

API Reference

# TargetEYE API

API Version: 0.16.0rc2

# INDEX

<b>1. AUDITLOGS</b>	<b>8</b>
1.1 GET /api/v1/auditlogs	8
1.2 GET /api/v1/auditlogs/{auditlog_id}	10
<b>2. CAMERAS</b>	<b>13</b>
2.1 GET /api/v1/cameras	13
2.2 POST /api/v1/cameras	14
2.3 POST /api/v1/cameras/search	16
2.4 GET /api/v1/cameras/{camera_id}	17
2.5 PUT /api/v1/cameras/{camera_id}	19
2.6 DELETE /api/v1/cameras/{camera_id}	21
<b>3. EVENTS</b>	<b>23</b>
3.1 GET /api/v1/events/{event_id}	23
3.2 GET /api/v1/events	26
3.3 GET /api/v1/eventmatches/{event_id}	30
3.4 GET /api/v1/eventmatches	33
<b>4. FACELISTS</b>	<b>36</b>
4.1 GET /api/v1/facelists	36
4.2 POST /api/v1/facelists	36
4.3 GET /api/v1/facelists/{facelist_id}	37
4.4 PUT /api/v1/facelists/{facelist_id}	38
4.5 DELETE /api/v1/facelists/{facelist_id}	40
4.6 GET /api/v1/facelists/{facelist_id}/faces	41
4.7 POST /api/v1/facelists/{facelist_id}/faces/{face_id}	42
4.8 DELETE /api/v1/facelists/{facelist_id}/faces/{face_id}	43
<b>5. FACES</b>	<b>45</b>
5.1 GET /api/v1/faces	45
5.2 POST /api/v1/faces	46
5.3 GET /api/v1/faces/{face_id}	47
5.4 PUT /api/v1/faces/{face_id}	49
5.5 DELETE /api/v1/faces/{face_id}	50
5.6 GET /api/v1/faces/{face_id}/images	51
5.7 POST /api/v1/faces/{face_id}/images	53
5.8 DELETE /api/v1/faces/{face_id}/images	54
<b>6. GROUPS</b>	<b>57</b>
6.1 GET /api/v1/groups	57
6.2 POST /api/v1/groups	57
6.3 GET /api/v1/groups/{group_id}	59
6.4 PUT /api/v1/groups/{group_id}	60
6.5 DELETE /api/v1/groups/{group_id}	61

6.6	GET /api/v1/groups/{group_id}/members	62
6.7	POST /api/v1/groups/{group_id}/members	63
6.8	DELETE /api/v1/groups/{group_id}/members	64
6.9	POST /api/v1/groups/search	65
6.10	GET /api/v1/groups/{group_id}/places	66
6.11	POST /api/v1/groups/{group_id}/places	67
6.12	DELETE /api/v1/groups/{group_id}/places	69
<b>7.</b>	<b>HELLO</b>	<b>70</b>
7.1	GET /api/v1/hello	70
<b>8.</b>	<b>ONVIF</b>	<b>71</b>
8.1	GET /api/v1/onvif/{camera_id}	71
8.2	GET /api/v1/onvif/{camera_id}/network	72
8.3	PUT /api/v1/onvif/{camera_id}/network	73
8.4	GET /api/v1/onvif/{camera_id}/users	75
8.5	PUT /api/v1/onvif/{camera_id}/users/{username}	76
8.6	GET /api/v1/onvif/{camera_id}/streams	78
8.7	PUT /api/v1/onvif/{camera_id}/stream/encoder	80
8.8	GET /api/v1/onvif/{camera_id}/image	81
8.9	PUT /api/v1/onvif/{camera_id}/image/brightness	84
8.10	PUT /api/v1/onvif/{camera_id}/image/contrast	85
8.11	PUT /api/v1/onvif/{camera_id}/image/saturation	87
8.12	PUT /api/v1/onvif/{camera_id}/image/sharpness	88
8.13	PUT /api/v1/onvif/{camera_id}/image/backlight	89
8.14	PUT /api/v1/onvif/{camera_id}/image/ir	91
8.15	PUT /api/v1/onvif/{camera_id}/image/exposure	92
8.16	PUT /api/v1/onvif/{camera_id}/image/dr	94
8.17	PUT /api/v1/onvif/{camera_id}/image/wb	95
8.18	GET /api/v1/onvif/{camera_id}/ptz	96
8.19	PUT /api/v1/onvif/{camera_id}/ptz	98
8.20	PUT /api/v1/onvif/{camera_id}/ptz/home	100
8.21	PUT /api/v1/onvif/{camera_id}/ptz/home/apply	101
8.22	GET /api/v1/onvif/{camera_id}/ptz/presets	103
8.23	POST /api/v1/onvif/{camera_id}/ptz/preset	104
8.24	DELETE /api/v1/onvif/{camera_id}/ptz/preset	105
8.25	POST /api/v1/onvif/{camera_id}/ptz/preset/apply	107
<b>9.</b>	<b>ORGANIZATIONS</b>	<b>109</b>
9.1	GET /api/v1/organizations	109
9.2	POST /api/v1/organizations	109
9.3	GET /api/v1/organizations/{organization_id}	110
9.4	PUT /api/v1/organizations/{organization_id}	111
9.5	DELETE /api/v1/organizations/{organization_id}	113
9.6	GET /api/v1/organizations/{organization_id}/members	114

9.7	GET	/api/v1/organizations/{organization_id}/groups	115
9.8	POST	/api/v1/organizations/search	116
<b>10.</b>	<b>ORION</b>		<b>118</b>
10.1	GET	/api/v1/orion/info	118
10.2	PUT	/api/v1/orion/config	118
<b>11.</b>	<b>PERMISSIONS</b>		<b>120</b>
11.1	POST	/api/v1/permissions	120
11.2	DELETE	/api/v1/permissions	121
<b>12.</b>	<b>PLACES</b>		<b>124</b>
12.1	GET	/api/v1/places	124
12.2	POST	/api/v1/places	125
12.3	POST	/api/v1/places/search	126
12.4	GET	/api/v1/places/{place_id}	127
12.5	PUT	/api/v1/places/{place_id}	128
12.6	DELETE	/api/v1/places/{place_id}	129
12.7	GET	/api/v1/places/{place_id}/plans	130
12.8	GET	/api/v1/places/{place_id}/cameras	131
<b>13.</b>	<b>PLANS</b>		<b>134</b>
13.1	GET	/api/v1/plans	134
13.2	POST	/api/v1/plans	135
13.3	POST	/api/v1/plans/search	135
13.4	GET	/api/v1/plans/{plan_id}	137
13.5	PUT	/api/v1/plans/{plan_id}	137
13.6	DELETE	/api/v1/plans/{plan_id}	138
13.7	GET	/api/v1/plans/{plan_id}/cameras	140
13.8	POST	/api/v1/plans/{plan_id}/cameras	141
13.9	DELETE	/api/v1/plans/{plan_id}/cameras	142
13.10	PUT	/api/v1/plans/{plan_id}/cameras/{camera_id}/position	143
<b>14.</b>	<b>PLATELISTS</b>		<b>146</b>
14.1	GET	/api/v1/platelist	146
14.2	POST	/api/v1/platelist	146
14.3	GET	/api/v1/platelist/{platelist_id}	147
14.4	PUT	/api/v1/platelist/{platelist_id}	148
14.5	DELETE	/api/v1/platelist/{platelist_id}	150
14.6	POST	/api/v1/platelist/{platelist_id}/plates/{plate_id}	151
14.7	DELETE	/api/v1/platelist/{platelist_id}/plates/{plate_id}	152
14.8	GET	/api/v1/platelist/{platelist_id}/plates	152
<b>15.</b>	<b>PLATES</b>		<b>154</b>
15.1	GET	/api/v1/plates	154
15.2	POST	/api/v1/plates	155

15.3	GET /api/v1/plates/{plate_id}	156
15.4	PUT /api/v1/plates/{plate_id}	157
15.5	DELETE /api/v1/plates/{plate_id}	159
<b>16.</b>	<b>ROLES</b>	<b>160</b>
16.1	GET /api/v1/roles	160
16.2	POST /api/v1/roles	160
16.3	GET /api/v1/roles/{role_id}	161
16.4	PUT /api/v1/roles/{role_id}	162
16.5	DELETE /api/v1/roles/{role_id}	163
16.6	GET /api/v1/roles/{role_id}/members	165
16.7	GET /api/v1/roles/{role_id}/permissions	166
<b>17.</b>	<b>SEARCH</b>	<b>169</b>
17.1	POST /api/v1/search/face	169
17.2	POST /api/v1/search/plate	170
<b>18.</b>	<b>SEGMENTS</b>	<b>173</b>
18.1	GET /api/v1/segments	173
18.2	GET /api/v1/segments/index	174
<b>19.</b>	<b>SNAPSHOTS</b>	<b>176</b>
19.1	GET /api/v1/snapshots/{event_id}	176
19.2	GET /api/v1/camerasnapshots/{camera_id}	177
<b>20.</b>	<b>STATS</b>	<b>179</b>
20.1	GET /api/v1/stats/cameras	179
20.2	GET /api/v1/stats/eventsbytype	179
20.3	GET /api/v1/stats/eventmatchesbypriority	181
20.4	GET /api/v1/stats/eventsintensity	182
<b>21.</b>	<b>STREAMS</b>	<b>184</b>
21.1	GET /api/v1/streams/{stream_id}	184
21.2	GET /api/v1/streams/{stream_id}/info	185
21.3	POST /api/v1/streams/verify	186
<b>22.</b>	<b>STREAMSERVERS</b>	<b>188</b>
22.1	GET /api/v1/streamservers	188
<b>23.</b>	<b>STREAMSTATUSES</b>	<b>189</b>
23.1	GET /api/v1/streamstatuses/summary	189
23.2	GET /api/v1/streamstatuses/{stream_id}	189
<b>24.</b>	<b>TOKEN</b>	<b>191</b>
24.1	POST /api/v1/token	191
24.2	POST /api/v1/token/refresh	192
24.3	POST /api/v1/token/revoke	193

<b>25. UPLOADS</b>	<b>195</b>
25.1 POST /api/v1/uploads	195
25.2 GET /api/v1/uploads/{upload_id}	196
25.3 GET /api/v1/uploads/{upload_id}/content	197
25.4 POST /api/v1/uploads/meta	198
<b>26. USERS</b>	<b>200</b>
26.1 GET /api/v1/users	200
26.2 POST /api/v1/users	201
26.3 GET /api/v1/users/me	202
26.4 PUT /api/v1/users/me	203
26.5 GET /api/v1/users/{user_id}	204
26.6 PUT /api/v1/users/{user_id}	205
26.7 DELETE /api/v1/users/{user_id}	207
26.8 PUT /api/v1/users/me/password	208
26.9 PUT /api/v1/users/me/email	208
26.10 POST /api/v1/users/search	209
<b>27. VIDECLIPS</b>	<b>212</b>
27.1 GET /api/v1/videoclips	212
27.2 POST /api/v1/videoclips	213
27.3 GET /api/v1/videoclips/{videoclip_id}	214
27.4 DELETE /api/v1/videoclips/{videoclip_id}	215

# Security and Authentication

## SECURITY SCHEMES

KEY	TYPE	DESCRIPTION
OAuth2PasswordBearer	oauth2	

# API

## 1. AUDITLOGS

### 1.1 GET /api/v1/auditlogs

**Get list of audit log entries, filtered by period, user ID and action type**

Get list of audit log entries, filtered by period and, optionally, by user ID and action type.

Returned entries are ordered by creation date in descending order.  
If no period provided, it will return entries for the last 24 hours.

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
user_ids	array of integer	
action_types	array of string <b>ALLOWED:</b> camera_create, camera_delete, camera_update, face_create, face_delete, face_update, facelist_create, facelist_delete, facelist_update, faceimage_create, faceimage_delete, facelist_face_create, facelist_face_delete, group_create, group_delete, group_update, group_user_create, group_user_delete, group_place_create, group_place_delete, organization_create, organization_delete, organization_update, permission_create, permission_delete, onvif_image_update, onvif_network_update, onvif_ptz_update, onvif_ptz_home_save, onvif_ptz_home_apply, onvif_ptz_preset_create, onvif_ptz_preset_delete, onvif_ptz_preset_apply, onvif_stream_update, onvif_user_update, orion_create, orion_delete, orion_update, place_create, place_delete, place_update, place_group_create, place_group_delete, plan_create, plan_delete, plan_update, plan_camera_create, plan_camera_delete, plan_camera_update, plate_create, plate_delete, plate_update, platelist_create, platelist_delete, platelist_update, platelist_plate_create, platelist_plate_delete, role_create, role_delete, role_update, upload_create, user_create, user_delete, user_update, user_login, user_logout, version_update, videoclip_create, videoclip_delete	
created_at_from	date-time	
created_at_to	date-time	
order_by	enum <b>ALLOWED:</b> created_at, -created_at	
organization_id	integer	
cursor	string	
limit	integer between 10 and 1000	

### RESPONSE

**STATUS CODE - 200:** List of audit log entries

**RESPONSE MODEL - application/json**

```
{  
  pagination* {
```



```

    cursor string
  }
  auditlogs* [{
    Array of object:
    action_type*      enum      ALLOWED:camera_create, camera_delete,
                                camera_update, face_create, face_delete,
                                face_update, facelist_create, facelist_delete,
                                facelist_update, faceimage_create,
                                faceimage_delete, facelist_face_create,
                                facelist_face_delete, group_create,
                                group_delete, group_update, group_user_create,
                                group_user_delete, group_place_create,
                                group_place_delete, organization_create,
                                organization_delete, organization_update,
                                permission_create, permission_delete,
                                onvif_image_update, onvif_network_update,
                                onvif_ptz_update, onvif_ptz_home_save,
                                onvif_ptz_home_apply, onvif_ptz_preset_create,
                                onvif_ptz_preset_delete, onvif_ptz_preset_apply,
                                onvif_stream_update, onvif_user_update,
                                orion_create, orion_delete, orion_update,
                                place_create, place_delete, place_update,
                                place_group_create, place_group_delete,
                                plan_create, plan_delete, plan_update,
                                plan_camera_create, plan_camera_delete,
                                plan_camera_update, plate_create, plate_delete,
                                plate_update, platelist_create,
                                platelist_delete, platelist_update,
                                platelist_plate_create, platelist_plate_delete,
                                role_create, role_delete, role_update,
                                upload_create, user_create, user_delete,
                                user_update, user_login, user_logout,
                                version_update, videoclip_create,
                                videoclip_delete
                                An enumeration.
    user_id*          integer
    client_ip         string
    meta* {
    }
    id*              string
    organization_id*  integer
    organization_name* string
    created_at*      string
    user_email*      string
    user_first_name* string
    user_last_name*  string
    user_avatar_id   string
    user_role_id*    integer
    user_role_name*  string
    user_avatar_url  string  1 to 65536 chars
    user_groups* [{
    Array of object:
      id*   integer
      name* string
    }]
    user_is_superuser* boolean
  }]

```

```
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

## 1.2 GET /api/v1/auditlogs/{auditlog\_id}

Get audit log entry by provided ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*auditlog_id	uuid	

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
organization_id	integer	

### RESPONSE

STATUS CODE - 200: Requested audit log entry

RESPONSE MODEL - application/json

```
{
  action_type*      enum      ALLOWED:camera_create, camera_delete, camera_update,
                    face_create, face_delete, face_update, facelist_create,
                    facelist_delete, facelist_update, faceimage_create,
                    faceimage_delete, facelist_face_create,
                    facelist_face_delete, group_create, group_delete,
                    group_update, group_user_create, group_user_delete,
                    group_place_create, group_place_delete,
                    organization_create, organization_delete,
                    organization_update, permission_create,
                    permission_delete, onvif_image_update,
                    onvif_network_update, onvif_ptz_update,
                    onvif_ptz_home_save, onvif_ptz_home_apply,
                    onvif_ptz_preset_create, onvif_ptz_preset_delete,
                    onvif_ptz_preset_apply, onvif_stream_update,
                    onvif_user_update, orion_create, orion_delete,
                    orion_update, place_create, place_delete, place_update,
```

place\_group\_create, place\_group\_delete, plan\_create,  
 plan\_delete, plan\_update, plan\_camera\_create,  
 plan\_camera\_delete, plan\_camera\_update, plate\_create,  
 plate\_delete, plate\_update, platelist\_create,  
 platelist\_delete, platelist\_update,  
 platelist\_plate\_create, platelist\_plate\_delete,  
 role\_create, role\_delete, role\_update, upload\_create,  
 user\_create, user\_delete, user\_update, user\_login,  
 user\_logout, version\_update, videoclip\_create,  
 videoclip\_delete

An enumeration.

```

user_id*           integer
client_ip          string
meta* {
}
id*               string
organization_id*  integer
organization_name* string
created_at*       string
user_email*       string
user_first_name*  string
user_last_name*   string
user_avatar_id    string
user_role_id*     integer
user_role_name*   string
user_avatar_url   string  1 to 65536 chars
user_groups* [{
  Array of object:
    id*   integer
    name* string
}]
user_is_superuser* boolean
}

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}

```

#### STATUS CODE - 500: Internal Server Error



## 2. CAMERAS

### 2.1 GET /api/v1/cameras

Get list of cameras

#### REQUEST

##### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
place_id	integer	
group_id	integer	
with_origin	boolean	
cursor	string	
limit	integer between 10 and 1000	

#### RESPONSE

STATUS CODE - 200: List of cameras

RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
  cameras* [{
    Array of object:
    name*          string    1 to 255 chars
    source* {
    }
    video_analytics {
    }
    video_storage {
    }
    video_streaming {
    }
    ptz {
    }
    place_id*      integer
    onvif_host     string    1 to 255 chars
    onvif_port     integer
    onvif_login    string    1 to 255 chars
    onvif_password string    1 to 255 chars
    scheduler_group string    1 to 64 chars
    scheduler_tags [string]  1 to 32 chars
    id*           string
    created_at*   string
    updated_at*   string
    snapshot_id   string
    snapshot_url  string    1 to 65536 chars
    origin {

```

```

        plan_id            integer
        plan_name          string
        place_id          integer
        place_name         string
        place_description  string
        place_latitude    number
        place_longitude    number
    }
    plan_id                integer
}]
}

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
      loc*  [string]
      msg*  string
      type* string
    }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

## 2.2 POST /api/v1/cameras

### Create new camera

#### REQUEST

##### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
with_origin	boolean	

##### REQUEST BODY - application/json

```

{
  name*           string    1 to 255 chars
  source* {
  }
  video_analytics {
  }
}

```

```

video_storage {
}
video_streaming {
}
ptz {
}
place_id*      integer
onvif_host     string  1 to 255 chars
onvif_port     integer
onvif_login    string  1 to 255 chars
onvif_password string  1 to 255 chars
scheduler_group string  1 to 64 chars
scheduler_tags [string] 1 to 32 chars
}

```

## RESPONSE

STATUS CODE - 200: Created camera

RESPONSE MODEL - application/json

```

{
  name*          string  1 to 255 chars
  source* {
  }
  video_analytics {
  }
  video_storage {
  }
  video_streaming {
  }
  ptz {
  }
  place_id*      integer
  onvif_host     string  1 to 255 chars
  onvif_port     integer
  onvif_login    string  1 to 255 chars
  onvif_password string  1 to 255 chars
  scheduler_group string  1 to 64 chars
  scheduler_tags [string] 1 to 32 chars
  id*           string
  created_at*   string
  updated_at*   string
  snapshot_id   string
  snapshot_url  string  1 to 65536 chars
  origin {
    plan_id      integer
    plan_name    string
    place_id     integer
    place_name   string
    place_description string
    place_latitude number
    place_longitude number
  }
  plan_id        integer
}

```

STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}
```

#### STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

#### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

---

## 2.3 POST /api/v1/cameras/search

### Search cameras by provided query string

Search cameras by ID or name

If UUID provided via query then camera with the exact ID is returned, otherwise the list of case-insensitive name inclusion matches is returned.

## REQUEST

### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
with_origin	boolean	
group_id	integer	
cursor	string	
limit	integer	between 10 and 1000

### REQUEST BODY - application/json

```
{
  query* string
}
```

## RESPONSE

### STATUS CODE - 200: List of found cameras

#### RESPONSE MODEL - application/json

```
{
  pagination* {
```



```

    cursor string
  }
  cameras* [{
    Array of object:
    name*      string    1 to 255 chars
    source* {
    }
    video_analytics {
    }
    video_storage {
    }
    video_streaming {
    }
    ptz {
    }
    place_id*  integer
    onvif_host string    1 to 255 chars
    onvif_port integer
    onvif_login string    1 to 255 chars
    onvif_password string  1 to 255 chars
    scheduler_group string  1 to 64 chars
    scheduler_tags [string] 1 to 32 chars
    id*        string
    created_at* string
    updated_at* string
    snapshot_id string
    snapshot_url string    1 to 65536 chars
    origin {
      plan_id      integer
      plan_name     string
      place_id     integer
      place_name    string
      place_description string
      place_latitude number
      place_longitude number
    }
    plan_id      integer
  }]
}

```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

## 2.4 GET /api/v1/cameras/{camera\_id}

## Get camera with provided ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
with_origin	boolean	

### RESPONSE

STATUS CODE - 200: Requested camera

RESPONSE MODEL - application/json

```
{
  name*           string    1 to 255 chars
  source* {
  }
  video_analytics {
  }
  video_storage {
  }
  video_streaming {
  }
  ptz {
  }
  place_id*       integer
  onvif_host      string    1 to 255 chars
  onvif_port      integer
  onvif_login     string    1 to 255 chars
  onvif_password  string    1 to 255 chars
  scheduler_group string    1 to 64 chars
  scheduler_tags  [string] 1 to 32 chars
  id*             string
  created_at*     string
  updated_at*     string
  snapshot_id     string
  snapshot_url    string    1 to 65536 chars
  origin {
    plan_id       integer
    plan_name     string
    place_id      integer
    place_name    string
    place_description string
    place_latitude number
    place_longitude number
  }
  plan_id         integer
}
```

STATUS CODE - 404: Not Found

### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

### STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

---

## 2.5 PUT /api/v1/cameras/{camera\_id}

### Update camera with provided ID

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

##### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
with_origin	boolean	

##### REQUEST BODY - application/json

```
{
  name*           string    1 to 255 chars
  source* {
  }
  video_analytics {
  }
  video_storage {
  }
  video_streaming {
  }
  ptz {
  }
  place_id*       integer
  onvif_host      string    1 to 255 chars
}
```

```

onvif_port      integer
onvif_login     string    1 to 255 chars
onvif_password  string    1 to 255 chars
scheduler_group string    1 to 64 chars
scheduler_tags  [string] 1 to 32 chars
}

```

## RESPONSE

STATUS CODE - 200: Updated camera

RESPONSE MODEL - application/json

```

{
  name*          string    1 to 255 chars
  source* {
  }
  video_analytics {
  }
  video_storage {
  }
  video_streaming {
  }
  ptz {
  }
  place_id*      integer
  onvif_host     string    1 to 255 chars
  onvif_port     integer
  onvif_login    string    1 to 255 chars
  onvif_password string    1 to 255 chars
  scheduler_group string    1 to 64 chars
  scheduler_tags [string] 1 to 32 chars
  id*           string
  created_at*   string
  updated_at*   string
  snapshot_id   string
  snapshot_url  string    1 to 65536 chars
  origin {
    plan_id      integer
    plan_name    string
    place_id     integer
    place_name   string
    place_description string
    place_latitude number
    place_longitude number
  }
  plan_id        integer
}

```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```

{
  status          enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code     integer   DEFAULT:400
  detail*        string
}

```

## STATUS CODE - 404: Not Found

### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

## STATUS CODE - 422: Validation Error

### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

## STATUS CODE - 500: Internal Server Error

### RESPONSE MODEL - application/json

---

## 2.6 DELETE /api/v1/cameras/{camera\_id}

### Delete specified camera

Delete specified camera by ID

It will automatically remove the camera from the plan it is attached to before deleting it.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

---

## RESPONSE

### STATUS CODE - 200: Camera successfully deleted

#### RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
}
```

```

                                An enumeration.
status_code integer DEFAULT:404
detail*     string
}

```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

---

# 3. EVENTS

## 3.1 GET /api/v1/events/{event\_id}

### Get recognition with provided ID

Get recognition by ID

Event type must be either 'face\_recognition', 'plate\_recognition', 'motion\_detection' or 'line\_crossing'.

Origin of event can be returned via setting 'with\_origin' query parameter to 'true'.

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*event_id	uuid	

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*event_type	enum	ALLOWED: face_recognition, plate_recognition, motion_detection, line_crossing
with_origin	boolean	

### RESPONSE

STATUS CODE - 200: Requested event

RESPONSE MODEL - application/json

```
ANY OF
OPTION 1 {
  origin {
    plan_id          integer
    plan_name        string
    place_id         integer
    place_name       string
    place_description string
    place_latitude   number
    place_longitude  number
    camera_id        string
    camera_name      string
  }
  streamserver_id* string
  stream_id*       string
  object_id*       string
  event_id*        string
  event_action*    string
  ts*              integer
  event_data* {
    detection* {
      bbox* [number]
    }
  }
}
```

```

    descriptor* {
        data* string
        norm* number
    }
    attributes {
    }
}
event_type*      enum      ALLOWED:face_recognition
}
OPTION 2{
    origin {
        plan_id          integer
        plan_name        string
        place_id         integer
        place_name       string
        place_description string
        place_latitude   number
        place_longitude  number
        camera_id        string
        camera_name      string
    }
    streamserver_id* string
    stream_id*       string
    object_id*       string
    event_id*        string
    event_action*    string
    ts*              integer
    event_data* {
        plate_number* {
            data* string
        }
        detection* {
            bbox* [number]
        }
    }
}
event_type*      enum      ALLOWED:plate_recognition
}
OPTION 3{
    origin {
        plan_id          integer
        plan_name        string
        place_id         integer
        place_name       string
        place_description string
        place_latitude   number
        place_longitude  number
        camera_id        string
        camera_name      string
    }
    streamserver_id* string
    stream_id*       string
    object_id*       string
    event_id*        string
    event_action*    string
    ts*              integer
    event_data* {
        motion* {

```



```

        roi_bbox* [number]
        roi_name* string
        area_fr* number
    }
    detection* {
        bbox* [number]
    }
    attributes {
        obj_class* enum ALLOWED:0, 1, 2
                        An enumeration.
    }
}
event_type*      enum      ALLOWED:motion_detection
}
OPTION 4{
    origin {
        plan_id          integer
        plan_name        string
        place_id         integer
        place_name       string
        place_description string
        place_latitude   number
        place_longitude  number
        camera_id        string
        camera_name      string
    }
    streamserver_id* string
    stream_id*       string
    object_id*       string
    event_id*        string
    event_action*    string
    ts*              integer
    event_data* {
        line_crossing* {
            line_coords* [number]
            direction*   enum      ALLOWED:0, 1, 2
                            An enumeration.
            line_name*   string
        }
        detection* {
            bbox* [number]
        }
        attributes {
            obj_class* enum ALLOWED:0, 1, 2
                            An enumeration.
        }
    }
}
event_type*      enum      ALLOWED:line_crossing
}

```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```

{
    status          enum      DEFAULT:invalid_request
                            ALLOWED:invalid_request
                            An enumeration.
    status_code    integer    DEFAULT:400
}

```

```
    detail*    string
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*    string
}
```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

---

## 3.2 GET /api/v1/events

### Get recognition events in time period

Get recognitions occurred in time period optionally filtered by stream or place ID

Delta between `ts_from` and `ts_to` cannot exceed 24 hours limit.

You can only provide either `stream_id` or `place_id` or `group_id` or none.

Event type must be either 'face\_recognition', 'plate\_recognition', 'motion\_detection' or 'line\_crossing'.

By default number of results is limited to 20. The limit cannot be lower than 10 or higher than 100.

Cursor to fetch the next page is returned in `Pagination-Cursor` header.

## REQUEST

### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*ts_from	date-time	
*ts_to	date-time	

NAME	TYPE	DESCRIPTION
*event_type	enum ALLOWED: face_recognition, plate_recognition, motion_detection, line_crossing	
stream_id	uuid	
place_id	integer	
group_id	integer	
with_origin	boolean	
cursor	string	
limit	integer between 10 and 1000	

## RESPONSE

STATUS CODE - 200: Requested events

RESPONSE MODEL - application/json

ANY OF

OPTION 1{

```
  pagination* {
    cursor string
  }
```

```
  events* [{
```

Array of object:

```
    origin {
      plan_id integer
      plan_name string
      place_id integer
      place_name string
      place_description string
      place_latitude number
      place_longitude number
      camera_id string
      camera_name string
    }
```

```
    streamserver_id* string
    stream_id* string
    object_id* string
    event_id* string
    event_action* string
    ts* integer
```

```
    event_data* {
      plate_number* {
        data* string
      }
      detection* {
        bbox* [number]
      }
    }
```

```
    event_type* enum ALLOWED:plate_recognition
```

```
  ]]
```

```
}
```

OPTION 2{

```
  pagination* {
    cursor string
```

```

}
events* [{
Array of object:
  origin {
    plan_id          integer
    plan_name        string
    place_id         integer
    place_name       string
    place_description string
    place_latitude   number
    place_longitude  number
    camera_id        string
    camera_name      string
  }
  streamserver_id* string
  stream_id*        string
  object_id*        string
  event_id*         string
  event_action*     string
  ts*               integer
  event_data* {
    detection* {
      bbox* [number]
    }
    descriptor {
      data* string
      norm* number
    }
    attributes {
    }
  }
  event_type*       enum    ALLOWED:face_recognition
}]
}

```

### OPTION 3{

```

  pagination* {
    cursor string
  }
  events* [{
Array of object:
    origin {
      plan_id          integer
      plan_name        string
      place_id         integer
      place_name       string
      place_description string
      place_latitude   number
      place_longitude  number
      camera_id        string
      camera_name      string
    }
    streamserver_id* string
    stream_id*        string
    object_id*        string
    event_id*         string
    event_action*     string
    ts*               integer
  }

```

```

event_data* {
  motion* {
    roi_bbox* [number]
    roi_name* string
    area_fr* number
  }
  detection* {
    bbox* [number]
  }
  attributes {
    obj_class* enum ALLOWED:0, 1, 2
                    An enumeration.
  }
}
event_type*      enum      ALLOWED:motion_detection
}]
}
OPTION 4{
  pagination* {
    cursor string
  }
  events* [{
    Array of object:
    origin {
      plan_id          integer
      plan_name        string
      place_id         integer
      place_name       string
      place_description string
      place_latitude   number
      place_longitude  number
      camera_id        string
      camera_name      string
    }
    streamserver_id* string
    stream_id*       string
    object_id*       string
    event_id*        string
    event_action*    string
    ts*              integer
    event_data* {
      line_crossing* {
        line_coords* [number]
        direction*   enum      ALLOWED:0, 1, 2
                        An enumeration.
        line_name*   string
      }
      detection* {
        bbox* [number]
      }
      attributes {
        obj_class* enum ALLOWED:0, 1, 2
                        An enumeration.
      }
    }
  }
  event_type*      enum      ALLOWED:line_crossing
}]

```

```
}
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

---

### 3.3 GET /api/v1/eventmatches/{event\_id}

#### Get event match with provided ID

Get event matches by ID

Event type must be either 'face\_recognition' or 'plate\_recognition'.

Origin of event match can be returned via setting 'with\_origin' query parameter to 'true'.

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*event_id	uuid	

---

##### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*event_type	enum	
	ALLOWED:	face_recognition, plate_recognition, motion_detection, line_crossing
with_origin	boolean	

---

#### RESPONSE

**STATUS CODE - 200: Requested event**

**RESPONSE MODEL - application/json**

ANY OF

OPTION 1{

```
origin {
  plan_id          integer
  plan_name        string
  place_id         integer
  place_name       string
  place_description string
  place_latitude   number
  place_longitude  number
  camera_id        string
  camera_name      string
}
streamserver_id*  string
stream_id*        string
object_id*        string
event_id*         string
event_action*     string
ts*               integer
event_data* {
  detection* {
    bbox* [number]
  }
  descriptor* {
    data* string
    norm* number
  }
  attributes {
  }
}
event_type*       enum      ALLOWED:face_recognition
match_data [{
  Array of object:
    face_id*       integer
    facelist_id*   integer
    similarity*    number   between -1 and 1
    face_name      string
    facelist_name* string
    priority       enum     ALLOWED:low, medium, high
                      An enumeration.
}]
}
```

OPTION 2{

```
origin {
  plan_id          integer
  plan_name        string
  place_id         integer
  place_name       string
  place_description string
  place_latitude   number
  place_longitude  number
  camera_id        string
  camera_name      string
}
streamserver_id*  string
stream_id*        string
object_id*        string
event_id*         string
```

```

event_action*    string
ts*              integer
event_data* {
  plate_number* {
    data* string
  }
  detection* {
    bbox* [number]
  }
}
event_type*      enum      ALLOWED:plate_recognition
match_data [{
  Array of object:
    plate_id*     integer
    platelist_id* integer
    plate_number  string
    similarity    number   between -1 and 1
    platelist_name* string
    priority      enum     ALLOWED:low, medium, high
                        An enumeration.
}]
}

```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:invalid_request
                        ALLOWED:invalid_request
                        An enumeration.
  status_code integer   DEFAULT:400
  detail*     string
}

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:not_found
                        ALLOWED:not_found
                        An enumeration.
  status_code integer   DEFAULT:404
  detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
  Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}

```

#### STATUS CODE - 500: Internal Server Error



### 3.4 GET /api/v1/eventmatches

#### Get event matches in time period

Get event matches occurred in time period optionally filtered by stream or place ID

Delta between `ts_from` and `ts_to` cannot exceed 24 hours limit.

Event type must be either 'face\_recognition' or 'plate\_recognition'.

You can only provide either `stream_id` or `place_id` or `group_id` or none.

By default number of results is limited to 20.  
The limit cannot be lower than 10 or higher than 100.

Cursor to fetch the next page is returned in `Pagination-Cursor` header.

#### REQUEST

##### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
* <code>ts_from</code>	date-time	
* <code>ts_to</code>	date-time	
* <code>event_type</code>	enum ALLOWED: face_recognition, plate_recognition, motion_detection, line_crossing	
<code>stream_id</code>	uuid	
<code>place_id</code>	integer	
<code>group_id</code>	integer	
<code>with_origin</code>	boolean	
<code>cursor</code>	string	
<code>limit</code>	integer between 10 and 1000	

#### RESPONSE

STATUS CODE - 200: Requested events

##### RESPONSE MODEL - application/json

###### ANY OF

###### OPTION 1 {

```

pagination* {
  cursor string
}

```

```

events* [{

```

Array of object:

```

  origin {
    plan_id           integer
    plan_name        string
    place_id         integer
  }
}

```

```

        place_name      string
        place_description string
        place_latitude  number
        place_longitude number
        camera_id       string
        camera_name     string
    }
    streamserver_id*   string
    stream_id*         string
    object_id*         string
    event_id*          string
    event_action*      string
    ts*                integer
    event_data* {
        detection* {
            bbox* [number]
        }
        descriptor {
            data* string
            norm* number
        }
        attributes {
        }
    }
    event_type*        enum        ALLOWED:face_recognition
    match_data [{
    Array of object:
        face_id*        integer
        facelist_id*    integer
        similarity*     number    between -1 and 1
        face_name       string
        facelist_name*  string
        priority        enum        ALLOWED:low, medium, high
                                An enumeration.
    }]
    }]
}
OPTION 2{
    pagination* {
        cursor string
    }
    events* [{
    Array of object:
        origin {
            plan_id      integer
            plan_name    string
            place_id     integer
            place_name   string
            place_description string
            place_latitude number
            place_longitude number
            camera_id    string
            camera_name  string
        }
        streamserver_id* string
        stream_id*       string
        object_id*       string

```

```

    event_id*      string
    event_action*  string
    ts*           integer
    event_data* {
      plate_number* {
        data* string
      }
      detection* {
        bbox* [number]
      }
    }
    event_type*    enum      ALLOWED:plate_recognition
    match_data [{
      Array of object:
        plate_id*      integer
        platelist_id*  integer
        plate_number    string
        similarity      number  between -1 and 1
        platelist_name* string
        priority        enum    ALLOWED:low, medium, high
                          An enumeration.
    }]
  }]
}

```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:invalid_request
                          ALLOWED:invalid_request
                          An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

# 4. FACELISTS

## 4.1 GET /api/v1/facelists

Get all face lists, optionally filtered by name

Get all face lists.

If list name was provided, it will return only face lists that match the name exactly.

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
name	string	

### RESPONSE

STATUS CODE - 200: Face lists

RESPONSE MODEL - application/json

```
{
  facelists* [{
    Array of object:
    name*      string  1 to 255 chars
    description string  max:2048 chars
    priority   enum    ALLOWED:low, medium, high
                An enumeration.
    id*        integer
    face_count* integer
  }]
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

## 4.2 POST /api/v1/facelists

Create new face list

Create the face list with provided data.

It will return 400 BAD REQUEST if the face list with provided name already exists.

## REQUEST

### REQUEST BODY - application/json

```
{
  name*      string 1 to 255 chars
  description string max:2048 chars
  priority   enum    ALLOWED:low, medium, high
                        An enumeration.
}
```

## RESPONSE

### STATUS CODE - 200: Created face list

#### RESPONSE MODEL - application/json

```
{
  name*      string 1 to 255 chars
  description string max:2048 chars
  priority   enum    ALLOWED:low, medium, high
                        An enumeration.
  id*        integer
  face_count* integer
}
```

### STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  status      enum    DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer DEFAULT:400
  detail*     string
}
```

### STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

---

## 4.3 GET /api/v1/facelists/{facelist\_id}

### Get face list with provided ID

Get the face list with provided ID.

It will return 404 NOT FOUND if the face list with provided ID was not found.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*facelist_id	integer	

## RESPONSE

**STATUS CODE - 200:** Requested face list

### RESPONSE MODEL - application/json

```
{
  name*      string  1 to 255 chars
  description string  max:2048 chars
  priority    enum    ALLOWED:low, medium, high
                An enumeration.
  id*        integer
  face_count* integer
}
```

**STATUS CODE - 404:** Not Found

### RESPONSE MODEL - application/json

```
{
  status      enum    DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

### RESPONSE MODEL - application/json

---

## 4.4 PUT /api/v1/facelists/{facelist\_id}

### Update face list with provided ID

Update the face list with provided ID.

It will return 400 BAD REQUEST if the face list with provided name already exists.

It will return 404 NOT FOUND if the face list with provided ID was not found.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*facelist_id	integer	

### REQUEST BODY - application/json

```
{
  name*      string 1 to 255 chars
  description string max:2048 chars
  priority    enum   ALLOWED:low, medium, high
                An enumeration.
}
```

## RESPONSE

### STATUS CODE - 200: Updated face list

#### RESPONSE MODEL - application/json

```
{
  name*      string 1 to 255 chars
  description string max:2048 chars
  priority    enum   ALLOWED:low, medium, high
                An enumeration.
  id*        integer
  face_count* integer
}
```

### STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  status      enum   DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer DEFAULT:400
  detail*     string
}
```

### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  status      enum   DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer DEFAULT:404
  detail*     string
}
```

### STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
```

```
    loc*  [string]
    msg*  string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

---

## 4.5 DELETE /api/v1/facelists/{facelist\_id}

### Delete specified face list

Delete the face list with provided ID.

It will return 400 BAD REQUEST if the face list has faces.

It will return 404 NOT FOUND if the face list with provided ID was not found.

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*facelist_id	integer	

---

### RESPONSE

**STATUS CODE - 200:** Face list successfully deleted

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error



## RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

## RESPONSE MODEL - application/json

## 4.6 GET /api/v1/facelists/{facelist\_id}/faces

### Get faces of the list

Get list of faces in a face list with provided ID.

It will return 404 NOT FOUND if the face list with provided ID was not found.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*facelist_id	integer	

## RESPONSE

STATUS CODE - 200: Faces attached to the specified list

### RESPONSE MODEL - application/json

```
{
  faces* [{
    Array of object:
    full_name string max:128 chars
    sex enum ALLOWED:m, f
    An enumeration.
    birth_date string
    document_number string max:128 chars
    id* integer
    images* [{
      Array of object:
      mime_type* string
      kind* enum ALLOWED:avatar, face_image, vehicle_image, plan,
      camera_snapshot, logo
      An enumeration.
      original_id string
      id* string
      created_at* string
      url string 1 to 65536 chars
      meta {
        detection {
          rect* {
            x_min* number
```

```

        y_min* number
        x_max* number
        y_max* number
    }
    landmarks* [{
        Array of object:
        x* number
        y* number
    }]
}
descriptor {
    data* string
    norm* number
}
}
}]
}]
}
}
}
}
}

```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```

{
  status           enum           DEFAULT:not_found
                                   ALLOWED:not_found
                                   An enumeration.
  status_code      integer        DEFAULT:404
  detail*          string
}

```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

## 4.7 POST /api/v1/facelists/{facelist\_id}/faces/{face\_id}

### Add face to the list

Add a face to a face list with provided ID.

It will return 400 BAD REQUEST if a face is already in a face list.

It will return 404 NOT FOUND if a face or face list with provided ID was not found.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*facelist_id	integer	
*face_id	integer	

## RESPONSE

**STATUS CODE - 200:** Face list face successfully added

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

## 4.8 DELETE /api/v1/facelists/{facelist\_id}/faces/{face\_id}

### Remove face from the list

Delete a face from a face list with provided ID.

It will return 404 NOT FOUND if a face or face list with provided ID was not found.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*facelist_id	integer	
*face_id	integer	

## RESPONSE

**STATUS CODE - 200:** Face list face successfully deleted

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status          enum          DEFAULT:not_found
                                ALLOWED:not_found
                                An enumeration.
  status_code    integer       DEFAULT:404
  detail*        string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

# 5. FACES

## 5.1 GET /api/v1/faces

### Get all faces optionally filtered by full name

Returns list of all the faces.

Result may be optionally filtered by full name.

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
full_name	string	

### RESPONSE

STATUS CODE - 200: List of all faces

#### RESPONSE MODEL - application/json

```
{
  faces* [{
    Array of object:
    full_name string max:128 chars
    sex enum ALLOWED:m, f
    An enumeration.
    birth_date string
    document_number string max:128 chars
    id* integer
    images* [{
      Array of object:
      mime_type* string
      kind* enum ALLOWED:avatar, face_image, vehicle_image, plan,
      camera_snapshot, logo
      An enumeration.
      original_id string
      id* string
      created_at* string
      url string 1 to 65536 chars
      meta {
        detection {
          rect* {
            x_min* number
            y_min* number
            x_max* number
            y_max* number
          }
          landmarks* [{
            Array of object:
            x* number
            y* number
          }]
        }
      }
    }
  }
}
```

```

        descriptor {
            data* string
            norm* number
        }
    }
}]]
}

```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

---

## 5.2 POST /api/v1/faces

### Create new face

Create new face

Date must be provided in ISO 8601 format:  
YYYY-MM-DD (i.e. 2012-07-21)

### REQUEST

**REQUEST BODY - application/json**

```

{
  full_name      string max:128 chars
  sex            enum    ALLOWED:m, f
                  An enumeration.
  birth_date     string
  document_number string max:128 chars
}

```

### RESPONSE

**STATUS CODE - 200: Created face**

**RESPONSE MODEL - application/json**

```

{
  full_name      string          max:128 chars
  sex            enum            ALLOWED:m, f
                  An enumeration.
  birth_date     string
  document_number string          max:128 chars
  id*           integer
  images* [{
    Array of object:

```

```

mime_type*  string
kind*       enum    ALLOWED:avatar, face_image, vehicle_image, plan,
                  camera_snapshot, logo
                  An enumeration.

original_id string
id*         string
created_at* string
url         string 1 to 65536 chars
meta {
  detection {
    rect* {
      x_min* number
      y_min* number
      x_max* number
      y_max* number
    }
    landmarks* [{
      Array of object:
      x* number
      y* number
    }]
  }
  descriptor {
    data* string
    norm* number
  }
}
}]
}

```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

### 5.3 GET /api/v1/faces/{face\_id}

#### Get face with provided ID

Get the face with provided ID.

It will return 404 NOT FOUND if the face with provided ID was not found.

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
------	------	-------------

NAME	TYPE	DESCRIPTION
*face_id	integer	

## RESPONSE

STATUS CODE - 200: Requested face

RESPONSE MODEL - application/json

```

{
  full_name      string          max:128 chars
  sex            enum            ALLOWED:m, f
                                   An enumeration.
  birth_date     string
  document_number string          max:128 chars
  id*            integer
  images* [{
    Array of object:
      mime_type* string
      kind*      enum    ALLOWED:avatar, face_image, vehicle_image, plan,
                        camera_snapshot, logo
                        An enumeration.
      original_id string
      id*         string
      created_at* string
      url         string 1 to 65536 chars
      meta {
        detection {
          rect* {
            x_min* number
            y_min* number
            x_max* number
            y_max* number
          }
          landmarks* [{
            Array of object:
              x* number
              y* number
            }]
        }
        descriptor {
          data* string
          norm* number
        }
      }
    }
  }]
}

```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```

{
  status      enum    DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
  detail*     string

```



```
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 5.4 PUT /api/v1/faces/{face\_id}

### Update face with provided ID

Update existing face

Date must be provided in ISO 8601 format:  
YYYY-MM-DD (i.e. 2012-07-21)

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*face_id	integer	

### REQUEST BODY - application/json

```
{
  full_name      string max:128 chars
  sex            enum   ALLOWED:m, f
                An enumeration.
  birth_date     string
  document_number string max:128 chars
}
```

## RESPONSE

STATUS CODE - 200: Updated face

RESPONSE MODEL - application/json

```
{
  full_name      string          max:128 chars
  sex            enum           ALLOWED:m, f
                An enumeration.
  birth_date     string
  document_number string          max:128 chars
  id*           integer
  images* [{
    Array of object:
    mime_type*  string
```

```

kind*          enum    ALLOWED:avatar, face_image, vehicle_image, plan,
                    camera_snapshot, logo
                    An enumeration.

original_id    string
id*           string
created_at*   string
url           string 1 to 65536 chars
meta {
  detection {
    rect* {
      x_min*   number
      y_min*   number
      x_max*   number
      y_max*   number
    }
    landmarks* [{
      Array of object:
      x*       number
      y*       number
    }]
  }
  descriptor {
    data*     string
    norm*     number
  }
}
}]
}

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
  status          enum    DEFAULT:not_found
                    ALLOWED:not_found
                    An enumeration.
  status_code     integer  DEFAULT:404
  detail*        string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

---

## 5.5 DELETE /api/v1/faces/{face\_id}

### Delete face with provided ID

Delete the face with provided ID.

It will return 404 NOT FOUND if the face with provided ID was not found.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*face_id	integer	

## RESPONSE

**STATUS CODE - 200:** Successful Response

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

## 5.6 GET /api/v1/faces/{face\_id}/images

**Get the list of images attached to a face with provided ID**

Get the list of images attached to a face with provided ID.

It will return 404 NOT FOUND if a face with provided ID was not found.

## REQUEST

## PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*face_id	integer	

## RESPONSE

**STATUS CODE - 200:** List of images attached to the specified face

**RESPONSE MODEL - application/json**

```
{
  faceimages* [{
    Array of object:
    mime_type* string
    kind* enum ALLOWED:avatar, face_image, vehicle_image, plan, camera_snapshot, logo
    An enumeration.
    original_id string
    id* string
    created_at* string
    url string 1 to 65536 chars
    meta {
      detection {
        rect* {
          x_min* number
          y_min* number
          x_max* number
          y_max* number
        }
        landmarks* [{
          Array of object:
          x* number
          y* number
        }]
      }
      descriptor {
        data* string
        norm* number
      }
    }
  }]
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status enum DEFAULT:not_found
  ALLOWED:not_found
  An enumeration.
  status_code integer DEFAULT:404
  detail* string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}

```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

## 5.7 POST /api/v1/faces/{face\_id}/images

### Add provided images to the face with provided ID

Add provided images to the face with provided ID.

It will return 400 BAD REQUEST if some of the provided images are already attached to face.

It will return 404 NOT FOUND if some of the provided images were not found.

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*face_id	integer	

#### REQUEST BODY - application/json

```

{
  image_ids* [string]
}

```

### RESPONSE

**STATUS CODE - 200:** List of images attached to the specified face

**RESPONSE MODEL - application/json**

```

{
  faceimages* [{
    Array of object:
    mime_type* string
    kind* enum ALLOWED:avatar, face_image, vehicle_image, plan,
               camera_snapshot, logo
               An enumeration.
    original_id string
    id* string
    created_at* string
    url string 1 to 65536 chars
    meta {
      detection {
        rect* {
          x_min* number
          y_min* number
          x_max* number
          y_max* number
        }
      }
    }
  }]
}

```

```

        landmarks* [{
          Array of object:
            x* number
            y* number
          }]
      }
      descriptor {
        data* string
        norm* number
      }
    }
  }]
}

```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

## 5.8 DELETE /api/v1/faces/{face\_id}/images

### Detach provided images from the face with provided ID

Remove provided images from the face with provided ID.

It will return 404 NOT FOUND if a face with provided ID was not found.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*face_id	integer	

### REQUEST BODY - application/json

```
{
  image_ids* [string]
}
```

## RESPONSE

STATUS CODE - 200: List of images attached to the specified face

### RESPONSE MODEL - application/json

```
{
  faceimages* [{
    Array of object:
    mime_type* string
    kind* enum ALLOWED:avatar, face_image, vehicle_image, plan,
               camera_snapshot, logo
               An enumeration.
    original_id string
    id* string
    created_at* string
    url string 1 to 65536 chars
    meta {
      detection {
        rect* {
          x_min* number
          y_min* number
          x_max* number
          y_max* number
        }
        landmarks* [{
          Array of object:
          x* number
          y* number
        }]
      }
      descriptor {
        data* string
        norm* number
      }
    }
  }]
}
```

STATUS CODE - 404: Not Found

### RESPONSE MODEL - application/json

```
{
  status enum DEFAULT:not_found
           ALLOWED:not_found
           An enumeration.
}
```

```
status_code integer DEFAULT:404
detail*     string
}
```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

---



# 6. GROUPS

## 6.1 GET /api/v1/groups

Get list of groups

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
cursor	string	
limit	integer	between 10 and 1000

### RESPONSE

STATUS CODE - 200: List of groups

RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
  groups* [{
    Array of object:
    name*          string  1 to 64 chars
    description*   string  max:2048 chars
    id*            integer
    organization_id* integer
    organization_name* string
    is_internal*   boolean
    user_count*    integer
    place_count*   integer
  }]
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

## 6.2 POST /api/v1/groups

## Create new group

Create the new group

Only superuser can explicitly set the organization ID.  
Otherwise it will be inferred from the current user.

Returns 400 BAD REQUEST in the following cases:

- Provided organization does not exist
- Organization was provided despite the user is not superuser

## REQUEST

### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
organization_id	integer	

### REQUEST BODY - application/json

```
{
  name*           string 1 to 64 chars
  description*    string max:2048 chars
}
```

## RESPONSE

### STATUS CODE - 200: Created group

#### RESPONSE MODEL - application/json

```
{
  name*           string 1 to 64 chars
  description*    string max:2048 chars
  id*            integer
  organization_id* integer
  organization_name* string
  is_internal*    boolean
  user_count*     integer
  place_count*    integer
}
```

### STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  status          enum    DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code     integer DEFAULT:400
  detail*        string
}
```

### STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
  }]
```

```
    type* string
  }}
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 6.3 GET /api/v1/groups/{group\_id}

Get group with provided ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*group_id	integer	

---

### RESPONSE

STATUS CODE - 200: Requested group

RESPONSE MODEL - application/json

```
{
  name*           string  1 to 64 chars
  description*    string  max:2048 chars
  id*             integer
  organization_id* integer
  organization_name* string
  is_internal*    boolean
  user_count*     integer
  place_count*    integer
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```
{
  status          enum      DEFAULT:not_found
                                ALLOWED:not_found
                                An enumeration.
  status_code     integer   DEFAULT:404
  detail*         string
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 6.4 PUT /api/v1/groups/{group\_id}

Update group with provided ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*group_id	integer	

---

#### REQUEST BODY - application/json

```
{
  name*           string 1 to 64 chars
  description*    string max:2048 chars
}
```

### RESPONSE

STATUS CODE - 200: Updated group

#### RESPONSE MODEL - application/json

```
{
  name*           string 1 to 64 chars
  description*    string max:2048 chars
  id*            integer
  organization_id* integer
  organization_name* string
  is_internal*   boolean
  user_count*    integer
  place_count*   integer
}
```

STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  status          enum    DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code     integer DEFAULT:400
  detail*        string
}
```

STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  status          enum    DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code     integer DEFAULT:404
  detail*        string
}
```

```
}
```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

---

## 6.5 DELETE /api/v1/groups/{group\_id}

### Delete group with provided ID

Delete the group with provided ID.

It will return 400 BAD REQUEST if the group has users.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*group_id	integer	

---

## RESPONSE

**STATUS CODE - 200: Group successfully deleted**

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  status          enum          DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code    integer       DEFAULT:400
  detail*        string
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  status          enum          DEFAULT:not_found
}
```

```

        ALLOWED: not_found
        An enumeration.
    status_code integer DEFAULT: 404
    detail*      string
}

```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

## 6.6 GET /api/v1/groups/{group\_id}/members

Get list of group members

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*group_id	integer	

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
cursor	string	
limit	integer	between 10 and 1000

### RESPONSE

**STATUS CODE - 200: List of group members**

**RESPONSE MODEL - application/json**

```

{
  pagination* {
    cursor string
  }
  users* [{
    Array of object:
    first_name* string max:32 chars
    last_name*  string max:32 chars
    avatar_id   string
    email*     string
    id*        integer
  }]
}

```

```

organization_id* integer
organization_name* string
avatar_url string 1 to 65536 chars
groups* [{
  Array of object:
    id* integer
    name* string
  }]
is_superuser* boolean
role_id* integer
}]
}

```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```

{
  status enum DEFAULT:not_found
  ALLOWED:not_found
  An enumeration.
  status_code integer DEFAULT:404
  detail* string
}

```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

## 6.7 POST /api/v1/groups/{group\_id}/members

### Add user to specified group

Add users to the specified group

It will return 400 BAD REQUEST in the following cases:

- \* Any of provided users is already in a group
- \* Any of provided users does not exist

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*group_id	integer	

#### REQUEST BODY - application/json

```
{
  user_ids* [integer]
}
```

## RESPONSE

**STATUS CODE - 200:** Group member successfully added

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

---

## 6.8 DELETE /api/v1/groups/{group\_id}/members

Remove user from specified group

### REQUEST

**PATH PARAMETERS**



NAME	TYPE	DESCRIPTION
*group_id	integer	

REQUEST BODY - application/json

```
{
  user_ids* [integer]
}
```

## RESPONSE

STATUS CODE - 200: Group member successfully deleted

RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

## 6.9 POST /api/v1/groups/search

### Search groups by provided query

Get list of groups filtered by query

Search is done over group name.

## REQUEST

### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
organization_id	integer	
cursor	string	
limit	integer	between 10 and 1000

REQUEST BODY - application/json

```
{
  query* string 2 to 255 chars
}
```

## RESPONSE

STATUS CODE - 200: List of found groups

RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
  groups* [{
    Array of object:
    name*          string  1 to 64 chars
    description*   string  max:2048 chars
    id*            integer
    organization_id* integer
    organization_name* string
    is_internal*   boolean
    user_count*    integer
    place_count*   integer
  }]
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 6.10 GET /api/v1/groups/{group\_id}/places

Get list of group places

### REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*group_id	integer	

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
cursor	string	
limit	integer	between 10 and 1000

---

## RESPONSE

**STATUS CODE - 200:** List of group places

**RESPONSE MODEL - application/json**

```
{
  pagination* {
    cursor string
  }
  places* [{
    Array of object:
    name*      string  1 to 255 chars
    description string  max:2048 chars
    latitude   number
    longitude  number
    id*        integer
    created_at* string
    updated_at* string
    plan_count* integer
    camera_count* integer
  }]
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*  [string]
    msg*  string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

---

### 6.11 POST /api/v1/groups/{group\_id}/places

#### Add place to specified group

Add places to the specified group

It will return 400 BAD REQUEST in the following cases:

\* Provided place already belongs to a group

\* Provided place does not exist

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*group_id	integer	

### REQUEST BODY - application/json

```
{
  place_ids* [integer]
}
```

## RESPONSE

**STATUS CODE - 200:** Group place successfully added

### RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 404:** Not Found

### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

## 6.12 DELETE /api/v1/groups/{group\_id}/places

Remove place from specified group

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*group_id	integer	

---

#### REQUEST BODY - application/json

```
{
  place_ids* [integer]
}
```

### RESPONSE

STATUS CODE - 200: Group place successfully deleted

#### RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

---

# 7. HELLO

## 7.1 GET /api/v1/hello

Hello

### REQUEST

No request parameters

### RESPONSE

**STATUS CODE - 200: Successful Response**

**RESPONSE MODEL - application/json**

```
{  
  message* string  
  version* string  
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

---

# 8. ONVIF

## 8.1 GET /api/v1/onvif/{camera\_id}

### Get camera information

Get camera information.

In addition, the following status codes are defined for 400 response:

- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

### RESPONSE

STATUS CODE - 200: Requested camera information

#### RESPONSE MODEL - application/json

```
{
  manufacturer*   string
  model*         string
  firmware_version* string
  serial_number*  string
  hardware_id*   string
}
```

STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  status          enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code     integer   DEFAULT:400
  detail*        string
}
```

STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  status          enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code     integer   DEFAULT:404
  detail*        string
}
```

STATUS CODE - 422: Validation Error

### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

### RESPONSE MODEL - application/json

STATUS CODE - 503: Service Unavailable

### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                          ALLOWED:temporarily_unavailable
                          An enumeration.
  status_code integer DEFAULT:503
  detail*     string
}
```

---

## 8.2 GET /api/v1/onvif/{camera\_id}/network

### Get camera network settings

Get camera network settings.

In addition, the following status codes are defined for 400 response:

- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

## RESPONSE

STATUS CODE - 200: Requested camera network settings

### RESPONSE MODEL - application/json

```
{
  http {
    port* integer
    enabled* boolean
  }
  https {
    port* integer
    enabled* boolean
  }
}
```



```

    rtsp {
      port*    integer
      enabled* boolean
    }
  }
}

```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

#### STATUS CODE - 503: Service Unavailable

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}

```

## 8.3 PUT /api/v1/onvif/{camera\_id}/network

### Update camera network settings

Update camera network settings.

In addition, the following status codes are defined for 400 response:

- onvif\_invalid\_argument - Invalid or insufficient arguments were provided
- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

### REQUEST BODY - application/json

```
{
  http integer
  https integer
  rtsp integer
}
```

## RESPONSE

**STATUS CODE - 200:** Update camera network settings status

### RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

### RESPONSE MODEL - application/json

```
{
  status enum DEFAULT:invalid_request
  ALLOWED:invalid_request
  An enumeration.
  status_code integer DEFAULT:400
  detail* string
}
```

**STATUS CODE - 404:** Not Found

### RESPONSE MODEL - application/json

```
{
  status enum DEFAULT:not_found
  ALLOWED:not_found
  An enumeration.
  status_code integer DEFAULT:404
  detail* string
}
```

**STATUS CODE - 422:** Validation Error

### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
```

```

    loc*   [string]
    msg*   string
    type*  string
  }
}

```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

**STATUS CODE - 503:** Service Unavailable

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}

```

## 8.4 GET /api/v1/onvif/{camera\_id}/users

### Get camera users information

Get camera users information.

In addition, the following status codes are defined for 400 response:

- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

## RESPONSE

**STATUS CODE - 200:** Requested camera users info

**RESPONSE MODEL - application/json**

```

{
  users* [ {
    Array of object:
    user_name*  string
    user_level* string
  } ]
}

```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
}

```

```

        status_code integer DEFAULT:400
        detail*     string
    }

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer DEFAULT:404
  detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

#### STATUS CODE - 503: Service Unavailable

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer DEFAULT:503
  detail*     string
}

```

## 8.5 PUT /api/v1/onvif/{camera\_id}/users/{username}

### Update camera user password

Update camera user password.

In addition, the following status codes are defined for 400 response:

- onvif\_invalid\_argument - Invalid or insufficient arguments were provided
- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
------	------	-------------

NAME	TYPE	DESCRIPTION
*camera_id	uuid	
*username	string	

**REQUEST BODY - application/json**

```
{
  password* string max:16 chars
}
```

**RESPONSE**

**STATUS CODE - 200:** Update camera user password status

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status          enum          DEFAULT:invalid_request
                                     ALLOWED:invalid_request
                                     An enumeration.
  status_code    integer       DEFAULT:400
  detail*        string
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status          enum          DEFAULT:not_found
                                     ALLOWED:not_found
                                     An enumeration.
  status_code    integer       DEFAULT:404
  detail*        string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

**STATUS CODE - 503:** Service Unavailable

## RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                  ALLOWED:temporarily_unavailable
                  An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}
```

## 8.6 GET /api/v1/onvif/{camera\_id}/streams

### Get camera streams information

Get camera streams information.

In addition, the following status codes are defined for 400 response:

- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

## RESPONSE

STATUS CODE - 200: Requested camera streams info

### RESPONSE MODEL - application/json

```
{
  streams* [{
    Array of object:
    name*      string
    link*      string      1 to 65536 chars
    current_encoder* enum    ALLOWED:H264, MPEG4, JPEG
                  An enumeration.
    current_resolution* enum  ALLOWED:320x240, 352x288, 640x360, 640x480, 704x576,
                  1280x720, 1280x960, 1920x1024, 1920x1080, 2304x1296,
                  2560x1440, 2688x1520
                  An enumeration.
    current_framerate* integer
    current_bitrate* integer
    current_gov_length integer
    supported_encoders* [{
    Array of object:
      name*      string
      supported_resolutions* [{
      Array of object:
        width* integer
        height* integer
      }]
    }
    framerate_range* {
      min_val* number
    }
  }
}
```

```

        max_val* number
    }
    bitrate_range* {
        min_val* number
        max_val* number
    }
    gov_length_range {
        min_val* number
        max_val* number
    }
    }
}]]
}

```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

#### STATUS CODE - 503: Service Unavailable

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
}

```

```

    status_code integer DEFAULT:503
    detail*     string
}

```

## 8.7 PUT /api/v1/onvif/{camera\_id}/stream/encoder

### Update camera stream encoder settings

Update camera stream encoder settings.

In addition, the following status codes are defined for 400 response:

- onvif\_invalid\_argument - Invalid or insufficient arguments were provided
- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

### REQUEST BODY - application/json

```

{
  stream_profile* enum    ALLOWED:MAIN_STREAM, SUB_STREAM_1, SUB_STREAM_2
                    An enumeration.
  encoder*        enum    ALLOWED:H264, MPEG4, JPEG
                    An enumeration.
  resolution*     enum    ALLOWED:320x240, 352x288, 640x360, 640x480, 704x576, 1280x720,
                    1280x960, 1920x1024, 1920x1080, 2304x1296, 2560x1440,
                    2688x1520
                    An enumeration.
  framerate*     integer
  bitrate*       integer
  gov_length     integer
}

```

## RESPONSE

STATUS CODE - 200: Update camera stream encoder settings status

### RESPONSE MODEL - application/json

```

{
  status string DEFAULT:ok
}

```

STATUS CODE - 400: Bad Request

### RESPONSE MODEL - application/json

```

{
  status          enum    DEFAULT:invalid_request
                    ALLOWED:invalid_request
                    An enumeration.
  status_code     integer DEFAULT:400
  detail*         string
}

```



## STATUS CODE - 404: Not Found

### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

## STATUS CODE - 422: Validation Error

### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

## STATUS CODE - 500: Internal Server Error

### RESPONSE MODEL - application/json

## STATUS CODE - 503: Service Unavailable

### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                  ALLOWED:temporarily_unavailable
                  An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}
```

---

## 8.8 GET /api/v1/onvif/{camera\_id}/image

### Get camera image information

Get camera image information.

In addition, the following status codes are defined for 400 response:

- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

## RESPONSE

## STATUS CODE - 200: Requested camera image information

### RESPONSE MODEL - application/json

```
{
  brightness {
    current_value* number
    supported_range* {
      min_val* number
      max_val* number
    }
  }
  color_saturation {
    current_value* number
    supported_range* {
      min_val* number
      max_val* number
    }
  }
  contrast {
    current_value* number
    supported_range* {
      min_val* number
      max_val* number
    }
  }
  sharpness {
    current_value* number
    supported_range* {
      min_val* number
      max_val* number
    }
  }
  backlight_compensation {
    current_mode*      enum      ALLOWED:ON, OFF
                        An enumeration.
    supported_modes*  [enum]    ALLOWED:ON, OFF
                        An enumeration.
  }
  ir_cut_filter {
    current_mode*      enum      ALLOWED:ON, OFF, AUTO
                        An enumeration.
    supported_modes*  [enum]    ALLOWED:ON, OFF, AUTO
                        An enumeration.
  }
  exposure {
    current_range* {
      min_val* number
      max_val* number
    }
    supported_range* {
      min_val* number
      max_val* number
    }
  }
  white_balance {
    current_mode*      enum      ALLOWED:AUTO, MANUAL
                        An enumeration.
    current_cr_gain_value* number
  }
}
```

```

current_cb_gain_value* number
supported_modes*      [enum] ALLOWED:AUTO, MANUAL
                        An enumeration.

supported_range* {
  cr_gain* {
    min_val* number
    max_val* number
  }
  cb_gain* {
    min_val* number
    max_val* number
  }
}
}
wide_dynamic_range {
  current_mode*      enum    ALLOWED:ON, OFF
                        An enumeration.

  current_value*    number

  supported_modes*  [enum]  ALLOWED:ON, OFF
                        An enumeration.

  supported_range* {
    min_val* number
    max_val* number
  }
}
}

```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```

{
  status      enum    DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.

  status_code integer  DEFAULT:400
  detail*    string
}

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
  status      enum    DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.

  status_code integer  DEFAULT:404
  detail*    string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
    loc*  [string]
    msg*  string
    type* string
  }]
}

```

```
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

STATUS CODE - 503: Service Unavailable

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}
```

---

## 8.9 PUT /api/v1/onvif/{camera\_id}/image/brightness

### Update camera image brightness

Update camera image brightness.

In addition, the following status codes are defined for 400 response:

- onvif\_invalid\_argument - Invalid or insufficient arguments were provided
- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

### REQUEST BODY - application/json

```
{
  value* number
}
```

## RESPONSE

STATUS CODE - 200: Updated camera image brightness status

RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer    DEFAULT:400
}
```

```
    detail*    string
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*    string
}
```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
      loc*  [string]
      msg*  string
      type* string
    }]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

#### STATUS CODE - 503: Service Unavailable

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                  ALLOWED:temporarily_unavailable
                  An enumeration.
  status_code integer    DEFAULT:503
  detail*    string
}
```

---

## 8.10 PUT /api/v1/onvif/{camera\_id}/image/contrast

### Update camera image contrast

Update camera image contrast.

In addition, the following status codes are defined for 400 response:

- onvif\_invalid\_argument - Invalid or insufficient arguments were provided
- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

REQUEST BODY - application/json

```
{
  value* number
}
```

## RESPONSE

STATUS CODE - 200: Updated camera image contrast status

RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

STATUS CODE - 503: Service Unavailable

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer  DEFAULT:503
}
```

```
    detail*    string
}
```

---

## 8.11 PUT /api/v1/onvif/{camera\_id}/image/saturation

### Update camera image color saturation

Update camera image color saturation.

In addition, the following status codes are defined for 400 response:

- onvif\_invalid\_argument - Invalid or insufficient arguments were provided
- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

### REQUEST BODY - application/json

```
{
  value* number
}
```

## RESPONSE

**STATUS CODE - 200:** Updated camera image color saturation status

### RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

### RESPONSE MODEL - application/json

```
{
  status          enum          DEFAULT:invalid_request
                                ALLOWED:invalid_request
                                An enumeration.
  status_code     integer       DEFAULT:400
  detail*         string
}
```

**STATUS CODE - 404:** Not Found

### RESPONSE MODEL - application/json

```
{
  status          enum          DEFAULT:not_found
                                ALLOWED:not_found
                                An enumeration.
  status_code     integer       DEFAULT:404
  detail*         string
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

STATUS CODE - 503: Service Unavailable

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer  DEFAULT:503
  detail*     string
}
```

---

## 8.12 PUT /api/v1/onvif/{camera\_id}/image/sharpness

### Update camera image sharpness

Update camera image sharpness.

In addition, the following status codes are defined for 400 response:

- onvif\_invalid\_argument - Invalid or insufficient arguments were provided
- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

### REQUEST BODY - application/json

```
{
  value* number
}
```

## RESPONSE

STATUS CODE - 200: Updated camera image sharpness status

RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```



```
}
```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

#### STATUS CODE - 503: Service Unavailable

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}
```

---

## 8.13 PUT /api/v1/onvif/{camera\_id}/image/backlight

### Update camera image backlight compensation

Update camera image backlight compensation.

In addition, the following status codes are defined for 400 response:

- onvif\_invalid\_argument - Invalid or insufficient arguments were provided
- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

### REQUEST BODY - application/json

```
{
  mode*  enum  ALLOWED:ON, OFF
          An enumeration.
}
```

## RESPONSE

**STATUS CODE - 200:** Updated camera image backlight compensation status

### RESPONSE MODEL - application/json

```
{
  status  string  DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 404:** Not Found

### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
      loc*  [string]
      msg*  string
      type* string
    }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

STATUS CODE - 503: Service Unavailable

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}
```

---

## 8.14 PUT /api/v1/onvif/{camera\_id}/image/ir

### Update camera image IR cut filter

Update camera image IR cut filter.

In addition, the following status codes are defined for 400 response:

- onvif\_invalid\_argument - Invalid or insufficient arguments were provided
- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

### REQUEST BODY - application/json

```
{
  mode* enum  ALLOWED:ON, OFF, AUTO
                An enumeration.
}
```

## RESPONSE

STATUS CODE - 200: Updated camera image IR cut filter status

RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}
```

```
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

#### STATUS CODE - 503: Service Unavailable

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                  ALLOWED:temporarily_unavailable
                  An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}
```

---

## 8.15 PUT /api/v1/onvif/{camera\_id}/image/exposure

### Update camera image exposure

Update camera image exposure.

In addition, the following status codes are defined for 400 response:

- onvif\_invalid\_argument - Invalid or insufficient arguments were provided
- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

REQUEST BODY - application/json

```
{
  value* number
}
```

## RESPONSE

STATUS CODE - 200: Updated camera image exposure status

RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

STATUS CODE - 503: Service Unavailable

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer  DEFAULT:503
}
```

```
    detail*    string
}
```

## 8.16 PUT /api/v1/onvif/{camera\_id}/image/dr

### Update camera image wide dynamic range

Update camera image wide dynamic range.

In addition, the following status codes are defined for 400 response:

- onvif\_invalid\_argument - Invalid or insufficient arguments were provided
- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

### REQUEST BODY - application/json

```
{
  mode    enum    ALLOWED:ON, OFF
           An enumeration.
  value   number
}
```

## RESPONSE

STATUS CODE - 200: Updated camera image wide dynamic range status

### RESPONSE MODEL - application/json

```
{
  status  string  DEFAULT:ok
}
```

STATUS CODE - 400: Bad Request

### RESPONSE MODEL - application/json

```
{
  status      enum    DEFAULT:invalid_request
               ALLOWED:invalid_request
               An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

STATUS CODE - 404: Not Found

### RESPONSE MODEL - application/json

```
{
  status      enum    DEFAULT:not_found
               ALLOWED:not_found
               An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

```
}
```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

**STATUS CODE - 503: Service Unavailable**

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:temporarily_unavailable
                  ALLOWED:temporarily_unavailable
                  An enumeration.
  status_code integer  DEFAULT:503
  detail*     string
}
```

---

## 8.17 PUT /api/v1/onvif/{camera\_id}/image/wb

### Update camera image white balance

Update camera image white balance.

In addition, the following status codes are defined for 400 response:

- onvif\_invalid\_argument - Invalid or insufficient arguments were provided
- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

### REQUEST BODY - application/json

```
{
  mode      enum      ALLOWED:AUTO, MANUAL
                  An enumeration.
  cr_gain   number
  cb_gain   number
}
```

## RESPONSE

**STATUS CODE - 200:** Updated camera image white balance status

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

**STATUS CODE - 503:** Service Unavailable

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}
```

---

## 8.18 GET /api/v1/onvif/{camera\_id}/ptz



## Get camera PTZ

Get camera PTZ.

In addition, the following status codes are defined for 400 response:

- onvif\_unsupported\_action - Action is not supported by the camera
- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

## RESPONSE

STATUS CODE - 200: Requested camera ptz information

RESPONSE MODEL - application/json

```
{
  pan_tilt* {
    current_position {
      x* number
      y* number
    }
    default_speed_range {
      x* number
      y* number
    }
    supported_range* {
      x {
        min_val* number
        max_val* number
      }
      y {
        min_val* number
        max_val* number
      }
    }
  }
  zoom* {
    default_speed_value number
    supported_range {
      x {
        min_val* number
        max_val* number
      }
      y {
        min_val* number
        max_val* number
      }
    }
  }
  max_num_of_presets* integer >=0
  home_supported* boolean
}
```

```
}
```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

#### STATUS CODE - 503: Service Unavailable

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                  ALLOWED:temporarily_unavailable
                  An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}
```

---

## 8.19 PUT /api/v1/onvif/{camera\_id}/ptz

### Update camera PTZ

Update camera PTZ.

In addition, the following status codes are defined for 400 response:

- onvif\_invalid\_argument - Invalid or insufficient arguments were provided
- onvif\_unauthenticated - Invalid login or password was provided
- onvif\_unsupported\_action - Action is not supported by the camera
- onvif\_not\_configured - Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

### REQUEST BODY - application/json

```
{
  move      enum      ALLOWED:UP, DOWN, LEFT, LEFT_UP, LEFT_DOWN, RIGHT, RIGHT_UP, RIGHT_DOWN
              An enumeration.
  zoom      enum      ALLOWED:ZOOM_IN, ZOOM_OUT
              An enumeration.
  step*     integer   between 1 and 10
}
```

## RESPONSE

### STATUS CODE - 200: PTZ update status

#### RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

### STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer   DEFAULT:400
  detail*     string
}
```

### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer   DEFAULT:404
  detail*     string
}
```

### STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
```

```

    msg*   string
    type*  string
  }]
}

```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

**STATUS CODE - 503:** Service Unavailable

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:temporarily_unavailable
                  ALLOWED:temporarily_unavailable
                  An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}

```

## 8.20 PUT /api/v1/onvif/{camera\_id}/ptz/home

### Save current PTZ position as home

Save current camera PTZ position as home.

It will return 400 BAD REQUEST with the following status codes:

- onvif\_invalid\_argument: Invalid or insufficient arguments were provided
- onvif\_unauthenticated: Invalid login or password was provided
- onvif\_unsupported\_action: Action is not supported by the camera
- onvif\_not\_configured: Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

## RESPONSE

**STATUS CODE - 200:** Current PTZ position saved as home status

**RESPONSE MODEL - application/json**

```

{
  status string DEFAULT:ok
}

```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}

```

```
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

#### STATUS CODE - 503: Service Unavailable

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                  ALLOWED:temporarily_unavailable
                  An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}
```

---

## 8.21 PUT /api/v1/onvif/{camera\_id}/ptz/home/apply

### Apply previously saved PTZ home position

Apply camera PTZ home position.

It will return 400 BAD REQUEST with the following status codes:

- `onvif_invalid_argument`: Invalid or insufficient arguments were provided
- `onvif_unauthenticated`: Invalid login or password was provided
- `onvif_unsupported_action`: Action is not supported by the camera
- `onvif_not_configured`: Insufficient credentials to access the camera were provided

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

## RESPONSE

**STATUS CODE - 200:** Applied previously saved PTZ home position status

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                          ALLOWED:invalid_request
                          An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                          ALLOWED:not_found
                          An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*  [string]
    msg*  string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

**STATUS CODE - 503:** Service Unavailable

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:temporarily_unavailable
                          ALLOWED:temporarily_unavailable
                          An enumeration.
  status_code integer  DEFAULT:503
  detail*     string
}
```

## 8.22 GET /api/v1/onvif/{camera\_id}/ptz/presets

### Get PTZ presets

Get list of camera PTZ presets.

It will return 400 BAD REQUEST with the following status codes:

- `onvif_invalid_argument`: Invalid or insufficient arguments were provided
- `onvif_unauthenticated`: Invalid login or password was provided
- `onvif_unsupported_action`: Action is not supported by the camera
- `onvif_not_configured`: Insufficient credentials to access the camera were provided

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

### RESPONSE

**STATUS CODE - 200:** Requested PTZ presets information

**RESPONSE MODEL - application/json**

```
{
  presets* [string]
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
  }
}
```

```
    type* string
  }}
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

**STATUS CODE - 503:** Service Unavailable

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}
```

---

## 8.23 POST /api/v1/onvif/{camera\_id}/ptz/preset

### Create PTZ preset

Create camera PTZ preset.

It will return 400 BAD REQUEST with the following status codes:

- `onvif_invalid_argument`: Invalid or insufficient arguments were provided
- `onvif_unauthenticated`: Invalid login or password was provided
- `onvif_unsupported_action`: Action is not supported by the camera
- `onvif_not_configured`: Insufficient credentials to access the camera were provided

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

#### REQUEST BODY - application/json

```
{
  preset_name* string max:64 chars
}
```

### RESPONSE

**STATUS CODE - 200:** Created PTZ preset status

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
```



```

        ALLOWED:invalid_request
        An enumeration.
    status_code integer DEFAULT:400
    detail*     string
}

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
    status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
    status_code integer DEFAULT:404
    detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
    detail [{
        Array of object:
        loc*  [string]
        msg*  string
        type* string
    }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

#### STATUS CODE - 503: Service Unavailable

##### RESPONSE MODEL - application/json

```

{
    status      enum      DEFAULT:temporarily_unavailable
                  ALLOWED:temporarily_unavailable
                  An enumeration.
    status_code integer DEFAULT:503
    detail*     string
}

```

## 8.24 DELETE /api/v1/onvif/{camera\_id}/ptz/preset

### Delete PTZ preset

Delete camera PTZ preset.

It will return 400 BAD REQUEST with the following status codes:

- onvif\_invalid\_argument: Invalid or insufficient arguments were provided
- onvif\_unauthenticated: Invalid login or password was provided
- onvif\_unsupported\_action: Action is not supported by the camera
- onvif\_not\_configured: Insufficient credentials to access the camera were provided

### REQUEST

## PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

## REQUEST BODY - application/json

```
{
  preset_name* string max:64 chars
}
```

## RESPONSE

STATUS CODE - 200: Deleted PTZ preset status

### RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

STATUS CODE - 400: Bad Request

### RESPONSE MODEL - application/json

```
{
  status          enum          DEFAULT:invalid_request
                                ALLOWED:invalid_request
                                An enumeration.
  status_code    integer       DEFAULT:400
  detail*        string
}
```

STATUS CODE - 404: Not Found

### RESPONSE MODEL - application/json

```
{
  status          enum          DEFAULT:not_found
                                ALLOWED:not_found
                                An enumeration.
  status_code    integer       DEFAULT:404
  detail*        string
}
```

STATUS CODE - 422: Validation Error

### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

STATUS CODE - 500: Internal Server Error

### RESPONSE MODEL - application/json

STATUS CODE - 503: Service Unavailable

### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                  ALLOWED:temporarily_unavailable
                  An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}
```

## 8.25 POST /api/v1/onvif/{camera\_id}/ptz/preset/apply

### Apply PTZ preset

Apply camera PTZ preset.

It will return 400 BAD REQUEST with the following status codes:

- `onvif_invalid_argument`: Invalid or insufficient arguments were provided
- `onvif_unauthenticated`: Invalid login or password was provided
- `onvif_unsupported_action`: Action is not supported by the camera
- `onvif_not_configured`: Insufficient credentials to access the camera were provided

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

#### REQUEST BODY - application/json

```
{
  preset_name* string max:64 chars
}
```

### RESPONSE

STATUS CODE - 200: Applied PTZ preset status

#### RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}
```

STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

#### STATUS CODE - 503: Service Unavailable

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                  ALLOWED:temporarily_unavailable
                  An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}
```

---

# 9. ORGANIZATIONS

## 9.1 GET /api/v1/organizations

Get list of organizations

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
cursor	string	
limit	integer	between 10 and 1000

### RESPONSE

STATUS CODE - 200: List of organizations

RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
  organizations* [{
    Array of object:
    name*      string  1 to 64 chars
    description* string  max:2048 chars
    logo_id    string
    id*        integer
    is_internal* boolean
    logo_url   string  1 to 65536 chars
  }]
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

## 9.2 POST /api/v1/organizations

Create new organization

## REQUEST

REQUEST BODY - application/json

```
{
  name*           string 1 to 64 chars
  description*    string max:2048 chars
  logo_id        string
}
```

## RESPONSE

STATUS CODE - 200: Created organization

RESPONSE MODEL - application/json

```
{
  name*           string 1 to 64 chars
  description*    string max:2048 chars
  logo_id        string
  id*            integer
  is_internal*   boolean
  logo_url       string 1 to 65536 chars
}
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
  status          enum    DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code     integer DEFAULT:400
  detail*        string
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 9.3 GET /api/v1/organizations/{organization\_id}

Get organization with provided ID

### REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
------	------	-------------

---

NAME	TYPE	DESCRIPTION
*organization_id	integer	

## RESPONSE

STATUS CODE - 200: Requested organization

RESPONSE MODEL - application/json

```
{
  name*      string  1 to 64 chars
  description* string  max:2048 chars
  logo_id    string
  id*        integer
  is_internal* boolean
  logo_url   string  1 to 65536 chars
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                          ALLOWED:not_found
                          An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

## 9.4 PUT /api/v1/organizations/{organization\_id}

Update organization with provided ID

### REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*organization_id	integer	

REQUEST BODY - application/json

```
{
  name*           string 1 to 64 chars
  description*    string max:2048 chars
  logo_id         string
}
```

## RESPONSE

**STATUS CODE - 200:** Updated organization

**RESPONSE MODEL - application/json**

```
{
  name*           string 1 to 64 chars
  description*    string max:2048 chars
  logo_id         string
  id*            integer
  is_internal*    boolean
  logo_url        string 1 to 65536 chars
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status          enum      DEFAULT:invalid_request
                                ALLOWED:invalid_request
                                An enumeration.
  status_code     integer   DEFAULT:400
  detail*         string
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status          enum      DEFAULT:not_found
                                ALLOWED:not_found
                                An enumeration.
  status_code     integer   DEFAULT:404
  detail*         string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**



## 9.5 DELETE /api/v1/organizations/{organization\_id}

### Delete organization with provided ID

Delete the organization with provided ID

It will return 400 BAD REQUEST if the organization has users or groups.

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*organization_id	integer	

### RESPONSE

**STATUS CODE - 200:** Organization was successfully deleted

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

## 9.6 GET /api/v1/organizations/{organization\_id}/members

Get list of organization members

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*organization_id	integer	

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
without_groups	boolean	
cursor	string	
limit	integer	between 10 and 1000

### RESPONSE

STATUS CODE - 200: List of organization members

RESPONSE MODEL - application/json

```
{
  members* [{
    Array of object:
    first_name*      string  max:32 chars
    last_name*       string  max:32 chars
    avatar_id        string
    email*           string
    id*              integer
    organization_id* integer
    organization_name* string
    avatar_url       string  1 to 65536 chars
    groups* [{
      Array of object:
      id*   integer
      name* string
    }]
    is_superuser*  boolean
    role_id*       integer
  }]
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
}
```

```
  detail*   string
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 9.7 GET /api/v1/organizations/{organization\_id}/groups

Get list of organization groups

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*organization_id	integer	

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
cursor	string	
limit	integer	between 10 and 1000

### RESPONSE

STATUS CODE - 200: List of organization groups

RESPONSE MODEL - application/json

```
{
  groups* [{
    Array of object:
    name*           string  1 to 64 chars
    description*    string  max:2048 chars
    id*             integer
    organization_id* integer
    organization_name* string
    is_internal*    boolean
    user_count*     integer
    place_count*    integer
  }]
}
```

## STATUS CODE - 404: Not Found

### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

## STATUS CODE - 422: Validation Error

### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

## STATUS CODE - 500: Internal Server Error

### RESPONSE MODEL - application/json

---

## 9.8 POST /api/v1/organizations/search

### Search organizations by provided query string

Search organizations by name.

List of case-insensitive name inclusion matches is returned.

## REQUEST

### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
cursor	string	
limit	integer	between 10 and 1000

---

### REQUEST BODY - application/json

```
{
  query* string
}
```

## RESPONSE

### STATUS CODE - 200: List of found organizations

### RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
}
```

```
organizations* [{  
  Array of object:  
  name*      string  1 to 64 chars  
  description* string  max:2048 chars  
  logo_id    string  
  id*        integer  
  is_internal* boolean  
  logo_url   string  1 to 65536 chars  
}]  
}
```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```
{  
  detail [{  
    Array of object:  
    loc* [string]  
    msg* string  
    type* string  
  }]  
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

---

# 10. ORION

## 10.1 GET /api/v1/orion/info

Get Orion service information

### REQUEST

No request parameters

### RESPONSE

STATUS CODE - 200: Provided Orion service information

RESPONSE MODEL - application/json

```
{
  service_version*  string
  software_version* string
  software_name*    string
  database_version* string
}
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer  DEFAULT:400
  detail*    string
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

STATUS CODE - 503: Service Unavailable

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer  DEFAULT:503
  detail*    string
}
```

---

## 10.2 PUT /api/v1/orion/config

Configure Orion integration

### REQUEST

REQUEST BODY - application/json

```
{
  wsdl_url* string 1 to 512 chars
}
```

```
login      string 1 to 255 chars
password   string 1 to 255 chars
}
```

## RESPONSE

**STATUS CODE - 200:** Configuration status

**RESPONSE MODEL - application/json**

```
{
  wsdl_url*    string 1 to 512 chars
  login        string 1 to 255 chars
  password     string 1 to 255 chars
  organization_id* integer
  updated_at*  string
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*  [string]
    msg*  string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

**STATUS CODE - 503:** Service Unavailable

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer  DEFAULT:503
  detail*     string
}
```

# 11. PERMISSIONS

## 11.1 POST /api/v1/permissions

Grant the permission to the role with provided ID

### REQUEST

REQUEST BODY - application/json

```
{
  role_id*    integer
  permission* enum    ALLOWED:auditlog_read, camera_create, camera_delete, camera_read,
                    camera_snapshot_read, camera_update, event_read,
                    faceimage_create, faceimage_delete, faceimage_read,
                    facelist_create, facelist_delete, facelist_face_create,
                    facelist_face_delete, facelist_face_read, facelist_read,
                    facelist_update, face_create, face_delete, face_read,
                    face_update, group_create, group_delete, group_place_create,
                    group_place_delete, group_place_read, group_read, group_update,
                    group_user_create, group_user_delete, group_user_read,
                    onvif_create, onvif_delete, onvif_read, onvif_update,
                    orion_create, orion_read, orion_update, orion_delete,
                    organization_create, organization_delete,
                    organization_groups_read, organization_members_read,
                    organization_read, organization_update, permission_create,
                    permission_delete, permission_read, place_create, place_delete,
                    place_group_create, place_group_delete, place_group_read,
                    place_plan_read, place_read, place_update, plan_camera_create,
                    plan_camera_delete, plan_camera_read, plan_camera_update,
                    plan_create, plan_delete, plan_read, plan_update,
                    platelist_create, platelist_delete, platelist_plate_create,
                    platelist_plate_delete, platelist_plate_read, platelist_read,
                    platelist_update, plate_create, plate_delete, plate_read,
                    plate_update, role_create, role_delete, role_member_read,
                    role_perm_read, role_read, role_update, search_face,
                    search_plate, segment_read, snapshot_read, stats_read,
                    stream_read, streamserver_read, streamstatus_read,
                    upload_create, upload_meta_read, upload_read, user_create,
                    user_delete, user_me_read, user_me_update, user_read,
                    user_update, videoclip_create, videoclip_delete, videoclip_read
                    An enumeration.
}
```

### RESPONSE

STATUS CODE - 200: Permission successfully granted

RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
```



```

    status      enum      DEFAULT:invalid_request
                                ALLOWED:invalid_request
                                An enumeration.
    status_code integer    DEFAULT:400
    detail*     string
}

```

#### STATUS CODE - 403: Forbidden

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:unauthorized
                                ALLOWED:unauthorized
                                An enumeration.
  status_code integer    DEFAULT:403
  detail*     string
}

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:not_found
                                ALLOWED:not_found
                                An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
      loc*   [string]
      msg*   string
      type*  string
    }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

## 11.2 DELETE /api/v1/permissions

Revoke the permission from the role with provided ID

### REQUEST

#### REQUEST BODY - application/json

```

{
  role_id*     integer
  permission*  enum      ALLOWED:auditlog_read, camera_create, camera_delete, camera_read,
                                camera_snapshot_read, camera_update, event_read,
                                faceimage_create, faceimage_delete, faceimage_read,
                                facelist_create, facelist_delete, facelist_face_create,
                                facelist_face_delete, facelist_face_read, facelist_read,
}

```

```

facelist_update, face_create, face_delete, face_read,
face_update, group_create, group_delete, group_place_create,
group_place_delete, group_place_read, group_read, group_update,
group_user_create, group_user_delete, group_user_read,
onvif_create, onvif_delete, onvif_read, onvif_update,
orion_create, orion_read, orion_update, orion_delete,
organization_create, organization_delete,
organization_groups_read, organization_members_read,
organization_read, organization_update, permission_create,
permission_delete, permission_read, place_create, place_delete,
place_group_create, place_group_delete, place_group_read,
place_plan_read, place_read, place_update, plan_camera_create,
plan_camera_delete, plan_camera_read, plan_camera_update,
plan_create, plan_delete, plan_read, plan_update,
platelist_create, platelist_delete, platelist_plate_create,
platelist_plate_delete, platelist_plate_read, platelist_read,
platelist_update, plate_create, plate_delete, plate_read,
plate_update, role_create, role_delete, role_member_read,
role_perm_read, role_read, role_update, search_face,
search_plate, segment_read, snapshot_read, stats_read,
stream_read, streamserver_read, streamstatus_read, upload_create,
upload_meta_read, upload_read, user_create, user_delete,
user_me_read, user_me_update, user_read, user_update,
videoclip_create, videoclip_delete, videoclip_read

```

An enumeration.

```

}

```

## RESPONSE

**STATUS CODE - 200:** Permission successfully revoked

**RESPONSE MODEL - application/json**

```

{
  status string DEFAULT:ok
}

```

**STATUS CODE - 403:** Forbidden

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:unauthorized
                ALLOWED:unauthorized
                An enumeration.
  status_code integer  DEFAULT:403
  detail*     string
}

```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

---

# 12. PLACES

## 12.1 GET /api/v1/places

Get list of places

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
latitude	number between -90 and 90	
longitude	number between -180 and 180	
radius	integer	
without_groups	boolean	
group_id	integer	

### RESPONSE

STATUS CODE - 200: List of places optionally filtered by given location and radius

#### RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
  places* [{
    Array of object:
    name*      string 1 to 255 chars
    description string max:2048 chars
    latitude   number
    longitude  number
    id*        integer
    created_at* string
    updated_at* string
    plan_count* integer
    camera_count* integer
  }]
}
```

STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  status      enum    DEFAULT:invalid_request
              ALLOWED:invalid_request
              An enumeration.
  status_code integer DEFAULT:400
  detail*     string
}
```

STATUS CODE - 422: Validation Error

### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

STATUS CODE - 500: Internal Server Error

### RESPONSE MODEL - application/json

---

## 12.2 POST /api/v1/places

Create new place

### REQUEST

#### REQUEST BODY - application/json

```
{
  name*       string 1 to 255 chars
  description string max:2048 chars
  latitude    number
  longitude   number
}
```

### RESPONSE

STATUS CODE - 200: Created place

#### RESPONSE MODEL - application/json

```
{
  name*       string 1 to 255 chars
  description string max:2048 chars
  latitude    number
  longitude   number
  id*         integer
  created_at* string
  updated_at* string
  plan_count* integer
  camera_count* integer
}
```

STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  status      enum    DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer DEFAULT:400
  detail*     string
}
```

STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

---

## 12.3 POST /api/v1/places/search

### Search places by provided query string

Search places by name.

List of case-insensitive name inclusion matches is returned.

## REQUEST

### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
without_groups	boolean	
cursor	string	
limit	integer	between 10 and 1000

---

### REQUEST BODY - application/json

```
{
  query* string
}
```

## RESPONSE

STATUS CODE - 200: List of found places

### RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
  places* [{
    Array of object:
    name* string 1 to 255 chars
    description string max:2048 chars
    latitude number
    longitude number
    id* integer
    created_at* string
    updated_at* string
    plan_count* integer
    camera_count* integer
  }]
}
```

```
  }]  
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{  
  detail [{  
    Array of object:  
    loc* [string]  
    msg* string  
    type* string  
  }]  
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 12.4 GET /api/v1/places/{place\_id}

Get place with provided ID

### REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*place_id	integer	

---

### RESPONSE

STATUS CODE - 200: Requested place

RESPONSE MODEL - application/json

```
{  
  name*          string  1 to 255 chars  
  description    string  max:2048 chars  
  latitude       number  
  longitude      number  
  id*           integer  
  created_at*   string  
  updated_at*   string  
  plan_count*   integer  
  camera_count* integer
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```
{  
  status          enum      DEFAULT:not_found  
                                     ALLOWED:not_found  
                                     An enumeration.  
  status_code    integer  DEFAULT:404  
  detail*        string
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 12.5 PUT /api/v1/places/{place\_id}

Update place with provided ID

### REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*place_id	integer	

---

REQUEST BODY - application/json

```
{
  name*      string 1 to 255 chars
  description string max:2048 chars
  latitude   number
  longitude  number
}
```

### RESPONSE

STATUS CODE - 200: Updated place

RESPONSE MODEL - application/json

```
{
  name*      string 1 to 255 chars
  description string max:2048 chars
  latitude   number
  longitude  number
  id*        integer
  created_at* string
  updated_at* string
  plan_count* integer
  camera_count* integer
}
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
  status      enum    DEFAULT:invalid_request
}
```



```

        ALLOWED:invalid_request
        An enumeration.
    status_code integer DEFAULT:400
    detail*     string
}

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer DEFAULT:404
  detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

## 12.6 DELETE /api/v1/places/{place\_id}

### Delete place with provided ID

Delete place with provided ID.

Returns 400 BAD REQUEST if place has attached plans.

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*place_id	integer	

### RESPONSE

#### STATUS CODE - 200: Place successfully deleted

##### RESPONSE MODEL - application/json

```

{
  status string DEFAULT:ok
}

```

#### STATUS CODE - 404: Not Found

### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

### STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

---

## 12.7 GET /api/v1/places/{place\_id}/plans

### Get place's list of plans

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*place_id	integer	

### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
cursor	string	
limit	integer	between 10 and 1000

## RESPONSE

### STATUS CODE - 200: List of plans

#### RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
  plans* [{
    Array of object:
    name*      string  1 to 255 chars
    place_id*  integer
  }]
}
```

```

    image_id*  string
    id*        integer
    image_url  string  1 to 65536 chars
    created_at* string
    updated_at* string
  }]
}

```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}

```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
    loc*  [string]
    msg*  string
    type* string
  }]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

## 12.8 GET /api/v1/places/{place\_id}/cameras

Get place's cameras

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*place_id	integer	

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
without_plans	boolean	
cursor	string	
limit	integer	between 10 and 1000

### RESPONSE

## STATUS CODE - 200: List of cameras

### RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
  cameras* [{
    Array of object:
    name*          string    1 to 255 chars
    source* {
    }
    video_analytics {
    }
    video_storage {
    }
    video_streaming {
    }
    ptz {
    }
    place_id*      integer
    onvif_host     string    1 to 255 chars
    onvif_port     integer
    onvif_login    string    1 to 255 chars
    onvif_password string    1 to 255 chars
    scheduler_group string    1 to 64 chars
    scheduler_tags [string] 1 to 32 chars
    id*           string
    created_at*   string
    updated_at*   string
    snapshot_id   string
    snapshot_url  string    1 to 65536 chars
    origin {
      plan_id      integer
      plan_name     string
      place_id     integer
      place_name    string
      place_description string
      place_latitude number
      place_longitude number
    }
    plan_id        integer
  }]
}
```

## STATUS CODE - 404: Not Found

### RESPONSE MODEL - application/json

```
{
  status      enum    DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

## STATUS CODE - 422: Validation Error

**RESPONSE MODEL - application/json**

```
{  
  detail [{  
    Array of object:  
    loc* [string]  
    msg* string  
    type* string  
  }]  
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

---

# 13. PLANS

## 13.1 GET /api/v1/plans

Get list of plans

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
group_id	integer	
cursor	string	
limit	integer	between 10 and 1000

### RESPONSE

STATUS CODE - 200: List of plans

RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
  plans* [{
    Array of object:
    name*      string  1 to 255 chars
    place_id*  integer
    image_id*  string
    id*        integer
    image_url  string  1 to 65536 chars
    created_at* string
    updated_at* string
  }]
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

## 13.2 POST /api/v1/plans

### Create new plan

#### REQUEST

##### REQUEST BODY - application/json

```
{
  name*      string  1 to 255 chars
  place_id*  integer
  image_id*  string
}
```

#### RESPONSE

##### STATUS CODE - 200: Created plan

##### RESPONSE MODEL - application/json

```
{
  name*      string  1 to 255 chars
  place_id*  integer
  image_id*  string
  id*        integer
  image_url  string  1 to 65536 chars
  created_at* string
  updated_at* string
}
```

##### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

##### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

##### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

---

## 13.3 POST /api/v1/plans/search

### Search plans by provided query string

Search plans by name.

List of case-insensitive name inclusion matches is returned.

## REQUEST

### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
group_id	integer	
cursor	string	
limit	integer between 10 and 1000	

### REQUEST BODY - application/json

```
{
  query* string
}
```

## RESPONSE

**STATUS CODE - 200:** List of found plans

### RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
  plans* [{
    Array of object:
    name*      string  1 to 255 chars
    place_id*  integer
    image_id*  string
    id*        integer
    image_url  string  1 to 65536 chars
    created_at* string
    updated_at* string
  }]
}
```

**STATUS CODE - 422:** Validation Error

### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

### RESPONSE MODEL - application/json



## 13.4 GET /api/v1/plans/{plan\_id}

Get plan with provided ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*plan_id	integer	

### RESPONSE

STATUS CODE - 200: Requested plan

RESPONSE MODEL - application/json

```
{
  name*      string  1 to 255 chars
  place_id*  integer
  image_id*  string
  id*        integer
  image_url  string  1 to 65536 chars
  created_at* string
  updated_at* string
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

## 13.5 PUT /api/v1/plans/{plan\_id}

Update plan with provided ID

### REQUEST

## PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*plan_id	integer	

### REQUEST BODY - application/json

```
{
  name*      string  1 to 255 chars
  place_id*  integer
  image_id*  string
}
```

## RESPONSE

### STATUS CODE - 200: Updated plan

#### RESPONSE MODEL - application/json

```
{
  name*      string  1 to 255 chars
  place_id*  integer
  image_id*  string
  id*        integer
  image_url  string  1 to 65536 chars
  created_at* string
  updated_at* string
}
```

### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

### STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

## 13.6 DELETE /api/v1/plans/{plan\_id}

Delete specified plan

Delete the plan with provided ID.

Returns 400 BAD REQUEST if the plan has cameras that belong to it.

## REQUEST

### PATH PARAMETERS

---

NAME	TYPE	DESCRIPTION
*plan_id	integer	

---

## RESPONSE

**STATUS CODE - 200:** Plan successfully deleted

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

---

## 13.7 GET /api/v1/plans/{plan\_id}/cameras

Get plan's cameras

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*plan_id	integer	

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
cursor	string	
limit	integer	between 10 and 1000

### RESPONSE

STATUS CODE - 200: List of cameras

RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
  cameras* [{
    Array of object:
    name*          string    1 to 255 chars
    source* {
    }
    video_analytics {
    }
    video_storage {
    }
    video_streaming {
    }
    ptz {
    }
    place_id*      integer
    onvif_host     string    1 to 255 chars
    onvif_port     integer
    onvif_login    string    1 to 255 chars
    onvif_password string    1 to 255 chars
    scheduler_group string    1 to 64 chars
    scheduler_tags [string] 1 to 32 chars
    id*            string
    created_at*    string
    updated_at*    string
    snapshot_id    string
    snapshot_url   string    1 to 65536 chars
    origin {
      plan_id      integer
      plan_name     string
      place_id     integer
    }
  }
}
```

```

        place_name      string
        place_description string
        place_latitude  number
        place_longitude number
    }
    plan_id*            integer
    x                   number    between 0 and 1
    y                   number    between 0 and 1
    camera_id*         string
}]]
}

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
  status      enum    DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
    loc*  [string]
    msg*  string
    type* string
  }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

## 13.8 POST /api/v1/plans/{plan\_id}/cameras

### Add cameras to plan

Add cameras with provided IDs to the plan.

Returns 400 BAD REQUEST in the following scenarios:

- At least one of the provided cameras was already added to this plan
- At least one of the provided cameras belongs to another place
- Plan or at least one of the provided cameras with provided ID does not exist

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*plan_id	integer	

#### REQUEST BODY - application/json

```
{
  camera_ids* [string]
}
```

## RESPONSE

**STATUS CODE - 200:** Cameras successfully added to the plan

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

---

## 13.9 DELETE /api/v1/plans/{plan\_id}/cameras

### Remove cameras from the plan

Remove cameras with provided IDs from the plan.

Returns 400 BAD REQUEST in the following scenarios:

- Plan with provided ID does not exist

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*plan_id	integer	

### REQUEST BODY - application/json

```
{
  camera_ids* [string]
}
```

## RESPONSE

**STATUS CODE - 200:** Cameras successfully removed from the plan

### RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 404:** Not Found

### RESPONSE MODEL - application/json

```
{
  status          enum          DEFAULT:not_found
                                ALLOWED:not_found
                                An enumeration.
  status_code    integer       DEFAULT:404
  detail*        string
}
```

**STATUS CODE - 422:** Validation Error

### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

### RESPONSE MODEL - application/json

## 13.10 PUT /api/v1/plans/{plan\_id}/cameras/{camera\_id}/position

### Update camera position on the plan

Update camera's position on the plan.

Returns 400 BAD REQUEST if camera does not belong to the specified plan.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*plan_id	integer	
*camera_id	uuid	

**REQUEST BODY - application/json**

```
{
  x number between 0 and 1
  y number between 0 and 1
}
```

**RESPONSE**

**STATUS CODE - 200: Camera with updated position on the plan**

**RESPONSE MODEL - application/json**

```
{
  x number between 0 and 1
  y number between 0 and 1
  camera_id* string
  plan_id* integer
}
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  status enum DEFAULT:invalid_request
  ALLOWED:invalid_request
  An enumeration.
  status_code integer DEFAULT:400
  detail* string
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  status enum DEFAULT:not_found
  ALLOWED:not_found
  An enumeration.
  status_code integer DEFAULT:404
  detail* string
}
```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500: Internal Server Error**





# 14. PLATELISTS

## 14.1 GET /api/v1/platelists

Get all plate lists, optionally filtered by name

Get all plate lists.

If list name was provided, it will return only plate lists that match the name exactly.

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
name	string	

### RESPONSE

STATUS CODE - 200: Plate lists

RESPONSE MODEL - application/json

```
{
  platelists* [{
    Array of object:
    name*      string  1 to 255 chars
    description string  max:2048 chars
    priority   enum    ALLOWED:low, medium, high
                An enumeration.
    id*        integer
    plate_count* integer
  }]
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

## 14.2 POST /api/v1/platelists

Create new plate list

Create new plate list with provided data.

It will return 400 BAD REQUEST if the plate list with provided name already exists.

## REQUEST

### REQUEST BODY - application/json

```
{
  name*      string  1 to 255 chars
  description string  max:2048 chars
  priority   enum    ALLOWED:low, medium, high
                        An enumeration.
}
```

## RESPONSE

### STATUS CODE - 200: Created plate list

#### RESPONSE MODEL - application/json

```
{
  name*      string  1 to 255 chars
  description string  max:2048 chars
  priority   enum    ALLOWED:low, medium, high
                        An enumeration.
  id*        integer
  plate_count* integer
}
```

### STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  status      enum    DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

### STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

---

## 14.3 GET /api/v1/platelists/{platelist\_id}

### Get plate list with provided ID

Get plate list with provided ID.

It will return 404 NOT FOUND if the plate list with provided ID was not found.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*platelist_id	integer	

## RESPONSE

**STATUS CODE - 200:** Requested plate list

**RESPONSE MODEL - application/json**

```
{
  name*      string  1 to 255 chars
  description string  max:2048 chars
  priority    enum    ALLOWED:low, medium, high
                An enumeration.
  id*        integer
  plate_count* integer
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum    DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

---

## 14.4 PUT /api/v1/platelists/{platelist\_id}

### Update plate list with provided ID

Update plate list with provided ID.

It will return 400 BAD REQUEST if the plate list with provided name already exists.

It will return 404 NOT FOUND if the plate list with provided ID was not found.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*platelist_id	integer	

### REQUEST BODY - application/json

```
{
  name*      string 1 to 255 chars
  description string max:2048 chars
  priority   enum    ALLOWED:low, medium, high
                An enumeration.
}
```

## RESPONSE

### STATUS CODE - 200: Updated plate list

#### RESPONSE MODEL - application/json

```
{
  name*      string 1 to 255 chars
  description string max:2048 chars
  priority   enum    ALLOWED:low, medium, high
                An enumeration.
  id*        integer
  plate_count* integer
}
```

### STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  status      enum    DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer DEFAULT:400
  detail*     string
}
```

### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  status      enum    DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer DEFAULT:404
  detail*     string
}
```

### STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
```

```
    loc*  [string]
    msg*  string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

---

## 14.5 DELETE /api/v1/platelists/{platelist\_id}

### Delete specified plate list

Delete the plate list with provided ID.

It will return 400 BAD REQUEST if the plate list has plates.

It will return 404 NOT FOUND if the plate list with provided ID was not found.

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*platelist_id	integer	

---

### RESPONSE

**STATUS CODE - 200:** Plate list successfully deleted

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*  [string]
    msg*  string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 14.6 POST /api/v1/platelists/{platelist\_id}/plates/{plate\_id}

### Add plate to the list

Add a plate to a plate list with provided ID.

It will return 400 BAD REQUEST if the plate is already in the plate list.  
It will return 404 NOT FOUND if a plate or plate list were not found.

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*platelist_id	integer	
*plate_id	integer	

---

### RESPONSE

STATUS CODE - 200: Plate list plate successfully added

RESPONSE MODEL - application/json

```
{
  status string DEFAULT:ok
}
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 14.7 DELETE /api/v1/platelists/{platelist\_id}/plates/{plate\_id}

### Remove plate from the list

Delete a plate from plate list.

It will return 404 NOT FOUND if a plate list with provided ID was not found.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*platelist_id	integer	
*plate_id	integer	

## RESPONSE

**STATUS CODE - 200:** Plate list plate successfully removed

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

---

## 14.8 GET /api/v1/platelists/{platelist\_id}/plates

### Get plates of the list

Get plates of a plate list with provided ID.



It will return 404 NOT FOUND if a plate list with provided ID was not found.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*platelist_id	integer	

## RESPONSE

**STATUS CODE - 200:** Plates attached to the specified list

### RESPONSE MODEL - application/json

```
{
  plates* [{
    Array of object:
    number*           string  max:32 chars
    owner_full_name   string  max:128 chars
    owner_document_number string  max:128 chars
    owner_birth_date  string
    vehicle_manufacturer string  max:32 chars
    vehicle_model     string  max:32 chars
    vehicle_type      string  max:32 chars
    vehicle_manufacture_date string
    vehicle_color     string  max:32 chars
    vehicle_image_id  string
    id*               integer
    vehicle_image_url string  1 to 65536 chars
  }]
}
```

**STATUS CODE - 404:** Not Found

### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

### RESPONSE MODEL - application/json

# 15. PLATES

## 15.1 GET /api/v1/plates

Get all plates optionally filtered by plate number

Returns list of all the plates.

Result may be optionally filtered by plate number.

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
number	string	

### RESPONSE

STATUS CODE - 200: List of all plates

RESPONSE MODEL - application/json

```
{
  plates* [{
    Array of object:
    number*           string  max:32 chars
    owner_full_name   string  max:128 chars
    owner_document_number string max:128 chars
    owner_birth_date  string
    vehicle_manufacturer string max:32 chars
    vehicle_model     string  max:32 chars
    vehicle_type      string  max:32 chars
    vehicle_manufacture_date string
    vehicle_color     string  max:32 chars
    vehicle_image_id  string
    id*               integer
    vehicle_image_url string  1 to 65536 chars
  }]
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

## 15.2 POST /api/v1/plates

### Create new plate

Create new plate.

Date must be provided in ISO 8601 format:  
YYYY-MM-DD (i.e. 2012-07-21)

### REQUEST

#### REQUEST BODY - application/json

```
{
  number*           string max:32 chars
  owner_full_name   string max:128 chars
  owner_document_number string max:128 chars
  owner_birth_date  string
  vehicle_manufacturer string max:32 chars
  vehicle_model     string max:32 chars
  vehicle_type      string max:32 chars
  vehicle_manufacture_date string
  vehicle_color     string max:32 chars
  vehicle_image_id  string
}
```

### RESPONSE

#### STATUS CODE - 200: Created plate

#### RESPONSE MODEL - application/json

```
{
  number*           string max:32 chars
  owner_full_name   string max:128 chars
  owner_document_number string max:128 chars
  owner_birth_date  string
  vehicle_manufacturer string max:32 chars
  vehicle_model     string max:32 chars
  vehicle_type      string max:32 chars
  vehicle_manufacture_date string
  vehicle_color     string max:32 chars
  vehicle_image_id  string
  id*              integer
  vehicle_image_url string 1 to 65536 chars
}
```

#### STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  status      enum    DEFAULT:invalid_request
              ALLOWED:invalid_request
              An enumeration.
  status_code integer DEFAULT:400
  detail*     string
}
```

#### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
```

```

status      enum      DEFAULT:not_found
              ALLOWED:not_found
              An enumeration.
status_code integer    DEFAULT:404
detail*     string
}

```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

## 15.3 GET /api/v1/plates/{plate\_id}

### Get plate with provided ID

Get a plate with provided ID.

It will return 404 NOT FOUND if the plate with provided ID was not found.

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*plate_id	integer	

### RESPONSE

**STATUS CODE - 200: Requested plate**

**RESPONSE MODEL - application/json**

```

{
  number*           string    max:32 chars
  owner_full_name   string    max:128 chars
  owner_document_number string  max:128 chars
  owner_birth_date  string
  vehicle_manufacturer string  max:32 chars
  vehicle_model     string    max:32 chars
  vehicle_type      string    max:32 chars
  vehicle_manufacture_date string
  vehicle_color     string    max:32 chars
  vehicle_image_id  string
  id*               integer
  vehicle_image_url string    1 to 65536 chars
}

```

## STATUS CODE - 404: Not Found

### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

## STATUS CODE - 422: Validation Error

### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

## STATUS CODE - 500: Internal Server Error

### RESPONSE MODEL - application/json

---

## 15.4 PUT /api/v1/plates/{plate\_id}

### Update plate with provided ID

Update existing plate.

It will return 404 NOT FOUND if the plate with provided ID was not found.  
It will return 400 BAD REQUEST if vehicle image with provided ID was not found or plate with provided number already exists.

Date must be provided in ISO 8601 format:  
YYYY-MM-DD (i.e. 2012-07-21)

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*plate_id	integer	

### REQUEST BODY - application/json

```
{
  number*           string max:32 chars
  owner_full_name   string max:128 chars
  owner_document_number string max:128 chars
  owner_birth_date  string
  vehicle_manufacturer string max:32 chars
  vehicle_model     string max:32 chars
  vehicle_type      string max:32 chars
  vehicle_manufacture_date string
  vehicle_color     string max:32 chars
}
```

```
    vehicle_image_id      string
  }
```

## RESPONSE

**STATUS CODE - 200:** Updated plate

**RESPONSE MODEL - application/json**

```
{
  number*           string   max:32 chars
  owner_full_name   string   max:128 chars
  owner_document_number string max:128 chars
  owner_birth_date  string
  vehicle_manufacturer string max:32 chars
  vehicle_model     string   max:32 chars
  vehicle_type      string   max:32 chars
  vehicle_manufacture_date string
  vehicle_color     string   max:32 chars
  vehicle_image_id  string
  id*               integer
  vehicle_image_url string   1 to 65536 chars
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

## 15.5 DELETE /api/v1/plates/{plate\_id}

### Delete plate with provided ID

Delete plate with provided ID.

It will return 404 NOT FOUND if the plate with provided ID was not found.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*plate_id	integer	

---

## RESPONSE

**STATUS CODE - 200:** Plate successfully deleted

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

---

# 16. ROLES

## 16.1 GET /api/v1/roles

Get list of roles

### REQUEST

No request parameters

### RESPONSE

STATUS CODE - 200: List of roles

RESPONSE MODEL - application/json

```
{
  roles* [{
    Array of object:
    name*      string  1 to 64 chars
    description* string max:2048 chars
    id*        integer
    is_internal* boolean
  }]
}
```

STATUS CODE - 403: Forbidden

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:unauthorized
                  ALLOWED:unauthorized
                  An enumeration.
  status_code integer  DEFAULT:403
  detail*     string
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 16.2 POST /api/v1/roles

Create new role

### REQUEST

REQUEST BODY - application/json

```
{
  name*      string  1 to 64 chars
  description* string max:2048 chars
}
```

### RESPONSE

STATUS CODE - 200: Created role

RESPONSE MODEL - application/json



```

{
  name*      string  1 to 64 chars
  description* string  max:2048 chars
  id*        integer
  is_internal* boolean
}

```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:unauthorized
                  ALLOWED:unauthorized
                  An enumeration.
  status_code integer  DEFAULT:403
  detail*     string
}

```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

## 16.3 GET /api/v1/roles/{role\_id}

Get role with provided ID

### REQUEST

**PATH PARAMETERS**

NAME	TYPE	DESCRIPTION
*role_id	integer	

### RESPONSE

**STATUS CODE - 200: Requested role**

**RESPONSE MODEL - application/json**

```

{
  name*      string  1 to 64 chars
  description* string  max:2048 chars
  id*        integer
  is_internal* boolean
}

```

**STATUS CODE - 403: Forbidden**

#### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:unauthorized
                  ALLOWED:unauthorized
                  An enumeration.
  status_code integer    DEFAULT:403
  detail*     string
}
```

#### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

#### STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

#### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

---

## 16.4 PUT /api/v1/roles/{role\_id}

### Update role with provided ID

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*role_id	integer	

##### REQUEST BODY - application/json

```
{
  name*       string 1 to 64 chars
  description* string max:2048 chars
}
```

#### RESPONSE

##### STATUS CODE - 200: Updated role

#### RESPONSE MODEL - application/json

```
{
  name*      string  1 to 64 chars
  description* string max:2048 chars
  id*        integer
  is_internal* boolean
}
```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

#### STATUS CODE - 403: Forbidden

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:unauthorized
                  ALLOWED:unauthorized
                  An enumeration.
  status_code integer  DEFAULT:403
  detail*     string
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*  [string]
    msg*  string
    type* string
  }]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

---

## 16.5 DELETE /api/v1/roles/{role\_id}

## Delete role with provided ID

Delete the role with provided ID.

It will return 400 BAD REQUEST if the role has users.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*role_id	integer	

## RESPONSE

**STATUS CODE - 200:** Role successfully deleted

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 403:** Forbidden

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:unauthorized
                ALLOWED:unauthorized
                An enumeration.
  status_code integer  DEFAULT:403
  detail*     string
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
```

```

detail [{
  Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}

```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

## 16.6 GET /api/v1/roles/{role\_id}/members

Get list of role members

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*role_id	integer	

### RESPONSE

STATUS CODE - 200: List of role members

RESPONSE MODEL - application/json

```

{
  members* [{
    Array of object:
      first_name* string max:32 chars
      last_name* string max:32 chars
      avatar_id string
      email* string
      id* integer
      organization_id* integer
      organization_name* string
      avatar_url string 1 to 65536 chars
      groups* [{
        Array of object:
          id* integer
          name* string
        }]
      is_superuser* boolean
      role_id* integer
    }]
}

```

STATUS CODE - 403: Forbidden

RESPONSE MODEL - application/json

```

{
  status enum DEFAULT:unauthorized
  ALLOWED:unauthorized
  An enumeration.
  status_code integer DEFAULT:403
}

```

```
    detail*    string
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

---

## 16.7 GET /api/v1/roles/{role\_id}/permissions

Get list of role permissions

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*role_id	integer	

### RESPONSE

#### STATUS CODE - 200: List of role permissions

##### RESPONSE MODEL - application/json

```
{
  permissions* [enum] ALLOWED:auditlog_read, camera_create, camera_delete, camera_read,
                    camera_snapshot_read, camera_update, event_read,
                    faceimage_create, faceimage_delete, faceimage_read,
                    facelist_create, facelist_delete, facelist_face_create,
                    facelist_face_delete, facelist_face_read, facelist_read,
                    facelist_update, face_create, face_delete, face_read,
                    face_update, group_create, group_delete, group_place_create,
                    group_place_delete, group_place_read, group_read, group_update,
                    group_user_create, group_user_delete, group_user_read,
                    onvif_create, onvif_delete, onvif_read, onvif_update,
}
```

```

    orion_create, orion_read, orion_update, orion_delete,
    organization_create, organization_delete,
    organization_groups_read, organization_members_read,
    organization_read, organization_update, permission_create,
    permission_delete, permission_read, place_create, place_delete,
    place_group_create, place_group_delete, place_group_read,
    place_plan_read, place_read, place_update, plan_camera_create,
    plan_camera_delete, plan_camera_read, plan_camera_update,
    plan_create, plan_delete, plan_read, plan_update,
    platelist_create, platelist_delete, platelist_plate_create,
    platelist_plate_delete, platelist_plate_read, platelist_read,
    platelist_update, plate_create, plate_delete, plate_read,
    plate_update, role_create, role_delete, role_member_read,
    role_perm_read, role_read, role_update, search_face,
    search_plate, segment_read, snapshot_read, stats_read,
    stream_read, streamserver_read, streamstatus_read,
    upload_create, upload_meta_read, upload_read, user_create,
    user_delete, user_me_read, user_me_update, user_read,
    user_update, videoclip_create, videoclip_delete, videoclip_read

```

An enumeration.

```

}

```

#### STATUS CODE - 403: Forbidden

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:unauthorized
                ALLOWED:unauthorized
                An enumeration.
  status_code integer    DEFAULT:403
  detail*     string
}

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
      loc*   [string]
      msg*   string
      type*  string
    }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json





# 17. SEARCH

## 17.1 POST /api/v1/search/face

Search face recognition events

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
group_id	integer	

#### REQUEST BODY - application/json

```
{
  descriptor* {
    data*   string
    norm*   number
    version* integer
  }
  threshold* number between 0 and 1
}
```

### RESPONSE

STATUS CODE - 200: List of matching face recognition events

#### RESPONSE MODEL - application/json

```
{
  events* [{
    Array of object:
    origin {
      plan_id           integer
      plan_name         string
      place_id          integer
      place_name        string
      place_description string
      place_latitude   number
      place_longitude   number
      camera_id         string
      camera_name       string
    }
    streamserver_id*  string
    stream_id*        string
    object_id*        string
    event_id*         string
    event_action*     string
    ts*               integer
    event_data* {
      detection* {
        bbox* [number]
      }
      descriptor {
        data* string
        norm* number
      }
    }
  }
}
```

```

    }
    attributes {
    }
}
event_type*      enum      ALLOWED:face_recognition
match_data [{
  Array of object:
    face_id*      integer
    facelist_id*  integer
    similarity*    number    between -1 and 1
    face_name      string
    facelist_name* string
    priority       enum      ALLOWED:low, medium, high
                                An enumeration.
  }]
similarity*      number    between -1 and 1
}]
}

```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```

{
  status          enum      DEFAULT:invalid_request
                                ALLOWED:invalid_request
                                An enumeration.
  status_code     integer    DEFAULT:400
  detail*         string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

#### STATUS CODE - 503: Service Unavailable

##### RESPONSE MODEL - application/json

```

{
  status          enum      DEFAULT:temporarily_unavailable
                                ALLOWED:temporarily_unavailable
                                An enumeration.
  status_code     integer    DEFAULT:503
  detail*         string
}

```

## Search plate recognition events

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
group_id	integer	

#### REQUEST BODY - application/json

```
{
  plate* {
    number* string
  }
}
```

### RESPONSE

STATUS CODE - 200: List of matching plate recognition events

#### RESPONSE MODEL - application/json

```
{
  events* [{
    Array of object:
    origin {
      plan_id integer
      plan_name string
      place_id integer
      place_name string
      place_description string
      place_latitude number
      place_longitude number
      camera_id string
      camera_name string
    }
    streamserver_id* string
    stream_id* string
    object_id* string
    event_id* string
    event_action* string
    ts* integer
    event_data* {
      plate_number* {
        data* string
      }
      detection* {
        bbox* [number]
      }
    }
    event_type* enum ALLOWED:plate_recognition
    match_data [{
      Array of object:
      plate_id* integer
      platelist_id* integer
      plate_number string
      similarity number between -1 and 1
      platelist_name* string
    }
  }
}
```

```

        priority      enum      ALLOWED:low, medium, high
                                An enumeration.
    ]]
    similarity*      number    between -1 and 1
}]
}

```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:invalid_request
                                ALLOWED:invalid_request
                                An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}

```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

**STATUS CODE - 503: Service Unavailable**

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:temporarily_unavailable
                                ALLOWED:temporarily_unavailable
                                An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}

```

---

# 18. SEGMENTS

## 18.1 GET /api/v1/segments

Get video segments

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*stream_id	uuid	
*ts_from	integer	
*ts_to	integer	

### RESPONSE

STATUS CODE - 200: List of requested segments

RESPONSE MODEL - application/json

```
{
  segments* [{
    Array of object:
    segment_url* string 1 to 65536 chars
    ts_start* integer
    ts_end* integer
    signature* string
  }]
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

STATUS CODE - 503: Service Unavailable

RESPONSE MODEL - application/json

```
{
  status          enum          DEFAULT:temporarily_unavailable
                                     ALLOWED:temporarily_unavailable
                                     An enumeration.
  status_code     integer      DEFAULT:503
  detail*         string
```

```
}
```

---

## 18.2 GET /api/v1/segments/index

### Get segments index file

Get url to stream segments index by `stream_id`

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*stream_id	uuid	

### RESPONSE

**STATUS CODE - 200:** Segments index file

#### RESPONSE MODEL - application/json

```
{
  index_url* string 1 to 65536 chars
  signature* string
}
```

**STATUS CODE - 404:** Not Found

#### RESPONSE MODEL - application/json

```
{
  status          enum          DEFAULT:not_found
                                     ALLOWED:not_found
                                     An enumeration.
  status_code     integer       DEFAULT:404
  detail*        string
}
```

**STATUS CODE - 422:** Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

#### RESPONSE MODEL - application/json

**STATUS CODE - 503:** Service Unavailable

#### RESPONSE MODEL - application/json

```
{
  status          enum          DEFAULT:temporarily_unavailable
                                     ALLOWED:temporarily_unavailable
}
```

An enumeration.

```
status_code integer DEFAULT:503
detail*     string
}
```

---

# 19. SNAPSHOTS

## 19.1 GET /api/v1/snapshots/{event\_id}

Get snapshot of requested event

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*event_id	uuid	

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*event_type	enum	
		ALLOWED: face_recognition, plate_recognition, motion_detection, line_crossing

### RESPONSE

STATUS CODE - 200: Snapshot of requested event

RESPONSE MODEL - application/json

undefined

RESPONSE MODEL - application/octet-stream

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

STATUS CODE - 503: Service Unavailable



## RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}
```

## 19.2 GET /api/v1/camerasnapshots/{camera\_id}

### Get snapshot of requested camera

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*camera_id	uuid	

#### RESPONSE

STATUS CODE - 200: Snapshot of requested camera

##### RESPONSE MODEL - application/json

undefined

##### RESPONSE MODEL - application/octet-stream

STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}
```

STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```
{
  detail [ {
    Array of object:
```

```
    loc*   [string]
    msg*   string
    type*  string
  }]
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

**STATUS CODE - 503: Service Unavailable**

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer    DEFAULT:503
  detail*     string
}
```

---

# 20. STATS

## 20.1 GET /api/v1/stats/cameras

Get number of total and active cameras

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
organization_id	integer	
group_id	integer	

### RESPONSE

STATUS CODE - 200: Information about cameras count

RESPONSE MODEL - application/json

```
{
  total_count* integer
  active_count* integer
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

STATUS CODE - 503: Service Unavailable

RESPONSE MODEL - application/json

```
{
  status          enum          DEFAULT:temporarily_unavailable
                  ALLOWED:temporarily_unavailable
                  An enumeration.
  status_code     integer      DEFAULT:503
  detail*         string
}
```

## 20.2 GET /api/v1/stats/eventsbytype

Get raw recognition stats in the provided time range

Get raw recognitions stats for the provided time period grouped by event\_type

For time ranges exceeding 24 hours stats periods will be grouped in days, otherwise - in hours.

## REQUEST

### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*ts_from	date-time	
*ts_to	date-time	
organization_id	integer	
group_id	integer	

## RESPONSE

**STATUS CODE - 200:** Requested event

**RESPONSE MODEL - application/json**

```
{
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
              ALLOWED:invalid_request
              An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
              ALLOWED:not_found
              An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 20.3 GET /api/v1/stats/eventmatchesbypriority

### Get raw recognition stats in the provided time range

Get recognition match stats for the provided time period grouped by match priority

For time ranges exceeding 24 hours stats periods will be grouped in days, otherwise - in hours.

## REQUEST

### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*ts_from	date-time	
*ts_to	date-time	
organization_id	integer	
group_id	integer	

## RESPONSE

STATUS CODE - 200: Requested event

RESPONSE MODEL - application/json

```
{
}
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}

```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

## 20.4 GET /api/v1/stats/eventsintensity

**Get event intensity in time period captured by stream with provided ID**

Get event intensity occurred in time period captured by stream with provided ID

Event type must be either 'face\_recognition', 'plate\_recognition', 'motion\_detection' or 'line\_crossing'.

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*stream_id	uuid	
*ts_from	date-time	
*ts_to	date-time	
organization_id	integer	
group_id	integer	

### RESPONSE

**STATUS CODE - 200:** Event intensity

**RESPONSE MODEL - application/json**

```

{
  meta {
    event_types {
    }
  }
  entries* [{
    Array of object:
    ts* string
    fr integer
    pr integer
    md integer
    lc integer
  }]
}

```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}
```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

---

# 21. STREAMS

## 21.1 GET /api/v1/streams/{stream\_id}

Get stream information by provided ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*stream_id	uuid	

### RESPONSE

STATUS CODE - 200: Information about requested stream

#### RESPONSE MODEL - application/json

```
{
  streamserver_id*  string
  streamserver_host* string
  status*           string
  ts*              integer
}
```

STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*    string
}
```

STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*  [string]
    msg*  string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

STATUS CODE - 503: Service Unavailable

#### RESPONSE MODEL - application/json

```
{
```



```

    status      enum      DEFAULT:temporarily_unavailable
                  ALLOWED:temporarily_unavailable
                  An enumeration.
    status_code integer    DEFAULT:503
    detail*     string
}

```

---

## 21.2 GET /api/v1/streams/{stream\_id}/info

Get stream information with provided ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*stream_id	uuid	

### RESPONSE

STATUS CODE - 200: Stream source information

RESPONSE MODEL - application/json

```

{
  frame_rate*  integer
  bit_rate*    integer
  frame_width* integer
  frame_height* integer
  codec*       string
}

```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}

```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}

```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```

{

```

```

detail [{
  Array of object:
    loc*   [string]
    msg*   string
    type*  string
}]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

**STATUS CODE - 503: Service Unavailable**

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer  DEFAULT:503
  detail*     string
}

```

## 21.3 POST /api/v1/streams/verify

Verify stream by provided RTSP link

### REQUEST

**REQUEST BODY - application/json**

```

{
  rtsp_link* string 1 to 512 chars
}

```

### RESPONSE

**STATUS CODE - 200: Stream verification status**

**RESPONSE MODEL - application/json**

```

{
  status string DEFAULT:ok
}

```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}

```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```

{
  detail [{

```

Array of object:

```
loc* [string]
msg* string
type* string
}]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

**STATUS CODE - 503: Service Unavailable**

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer  DEFAULT:503
  detail*     string
}
```

---

## 22. STREAMSERVERS

### 22.1 GET /api/v1/streamservers

Get information about stream servers state

#### REQUEST

No request parameters

#### RESPONSE

**STATUS CODE - 200:** List of stream servers state

**RESPONSE MODEL - application/json**

```
{
  streamservers* [{
    Array of object:
    streamserver_id* string
    group             string
    tags*             [string]
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

**STATUS CODE - 503:** Service Unavailable

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer  DEFAULT:503
  detail*     string
}
```

---

## 23. STREAMSTATUSES

### 23.1 GET /api/v1/streamstatuses/summary

Get uptime statistics for all streams

#### REQUEST

No request parameters

#### RESPONSE

STATUS CODE - 200: Information about stream uptime statistics

RESPONSE MODEL - application/json

```
{
  stream_status_summary* [{
    Array of object:
    stream_id*    string
    stream_name*  string 1 to 255 chars
    lifetime*     number
    availability* number between 0 and 100
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

STATUS CODE - 503: Service Unavailable

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:temporarily_unavailable
                ALLOWED:temporarily_unavailable
                An enumeration.
  status_code integer  DEFAULT:503
  detail*     string
}
```

---

### 23.2 GET /api/v1/streamstatuses/{stream\_id}

Get downtime statistics for specified stream

#### REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*stream_id	uuid	

#### RESPONSE

STATUS CODE - 200: Information about stream downtime periods

RESPONSE MODEL - application/json

```

{
  downtime_stats* [{
    Array of object:
    ts*          string
    downtime_start* string
    downtime_end* string
    reason       string 1 to 255 chars
  }]
}

```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}

```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

**STATUS CODE - 503: Service Unavailable**

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:temporarily_unavailable
                  ALLOWED:temporarily_unavailable
                  An enumeration.
  status_code integer  DEFAULT:503
  detail*     string
}

```

---

# 24. TOKEN

## 24.1 POST /api/v1/token

### Obtain OAuth2 token

Obtain OAuth2 token by providing valid user credentials:

- `grant_type`: Must always be set to 'password'
- `username`: Provide username
- `password`: Provide user password

Obtained `access_token` may be later used to access protected resources, use appropriate HTTP header:

Authorization: Bearer `obtained_access_token`

The following status codes are defined for 401 response:

- `token_invalid_credentials` - Invalid credentials were provided

### REQUEST

#### FORM DATA PARAMETERS

NAME	TYPE	DESCRIPTION
<code>grant_type</code>	string	
<code>username</code>	string	
<code>password</code>	string	
<code>scope</code>	string	
<code>client_id</code>	string	
<code>client_secret</code>	string	

### RESPONSE

STATUS CODE - 200: Obtained token

#### RESPONSE MODEL - application/json

```
{
  access_token*  string  max:4096 chars
  refresh_token* string  max:4096 chars
  token_type*    string  PATTERN:bearer
  expires_in*    integer
}
```

STATUS CODE - 401: Unauthorized

#### RESPONSE MODEL - application/json

```
{
  status*        enum      ALLOWED:token_invalid_credentials
                  An enumeration.
  status_code    integer  DEFAULT:401
  detail*        string
}
```

```
}
```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

---

## 24.2 POST /api/v1/token/refresh

### Refresh OAuth2 token

Refresh OAuth2 access token by providing valid refresh token:

- `grant_type`: Must always be set to 'refresh\_token'
- `refresh_token`: Valid refresh token

Obtained `access_token` may be later used to access protected resources, use appropriate HTTP header:

Authorization: Bearer `obtained_access_token`

The following status codes are defined for 400 response:

- `token_refresh_error` - Any error related to token refreshing

## REQUEST

### FORM DATA PARAMETERS

NAME	TYPE	DESCRIPTION
<code>grant_type</code>	string	
<code>refresh_token</code>	string	

## RESPONSE

**STATUS CODE - 200: Refreshed token**

**RESPONSE MODEL - application/json**

```
{
  access_token* string max:4096 chars
  refresh_token* string max:4096 chars
  token_type* string PATTERN:bearer
  expires_in* integer
}
```

**STATUS CODE - 400: Bad Request**



#### RESPONSE MODEL - application/json

```
{
  status*      enum      ALLOWED:token_refresh_error
                  An enumeration.
  status_code  integer   DEFAULT:400
  detail*      string
}
```

#### STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

#### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

---

## 24.3 POST /api/v1/token/revoke

### Revoke OAuth2 token

Revoke OAuth2 token by providing revocable token:

- `token`: Revocable token (access or refresh token)
- `token_type_hint`: Used to specify which token type is going to be revoked. If not set, this will be determined automatically.

If access token was provided, revokes it. If refresh token was provided, revokes it and all the corresponding access tokens.

The following status codes are defined for 400 response:

- `token_revoke_error` - Any error related to token revocation

## REQUEST

### FORM DATA PARAMETERS

NAME	TYPE	DESCRIPTION
<code>token</code>	string	
<code>token_type_hint</code>	string	An enumeration.

## RESPONSE

STATUS CODE - 204: Token revoked successfully

STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  status*      enum      ALLOWED:token_revoke_error
                An enumeration.
  status_code  integer   DEFAULT:400
  detail*      string
}
```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

---

# 25. UPLOADS

## 25.1 POST /api/v1/uploads

### Upload file

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*kind	enum	ALLOWED: avatar, face_image, vehicle_image, plan, camera_snapshot, logo

#### FORM DATA PARAMETERS

NAME	TYPE	DESCRIPTION
upload_data	string(binary)	

### RESPONSE

STATUS CODE - 200: Created upload

#### RESPONSE MODEL - application/json

```
{
  uploads* [{
    mime_type* string
    kind* enum ALLOWED:avatar, face_image, vehicle_image, plan,
               camera_snapshot, logo
               An enumeration.
    original_id string
    id* string
    created_at* string
    url string 1 to 65536 chars
    meta {
      detection {
        rect* {
          x_min* number
          y_min* number
          x_max* number
          y_max* number
        }
        landmarks* [{
          Array of object:
          x* number
          y* number
        }]
      }
      descriptor {
        data* string
        norm* number
      }
    }
  }
}
```

```
}]
}
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

---

## 25.2 GET /api/v1/uploads/{upload\_id}

Request upload with provided ID

### REQUEST

**PATH PARAMETERS**

NAME	TYPE	DESCRIPTION
*upload_id	uuid	

### RESPONSE

**STATUS CODE - 200: Requested upload**

**RESPONSE MODEL - application/json**

```
{
  mime_type* string
  kind*      enum      ALLOWED:avatar, face_image, vehicle_image, plan, camera_snapshot,
                  logo
                  An enumeration.
  original_id string
  id*        string
  created_at* string
  url        string  1 to 65536 chars
  meta {
    detection {
```

```

    rect* {
      x_min* number
      y_min* number
      x_max* number
      y_max* number
    }
    landmarks* [{
      Array of object:
      x* number
      y* number
    }]
  }
  descriptor {
    data* string
    norm* number
  }
}

```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}

```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```

{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

## 25.3 GET /api/v1/uploads/{upload\_id}/content

Download requested file

### REQUEST

**PATH PARAMETERS**

NAME	TYPE	DESCRIPTION
*upload_id	uuid	

## RESPONSE

STATUS CODE - 200: Requested file

RESPONSE MODEL - application/json

undefined

RESPONSE MODEL - application/octet-stream

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```
{
  status      enum      DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 25.4 POST /api/v1/uploads/meta

Get meta of uploaded object

### REQUEST

FORM DATA PARAMETERS

NAME	TYPE	DESCRIPTION
upload_data	string(binary)	

### RESPONSE

STATUS CODE - 200: Uploaded object meta

RESPONSE MODEL - application/json

```
{
  meta* [{
    Array of object:
    detection {
      rect* {

```

```

        x_min* number
        y_min* number
        x_max* number
        y_max* number
    }
    landmarks* [{
    Array of object:
        x* number
        y* number
    }]
}
descriptor {
    data* string
    norm* number
}
}]
}

```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```

{
    status      enum      DEFAULT:invalid_request
                  ALLOWED:invalid_request
                  An enumeration.
    status_code integer  DEFAULT:400
    detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
    detail [{
    Array of object:
        loc* [string]
        msg* string
        type* string
    }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

# 26. USERS

## 26.1 GET /api/v1/users

Get list of users

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
without_groups	boolean	
cursor	string	
limit	integer	between 10 and 1000

### RESPONSE

STATUS CODE - 200: List of users

RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
  users* [{
    Array of object:
    first_name* string max:32 chars
    last_name* string max:32 chars
    avatar_id string
    email* string
    id* integer
    organization_id* integer
    organization_name* string
    avatar_url string 1 to 65536 chars
    groups* [{
      Array of object:
      id* integer
      name* string
    }]
    is_superuser* boolean
    role_id* integer
  }]
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```



```
}]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 26.2 POST /api/v1/users

### Create new user

Create new user

Only superuser can explicitly set the organization ID.  
Otherwise it will be inferred from the current user.  
Only one group ID can be provided, HTTP 422 will be returned  
if group\_ids contains more than one group ID.

Returns 400 BAD REQUEST in the following cases:

- User with provided email already exists
- Some of provided groups do not exist
- Provided avatar does not exist
- Provided organization does not exist
- Organization was provided despite the user is not superuser

## REQUEST

### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
organization_id	integer	

---

### REQUEST BODY - application/json

```
{
  first_name* string    max:32 chars
  last_name*  string    max:32 chars
  avatar_id   string
  email*     string
  password*  string    max:128 chars
  group_ids* [integer]
  role_id*   integer
}
```

## RESPONSE

STATUS CODE - 200: Created user

RESPONSE MODEL - application/json

```
{
  first_name*   string    max:32 chars
  last_name*    string    max:32 chars
  avatar_id     string
  email*       string
  id*          integer
  organization_id* integer
  organization_name* string
  avatar_url    string    1 to 65536 chars
  groups*      [{
```

Array of object:

```
    id*   integer
    name* string
  }]
  is_superuser*   boolean
  role_id*        integer
}
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:invalid_request
                ALLOWED:invalid_request
                An enumeration.
  status_code integer  DEFAULT:400
  detail*     string
}
```

**STATUS CODE - 422: Validation Error**

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

---

## 26.3 GET /api/v1/users/me

**Get currently authenticated user**

### REQUEST

No request parameters

### RESPONSE

**STATUS CODE - 200: Authenticated user**

**RESPONSE MODEL - application/json**

```
{
  first_name*   string  max:32 chars
  last_name*    string  max:32 chars
  avatar_id     string
  email*        string
  id*           integer
  organization_id* integer
  organization_name* string
  avatar_url    string  1 to 65536 chars
  groups*      [{
    Array of object:
```

```

    id*    integer
    name*  string
  }]
  is_superuser*  boolean
  role_id*       integer
  permissions*   [enum]  ALLOWED:auditlog_read, camera_create, camera_delete,
                        camera_read, camera_snapshot_read, camera_update,
                        event_read, faceimage_create, faceimage_delete,
                        faceimage_read, facelist_create, facelist_delete,
                        facelist_face_create, facelist_face_delete,
                        facelist_face_read, facelist_read, facelist_update,
                        face_create, face_delete, face_read, face_update,
                        group_create, group_delete, group_place_create,
                        group_place_delete, group_place_read, group_read,
                        group_update, group_user_create, group_user_delete,
                        group_user_read, onvif_create, onvif_delete, onvif_read,
                        onvif_update, orion_create, orion_read, orion_update,
                        orion_delete, organization_create, organization_delete,
                        organization_groups_read, organization_members_read,
                        organization_read, organization_update,
                        permission_create, permission_delete, permission_read,
                        place_create, place_delete, place_group_create,
                        place_group_delete, place_group_read, place_plan_read,
                        place_read, place_update, plan_camera_create,
                        plan_camera_delete, plan_camera_read,
                        plan_camera_update, plan_create, plan_delete, plan_read,
                        plan_update, platelist_create, platelist_delete,
                        platelist_plate_create, platelist_plate_delete,
                        platelist_plate_read, platelist_read, platelist_update,
                        plate_create, plate_delete, plate_read, plate_update,
                        role_create, role_delete, role_member_read,
                        role_perm_read, role_read, role_update, search_face,
                        search_plate, segment_read, snapshot_read, stats_read,
                        stream_read, streamserver_read, streamstatus_read,
                        upload_create, upload_meta_read, upload_read,
                        user_create, user_delete, user_me_read, user_me_update,
                        user_read, user_update, videoclip_create,
                        videoclip_delete, videoclip_read
                        An enumeration.
}

```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

## 26.4 PUT /api/v1/users/me

Update currently authenticated user

### REQUEST

**REQUEST BODY - application/json**

```

{
  first_name*  string  max:32 chars
  last_name*   string  max:32 chars
  avatar_id    string
}

```

## RESPONSE

STATUS CODE - 200: Updated authenticated user

RESPONSE MODEL - application/json

```
{
  first_name*      string  max:32 chars
  last_name*       string  max:32 chars
  avatar_id        string
  email*           string
  id*              integer
  organization_id* integer
  organization_name* string
  avatar_url       string  1 to 65536 chars
  groups* [ {
    Array of object:
      id*  integer
      name* string
    } ]
  is_superuser*   boolean
  role_id*        integer
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [ {
    Array of object:
      loc* [string]
      msg*  string
      type* string
    } ]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 26.5 GET /api/v1/users/{user\_id}

Get user with provided ID

### REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*user_id	integer	

### RESPONSE

STATUS CODE - 200: Requested user

RESPONSE MODEL - application/json

```
{
  first_name*      string  max:32 chars
```

```

last_name*      string  max:32 chars
avatar_id       string
email*          string
id*             integer
organization_id* integer
organization_name* string
avatar_url      string  1 to 65536 chars
groups* [{
  Array of object:
    id* integer
    name* string
}]
is_superuser*   boolean
role_id*        integer
}

```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```

{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}

```

#### STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

```

{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}

```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

## 26.6 PUT /api/v1/users/{user\_id}

Update user with provided ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*user_id	integer	

#### REQUEST BODY - application/json

```

{
  first_name* string  max:32 chars
}

```

```

last_name*  string    max:32 chars
avatar_id   string
email*      string
password    string    max:128 chars
group_ids*  [integer]
role_id*    integer
}

```

## RESPONSE

**STATUS CODE - 200:** Updated user

**RESPONSE MODEL - application/json**

```

{
  first_name*  string    max:32 chars
  last_name*   string    max:32 chars
  avatar_id    string
  email*       string
  id*          integer
  organization_id* integer
  organization_name* string
  avatar_url   string    1 to 65536 chars
  groups*     [{
    Array of object:
      id*      integer
      name*    string
    }]
  is_superuser* boolean
  role_id*    integer
}

```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:invalid_request
              ALLOWED:invalid_request
              An enumeration.
  status_code integer    DEFAULT:400
  detail*     string
}

```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```

{
  status      enum      DEFAULT:not_found
              ALLOWED:not_found
              An enumeration.
  status_code integer    DEFAULT:404
  detail*     string
}

```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```

{
  detail [ {
    Array of object:

```

```
    loc*  [string]
    msg*  string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

---

## 26.7 DELETE /api/v1/users/{user\_id}

Delete user with provided ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*user_id	integer	

---

### RESPONSE

**STATUS CODE - 200:** User deleted successfully

**RESPONSE MODEL - application/json**

```
{
  status string DEFAULT:ok
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*  [string]
    msg*  string
    type* string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

---

## 26.8 PUT /api/v1/users/me/password

Update currently authenticated user password

### REQUEST

REQUEST BODY - application/json

```
{
  old_password* string max:128 chars
  new_password* string max:128 chars
}
```

### RESPONSE

STATUS CODE - 200: Updated authenticated user password

RESPONSE MODEL - application/json

```
{
  first_name*      string  max:32 chars
  last_name*       string  max:32 chars
  avatar_id        string
  email*           string
  id*              integer
  organization_id* integer
  organization_name* string
  avatar_url       string  1 to 65536 chars
  groups* [{
    Array of object:
      id* integer
      name* string
    }]
  is_superuser*    boolean
  role_id*         integer
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
      loc* [string]
      msg* string
      type* string
    }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 26.9 PUT /api/v1/users/me/email

Update currently authenticated user email

### REQUEST

REQUEST BODY - application/json

```
{
```



```
password* string max:128 chars
email*    string
}
```

## RESPONSE

**STATUS CODE - 200:** Updated authenticated user email

**RESPONSE MODEL - application/json**

```
{
  first_name*    string  max:32 chars
  last_name*     string  max:32 chars
  avatar_id      string
  email*         string
  id*            integer
  organization_id* integer
  organization_name* string
  avatar_url     string  1 to 65536 chars
  groups* [ {
    Array of object:
      id*    integer
      name*  string
    } ]
  is_superuser* boolean
  role_id*    integer
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [ {
    Array of object:
      loc* [string]
      msg* string
      type* string
    } ]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

---

## 26.10 POST /api/v1/users/search

### Search users by provided query

Get list of users filtered by query.

If query is valid email, the search is done over user emails.  
Otherwise search is done over two fields:

- first\_name + ' ' + last\_name
- last\_name + ' ' + first\_name

Length of query must be in range [2;255]

## REQUEST

## QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
without_groups	boolean	
organization_id	integer	
cursor	string	
limit	integer	between 10 and 1000

## REQUEST BODY - application/json

```
{
  query* string 2 to 255 chars
}
```

## RESPONSE

### STATUS CODE - 200: List of found users

#### RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
  users* [{
    Array of object:
    first_name* string max:32 chars
    last_name* string max:32 chars
    avatar_id string
    email* string
    id* integer
    organization_id* integer
    organization_name* string
    avatar_url string 1 to 65536 chars
    groups* [{
      Array of object:
      id* integer
      name* string
    }]
    is_superuser* boolean
    role_id* integer
  }]
}
```

### STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

### STATUS CODE - 500: Internal Server Error



# 27. VIDECLIPS

## 27.1 GET /api/v1/videoclips

Get a list of video clips

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
camera_id	uuid	
status	enum ALLOWED: queued, running, failed, aborted, done	
created_at_start	date-time	
created_at_end	date-time	

### RESPONSE

STATUS CODE - 200: List of video clips

RESPONSE MODEL - application/json

```
{
  pagination* {
    cursor string
  }
  videoclips* [{
    Array of object:
    camera_id* string
    range_start* string
    range_end* string
    id* string
    camera_name* string
    created_at* string
    finished_at string
    progress number DEFAULT:0
    result {
      url* string 1 to 65536 chars
    }
    status* enum ALLOWED:queued, running, failed, aborted, done
    An enumeration.
  }]
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

```
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

---

## 27.2 POST /api/v1/videoclips

### Create a new video clip

Schedule to create a new video clip for selected camera within given time range.

The following status codes are defined for 400 response:

- `videoclip_invalid_time_range` - Attempt to create a video clip with
- time range that is outside of camera's archive duration

The following status codes are defined for 404 response:

- `videoclip_camera_not_found` - Attempt to create a video clip
- for camera that does not exist

The following status codes are defined for 429 response:

- `videoclip_user_quota_exceeded` - Attempt to create a video clip when user already reached
- limit on scheduled video clips

## REQUEST

**REQUEST BODY - application/json**

```
{
  camera_id*   string
  range_start* string
  range_end*   string
}
```

## RESPONSE

**STATUS CODE - 200:** Created video clip

**RESPONSE MODEL - application/json**

```
{
  camera_id*   string
  range_start* string
  range_end*   string
  id*          string
  camera_name* string
  created_at*  string
  finished_at  string
  progress     number  DEFAULT:0
  result {
    url* string 1 to 65536 chars
  }
  status*      enum    ALLOWED:queued, running, failed, aborted, done
                An enumeration.
}
```

**STATUS CODE - 400:** Bad Request

#### RESPONSE MODEL - application/json

```
{
  status*      enum      ALLOWED:videoclip_invalid_time_range
                  An enumeration.
  status_code  integer    DEFAULT:400
  detail*      string
}
```

#### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  status*      enum      ALLOWED:videoclip_camera_not_found
                  An enumeration.
  status_code  integer    DEFAULT:404
  detail*      string
}
```

#### STATUS CODE - 422: Validation Error

#### RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

#### STATUS CODE - 429: Too Many Requests

#### RESPONSE MODEL - application/json

```
{
  status*      enum      ALLOWED:videoclip_user_quota_exceeded
                  An enumeration.
  status_code  integer    DEFAULT:429
  detail*      string
}
```

#### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

---

## 27.3 GET /api/v1/videoclips/{videoclip\_id}

Get video clip with provided ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*videoclip_id	uuid	

### RESPONSE

STATUS CODE - 200: Requested video clip

RESPONSE MODEL - application/json

```
{
  camera_id*   string
  range_start* string
  range_end*   string
  id*          string
  camera_name* string
  created_at*  string
  finished_at  string
  progress     number DEFAULT:0
  result {
    url* string 1 to 65536 chars
  }
  status*      enum    ALLOWED:queued, running, failed, aborted, done
                An enumeration.
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```
{
  status      enum    DEFAULT:not_found
                ALLOWED:not_found
                An enumeration.
  status_code integer DEFAULT:404
  detail*     string
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
    loc* [string]
    msg* string
    type* string
  }]
}
```

STATUS CODE - 500: Internal Server Error

RESPONSE MODEL - application/json

---

## 27.4 DELETE /api/v1/videoclips/{videoclip\_id}

Delete video clip with requested ID

### REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*videoclip_id	uuid	

---

## RESPONSE

**STATUS CODE - 204:** Video clip successfully deleted

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  status      enum      DEFAULT:not_found
                  ALLOWED:not_found
                  An enumeration.
  status_code integer  DEFAULT:404
  detail*     string
}
```

**STATUS CODE - 422:** Validation Error

**RESPONSE MODEL - application/json**

```
{
  detail [{
    Array of object:
    loc*   [string]
    msg*   string
    type*  string
  }]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

---



