ROG server API

Authorization

Request:

POST /api/v2/sessions

Payload:

application/json: {“session”: {“email”: string, “password”: string}}

Response:

application/json: {“jwt”: string, “id”: int}

Get cameras

Request:

GET /api/v2/me/cameras

Authorization: jwt\_token

Response:

application/json: {

“data”: [

{“id”: string, “rtspUrl”: string},

…

]}

BVC backend API

Authorization

Request:

POST /api/auth

Payload: application/json: {“username”: string, “password”: string}

Response: application/json: {“access\_token”: string}

use this token in request header: “Authorization: JWT access\_token”

Get cameras

Request:

GET /api/user/<user\_id>/cameras

GET /api/cameras/<camera\_id>

GET /api/camera/<camera\_id>

Jwt required

Response: json:

{“cameras”: [ {“id”: int, “url”: string}, …]}

{“camera”: {“id”: int, “name”: string, “url”: string, “enabled”: bool, “connectedOnce”: bool}}

Add cameras

Request:

POST /api/user/<user\_id>/cameras

Authorization: JWT access\_token

Payload: json: [{“id”: int, “name”: string, “url”: string, “enabled”: bool}, …]

Response: 200, json: {}

Request:

POST /api/user/<user\_id>/camera

Authorization: JWT access\_token

Payload: json: {“id”: int, “name”: string, “url”: string, “enabled”: bool}

Response: 200, json: {}

Set cameras

Request:

PUT /api/user/<user\_id>/cameras

Authorization: JWT access\_token

Payload: json: [{“id”: int, “name”: string, “url”: string, “enabled”: bool}, …]

Response: 204, json: {}

Delete cameras

Request: DELETE /api/cameras/<camera\_id>

jwt required

Payload: json: { }

Response: 204, json: {}

Get camera config

Request:

GET /api/cameras/<camera\_id>/alerts

GET /api/cameras/<camera\_id>/alerts/<alert\_id>

Response: json:

{“alerts”: [ {

“id”: string,

"type": "VW"|"RA"|"LD" - alert type, virtual wall | restricted area | loitering detection

"duration": float - seconds for LA alert, only for LA

"direction": "R"|"L"|"B" - only for VW, direction: right, left, both

"points": array of points, where point - array of floats: [x, y]

}, … ]}

Put camera config

Payload: json:

{"alert": {

"type": "VW"|"RA"|"LD" - alert type, virtual wall | restricted area | loitering detection

"duration": float - seconds for LA alert, only for LA

"direction": "R"|"L"|"B" - only for VW, direction: right, left, both

"points": array of points, where point - array of floats: [x, y]

}}

Request: PUT /api/cameras/<camera\_id>/alerts/<alert\_id>

Response: 204

Request: POST /api/cameras/<camera\_id>/alerts

Response: 201, json: {“alert”: {“id: string”}}

Remove alert

DELETE /api/camera/<camera\_id>/alerts/<alert\_id>

DELETE /api/camera/<camera\_id>/alerts

Response: 204

Enable/Disable camera

PUT /api/camera/<camera\_id>/enabled

Payload: json: {“value”: bool}

Response: 204

Camera properties

GET /api/camera/<camera\_id>/enabled

GET /api/camera/<camera\_id>/connectedOnce

Response: 200, json: {“value”: bool}

Generated alert

POST /api/alerts/

Payload: json: {“camera\_id”: int, … 2DO}

Recognizer status

POST /api/reco\_status

Payload: json: {‘fps’:float, ‘cameras’: [{‘id’:int, ‘fps1’:float, ‘fps2’:float}, …]}

Response: 204

Recognizer stops working

POST /api/reco\_end

Payload: json: {}

Response: 204