to conveniently modify several users at the same time.

configure/users/{user_id}

Detailed user description

Example of a request

```
GET http://192.168.100.50:8080/configure/users/4dbb3d27-51a4-4a94-92e3-fbe3a522431b
```

Example of a response

```
{
  "Id": "4dbb3d27-51a4-4a94-92e3-fbe3a522431b",
  "GroupId": "a03f6f5b-7f43-4537-89ea-77eee1e2d15a",
  "Login": "root",
  "UserType": "Standart",
  "UserBlocked": false,
  "Comment": null
}
```

Note: UserType field has Standart value for ordinary users and ActiveDirectory value for ActiveDirectory users.

Deleting existing user

Example of a request

```
DELETE http://192.168.100.50:8080/configure/users/bd22f81d-ce66-43b7-a6a1-57cef8b6bf03
```


configure/viewprofiles

Brief description of all profiles

Example of a request

```
GET http://192.168.100.50:8080/configure/viewprofiles
```

Example of a response

```
[
  {
    "Id": "ddf047bf-70d1-482d-b1fa-fdc38207e148",
    "Name": "New profile 1",
    "GridType": "GridType6",
    "ChannelsPositions": [
      {
        "GridIndex": 0,
        "ChannelId": "501bd18f-6b09-491f-a1f4-8d5545f7f0fa"
      },
      {
        "GridIndex": 2,
        "ChannelId": "405c1bbd-f0d5-4a3f-b8de-e238e021f6c3"
      }
    ]
  },
  {
    "Id": "ccf047bf-70d1-482d-b1fa-fdc38207e148",
    "Name": "New profile 222",
    "GridType": "GridType6",
    "ChannelsPositions": [
      {
        "GridIndex": 0,
        "ChannelId": "501bd18f-6b09-491f-a1f4-8d5545f7f0fa"
      },
      {
        "GridIndex": 1,
        "ChannelId": "405c1bbd-f0d5-4a3f-b8de-e238e021f6c3"
      }
    ]
  }
]
```

```
]
},
{
  "Id": "a932fbdd-bbb6-40fa-9b7f-bf27dd2067de",
  "Name": "New profile 223",
  "GridType": "GridType6",
  "ChannelsPositions": [
    {
      "GridIndex": 0,
      "ChannelId": "501bd18f-6b09-491f-a1f4-8d5545f7f0fa"
    },
    {
      "GridIndex": 1,
      "ChannelId": "405c1bbd-f0d5-4a3f-b8de-e238e021f6c3"
    }
  ]
}
]
```

Adding new profiles

Example of a request

POST <http://192.168.100.50:8080/configure/viewprofiles>

POST request body must contain an array of JSON objects corresponding to the profiles being added (see the [configure/viewprofiles/{viewprofile_id}](#) resource).

Description of a JSON object of the profile being added:

Parameter	Description
Name	Name of profile
GridType	Type of grid. It can have one of the following values: GridType1, GridType2, GridType4, GridType3x2, GridType6, GridType7, GridType4x2, GridType8, GridType9, GridType10, GridType12, GridType13, GridType16, GridType17, GridType20, GridType24, GridType25, GridType30, GridType36, GridType64, GridType100, GridType110, GridType1x4, GridType256, GridType262, GridType2X4, GridType12X11
ChannelsPositions	The array that determines a cell where the certain camera is located.

Description of JSON objects contained in the ChannelsPositions array:

Parameter	Description
GridIndex	Index in grid (reckoned from the upper left corner), begins with 0
ChannelId	ID of camera located in corresponding cell

Example of a POST request body for adding two profiles:

```
[
  {
    "Name": "Main Profile",
    "GridType": "GridType6",
    "ChannelsPositions": [
      {
        "GridIndex": 0,
        "ChannelId": "501bd18f-6b09-491f-a1f4-8d5545f7f0fa"
      },
      {
        "GridIndex": 2,
```

```

        "ChannelId": "405c1bbd-f0d5-4a3f-b8de-e238e021f6c3"
    }
  ]},
  {
    "GridType": "GridType16",
    "ChannelsPositions": [
      {
        "GridIndex": 0,
        "ChannelId": "501bd18f-6b09-491f-a1f4-8d5545f7f0fa"
      },
      {
        "GridIndex": 2,
        "ChannelId": "405c1bbd-f0d5-4a3f-b8de-e238e021f6c3"
      }
    ]
  }
]

```

Example of a response

```

{
  "CreatedObjectsIds": [
    "781f76ac-0995-46a3-92c1-5d960edb85c4",
    "0689209c-68af-4b57-b2a3-bcb9cf2e0906"
  ]
}

```

Modifying existing profiles

Example of a request

```
PUT http://192.168.100.50:8080/configure/viewprofiles
```

PUT request body must contain JSON objects corresponding with the `configure/viewprofiles/{viewprofile_id}` resource and mandatorily containing the profile's ID and other information to be modified.

POST request body must contain an array of JSON objects corresponding to the profiles being added (see the `configure/viewprofiles/{viewprofile_id}` resource).

Description of a JSON object of the profile being added:

Parameter	Description
* Id	Profile ID
Name	Name of profile
GridType	Type of grid. It can have one of the following values: GridType1, GridType2, GridType4, GridType3x2, GridType6, GridType7, GridType4x2, GridType8, GridType9, GridType10, GridType12, GridType13, GridType16, GridType17, GridType20, GridType24, GridType25, GridType30, GridType36, GridType64, GridType100, GridType110, GridType1x4, GridType256, GridType262, GridType2X4, GridType12X11
ChannelsPositions	The array that determines a cell where the certain camera is located.

* Mandatory field

Description of JSON objects contained in the ChannelsPositions array:

Parameter	Description
GridIndex	Index in grid (reckoned from the upper left corner), begins with 0
ChannelId	ID of camera located in corresponding cell

Example of a PUT request where the profile is renamed and the location of cameras in the profile is modified:

```
[
  {
    "Id": "781f76ac-0995-46a3-92c1-5d960edb85c4",
    "Name": "Renamed Profile",
    "ChannelsPositions": [
```

```
    {
      "GridIndex": 3,
      "ChannelId": "501bd18f-6b09-491f-a1f4-8d5545f7f0fa"
    },
    {
      "GridIndex": 4,
      "ChannelId": "405c1bbd-f0d5-4a3f-b8de-e238e021f6c3"
    }
  ]
}
```

Note: a PUT request is used for modifying multiple profiles, but it modifies not the whole array of profiles but only those for which the identifiers were specified. It is a deviation from REST, but it allows to conveniently modify several profiles at the same time.

configure/viewprofiles/{viewprofile_id}

Profile information

Example of a request

```
GET http://192.168.100.50:8080/configure/viewprofiles/ddf047bf-70d1-482d-b1fa-fdc38207e148
```

Example of a response

```
{
  "Id": "ddf047bf-70d1-482d-b1fa-fdc38207e148",
  "Name": "New profile 1",
  "GridType": "GridType6",
  "ChannelsPositions": [
    {
      "GridIndex": 0,
      "ChannelId": "501bd18f-6b09-491f-a1f4-8d5545f7f0fa"
    },
    {
      "GridIndex": 2,
      "ChannelId": "405c1bbd-f0d5-4a3f-b8de-e238e021f6c3"
    }
  ]
}
```

Deleting an existing profile

Example of a request

```
DELETE http://192.168.100.50:8080/configure/viewprofiles/781f76ac-0995-46a3-92c1-5d960edb85c4
```

configure/servers

Brief information about all servers

Example of a request

```
GET http://192.168.100.50:8080/configure/servers
```

Example of a response

```
[
  {
    "Id": "cd81195c-9c0e-42f0-8535-c55af9c35161",
    "Name": "Server 1",
    "Hostname": "192.168.100.50",
    "Port": 8080,
    "IsReplicationServer": false,
    "UseFastReservation": false,
    "TotalLicParams": {
      "Channels": 400,
      "SoundChannels": 400,
      "PtzChannels": 400,
      "FaceRecognitionChannels": 16,
      "PlateRecognCompleteChannels": 400,
      "PlateRecognLightChannels": 0,
      "PeopleCounterChannels": 400,
      "ReservedChannels": 400
    },
    "UsedLicParams": {
      "Channels": 2,
      "SoundChannels": 0,
      "PtzChannels": 0,
      "FaceRecognitionChannels": 0,
      "PlateRecognCompleteChannels": 0,
      "PlateRecognLightChannels": 0,
      "PeopleCounterChannels": 0,
      "ReservedChannels": 0
    }
  },
]
```



```
{
  "Id": "9162138b-046a-48ed-82ed-6c6c7f6a52ba",
  "Name": "Server 2",
  "Hostname": "10.1.1.86",
  "Port": 8081,
  "IsReplicationServer": false,
  "UseFastReservation": false,
  "TotalLicParams": {
    "Channels": 400,
    "SoundChannels": 400,
    "PtzChannels": 400,
    "FaceRecognitionChannels": 16,
    "PlateRecognCompleteChannels": 400,
    "PlateRecognLightChannels": 0,
    "PeopleCounterChannels": 400,
    "ReservedChannels": 400
  },
  "UsedLicParams": {
    "Channels": 2,
    "SoundChannels": 0,
    "PtzChannels": 0,
    "FaceRecognitionChannels": 0,
    "PlateRecognCompleteChannels": 0,
    "PlateRecognLightChannels": 0,
    "PeopleCounterChannels": 0,
    "ReservedChannels": 0
  }
}
1
```

configure/servers/{server_id}

Detailed server information

Example of a request

```
GET http://192.168.100.50:8080/configure/servers/cd81195c-9c0e-42f0-8535-c55af9c35161
```

Example of a response

```
{
  "Id": "cd81195c-9c0e-42f0-8535-c55af9c35161",
  "Name": "Server 1",
  "Hostname": "192.168.100.50",
  "Port": 8080,
  "IsReplicationServer": false,
  "UseFastReservation": false,
  "TotalLicParams": {
    "Channels": 400,
    "SoundChannels": 400,
    "PtzChannels": 400,
    "FaceRecognitionChannels": 16,
    "PlateRecognCompleteChannels": 400,
    "PlateRecognLightChannels": 0,
    "ReservedChannels": 400
  },
  "UsedLicParams": {
    "Channels": 2,
    "SoundChannels": 0,
    "PtzChannels": 0,
    "FaceRecognitionChannels": 0,
    "PlateRecognCompleteChannels": 0,
    "PlateRecognLightChannels": 0,
    "ReservedChannels": 0
  }
}
```

configure/servers/{server_id}/channels

Brief information regarding cameras of the server

Example of a request

```
GET http://192.168.100.50:8080/configure/servers/cd81195c-9c0e-42f0-8535-c55af9c35161/channels
```

Example of a response

```
[
  {
    "Id": "428d7aff-2e4a-46df-acff-0550cd827cd3",
    "Name": "Channel 1",
    "Disabled": false,
    "ServerBindingsSettings": {
      "OwnerServerId": "cd81195c-9c0e-42f0-8535-c55af9c35161",
      "ReservedServerId": "00000000-0000-0000-0000-000000000000",
      "ReplicationServerId": "00000000-0000-0000-0000-000000000000"
    },
    "ParentSecObjectId": "1f2c7319-9e87-46a2-be33-61bf2872b7df"
  },
  {
    "Id": "56eeec10-c6ac-4257-8821-907dd85013ff",
    "Name": "Channel 3",
    "Disabled": false,
    "ServerBindingsSettings": {
      "OwnerServerId": "cd81195c-9c0e-42f0-8535-c55af9c35161",
      "ReservedServerId": "00000000-0000-0000-0000-000000000000",
      "ReplicationServerId": "00000000-0000-0000-0000-000000000000"
    },
    "ParentSecObjectId": "1f2c7319-9e87-46a2-be33-61bf2872b7df"
  }
]
```

configure/devicemanufacturers

Supported camera models

Below is the abridged version of the response, with only the beginning of the list of camera models. The full response contains the complete list of camera models.

Example of a request

```
GET http://192.168.100.50:8080/configure/devicemanufacturers
```

Example of a response

```
[
  {
    "Name": "3S",
    "DeviceModels": [
      {
        "Id": "4c2c3083-21d9-4f32-ae8e-42903ea64241",
        "Name": "N10xx",
        "AltStreamSupported": true,
        "MultiChannelServerSupported": false,
        "DeviceArchiveSupported": false,
        "PtzSupported": false,
        "SoundReceivingSupported": true,
        "SoundTransmittingSupported": false,
        "IoProcessingEnabledSupported": true,
        "MainVideoStreamFormatsSupported": [
          "MJPEG",
          "H264"
        ],
        "AltVideoStreamFormatsSupported": [
          "MJPEG",
          "H264"
        ],
        "DefaultExternalNetworkPorts": [
          554
        ]
      }
    ]
  }
]
```

```
]
},
{
  "Name": "ABRON",
  "DeviceModels": [
    {
      "Id": "d5d62eba-a314-4fd0-a9c4-b29ae7d0f356",
      "Name": "ABC-i(2xx,4xx,6xx)(P,VP,VRP)",
      "AltStreamSupported": true,
      "MultiChannelServerSupported": false,
      "DeviceArchiveSupported": false,
      "PtzSupported": false,
      "SoundReceivingSupported": true,
      "SoundTransmittingSupported": true,
      "IoProcessingEnabledSupported": true,
      "MainVideoStreamFormatsSupported": [
        "H264",
        "MJPEG"
      ],
      "AltVideoStreamFormatsSupported": [
        "H264",
        "MJPEG"
      ],
      "DefaultExternalNetworkPorts": []
    },
    {
      "Id": "ebe511c5-31fd-48d9-bd0d-00a0ef22a9e2",
      "Name": "ABC-i(31x,41x,61x)(FR,FP,VP,VRP)",
      "AltStreamSupported": true,
      "MultiChannelServerSupported": false,
      "DeviceArchiveSupported": false,
      "PtzSupported": false,
      "SoundReceivingSupported": false,
      "SoundTransmittingSupported": false,
      "IoProcessingEnabledSupported": false,
      "MainVideoStreamFormatsSupported": [
        "MPEG4_Part2",
        "H264",
        "MJPEG"
      ]
    }
  ]
}
```

```

    ],
    "AltVideoStreamFormatsSupported": [
        "MPEG4_Part2",
        "H264",
        "MJPEG"
    ],
    "DefaultExternalNetworkPorts": []
},
{
    "Id": "2d21b065-bb0c-4bff-8096-25feb89d5093",
    "Name": "ABR",
    "AltStreamSupported": true,
    "MultiChannelServerSupported": true,
    "DeviceArchiveSupported": false,
    "PtzSupported": false,
    "SoundReceivingSupported": true,
    "SoundTransmittingSupported": false,
    "IoProcessingEnabledSupported": false,
    "MainVideoStreamFormatsSupported": [
        "H264"
    ],
    "AltVideoStreamFormatsSupported": [
        "H264"
    ],
    "DefaultExternalNetworkPorts": [
        554
    ]
}
]
},
...
]

```

configure/monitoring

Monitoring settings information

Example of a request

```
GET http://192.168.100.50:8080/configure/monitoring
```

Example of a response

```
{
  "MonitoringEnabled": "true",
  "Password": "",
  "MonitoringUrl": "http://192.168.100.115:8889",
  "MonitoringNewServerEnabled": true,
  "MonitoringPushIntervalSeconds": 5,
  "ServerIds": [
    "fa289c64-05fc-4b1c-a367-727f02578c07"
  ]
  "ProxySettings":
  {
    "ProxyServerEnabled": "true",
    "ProxyServerId": "fa289c64-05fc-4b1c-a367-727f02578c07"
  }
}
```

Modifying monitoring settings

Example of a request

```
PUT http://192.168.100.50:8080/configure/monitoring
```

Description of a JSON object monitoring setting:

Parameter		Description	Type
*Password		monitoring server password	string
*MonitoringUrl		server monitoring address	string
*MonitoringNewServerEnabled		server list adding mode (true - new servers are automatically added, false - new servers are not added for monitoring).	bool
*ServerIds		list of server IDs to monitor	array
MonitoringEnabled		monitoring enabled or not	bool
*MonitoringPushIntervalSeconds		interval for sending information from the monitoring agent to the monitoring server in milliseconds. Possible values are 2, 5, 10, 30, 45 seconds. (Default interval will be 2 seconds)	int
ProxySettings	ProxyServerEnabled*	monitoring data proxying enabled or not	bool
	ProxyServerId*	ID of server that is used to proxy monitoring data	Guid

* Mandatory field

Example of a request body for modifying monitoring settings:

```
{
  "MonitoringEnabled": "true",
  "Password": "",
  "MonitoringUrl": "http://192.168.100.115:8889",
  "MonitoringNewServerEnabled": true,
  "MonitoringPushIntervalSeconds": 5,
  "ServerIds": [
    "fa289c64-05fc-4b1c-a367-727f02578c07"
  ]
}
```



```
]
"ProxySettings":
{
  "ProxyServerEnabled": "true",
  "ProxyServerId": "fa289c64-05fc-4b1c-a367-727f02578c07"
}
}
```

api/channels

Brief information regarding all cameras bound to server

Example of a request

```
GET http://192.168.100.50:8080/configure/channels/
```

Example of a response

```
[
  {
    "Id": "428d7aff-2e4a-46df-acff-0550cd827cd3",
    "Name": "Channel 1",
    "Disabled": false,
    "ServerBindingsSettings": {
      "OwnerServerId": "cd81195c-9c0e-42f0-8535-c55af9c35161",
      "ReservedServerId": "00000000-0000-0000-0000-000000000000",
      "ReplicationServerId": "00000000-0000-0000-0000-000000000000"
    },
    "ParentSecObjectId": "1f2c7319-9e87-46a2-be33-61bf2872b7df"
  },
  {
    "Id": "44afcd50-7b7c-4f91-89f5-5385acd66c96",
    "Name": "Channel 2",
    "Disabled": false,
    "ServerBindingsSettings": {
      "OwnerServerId": "9162138b-046a-48ed-82ed-6c6c7f6a52ba",
      "ReservedServerId": "00000000-0000-0000-0000-000000000000",
      "ReplicationServerId": "00000000-0000-0000-0000-000000000000"
    },
    "ParentSecObjectId": "25f867d0-92d5-4571-86da-403ea9458501"
  },
  {
    "Id": "56eeec10-c6ac-4257-8821-907dd85013ff",
    "Name": "Channel 3",
    "Disabled": false,
    "ServerBindingsSettings": {
      "OwnerServerId": "cd81195c-9c0e-42f0-8535-c55af9c35161",
```

```
        "ReservedServerId": "00000000-0000-0000-0000-000000000000",
        "ReplicationServerId": "00000000-0000-0000-0000-000000000000"
    },
    "ParentSecObjectId": "1f2c7319-9e87-46a2-be33-61bf2872b7df"
},
{
    "Id": "f6ffec63-3255-47de-85d6-99692ebace41",
    "Name": "Channel 4",
    "Disabled": false,
    "ServerBindingsSettings": {
        "OwnerServerId": "9162138b-046a-48ed-82ed-6c6c7f6a52ba",
        "ReservedServerId": "00000000-0000-0000-0000-000000000000",
        "ReplicationServerId": "00000000-0000-0000-0000-000000000000"
    },
    "ParentSecObjectId": "25f867d0-92d5-4571-86da-403ea9458501"
}
```

```
]
```

api/channels/{channel_id}

Brief camera information

Example of a request

```
GET http://192.168.100.50:8080/api/channels/7d69e586-25c6-470e-a368-5570d57b2631
```

Example of a response

```
{
  "Id": "7d69e586-25c6-470e-a368-5570d57b2631",
  "Name": "Channel 1",
  "Disabled": false,
  "ServerBindingsSettings": {
    "OwnerServerId": "cd81195c-9c0e-42f0-8535-c55af9c35161",
    "ReservedServerId": "00000000-0000-0000-0000-000000000000",
    "ReplicationServerId": "00000000-0000-0000-0000-000000000000"
  },
  "ParentSecObjectId": "1f2c7319-9e87-46a2-be33-61bf2872b7df"
}
```

api/channels/{channel_id}/status

Camera operating capacity status (reception bitrate, recording bitrate, issued detected)

Camera operating capacity status corresponds with the status gathered by the ULTRA Monitoring. The status can be obtained even when the ULTRA Monitoring is disabled.

Example of a request

```
GET http://192.168.100.50:8080/api/channels/7d69e586-25c6-470e-a368-5570d57b2631/status
```

Example of a response

```
{
  "Id": "7d69e586-25c6-470e-a368-5570d57b2631",
  "Name": "Camera 1",
  "AltEnabled": false,
  "ReceiveVideoMbps": 0.8812103271484375,
  "ReceiveAltVideoMbps": 0,
  "WriteVideoMbps": 0,
  "Status": "Ok"
}
```

Description of a JSON object of the response:

Parameter	Description
Id	Camera ID
Name	Name of camera
AltEnabled	Indicates that alternative stream is enabled on camera
ReceiveVideoMbps	Main stream reception bitrate, Mbps
ReceiveAltVideoMbps	Alternative stream reception bitrate, Mbps

WriteVideoMbps	Bitrate of recording to archive, Mbps
StreamsStatuses	Status by camera streams
Status	Status of issues detected on camera: OK – no issues, ErrorMain – issues with main stream, ErrorAlt –issues with alternative stream, ErrorSound – issues with sound, Wait – status not determined yet

REST API for face recognition

Starting from Version 1.7 of Eocortex Ultra, the REST API for working with Eocortex face database is available. All the main operations with the database are supported. The term “face” means an entry regarding a person in the Eocortex face database.

A possibility to perform synchronization of the databases of Eocortex and external software has been added. The common element unifying the Eocortex database and the external application database is the `external_id` field. A 36-symbol string is used for filling out this field. It is also possible to use the `id` field for the identification of the database objects instead of using the `external_id` external identifier. The `id` field is the internal identifier of the Eocortex face database.

The external ids can be specified for both faces and groups. The internal identifier cannot be modified.

The work with the following face recognition modules is supported:

- Eocortex Light
- Eocortex Complete

Requirements

For editing a face database, it is required (PUT, POST, DELETE requests) that all the requests be fulfilled by a user with the right to edit the data of the intelligent modules.

For Eocortex Ultra, the user on behalf of whom the requests are being made must have access to the module for any query to the face database of a particular module. Additionally, API is not available for the users with Operator rights in Eocortex Ultra.

Requirements for the photos to be uploaded

- The image must contain a single face located in the center of the frame.
- The image must be square or close to being square.
- The image must be in color.
- The height of a face must be at least 25% of the height of the image.
- The facial expression must be neutral, eyes open, mouth closed.
- There must be free space all around the face (the facial contours must not touch the edges).



The module lets you add entries to the face database with images that do not meet the requirements of the module, or without any of them at all. Such images will be marked with the icon in the database and will not be used by the module as a recognition sample. To add a record with an unrecognizable sample, set the *force* parameter in the query body to *true*.

GET /api/faceconfig

To obtain the list of all the available face recognition modules

Example of a request

```
http://localhost:8080/api/faceconfig
```

Example of a response

```
{
  "faces_modules": [
    {
      "name": "basic",
      "enabled": true
    },
    {
      "name": "expert",
      "enabled": false
    }
  ]
}
```

GET /api/faces

To obtain the list of all the faces from the database

Example of a request

```
http://localhost:8080/api/faces?offset=0&portion=50&module=complete
```

Parameters

Parameter	Description
offset	Non-mandatory parameter. It specifies the number of a face starting from which it is needed to obtain the list of faces. The default value is 0.
portion	Non-mandatory parameter. The number of faces that needs to be obtained. The default value is 1000, the maximum value equals the number of faces in the database.
module	Mandatory parameter. The name of the face recognition module to work with. The list of the available modules can be viewed using the /api/faceconfig request.
filter	<p><attribute_name> <logical comparison operator> '<assumed value>', where</p> <ul style="list-style-type: none"><attribute_name> - name of attribute (field) of entity. external_id and group_id are available;<logical comparison operator> - '=' operator available for searching for the entrance to the collections;'<assumed value>' – ID value; <p><u>Important: the parameter must correspond with the URL encoding</u></p> <p>Example of requests with filtration:</p> <p>http://localhost:8080/api/faces?offset=0&portion=50&module=basic&filter=external_id='123'</p> <p>http://localhost:8080/api/faces?offset=0&portion=50&module=basic&filter=</p> <p>The filters can be combined using AND operator, e.g. search for people belonging to several groups:</p> <p>http://localhost:8080/api/faces?offset=0&portion=50&module=basic&filter=group_id='05a21e32-51e5-405e-b4cf-3082796c8703' AND group_id='2090779b-9091-4989-a5f8-92d1920d8291'</p>

Example of displaying in URL encoding:

http://localhost:8080/api/faces?offset=0&portion=50&module=basic&filter=group_id%3D%2705a21e32-51e5-405e-b4cf-3082796c8703%27

Example of a response

```
{
  "offset": 0,
  "portion": 50,
  "total_count": 153,
  "faces": [
    {
      "id": "d1bd0147-e3d7-4bcd-a19c-e9a1acafaadd",
      "external_id": "0",
      "first_name": "John",
      "patronymic": "B.",
      "second_name": "Smith",
      "additional_info": "\"Sacred Motors\" CEO",
      "modification_time": "2019-05-28T05:58:17.994Z"
    },
    {
      "id": "59e5994d-85a5-4ca8-b4b5-1b660bb45e2a",
      "external_id": "1",
      "first_name": "William",
      "patronymic": "W.",
      "second_name": "Woodman",
      "additional_info": "",
      "modification_time": "2019-06-13T22:07:41.045Z"
    }
  ]
}
```



modification_time is the UTC time when a face was added or updated in the database.

POST /api/faces

To add a new face to the database. The response includes full description of the face.

Example of a request

```
http://localhost:8080/api/faces?module=complete
```

Parameters

Parameter	Description
module	Mandatory parameter. Name of face recognition module to work with. List of available modules can be obtained using /api/faceconfig request.

Headers

Parameter	Description
Content-Type	application/json

Body (application/json)

```
{
  "external_id" : "0",
  "first_name": "John",
  "patronymic": "B.",
  "second_name": "Smith",
  "additional_info": "\"Sacred Motors\" CEO",
  "groups": [
    {
      "id": "4cf2c1bb-707a-4455-a00b-c7d114eae96f"
```



```


"/9j/4AAQSkZJRgABAQEASABIAAD/4QAIhXhpZgAATU0AKgAAAAgAAQESAAMAAAABAAEAAAAAAAD/2wBDAAIBAQIBAQICAgICAgICAwUDAwMDAwYEBAMFBwYHBwGBwICQs
JCAGKCAcHCg0KCgsMDAwMBwkODw0MDgsMDAz/2wBDAQICAgMDAwYDAwYMCacIDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAz/wAA
RCABGAE8DASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRV
S0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMX
Gx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJ
BUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqO
kpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn60nq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwDqvs1Hk/u6uf8ALOmRw+bX3B8eMt4qk8r5PuU6OLyqP/R
dAE3/AC0+5Tk6VDUKXagBr9ag/wBVJVnyv++qJYv3lAFZ4vNqtLFmSrkv3Kh8mgC0nWm0UUGgv1fvKmjiipnm+9PjloMxH6UJ0ofpQnSgAz+8qT7J+7pn1fWn/AMSAAniaPopo
VGygChL/rKmji/d1NJFSJ0oAb/rY6Z5X0qaPZ5aUSRUGgyiSKnwf6xqfJL+8oMxn1fWnxxebRF9ypo4vKjoAIu1Q3H+tp8f+sp8kX7ugCnHvqxUUKVD76AK1v8AwJVyOoY7W
rX/ACzoAdHsqGSL95T44v3lPnoAI/3VP8r60W/9Kl0AbF9+mS/6ymW/7qSPO/3u+gBn1fWnvF+7qbyamji/d0AYsYkL/eqxRRQAU5+1FFACxxc1LRRQAPh1c/xU3/1pRRQBN
L3p6y+YKKKAP//Z"
  ]
}

```



The *face_images* array must contain at least one photo. All the other fields may be empty or *null*. The photos CAN be transferred in the following formats: jpg, png and bmp. The photos MUST be coded in base64 string.



The module lets you add entries to the face database with images that do not meet the requirements of the module, or without any of them at all. Such images will be marked with the  icon in the database and will not be used by the module as a recognition sample. To add a record with an unrecognizable sample, set the *force* parameter in the query body to *true*.

GET /api/faces/<id>

To obtain the detailed description of a face together with the images from the database coded in base64

Example of a request

```
http://localhost:8080/api/faces/d1bd0147-e3d7-4bcd-a19c-e9a1acafaadd?module=complete&onlymainsample=true
```

Parameters

Parameter	Description
module	Mandatory parameter. Name of face recognition module to work with. List of available modules can be obtained using /api/faceconfig request.
onlymainsample	Non-mandatory parameter. Allows to download a reference image of a face only. Values available: true and false

Example of a response

```
{
  "id": "d1bd0147-e3d7-4bcd-a19c-e9a1acafaadd",
  "external_id": "0",
  "first_name": "John",
  "patronymic": "B.",
  "second_name": "Smith",
  "additional_info": "\"Sacred Motors\" CEO",
  "modification_time": "2019-05-28T05:58:17.994Z",
  "groups": [
    "4cf2c1bb-707a-4455-a00b-c7d114eae96f"
  ],
  "face_images": [
```

```
"/9j/4AAQSkZJRgABAQEASABIAAD/4QAIhXhpZgAATU0AKgAAAAgAAQESAAMAAAABAAEAAAAAAD/2wBDAAIBAQIBAQICAgICAgICAwUDAwMDAwYEBAMFBwYHBwcGBwcICQsJCAgKCACgHCg0KCGsMDAwMBWkODw0MDgsMDAz/2wBDAQICAgMDAwYDAwYMCACIDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAz/wAARCABGAE8DASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn60nq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwDqvs1Hk/u6uf8ALOmRw+bX3B8eMt4qk8r5PuU6OLyqP/RdAE3/AC0+5Tk6VDUKXagBr9ag/wBVJVnyv++qJYv3lAFZ4vNqtLFmSrkv3Kh8mgC0nWm0UUGgv1fvKmjiipnm+9PjloMxH6UJ0ofpQnSgAz+8qT7J+7pn1fWn/AMGygChL/rKmji/d1NJFSJ0oAb/rY6Z5X0qaPZ5aUSRUGgyiSKnwf6xqfJL+8oMxn1fWnxxebRF9ypo4vKjoAIu1Q3H+tp8f+sp8kX7ugCnHvqxUUKVD76AK1v8AwJVyOoY7WrX/ACzoAdHsqGSL95T44v3lPnoAI/3VP8r60W/9Kl0AbF9+mS/6ymW/7qSpo/3u+gBn1fWnVf+7qbyamji/d0AYsYkL/eqxRRQAU5+1FFACxxc1LRRQAPH1c/xU3/1pRRQBNL3p6y+YKKKAP//Z"
]
```



modification_time is the UTC time when a face was added or updated in the database.

PUT /api/faces/<id>

To update an existing face in the database. Full description is returned in response.

Example of a request

```
http://localhost:8080/api/faces/d1bd0147-e3d7-4bcd-a19c-e9a1acafaadd?module=complete
```

Parameters

Parameter	Description
module	Mandatory parameter. Name of face recognition module to work with. List of available modules can be obtained using /api/faceconfig request.

Headers

Parameter	Description
Content-Type	application/json

Body (application/json)

```
{
  "external_id" : "3",
  "first_name" : "John",
  "patronymic" : "B.",
  "second_name" : "Smith",
  "groups" : [
    {
      "id" : "4cf2c1bb-707a-4455-a00b-c7d114eae96f"
    }
  ],
}
```

```
"additional_info" : "\"Sacred Motors\" CEO",
"face_images": [
```

```
"/9j/4AAQSkZJRgABAQEASABIAAD/4QAIrXhpZgAATU0AKgAAAAgAAQESAAMAAAABAAEAAAAAAAD/2wBDAAIBAQIBAQICAgICAgICAwUDAwMDAwYEBAMFBwYHBwGBwICQs
JCAGKCAcHCg0KCgsMDAwMBwkODw0MDgsMDAz/2wBDAAQICAgMDAwYDAwYMCacIDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAz/wAA
RCABGAE8DASIAAhEBAXEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRV
S0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMX
Gx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJ
BUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqO
kpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn60nq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwDqvs1Hk/u6uf8ALOmRw+bX3B8eMt4qk8r5PuU6OLyqP/R
dAE3/AC0+5Tk6VDUKXagBr9ag/wBVJVnyv++qJYv3lAFZ4vNqtLFmSrkv3Kh8mgC0nWm0UUGgv1fvKmjiPnm+9PjloMxH6UJ0ofpQnSgAz+8qT7J+7pn1fwn/AMGygChL/rK
mji/d1NJFSJ0oAb/rY6Z5X0qaPZ5aUSRUGgyiSKnwf6xqfJL+8oMxn1fWnxxebRF9ypo4vKjoAIu1Q3H+tp8f+sp8kX7ugCnHvqxUUKVD76AK1v8AwJVyOoY7WrX/ACzoAdH
sqGSL95T44v3lPnoAI/3VP8r60W/9Kl0AbF9+mS/6ymW/7qSpo/3u+gBn1fWnvF+7qbyamji/d0AYsYkL/eqxRRQAU5+lFFACxxc1LRRQAPH1c/xU3/lpRRQBNL3py+YKKK
AP//Z"
]
}
```

Example of a response

```
{
  "id" : "d1bd0147-e3d7-4bcd-a19c-e9a1acafaadd",
  "external_id" : "3",
  "first_name": "John",
  "patronymic": "B.",
  "second_name": "Smith",
  "groups": [
    {
      "id": "4cf2c1bb-707a-4455-a00b-c7d114eae96f"
    }
  ],
  "additional_info": "\"Sacred Motors\" CEO",
  "modification_time": "2019-05-28T05:58:17.994Z",
```

```


"face_images": [
"/9j/4AAQSkZJRgABAQEASABIAAD/4QAIrXhpZgAATU0AKgAAAAGAAQESAAMAAAABAAEAAAAAAD/2wBDAAIBAQIBAQICAgICAgICAwUDAwMDAwYEBAMFBwYHBwcGBwcICQs
JCAGKCAcHCg0KCgsMDAwMBwkODw0MDgsMDAz/2wBDAQICAgMDAwYDAwYMCAcIDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAz/wAA
RCABGAIE8DASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRV
S0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMX
Gx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJ
BUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVlYWVnpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqO
kpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn60nq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwDqvs1Hk/u6uf8ALOmRw+bX3B8eMt4qk8r5PuU6OLyqP/R
dAE3/AC0+5Tk6VDUKXagBr9ag/wBVJVnyv++qJYv3lAFZ4vNqtLFmSrkv3Kh8mgC0nWm0UUGgv1fvKmjiipnm+9PjloMxH6UJ0ofpQnSgAz+8qT7J+7pn1fwn/AMGygChL/rK
mji/d1NJFSJ0oAb/rY6Z5X0qaPZ5aUSRUGgyiSKnw6xqfJL+8oMxn1fWnxxebRF9ypo4vKjoAIu1Q3H+tp8f+sp8kX7ugCnHvqxUUKVD76AK1v8AwJVyOoY7WrX/ACzoAdH
sqGSL95T44v3lPnoAI/3VP8r60W/9Kl0AbF9+mS/6ymW/7qSpo/3u+gBn1fWnvF+7qbyamji/d0AYsYkL/eqxRRQAU5+1FFACxxc1LRRQAPH1c/xU3/1pRRQBNL3p6y+YKKK
AP//Z"
]
}

```



The *face_images* array must contain at least one photo. All the other fields may be empty or *null*. The photos CAN be transferred in the following formats: jpg, png and bmp. The photos MUST be coded in base64 string.



The module lets you add entries to the face database with images that do not meet the requirements of the module, or without any of them at all. Such images will be marked with the  icon in the database and will not be used by the module as a recognition sample. To add a record with an unrecognizable sample, set the *force* parameter in the query body to *true*.

DELETE /api/faces/<id>

To delete a face with the specified identifier from the database.

Example of a request

<code>http://localhost:8080/api/faces/d1bd0147-e3d7-4bcd-a19c-e9a1acafaadd?module=complete</code>

Parameters

Parameter	Description
module	Name of face recognition module to work with. List of available modules can be obtained using <i>/api/faceconfig</i> request.

GET /api/faces-groups

To obtain a list of the groups of faces from the database.

Example of a request

```
http://localhost:8080/api/faces-groups?offset=0&portion=10&module=complete
```

Parameters

Parameter	Description
offset	Non-mandatory parameter. It specifies a number of the group of faces starting from which it is needed to obtain the list of groups. The default value is 0.
portion	Non-mandatory parameter. The number of the groups of faces that needs to be obtained. The default value is 1000, the maximum value equals the number of faces in the database.
module	Mandatory parameter. The name of the face recognition module to work with. The list of the available modules can be viewed using the /api/faceconfig request.
filter	<p><attribute_name> <logical comparison operator> '<assumed value>', where</p> <ul style="list-style-type: none"><attribute_name> - name of attribute (field) of entity. external_id is available;<logical comparison operator> - '=' operator available for searching for the entrance to the collections;'<assumed value>' – ID value; <p><u>Important: the parameter must correspond with the URL encoding</u></p> <p>Example of the requests with filtration:</p> <p><u>http://localhost:8080/api/faces-groups?offset=0&portion=10&module=basic&filter=external_id='123'</u></p> <p>Example of displaying in URL encoding:</p> <p><u>http://localhost:8080/api/faces-groups?offset=0&portion=10&module=basic&filter=external_id%3D%27123%27</u></p>

Example of a response

```
{
```

```
{
  "offset": 0,
  "portion": 2,
  "total_count": 2,
  "groups": [
    {
      "id" : "4cf2c1bb-707a-4455-a00b-c7d114eae96f",
      "external_id" : "0",
      "name" : "Employess",
      "intercept" : false,
      "modification_time": "2019-05-28T05:58:17.994Z"
    },
    {
      "id" : "d2282515-520f-4d57-9cee-8674de0eb0cb",
      "external_id" : "1",
      "name" : "Thieves",
      "intercept" : true,
      "modification_time": "2019-05-28T05:58:17.994Z"
    }
  ]
}
```



modification_time is the UTC time when a face was added or updated in the database.

POST /api/faces-groups

To add a group of faces into database. The complete description of the group is returned in a response.

Example of a request

```
http://localhost:8080/api/faces-groups?module=complete
```

Parameters

Parameter	Description
module	Mandatory parameters. Name of the face recognition module to work with. Available modules can be viewed using the <i>/api/faceconfig</i> request.

Headers

Parameter	Description
Content-Type	application/json

Body (application/json)

```
{
  "external_id": "0",
  "name": "Employess",
  "intercept": false,
  "color": "ff00ff00"
}
```

Example of a response

```
{
  "id" : "4cf2c1bb-707a-4455-a00b-c7d114eae96f",
  "external_id" : "0",
  "name" : "Employess",
  "color" : "ffffffff",
  "intercept" : false,
  "modification_time": "2019-05-28T05:58:17.994Z"
}
```



The *name* field must be filled mandatorily. All the other fields may be empty or *null*. The *hex* value of a *string* type is used for the *color* field (for *argb* color).

GET /api/faces-groups/<id>

To obtain a description of the group.

Example of a request

```
http://localhost:8080/api/faces-groups/4cf2c1bb-707a-4455-a00b-c7d114eae96f?module=complete
```

Parameters

Parameter	Description
module	Mandatory parameter. Name of face recognition module to work with. Available modules can be viewed using <i>/api/faceconfig</i> request.

Example of response

```
{
  "id" : "4cf2c1bb-707a-4455-a00b-c7d114eae96f",
  "external_id" : "0",
  "name" : "Employess",
  "color" : "ffffffff",
  "intercept" : false,
  "modification_time": "2019-05-28T05:58:17.994Z"
}
```

PUT /api/faces-groups/<id>

To update an existing group of faces in the database. Full description is returned in response.

Example of a request

```
http://localhost:8080/api/faces-groups/4cf2c1bb-707a-4455-a00b-c7d114eae96?module=complete
```

Parameters

Parameter	Description
module	Mandatory parameter. Name of face recognition module to work with. Available modules can be viewed using <i>/api/faceconfig</i> request.

Headers

Parameter	Description
Content-Type	application/json

Body (application/json)

```
{
  "external_id" : "0",
  "name" : "Employess",
  "color" : "ffffffff",
  "intercept" : false
}
```

Example of a response

```
{
  "id": "4cf2c1bb-707a-4455-a00b-c7d114eae96f",
  "external_id" : "0",
  "name": "Employess",
  "color": "ffffffff",
  "intercept": false,
  "modification_time": "2019-05-28T05:58:17.994Z"
}
```



The *name* field must be filled mandatorily. All the other fields may be empty or *null*. The *hex* value of a *string* type is used for the *color* field (for *argb* color).

DELETE /api/faces-groups/<id>

To delete a group of faces from the database.

Example of a request

```
http://localhost:8080/api/faces-groups/4cf2c1bb-707a-4455-a00b-c7d114eae96?module=complete
```

Parameters

Parameter	Description
module	Mandatory parameter. Name of face recognition module to work with. Available modules can be viewed using <i>/api/faceconfig</i> request.



When a group is deleted, all its members exit the group without being deleted.

REST API interaction with License Plate database

Starting from version 3.4 of Eocortex, the REST API functionality for working with the license plate database has become available. It supports all basic operations with the database: getting, adding, modifying and deleting records about vehicles and groups.

The following modules are supported:

- Eocortex LPR Light
- Eocortex LPR Complete

Below is the description of the resources used for working with license plates database:

Resource	Description	GET	POST	PUT	DELETE
api/carconfig	Type of query for obtaining a list of available license plate recognition modules	+	n/a	n/a	n/a
api/cars	Category of queries for obtaining information about vehicles already added to the database and adding new entries	+	+	n/a	n/a
api/cars/<id>	Type of query for getting/adding/modifying/deleting detailed information about a specific vehicle record in the database	+	n/a	+	+
api/cars-groups	Category of queries for getting a list of license plate groups and adding new groups	+	+	n/a	n/a
api/cars-groups/<id>	Type of query for obtaining/changing/deleting detailed information about a specific group of license plates	+	n/a	+	+

Use cases

Synchronization of license plates database with external systems

It is possible to synchronize the license plate database with the third-party software. The common element that unites Eocortex database with the third-party software database is the "**external_id**" field. A 36-symbol string is used to fill in this field. It is allowed to not use the external identifier "**external_id**" at all, but use the ID field, which works as an internal identifier in the Eocortex license plate database, to identify database objects.

External ID can be set for both vehicles and groups.

The internal identifier cannot be changed.

Requirements, limitations, recommendations

Compatibility with previous versions

Starting from version 3.4, the **ExternalId** and **ExternalSystemId** fields have been added to the license plate recognition events and entries in the database. This change affects compatibility with previous versions of the software in case of system downgrade. If the system downgrade is needed, it is recommended to restore the database from the backup. But please do remember that restoring a database from a backup only partially solves the issue: database entries created by previous versions will be viewable in the Client application, but all entries created by version 3.4 will be lost.

To restore the database, follow the "Restoring a database from backups" instruction from the "Service operations with FirebirdDBMS" guide.

Requirements

To edit the license plate database, it is necessary to execute all queries (PUT, POST, DELETE) under a user with the "Editing face database and license plate database data" user right granted.

For **Eocortex Enterprise/Ultra** version, the user must have access to the module itself to perform any queries to the database of this module.

Requests

GET /api/carconfig

Get a list of all available license plate recognition modules.

Request example

```
http://localhost:8080/api/carconfig
```

Response example

```
{
  "auto_modules":
    { "name":
      "complete",
      "enabled": true
    },
    {
      "name": "light",
      "enabled": false
    }
  }
}
```

GET /api/cars

Get a list of all vehicles from the license plate database.

Request example

```
http://localhost:8080/api/cars?offset=0&portion=50
```

Parameters

offset	Optional parameter. Specifies the sequence number of the entry from which you want to get the list of cars. The default value is 0.
portion	Optional parameter. The number of vehicle records to retrieve. The default value is 1000, the maximum value is equal to the number of vehicle records in the database.
filter	Optional parameter. The filter syntax is as follows: <attribute_name> <logical comparison operator> '<assumed value>', where <attribute_name> – name of attribute (field) of entity. Parameter value options: <ul style="list-style-type: none">● group_id – filter by group ID;● first_name – filter by First Name line;● last_name – filter by Last Name line;● patronymic – filter by Patronymic line;● license_plate_number – filter by license plate number;● required_ratio – the required percentage of coincidence of the full name or plate number to include the vehicle

in the list (solid value from 0 to 100, default 100). The higher will be value, the more accurate will be search;

- **external_id** – identifier of the record from the third-party software integrated with Eocortex;
- **external_sys_id** – identifier of the third-party software integrated with Eocortex. It is used by the database synchronization functionality from an external system (this functionality has not yet been implemented for car numbers). The maximum length is 64 characters.
- **external_owner_id** – vehicle owner identifier used by third-party software integrated with Eocortex. It can be used for integration with external systems for 2FA tasks. Vehicle owner identifier may not be unique for entries in the database to make such tasks, as "One person can have several vehicles, but one pass" possible. The maximum length is 64 characters.

<logical comparison operator> – operator '=' to search for matching entities in the database;

'<assumed value>' – value for the parameter used;

Note. Parameter must match URL encoding

Request samples with filtering:

http://localhost:8080/api/cars?offset=0&portion=50&module=basic&filter=external_id='123'

Filters can be combined using the "AND" operator, such as searching for vehicles belonging to multiple groups:

http://localhost:8080/api/cars?offset=0&portion=50&module=basic&filter=group_id='05a21e32-51e5-405e-b4cf-3082796c8703' AND group_id='2090779b-9091-4989-a5f8-92d1920d8291'

But several filters cannot be set at once for the first_name, last_name, patronymic parameters.

Request sample in URL encoding:

http://localhost:8080/api/cars?offset=0&portion=50&module=basic&filter=group_id%3D%2705a21e32-51e5-405e-b4cf-3082796c8703%27

dbid	<p>Optional parameter.</p> <p>Available only for the License Plate Recognition Complete module.</p> <p>Identifier of the specific license plate database.</p> <p>Multiple license plate databases can be created in Eocortex for the License Plate Recognition module. If this parameter is specified, the action is performed with a unique database that corresponds the identifier used. If the parameter is not specified, then the action is performed with the common database.</p> <p>More information about this feature can be found in the Eocortex Admin Guide.</p>
-------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Response example

```
{
  "offset": 0,
  "portion": 2,
  "total_count": 2,
  "plates": [
    {
      "id": "74d8b8cb-ca9c-4a6f-af8c-f990dad24325",
      "external_id": "",
      "license_plate_number": "0529KH70",
      "additional_info": "Custom additional note",
      "modification_time": "0001-01-01T00:00:00"
    },
    {
      "id": "c25ddd15-53a3-47f4-8471-3b5425a983f9",
      "external_id": "",
      "license_plate_number": "K999YB70",
      "additional_info": "No",
      "modification_time": "0001-01-01T00:00:00"
    }
  ]
}
```

Note

modification_time - time in UTC when the record was added or updated in the license plate database.

POST /api/cars

Adds a record about a new vehicle to the license plate database. A full description of the vehicle is returned in response.

Request example

```
http://localhost:8080/api/cars
```

Parameters

dbid	Optional parameter. Available only for the License Plate Recognition Complete module. Identifier of the specific license plate database. Multiple license plate databases can be created in Eocortex for the License Plate Recognition module. If this parameter is specified, the action is performed with a unique database that corresponds the identifier used. If the parameter is not specified, then the action is performed with the common database. More information about this feature can be found in the Eocortex Admin Guide.
-------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Headers

Content-Type	application/json
--------------	------------------

Body (application/json)

```
{
  "owner": {
    "first_name": "John",
    "second_name": "Cash",
    "third_name": "Ray"
  },
  "external_id": "0",
  "license_plate_number": "0529KH34",
  "additional_info": "Custom note",
}
```

```
"model": "Rolls-Royce Silver Shadow",
"color": "Black",
"groups": [
{
  "id": "8d7223c9-b9ff-42b8-8fe8-ec7c5a6efa67"
}
]
}
```

Response example

```
{
  "owner": {
    "first_name": "John",
    "second_name": "Cash",
    "third_name": "Ray"
  },
  "id": "f23fddff-d58f-4d4e-9b31-ba1b513f537d",
  "external_id": "0",
  "license_plate_number": "0529H34",
  "additional_info": "Custom note",
  "modification_time": "0001-01-01T00:00:00",
  "model": "Rolls-Royce Silver Shadow",
  "color": "Black",
  "groups": [
    {
      "id": "8d7223c9-b9ff-42b8-8fe8-ec7c5a6efa67"
    }
  ]
}
```

GET /api/cars/<id>

Get a detailed description of the vehicle from the license plate database.

Request example

http://localhost:8080/api/cars/74d8b8cb-ca9c-4a6f-af8c-f990dad24325

Parameters

dbid	<p>Optional parameter.</p> <p>Available only for the License Plate Recognition Complete module.</p> <p>Identifier of the specific license plate database.</p> <p>Multiple license plate databases can be created in Eocortex for the License Plate Recognition module. If this parameter is specified, the action is performed with a unique database that corresponds the identifier used. If the parameter is not specified, then the action is performed with the common database.</p> <p>More information about this feature can be found in the Eocortex Admin Guide.</p>
------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Response example

```
{
  "owner": {
    "first_name": "John",
    "second_name": "Cash",
    "third_name": "Ray"
  },
  "id": "74d8b8cb-ca9c-4a6f-af8c-f990dad24325",
  "external_id": "",
  "license_plate_number": "0529KH70",
  "additional_info": "Custom note",
  "modification_time": "0001-01-01T00:00:00",
  "model": "Rolls-Royce Silver Shadow",
  "color": "Black",
  "groups": [
```

```
{  
  "id": "8d7223c9-b9ff-42b8-8fe8-ec7c5a6efa67"  
}  
]  
}
```

Note

modification_time - time in UTC when the record was added or updated in the license plate database.

PUT /api/cars/<id>

Updating an existing vehicle record in the license plate database. A full description is returned in response.

Request example

```
http://localhost:8080/api/cars/d1bd0147-e3d7-4bcd-a19c-e9a1acafaadd
```

Parameters

dbid	<p>Optional parameter.</p> <p>Available only for the License Plate Recognition Complete module.</p> <p>Identifier of the specific license plate database.</p> <p>Multiple license plate databases can be created in Eocortex for the License Plate Recognition module. If this parameter is specified, the action is performed with a unique database that corresponds the identifier used. If the parameter is not specified, then the action is performed with the common database.</p> <p>More information about this feature can be found in the Eocortex Admin Guide.</p>
-------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Headers

Content-Type	application/json
--------------	------------------

Body (application/json)

<pre>{ "owner": { "first_name": "John", "second_name": "Cash", "third_name": "Ray" }, "id": "74d8b8cb-ca9c-4a6f-af8c-f990dad24325", "external_id": "", "license_plate_number": "JCASH1",</pre>

```
"additional_info": "Singer",
"modification_time": "0001-01-01T00:00:00",
"model": "Rolls-Royce Silver Shadow",
"color": "Black",
"groups": [
{
  "id": "8d7223c9-b9ff-42b8-8fe8-ec7c5a6efa67"
}
]
}
```

Response example

```
{
  "owner": {
    "first_name": "John",
    "second_name": "Cash",
    "third_name": "Ray"
  },
  "id": "74d8b8cb-ca9c-4a6f-af8c-f990dad24325",
  "external_id": "",
  "license_plate_number": "JCASH1",
  "additional_info": "Singer",
  "modification_time": "0001-01-01T00:00:00",
  "model": "Rolls-Royce Silver Shadow",
  "color": "Black",
  "groups": [
    {
      "id": "8d7223c9-b9ff-42b8-8fe8-ec7c5a6efa67"
    }
  ]
}
```

DELETE /api/cars/

Delete a record about a vehicle with the specified identifier from the database.

Request example

```
http://localhost:8080/api/cars/74d8b8cb-ca9c-4a6f-af8c-f990dad24325
```

Parameters

dbid	<p>Optional parameter.</p> <p>Available only for the License Plate Recognition Complete module.</p> <p>Identifier of the specific license plate database.</p> <p>Multiple license plate databases can be created in Eocortex for the License Plate Recognition module. If this parameter is specified, the action is performed with a unique database that corresponds the identifier used. If the parameter is not specified, then the action is performed with the common database.</p> <p>More information about this feature can be found in the Eocortex Admin Guide.</p>
-------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

GET /api/cars-groups

Get a list of vehicle groups from the license plate database.

Request example

```
http://localhost:8080/api/cars-groups?offset=0&portion=10
```

Parameters

offset	Optional parameter. Specifies the sequence number of the entry from which you want to get the list of groups. The default is 0.
portion	Optional parameter. The number of groups to receive. The default value is 1000, the maximum value is equal to the number of vehicle records in the database.
filter	Optional parameter. The filter syntax is as follows: <attribute_name> <logical comparison operator> '<assumed value>', where <attribute_name> - name of attribute (field) of entity. Parameter value options: <ul style="list-style-type: none">● external_id – identifier of the record from the third-party software integrated with Eocortex;● external_sys_id – identifier of the third-party software integrated with Eocortex. It is used by the database synchronization functionality from an external system. The maximum length is 64 characters. <logical comparison operator> – operator "=" to search for matching entities in the database;

	<p>'<assumed value>' – value for the parameter used;</p> <p><u>Note. Parameter must match URL encoding</u></p> <p>Request samples with filtering:</p> <p>http://localhost:8080/api/cars-groups?offset=0&portion=10&module=basic&filter=external_id='123'</p> <p>Request sample in URL encoding:</p> <p>http://localhost:8080/api/cars-groups?offset=0&portion=10&module=basic&filter=external_id%3D%27123%27</p>
dbid	<p>Optional parameter.</p> <p>Available only for the License Plate Recognition Complete module.</p> <p>Identifier of the specific license plate database.</p> <p>Multiple license plate databases can be created in Eocortex for the License Plate Recognition module. If this parameter is specified, the action is performed with a unique database that corresponds the identifier used. If the parameter is not specified, then the action is performed with the common database.</p> <p>More information about this feature can be found in the Eocortex Admin Guide.</p>

Response example

```
{
  "offset": 0,
  "portion": 3,
  "total_count": 3,
  "groups": [
    {
      "id": "8d7223c9-b9ff-42b8-8fe8-ec7c5a6efa67",
      "external_id": "",
      "name": "White list",
      "intercept": false,
      "open_barrier": false,
      "modification_time": "2021-02-14T02:26:01.423Z"
    }
  ]
}
```

```
  },
  {
    "id": "fbf579b5-b171-40ea-acd7-e4680ce3b962",
    "external_id": "",
    "name": "Interception",
    "intercept": true,
    "open_barrier": false,
    "modification_time": "2021-02-14T02:26:01.423Z"
  },
  {
    "id": "976cff8b-6ad3-4153-bf57-3321836c8006",
    "external_id": "",
    "name": "Open barrier gate",
    "intercept": false,
    "open_barrier": true,
    "modification_time": "2021-02-14T02:26:01.423Z"
  }
]
```

Note

modification_time – time in UTC when the record was added or updated in the license plate database.

POST /api/cars-groups

Adds a group to the license plate database. A full description of the group is returned in response.

Request example

```
http://localhost:8080/api/cars-groups
```

Parameters

dbid	<p>Optional parameter.</p> <p>Available only for the License Plate Recognition Complete module.</p> <p>Identifier of the specific license plate database.</p> <p>Multiple license plate databases can be created in Eocortex for the License Plate Recognition module. If this parameter is specified, the action is performed with a unique database that corresponds the identifier used. If the parameter is not specified, then the action is performed with the common database.</p> <p>More information about this feature can be found in the Eocortex Admin Guide.</p>
-------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Headers

Content-Type	application/json
--------------	------------------

Body (application/json)

```
{
  "external_id": "3",
  "name": "White list",
  "intercept": false,
  "open_barrier": false,
  "color": "ff00ff00"
}
```

Response example

```
{
  "id": "ce25fa01-f49c-41c2-ac9d-7653ab900dd5",
  "external_id": "3",
  "name": "White list",
  "intercept": false,
  "open_barrier": false,
  "modification_time": "2021-02-16T17:37:27.3336743Z",
  "color": "ff00ff00"
}
```

Note

The *Name* field must be filled for each request, all other fields may be empty or *null*. For the *Color* field *string* type hex value (for *argb* color) must be used.

GET /api/cars-groups/<id>

Get a description of the group.

Request example

```
http://localhost:8080/api/cars-groups/8d7223c9-b9ff-42b8-8fe8-ec7c5a6efa67
```

Parameters

dbid	<p>Optional parameter.</p> <p>Available only for the License Plate Recognition Complete module.</p> <p>Identifier of the specific license plate database.</p> <p>Multiple license plate databases can be created in Eocortex for the License Plate Recognition module. If this parameter is specified, the action is performed with a unique database that corresponds the identifier used. If the parameter is not specified, then the action is performed with the common database.</p> <p>More information about this feature can be found in the Eocortex Admin Guide.</p>
-------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Response example

```
{
  "id": "8d7223c9-b9ff-42b8-8fe8-ec7c5a6efa67",
  "external_id": "",
  "name": "White list",
  "intercept": false,
  "open_barrier": false,
  "modification_time": "2021-02-14T02:26:01.423Z",
  "color": "ff00ff00"
}
```


PUT /api/cars-groups/<id>

Updates an existing group in the license plate database. A full description of the group is returned in response.

Request example

```
http://localhost:8080/api/cars-groups/ce25fa01-f49c-41c2-ac9d-7653ab900dd5
```

Parameters

dbid	Optional parameter. Available only for the License Plate Recognition Complete module. Identifier of the specific license plate database. Multiple license plate databases can be created in Eocortex for the License Plate Recognition module. If this parameter is specified, the action is performed with a unique database that corresponds the identifier used. If the parameter is not specified, then the action is performed with the common database. More information about this feature can be found in the Eocortex Admin Guide.
-------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Headers

Content-Type	application/json
--------------	------------------

Body (application/json)

```
{
  "external_id": "4",
  "name": "White list",
  "intercept": false,
  "open_barrier": false,
  "color": "ff00ff00"
}
```

Response example

```
{
  "id": "ce25fa01-f49c-41c2-ac9d-7653ab900dd5",
  "external_id": "4",
  "name": "White list",
  "intercept": false,
  "open_barrier": false,
  "modification_time": "2021-02-16T17:44:32.833567Z",
  "color": "ff00ff00"
}
```

Note

The *Name* field must be filled for each request, all other fields may be empty or *null*. For the *Color* field *string* type hex value (for *argb* color) must be used.

DELETE /api/cars-groups/

Remove a group from the license plate database.

Request example

```
http://localhost:8080/api/cars-groups/ce25fa01-f49c-41c2-ac9d-7653ab900dd5
```

Parameters

dbid	<p>Optional parameter.</p> <p>Available only for the License Plate Recognition Complete module.</p> <p>Identifier of the specific license plate database.</p> <p>Multiple license plate databases can be created in Eocortex for the License Plate Recognition module. If this parameter is specified, the action is performed with a unique database that corresponds the identifier used. If the parameter is not specified, then the action is performed with the common database.</p> <p>More information about this feature can be found in the Eocortex Admin Guide.</p>
-------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Note

When you delete a group, all members automatically leave the group without deleting them.

REST API for getting data from the Object Classification and Counting module

Starting from version 4.0 of Eocortex, with the advent of the Object Classification and Counting module, the REST API for obtaining data from this module is available. The REST API is necessary because the events received from the database don't provide the user with readable information about the data accumulated by the module.

Description of the resources used to get data from the Object Classification and Counting module:

Resource	Description	GET
api/objects_counting/current_counters	Obtaining current counter values for lines and areas. The data is the same as what is seen in the Eocortex Client in real time.	+
api/objects_counting/report	Obtaining data on the number of objects that crossed the lines for the specified time interval.	+

GET /api/object_counting/current_counters

Get the current counter values for lines and areas. The data is the same as what is seen in the Eocortex Client in real time.

Request example

```
http://127.0.0.1:8080/api/objects_counting/current_counters?login=root&password=&channelId=9e2cd158-f739-4715-89fb-a460147fc33b
```

Parameters

ChannelId	Optional parameter. Camera ID for which to get data. If not set, the data is output as a list for all cameras on which the module is running.
------------------	------------------------------------------------------------------------------------------------------------------------------------------------------

Response example

```
{
  "Id": "9e2cd158-f739-4715-89fb-a460147fc33b",
  "ChannelName": "Plant",
  "Zones": [
    {
      "Id": "0af772da-9993-4a39-a785-4e7c89efe55a",
      "Name": "Crossing of the line",
      "Type": "LineSection",
      "CurrentCounts": {
        "Person": 16,
        "Animal": 7
      }
    },
    {
      "Id": "d98c95d6-9e6a-4a5e-80a4-184826846b17",
      "Name": "Counting objects in area",
      "Type": "Zone",
      "CurrentCounts": {
        "Person": 3,
```

```
"Car": 0,  
"Bus": 0,  
"Truck": 0,  
"Motorbike": 0,  
"Animal": 0
```

```
}
```

```
}
```

```
]
```

```
}
```

GET /api/object_counting/report

Get data on the number of objects that crossed the lines for a specified time interval.

Request example

```
http://127.0.0.1:8080/api/objects_counting/report?login=root&password=&channelId=9e2cd158-f739-4715-89fb-a460147fc33b&startTime="2022.12.22 18:00:00"&endTime="2022.12.22 18:15:00"
```

Parameters

ChannelId	Optional parameter. Camera ID for which to get data. If not set, the data is output as a list for all cameras on which the module is running.
StartTime	Search start time (local to the server) in the format dd.MM.yyyy HH:mm:ss.fff
EndTime	Search end time (local to the server) in the format dd.MM.yyyy HH:mm:ss.fff

Response example

```
{
  "Id": "9e2cd158-f739-4715-89fb-a460147fc33b",
  "ChannelName": "Plant",
  "Zones": [
    {
      "Id": "0af772da-9993-4a39-a785-4e7c89efe55a",
      "Name": "Crossing of the line",
      "StartCounts": {
        "Person": 9,
        "Animal": 4
      },
      "TotalCounts": {
        "Person": 11,
        "Animal": 3
      }
    }
  ]
}
```

```
}  
  ]  
    }  
      }
```

StartCounts - values at the beginning of the search period, **Total Counts** - the number of objects crossing the line during the period.

Note

This command returns counters only for lines. Counters for areas are not available, since counting for a period is meaningless for them.