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>

Jwt required

Response: json:

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

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

Set cameras

Request:

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

GET /api/cameras/<camera\_id>

Authorization: JWT access\_token

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

Response: json: {}

Delete cameras

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

jwt required

Payload: json: { }

Response: json: 204: {}

Get camera config

Request:

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

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

Response: json:

{“alerts”: [ {

“id”: string,

“type”: string, // Loitering Detection, Virtual Wall, Restricted Area

“data”: {<alert type specific data>}

}, … ]}

Put camera config

Request:

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

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

Payload: json: {“alert”: {“type”: string, “data”: {<alert type specific data>}}}

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

Remove alert

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

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

Enable/Disable camera

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

Payload: json: {“enabled”: bool}

Generated alert

POST /api/alerts/

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