raspiCamSrv

API for Raspberry Pi Camera Server (raspiCamSrv) https://github.com/signag/raspi-cam-srv

Security: JSON Web Tokens (JWT)

POST api login

<base_url>/api/login

Client login.

Returns: Access Token and Refresh Token

Body raw (json)

json

```json
{
    "username": "<user>",
    "password": "<password>"
}
```

POST api refresh

<base_url>/api/refresh

Refresh of Access Token

Authentication: Refresh Token

Response: Access Token

AUTHORIZATION Bearer Token

Token	<refresh_token>

GET api protected

```
<base_url>/api/protected
```

Dummy API for testing purposes

AUTHORIZATION Bearer Token

Token	<access_token>

GET api take_photo

```
<base_url>/api/take_photo
```

Take photo with active camera

AUTHORIZATION Bearer Token

Token	<access_token>

GET api take_photo 2

```
<base_url>/api/take_photo2
```

Take photo with second (non-active) camera

AUTHORIZATION Bearer Token

Token	<access_token>

GET api take_photo both

```
<base_url>/api/take_photo_both
```

Take photos simultaneously with both cameras.

The photos will have the same file name, but are stored in camera-specific subfolders.

.

.

AUTHORIZATION Bearer Token

Token	<access_token>

GET api take_raw_photo

<base_url>/api/take_raw_photo

Take raw photo with active camera.

In addition to the raw photo, also a normal photo is stored.

AUTHORIZATION Bearer Token

Token	<access_token>

GET api take_raw_photo2

<base_url>/api/take_raw_photo2

Take raw photo with second (non-active) camera.

In addition to the raw photo, also a normal photo is stored.

AUTHORIZATION Bearer Token

Token	<access_token>

GET api take_raw_photo both

<base_url>/api/take_raw_photo_both

Take raw photos simultaneously with both cameras.

The photos will have the same file name, but are stored in camera-specific subfolders.

In addition to the raw photo, also a normal photo is stored.

.

.

AUTHORIZATION Bearer Token

Token	<access_token>

GET api record video

<base_url>/api/record_video

Record video with fixed duration using the active camera.

Data: video duration.

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

json

```json
{
    "duration": 30
}
```

GET api record video until stop

<base_url>/api/record_video

Start recording a video with active camera.

Recording must be stopped with **api stop video.**

AUTHORIZATION Bearer Token

Token	<token>

GET api stop video

<base_url>/api/stop_video

Stop video recording with the active camera.

AUTHORIZATION Bearer Token

Token	<token>

GET api record video 2

<base_url>/api/record_video2

Record video with fixed duration using the second (non-active) camera.

Data: video duration.

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

json

```json
{
    "duration": 30
}
```

GET api record video 2 until stop

<base_url>/api/record_video2

Start recording a video with the second (non-active) camera.

Recording must be stopped with **api stop video2.**

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

json

```json
{
}
```

GET api stop video 2

<base_url>/api/stop_video2

Stop recording a video with the second camera.

AUTHORIZATION Bearer Token

Token	<token>

GET api record video both

<base_url>/api/record_video_both

Simultaneously record fixed-length videos with both cameras.

The videos will get the same name but will be stored in camera-specific subdirectories.

Data: video duration.

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

json

```
{
    "duration": 30
}
```

GET api record video both until stop

<base_url>/api/record_video_both

Start simultaneously recording videos with both cameras.

The videos will get the same name but will be stored in camera-specific subdirectories.

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

```json
{
}
```

GET api stop video both

<base_url>/api/stop_video_both

Stop recording videos with both cameras

AUTHORIZATION Bearer Token

Token	<token>

GET api switch cameras

<base_url>/api/switch_cameras

Switch cameras for systems with 2 cameras.

AUTHORIZATION Bearer Token

Token	<access_token>

GET api start motion detection

<base_url>/api/start_triggered_capture

Start motion detection

AUTHORIZATION Bearer Token

Token	<access_token>

GET api stop motion detection

<base_url>/api/stop_triggered_capture

Stop motion detection

AUTHORIZATION Bearer Token

Token	<access_token>

GET api info

<base_url>/api/info

Get status information from server:

Plain Text

```
{
    "message": {
        "active_camera": "Camera 0 (imx708)",
        "cameras": [
            {
                "active": true,
                "is_usb": false,
                "model": "imx708",
                "num": 0,
                "status": "open - started - current Sensor Mode: 2"
            },
```

AUTHORIZATION Bearer Token

Token	<access_token>

GET api probe

<base_url>/api/probe

Probe a set of object properties.

You need to specify an object through one of the singleton base classes (Camera(), CameraCfg(), MotionDetector(), PhotoSeriesCfg() or TriggerHandler()) and then specify valid properties with dot-notation.

Note: Not all properties might be JSON-serializable.

AUTHORIZATION Bearer Token

Token	<access_token>

Body raw (json)

json

```json
{
    "properties": [
        {
            "property": "Camera().event"
        },
        {
            "property": "Camera().event2"
        },
        {
            "property": "Camera().last_access"
        },
        {
            "property": "Camera().last_access2"
        },
        {
            "property": "Camera().threadLock.locked()"
        },
        {
            "property": "Camera().thread2Lock.locked()"
        },
        {
            "property": "CameraCfg().serverConfig.error"
        },
        {
            "property": "CameraCfg().serverConfig.error2"
        },
        {
            "property": "CameraCfg().serverConfig.errorc2"
        },
        {
            "property": "CameraCfg().serverConfig.errorc22"
        }
    ]
}
```