

# ANT-37500 API guide for firmware 3.3.1

## Contents

Document info.....	2
1. Introduction.....	9
2. EXAMPLE PYTHON CODE (v2.7.13) .....	9
3. Changes .....	12
4. PUT /api/camera .....	13
5. DELETE /api/camera/<idx:int> .....	14
6. GET /api/camera/<idx:int> .....	15
7. PUT /api/camera/<idx:int>.....	16
8. GET /api/camera/discover/<idx:int> .....	17
9. DELETE /api/cameras.....	18
10. GET /api/cameras .....	19
11. PUT /api/cameras .....	20
12. PUT /api/cameras/restart .....	21
13. GET /api/display .....	22
14. GET /api/display/modes .....	23
15. POST /api/login .....	24
16. POST /api/logout .....	25
17. GET /api/network/<iface> .....	26
18. PUT /api/network/<iface> .....	27
19. PUT /api/network/restart/<iface> .....	28
20. PUT /api/network/wifi/connect/<iface> .....	29
21. GET /api/network/wifi/scan.....	30
22. GET /api/onvif/cameras.....	31
23. GET /api/onvif/profiles .....	32
24. GET /api/system/config .....	33
25. POST /api/system/config .....	35
26. PUT /api/system/config .....	36
27. DELETE /api/system/config/<v>.....	37
28. GET /api/system/config/<v>.....	38
29. PUT /api/system/config/views .....	39
30. PUT /api/system/factoryreset.....	40
31. GET /api/system/info .....	41
32. PUT /api/system/reboot.....	42
33. PUT /api/system/restart .....	43
34. DELETE /api/system/ssl.....	44
35. POST /api/system/ssl/cert .....	45
36. POST /api/system/ssl/key.....	46
37. PUT /api/system/user.....	47
38. GET /api/view .....	48
39. GET /api/view/<v:int> .....	49
40. PUT /api/view/<v:int>/play .....	50
41. GET /api/view/default .....	51
42. PUT /api/view/default.....	52
43. GET /api/view/mode .....	53

---

44.	PUT /api/view/mode .....	54
45.	GET /api/view/status .....	55
46.	GET /api/views.....	56
47.	Definitions .....	57

For further help and advice please contact Antrica on:  
 Email: [support@antrica.com](mailto:support@antrica.com)  
 Phone: +44 1628 626098 and ask for Technical support

## Document info

Version	date	author	Comments
0.1	18-Aug-23	David M	Initial draft for release 3.3.1 – based on 3.2.0

## Spotbox API Version 3.3.x

### Table of Contents

Document info.....	2
1. Introduction.....	9
2. EXAMPLE PYTHON CODE (v2.7.13) .....	9
2.1. Installing on Windows .....	9
3. Changes.....	12
3.1. V.1.1.1.....	12
3.2. V.2.0.x.....	12
3.3. V.3.1.x.....	12
3.4. V.3.3.x.....	12
4. PUT /api/camera .....	13
4.1. request ( <i>json object</i> ): .....	13
4.2. response ( <i>json object</i> ): .....	13
4.3. Example:.....	13
4.3.1. request: .....	13
4.3.2. response: .....	13
5. DELETE /api/camera/<idx:int> .....	14
5.1. response ( <i>json object</i> ): .....	14
5.2. Example:.....	14
5.2.1. request: .....	14
5.2.2. response: .....	14
6. GET /api/camera/<idx:int> .....	15
6.1. response ( <i>json object</i> ): .....	15
6.2. Example:.....	15
6.2.1. request: .....	15
6.2.2. response: .....	15
7. PUT /api/camera/<idx:int>.....	16
7.1. request ( <i>json object</i> ): .....	16
7.2. response ( <i>json object</i> ): .....	16
7.3. Example:.....	16
7.3.1. request: .....	16
7.3.2. response: .....	16
8. GET /api/camera/discover/<idx:int>.....	17
8.1. response ( <i>json object</i> ): .....	17
8.2. Example:.....	17
8.2.1. request: .....	17
8.2.2. response: .....	17
9. DELETE /api/cameras.....	18
9.1. response ( <i>json object</i> ): .....	18
9.2. Example:.....	18
9.2.1. request: .....	18
9.2.2. response: .....	18
10. GET /api/cameras.....	19
10.1. response ( <i>json object</i> ): .....	19
10.2. Example:.....	19
10.2.1. request: .....	19

10.2.2. response: .....	19
11. PUT /api/cameras .....	20
11.1. request ( <i>json object</i> ): .....	20
11.2. response ( <i>json object</i> ): .....	20
11.3. Example:.....	20
11.3.1. request: .....	20
11.3.2. response: .....	20
12. PUT /api/cameras/restart .....	21
12.1. response ( <i>json object</i> ): .....	21
12.2. Example:.....	21
12.2.1. request: .....	21
12.2.2. response: .....	21
13. GET /api/display .....	22
13.1. response ( <i>json object</i> ): .....	22
13.2. Example:.....	22
13.2.1. request: .....	22
13.2.2. response: .....	22
14. GET /api/display/modes .....	23
14.1. response ( <i>json object</i> ): .....	23
14.2. Example:.....	23
14.2.1. request: .....	23
14.2.2. response: .....	23
15. POST /api/login .....	24
15.1. request ( <i>json object</i> ): .....	24
15.2. response ( <i>json object</i> ): .....	24
15.3. Example:.....	24
15.3.1. request: .....	24
15.3.2. response: .....	24
16. POST /api/logout .....	25
16.1. response ( <i>json object</i> ): .....	25
16.2. Example:.....	25
16.2.1. request: .....	25
16.2.2. response: .....	25
17. GET /api/network/<iface> .....	26
17.1. response ( <i>json object</i> ): .....	26
17.2. Example:.....	26
17.2.1. request: .....	26
17.2.2. response: .....	26
18. PUT /api/network/<iface> .....	27
18.1. request ( <i>json object</i> ): .....	27
18.2. response ( <i>json object</i> ): .....	27
18.3. Example:.....	27
18.3.1. request: .....	27
18.3.2. response: .....	27
19. PUT /api/network/restart/<iface> .....	28
19.1. response ( <i>json object</i> ): .....	28
19.2. Example:.....	28
19.2.1. request: .....	28
19.2.2. response: .....	28
20. PUT /api/network/wifi/connect/<iface> .....	29
20.1. request ( <i>json object</i> ): .....	29
20.2. response ( <i>json object</i> ): .....	29

20.3.	Example:	29
20.3.1.	request:	29
20.3.2.	response:	29
21.	GET /api/network/wifi/scan	30
21.1.	response ( <i>json object</i> ):	30
21.2.	Example:	30
21.2.1.	request:	30
21.2.2.	response:	30
22.	GET /api/onvif/cameras	31
22.1.	response ( <i>json object</i> ):	31
22.2.	Example:	31
22.2.1.	request:	31
22.2.2.	response:	31
23.	GET /api/onvif/profiles	32
23.1.	params:	32
23.2.	response ( <i>json object</i> ):	32
23.3.	Example:	32
23.3.1.	request:	32
23.3.2.	response:	32
24.	GET /api/system/config	33
24.1.	response ( <i>json object</i> ):	33
24.2.	Example:	33
24.2.1.	request:	33
24.2.2.	response:	33
25.	POST /api/system/config	35
25.1.	request ( <i>Multipart-Encoded JSON File</i> ):	35
25.2.	response ( <i>json object</i> ):	35
25.3.	Example:	35
25.3.1.	request:	35
26.	PUT /api/system/config	36
26.1.	request ( <i>json object</i> ):	36
26.2.	response ( <i>json object</i> ):	36
26.3.	Example:	36
26.3.1.	request:	36
26.3.2.	response:	36
27.	DELETE /api/system/config/<v>	37
27.1.	response ( <i>json object</i> ):	37
27.2.	Example:	37
27.2.1.	request:	37
27.2.2.	response:	37
28.	GET /api/system/config/<v>	38
28.1.	response ( <i>json object</i> ):	38
28.2.	Example:	38
28.2.1.	request:	38
28.2.2.	response:	38
29.	PUT /api/system/config/views	39
29.1.	Example:	39
29.1.1.	request:	39
29.1.2.	response:	39
30.	PUT /api/system/factoryreset	40
30.1.	response ( <i>json object</i> ):	40
30.2.	Example:	40

30.2.1. request: .....	40
30.2.2. response: .....	40
31. GET /api/system/info .....	41
31.1. response ( <i>json object</i> ): .....	41
31.2. Example: .....	41
31.2.1. request: .....	41
31.2.2. response: .....	41
32. PUT /api/system/reboot.....	42
32.1. response ( <i>json object</i> ): .....	42
32.2. Example: .....	42
32.2.1. request: .....	42
32.2.2. response: .....	42
33. PUT /api/system/restart .....	43
33.1. response ( <i>json object</i> ): .....	43
33.2. Example: .....	43
33.2.1. request: .....	43
33.2.2. response: .....	43
34. DELETE /api/system/ssl.....	44
34.1. response ( <i>json object</i> ): .....	44
34.2. Example: .....	44
34.2.1. request: .....	44
34.2.2. response: .....	44
35. POST /api/system/ssl/cert .....	45
35.1. request ( <i>Multipart-Encoded File</i> ): .....	45
35.2. response ( <i>json object</i> ): .....	45
35.3. Example: .....	45
35.3.1. request: .....	45
35.3.2. response: .....	45
36. POST /api/system/ssl/key.....	46
36.1. request ( <i>Multipart-Encoded File</i> ): .....	46
36.2. response ( <i>json object</i> ): .....	46
36.3. Example: .....	46
36.3.1. request: .....	46
36.3.2. response: .....	46
37. PUT /api/system/user.....	47
37.1. request ( <i>json object</i> ): .....	47
37.2. response ( <i>json object</i> ): .....	47
37.3. Example: .....	47
37.3.1. request: .....	47
37.3.2. response: .....	47
38. GET /api/view .....	48
38.1. response ( <i>json object</i> ): .....	48
38.2. Example: .....	48
38.2.1. request: .....	48
38.2.2. response: .....	48
39. GET /api/view/<v:int> .....	49
39.1. response ( <i>json object</i> ): .....	49
39.2. Example: .....	49
39.2.1. request: .....	49
39.2.2. response: .....	49
40. PUT /api/view/<v:int>/play .....	50
40.1. request ( <i>json object</i> ): .....	50

40.2.	response ( <i>json object</i> ): .....	50
40.3.	Example:.....	50
40.3.1.	request: .....	50
40.3.2.	response: .....	50
41.	GET /api/view/default .....	51
41.1.	response ( <i>json object</i> ): .....	51
41.2.	Example:.....	51
41.2.1.	request: .....	51
41.2.2.	response: .....	51
42.	PUT /api/view/default.....	52
42.1.	request ( <i>json object</i> ): .....	52
42.2.	Example:.....	52
42.2.1.	request: .....	52
42.2.2.	response: .....	52
43.	GET /api/view/mode .....	53
43.1.	response ( <i>json object</i> ): .....	53
43.2.	Example:.....	53
43.2.1.	request: .....	53
43.2.2.	response: .....	53
44.	PUT /api/view/mode .....	54
44.1.	request ( <i>json object</i> ): .....	54
44.2.	response ( <i>json object</i> ): .....	54
44.3.	Example:.....	54
44.3.1.	request: .....	54
44.3.2.	response: .....	54
45.	GET /api/view/status .....	55
45.1.	response ( <i>json object</i> ): .....	55
45.2.	Example:.....	55
45.2.1.	request: .....	55
45.2.2.	response: .....	55
46.	GET /api/views.....	56
46.1.	response ( <i>json object</i> ): .....	56
46.2.	Example:.....	56
46.2.1.	request: .....	56
46.2.2.	response: .....	56
47.	Definitions .....	57
47.1.	GridModesT .....	57
47.2.	ProfileT .....	57
47.3.	ViewListT.....	57
47.4.	ViewCamT .....	57
47.5.	ViewsT .....	57
47.6.	CameraGridsT .....	58
47.7.	CameraPropsT.....	58
47.8.	CameraT .....	58
47.9.	NetStateT.....	59
47.10.	NetSecurity.....	59
47.11.	NetAgentParam .....	59
47.12.	NetIpv4.....	59
47.13.	NetworkInterfaceT .....	59
47.14.	cameras .....	59
47.15.	network.....	60
47.16.	GridFontE .....	60

47.17. GridFontT .....	60
47.18. DefaultViewT .....	60
47.19. WebUIT .....	60
47.20. RtspStreamT .....	60
47.21. RtspEncProps .....	60
47.22. RtspStreamAddress .....	61
47.23. RtspSvrT .....	61
47.24. user_config .....	61
47.25. display .....	61
47.26. https_enable .....	61

## 1. Introduction

The Spotbox provides an API for managing the device through HTTP requests. The API methods are implemented as a set of http routes.

Each method take parameters as part of the URL or as a JSON object in the body of the request.

All calls will return a JSON object with one of two properties:

```
{"result": <result object> } // Successful result
```

or

```
{"error": "ERROR_CODE", "msg": "message"} // Error result
```

The API is protected with a basic authorisation mechanism which requires the “/api/login” method to be called before any other methods are used.

On successful login a cookie is returned which must be provided for subsequent calls to be successful. The cookie will be removed by the “/api/logout” method.

## 2. EXAMPLE PYTHON CODE (v2.7.13)

There is a simple python application "spotbox\_api.py" provided to demonstrate sending API commands to the Spotbox.

### 2.1. Installing on Windows

The application requires python and an extra python module to be installed. To install python on you host computer download and install the following file:

<https://www.python.org/ftp/python/2.7.13/python-2.7.13.msi>

NOTE: Make sure the option to add python.exe to Path is selected:



The open a command prompt and confirm python is installed correctly:

```
c:\spotbox>python -V
Python 2.7.13

c:\spotbox>
```

Next install the "requests" module:

```
c:\spotbox>pip install -sprequests
Collecting requests
  Downloading requests-2.18.1-py2.py3-none-any.whl (88kB)
    100% |#####| 92kB 1.5MB/s
Collecting chardet<3.1.0,>=3.0.2 (from requests)
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133kB)
    100% |#####| 143kB 1.3MB/s
Collecting certifi>=2017.4.17 (from requests)
  Downloading certifi-2017.4.17-py2.py3-none-any.whl (375kB)
    100% |#####| 378kB 1.6MB/s
Collecting idna<2.6,>=2.5 (from requests)
  Downloading idna-2.5-py2.py3-none-any.whl (55kB)
    100% |#####| 61kB 1.3MB/s
Collecting urllib3<1.22,>=1.21.1 (from requests)
  Downloading urllib3-1.21.1-py2.py3-none-any.whl (131kB)
    100% |#####| 133kB 1.1MB/s
Installing collected packages: chardet, certifi, idna, urllib3, requests
Successfully installed certifi-2017.4.17 chardet-3.0.4 idna-2.5 requests-2.18.1 urllib3-1.21.1

c:\spotbox>
```

You should now be able to send commands to the Spotbox.

Running the command provides an interactive prompt. To send an API request type in the HTTP command ('get', 'put', 'post', 'delete') followed by the api route then the JSON parameter if required.

NOTE: Don't forget to always log into the spotbox as the first command.

For example:

```
c:\spotbox>spotbox_api.py -h 10.0.1.133
> post /api/login {"user":"admin", "password":"admin"}
request URL:
http://10.0.1.133/api/login

response:

> get /api/view/mode
request URL:
http://10.0.1.133/api/view/mode

response:
{"result": "grid_2x2"}
> put /api/view/mode {"mode":"grid_4x4"}
request URL:
http://10.0.1.133/api/view/mode

response:
{"result": "OK"}
> get /api/view/mode
request URL:
http://10.0.1.133/api/view/mode

response:
{"result": "grid_4x4"}
>
```

## 3. Changes

### 3.1. V.1.1.1

The following routes have been added:

```
GET /api/display/modes
POST /api/system/config
POST /api/system/ssl/key
POST /api/system/ssl/cert
```

### 3.2. V.2.0.x

The following routes have been added:

```
DELETE /api/system/ssl
```

### 3.3. V.3.1.x

The following routes have been added:

```
PUT /api/network/<iface>
PUT /api/network/restart/<iface>
GET /api/network/wifi/scan
PUT /api/network/wifi/connect/<iface>
```

Also to support WIFI the format of the 'network' config value has been changed. The old network config will be accepted and be transformed to the new format.

### 3.4. V.3.3.x

The following routes have been added:

```
GET /api/camera/discover/<idx>
GET /api/system/config/<item>
PUT /api/system/config/<item>
DELETE /api/system/config/<item>
```

The following config entry has been added:

```
/api/system/config/views
```

The following config entry has been modified:

```
/api/system/config/cameras
```

## 4. PUT /api/camera

Append new camera to camera list, return camera number

### 4.1. request (json object):

- *camera*: Refer to `#/definitions/CameraT`.

### 4.2. response (json object):

- *result* (integer)

### 4.3. Example:

#### 4.3.1. request:

**PUT /api/camera**

```
{"camera": {"label": "C3", "tag": "C3", "uri": "rtsp://192.168.86.13:554/media/stream.sdp?profile=Profile100", "username": "admin", "password": "**password**"}
}
```

#### 4.3.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": 2}
```

## 5. DELETE /api/camera/<idx:int>

Delete camera *idx*

### 5.1. response (*json object*):

- *result*
  - Any of
    - *integer*
    - *null*

### 5.2. Example:

#### 5.2.1. request:

```
DELETE /api/camera/2
```

#### 5.2.2. response:

```
200 OK  
Content-Type: application/json  
{"result": 2}
```

## 6. GET /api/camera/<idx:int>

Get camera definition for camera *idx*

### 6.1. response (*json object*):

- *result*: Refer to *#/definitions/CameraT*.

### 6.2. Example:

#### 6.2.1. request:

```
GET /api/camera/0
```

#### 6.2.2. response:

```
200 OK
```

```
Content-Type: application/json
```

```
{"result": {"tag": "C1", "uri": "rtsp://192.168.86.10", "label": "C1", "devinfo": "rtsp:", "idx": 0, "lab_position": 0, "password": "**password**", "restart": 0, "username": "user", "zoom": 0, "props": {"audio": 0, "latency": 0, "protocols": 0, "timeout": 0}, "profiles": []}}
```

## 7. PUT /api/camera/<idx:int>

Update camera definition for camera *idx*

### 7.1. request (json object):

- *camera*: Refer to `#/definitions/CameraT`.

### 7.2. response (json object):

- *result* (integer)

### 7.3. Example:

#### 7.3.1. request:

**PUT /api/camera/2**

```
{"camera": {"label": "NEW LABEL", "tag": "new"}}
```

#### 7.3.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": 2}
```

## 8. GET /api/camera/discover/<idx:int>

Probe camera stream returning its profiles if onvif or simplified profile for other sources

### 8.1. response (json object):

- *result* (object)
  - *devinfo* (string)
  - *profiles* (array)
    - Items: Refer to *#/definitions/ProfileT*.

### 8.2. Example:

#### 8.2.1. request:

```
GET /api/camera/discover/0
```

#### 8.2.2. response:

```
200 OK
```

```
Content-Type: application/json
```

```
{"result": {"devinfo": "rtsp:", "profiles": [{"name": "video", "codec": "video/x-h264", "res_width": 1920, "res_height": 1080, "fps": "15", "uri": "rtsp://user:a1b2c3d4@192.168.86.10"}]}}
```

## 9. DELETE /api/cameras

Delete existing list of cameras

### 9.1. response (*json object*):

- `result: 0.`

### 9.2. Example:

#### 9.2.1. request:

**DELETE /api/cameras**

#### 9.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": 0}
```

## 10.GET /api/cameras

Get list of camera settings

### 10.1. response (*json object*):

- *result*: Refer to `#/definitions/cameras`.

### 10.2. Example:

#### 10.2.1. request:

```
GET /api/cameras
```

#### 10.2.2. response:

```
200 OK
```

```
Content-Type: application/json
```

```
{"result": [{"tag": "C1", "uri": "rtsp://192.168.86.10", "label": "C1", "devinfo": "rtsp:", "idx": 0, "lab_position": 0, "password": "**password**", "restart": 0, "username": "user", "zoom": 0, "props": {"audio": 0, "latency": 0, "protocols": 0, "timeout": 0}, "profiles": []}, {"tag": "C2", "uri": "rtsp://192.168.86.11", "label": "C2", "devinfo": "rtsp:", "idx": 1, "lab_position": 0, "password": "**password**", "restart": 0, "username": "user", "zoom": 0, "props": {"audio": 0, "latency": 0, "protocols": 0, "timeout": 0}, "profiles": []}, {"tag": "C3", "uri": "rtsp://192.168.86.13:554/media/stream.sdp?profile=Profile100", "label": "C3", "devinfo": "rtsp:", "idx": 2, "lab_position": 0, "password": "**password**", "restart": 0, "username": "admin", "zoom": 0, "props": {"audio": 0, "latency": 0, "protocols": 0, "timeout": 0}, "profiles": []}]}
```

## 11.PUT /api/cameras

Set new list of cameras to be displayed

### 11.1. request (json object):

- **cameras:** Refer to `#/definitions/cameras`.

### 11.2. response (json object):

- **result:** `OK`.

### 11.3. Example:

#### 11.3.1. request:

**PUT /api/cameras**

```
{"cameras": [{"tag": "C1", "label": "C1", "uri": "rtsp://192.168.86.10", "user name": "user", "password": "**password**", "props": {"audio": 0}}, {"label": "C2", "tag": "C2", "uri": "rtsp://192.168.86.11", "username": "user", "password": "**password**"}]}
```

#### 11.3.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": "OK"}
```

## 12.PUT /api/cameras/restart

restart playing cameras in a view

### 12.1. response (*json object*):

- *result*: OK.

### 12.2. Example:

#### 12.2.1. request:

**PUT /api/cameras/restart**

#### 12.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": "OK"}
```

## 13.GET /api/display

Get current display resolution

### 13.1. response (*json object*):

- *result* (*string*)

### 13.2. Example:

#### 13.2.1. request:

**GET /api/display**

#### 13.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": "1920x1080@60"}
```

## 14.GET /api/display/modes

Get current resolutions supported by display

### 14.1. response (*json object*):

- *result* (array)
  - Items (*string*)

### 14.2. Example:

#### 14.2.1. request:

**GET /api/display/modes**

#### 14.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{
  "result": [
    "640x480@75", "640x480@73", "640x480@67", "640x480@60", "720x400@70",
    "720x576@50", "720x480@60", "800x600@60", "800x600@75", "800x600@72", "800x600@56",
    "832x624@75", "1024x768@60", "1024x768@70", "1024x768@75", "1152x864@75", "1280x1024@60",
    "1280x960@60", "1280x800@60", "1280x720@60", "1280x1024@75", "1280x720@50",
    "1440x900@60", "1440x900@75", "1600x1200@60", "1680x1050@60", "1920x1080@50",
    "1920x1080@60"
  ]
}
```

## 15.POST /api/login

login to session

### 15.1. request (*json object*):

- *user* (string)
- *password* (string)

### 15.2. response (*json object*):

- *result*: OK.

### 15.3. Example:

#### 15.3.1. request:

**POST /api/login**

```
{"user": "admin", "password": "**password**"}
```

#### 15.3.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": "OK"}
```

## 16.POST /api/logout

Logout of a session

### 16.1. response (*json object*):

- *result*: OK.

### 16.2. Example:

#### 16.2.1. request:

POST /api/logout

#### 16.2.2. response:

200 OK

Content-Type: application/json

```
{"result": "OK"}
```

## 17.GET /api/network/<iface>

Get network configuration for *iface*

### 17.1. response (*json object*):

- *result*: Refer to `#/definitions/NetworkInterfaceT`.

### 17.2. Example:

#### 17.2.1. request:

```
GET /api/network/eth0
```

#### 17.2.2. response:

```
200 OK
```

```
Content-Type: application/json
```

```
{"result": {"Type": "Wired", "Available": true, "Enabled": true, "AgentParams": [], "Connect": "", "DNS": "192.168.86.1", "IPv4": {"Address": "192.168.86.222", "Gateway": "192.168.86.1", "Method": "static", "Netmask": "255.255.0.0"}, "NTP": "192.168.0.1", "State": "ready", "Address": "4A:61:4C:80:00:00"}}
```

## 18.PUT /api/network/<iface>

Set network configuration for *iface*

### 18.1. request (json object):

- *config*: Refer to `#/definitions/NetworkInterfaceT`.

### 18.2. response (json object):

- *result*: `OK`.

### 18.3. Example:

#### 18.3.1. request:

```
PUT /api/network/wlan0
{"config": {"Enabled": true, "Connect": "", "IPv4": {"Method": "dhcp"}, "Agent
Params": [{"ssid": "pumpkane-g", "passphrase": "***password***"}]}
```

#### 18.3.2. response:

```
200 OK
Content-Type: application/json
{"result": "OK"}
```

## 19.PUT /api/network/restart/<iface>

Restart network interface *iface*

### 19.1. response (*json object*):

- *result*: OK.

### 19.2. Example:

#### 19.2.1. request:

```
PUT /api/network/restart/wlan0
```

#### 19.2.2. response:

```
200 OK
```

```
Content-Type: application/json
```

```
{"result": "OK"}
```

## 20.PUT /api/network/wifi/connect/<iface>

Attempt connection for interface *iface*

### 20.1. request (json object):

- *connect* (['string', 'null'])

### 20.2. response (json object):

- *result*: OK.

### 20.3. Example:

#### 20.3.1. request:

```
PUT /api/network/wifi/connect/wlan0
{"connect": "pumpplane-g"}
```

#### 20.3.2. response:

```
200 OK
Content-Type: application/json
{"result": "OK"}
```

## 21.GET /api/network/wifi/scan

Return available wifi networks

### 21.1. response (*json object*):

- *result*
  - Any of
    - *object*
      - *wLan0* (*array*)
        - *Items* (*object*)
          - *Name* (*string*)
          - *State*: Refer to [#/definitions/NetStateT](#).
          - *Strength* (*integer*)
          - *Security* (*array*)
            - *Items*: Refer to [#/definitions/NetSecurity](#).
  - *null*

### 21.2. Example:

#### 21.2.1. request:

**GET /api/network/wifi/scan**

#### 21.2.2. response:

**200 OK**  
**Content-Type: application/json**  
 {"result": {}}

## 22.GET /api/onvif/cameras

Search network for available onvif cameras

### 22.1. response (*json object*):

- *result* (*array*)
  - *Items* (*string*)

### 22.2. Example:

#### 22.2.1. request:

**GET /api/onvif/cameras**

#### 22.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": ["192.168.86.10", "192.168.86.11", "192.168.86.13:80", "192.168.86.222:8000"]}
```

## 23.GET /api/onvif/profiles

Query onvif cameras for available profiles

### 23.1. params:

- *uri* (string)
- *username* (string)
- *password* (string)

### 23.2. response (json object):

- *result* (array)
  - Items: Refer to `#/definitions/ProfileT`.

### 23.3. Example:

#### 23.3.1. request:

```
GET /api/onvif/profiles?uri=192.168.86.10&username=user&password=a1b2c3d4
```

#### 23.3.2. response:

200 OK

Content-Type: application/json

```
{
  "result": [
    {
      "name": "Multicast:mainStream",
      "codec": "H264",
      "res_width": "1920",
      "res_height": "1080",
      "fps": "15",
      "uri": "rtsp://192.168.86.10:554/Streaming/Channels/101?transportmode=mc&profile=Profile_1"
    },
    {
      "name": "Multicast:subStream",
      "codec": "H264",
      "res_width": "640",
      "res_height": "480",
      "fps": "20",
      "uri": "rtsp://192.168.86.10:554/Streaming/Channels/102?transportmode=mc&profile=Profile_2"
    },
    {
      "name": "Unicast:mainStream",
      "codec": "H264",
      "res_width": "1920",
      "res_height": "1080",
      "fps": "15",
      "uri": "rtsp://192.168.86.10:554/Streaming/Channels/101?transportmode=unic&profile=Profile_1"
    },
    {
      "name": "Unicast:subStream",
      "codec": "H264",
      "res_width": "640",
      "res_height": "480",
      "fps": "20",
      "uri": "rtsp://192.168.86.10:554/Streaming/Channels/102?transportmode=unic&profile=Profile_2"
    }
  ]
}
```

## 24.GET /api/system/config

Get current system configuration

### 24.1. response (json object):

- *result* (object)
  - *cameras*: Refer to `#/definitions/cameras`.
  - *views*: Refer to `#/definitions/ViewsT`.
  - *network*: Refer to `#/definitions/network`.
  - *display*: Refer to `#/definitions/display`.
  - *default\_view*: Refer to `#/definitions/DefaultViewT`.
  - *grid\_font*: Refer to `#/definitions/GridFontT`.
  - *https\_enable*: Refer to `#/definitions/https_enable`.
  - *user\_config*: Refer to `#/definitions/user_config`.
  - *rtsp\_svr*: Refer to `#/definitions/RtspSvrT`.
  - *web\_ui*: Refer to `#/definitions/WebUIT`.

### 24.2. Example:

#### 24.2.1. request:

**GET /api/system/config**

#### 24.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{
  "result": {
    "cameras": [
      {
        "tag": "C1",
        "uri": "rtsp://192.168.86.10",
        "label": "C1",
        "devinfo": "rtsp:",
        "idx": 0,
        "lab_position": 0,
        "password": "***password**",
        "restart": 0,
        "username": "user",
        "zoom": 0,
        "props": {
          "audio": 0,
          "latency": 0,
          "protocols": 0,
          "timeout": 0
        },
        "profiles": [
          {
            "enabled": true,
            "name": "video",
            "codec": "video/x-h264",
            "uri": "rtsp://user:a1b2c3d4@192.168.86.10",
            "fps": "15",
            "res_height": 1080,
            "res_width": 1920
          },
          {
            "tag": "C2",
            "uri": "onvif://192.168.86.11",
            "label": "C2",
            "devinfo": "onvif:HIKVISION,DS-2CD2042WD-I",
            "idx": 1,
            "lab_position": 0,
            "password": "***password**",
            "restart": 0,
            "username": "user",
            "zoom": 0,
            "props": {
              "audio": 0,
              "latency": 0,
              "protocols": 0,
              "timeout": 0
            },
            "profiles": [
              {
                "enabled": false,
                "name": "Multicast:mainStream",
                "codec": "H264",
                "uri": "rtsp://192.168.86.11:554/Streaming/Channels/101?transportmode=mcast&profile=Profile_1",
                "fps": "20",
                "res_height": "1296",
                "res_width": "2304"
              },
              {
                "enabled": false,
                "name": "Multicast:subStream",
                "codec": "H264",
                "uri": "rtsp://192.168.86.11:554/Streaming/Channels/102?transportmode=mcast&profile=Profile_2",
                "fps": "20",
                "res_height": "360",
                "res_width": "640"
              },
              {
                "enabled": true,
                "name": "Unicast:mainStream",
                "codec": "H264",
                "uri": "rtsp://192.168.86.11:554/Streaming/Channels/101?transportmode=unicast&profile=Profile_1",
                "fps": "20",
                "res_height": "1296",
                "res_width": "2304"
              },
              {
                "enabled": true,
                "name": "Unicast:subStream",
                "codec": "H264",
                "uri": "rtsp://192.168.86.11:554/Streaming/Channels/102?transportmode=unicast&profile=Profile_2",
                "fps": "20",
                "res_height": "360",
                "res_width": "640"
              }
            ]
          },
          {
            "tag": "C3",
            "uri": "onvif://192.168.86.13:80",
            "label": "",
            "devinfo": "onvif:PELCO,IMV529-1ERS",
            "idx": 2,
            "lab_position": 0,
            "password": "***password**",
            "restart": 0,
            "username": "admin",
            "zoom": 0,
            "props": {
              "audio": 0,
              "latency": 0,
              "protocols": 0,
              "timeout": 0
            },
            "profiles": [
              {
                "enabled": false,
                "name": "Multicast:Profile100",
                "codec": "H264",
                "uri": "rtsp://192.168.86.13:554/media/stream.sdp?profile=Profile100",
                "fps": "20",
                "res_height": "1920",
                "res_width": "2560"
              },
              {
                "enabled": false,
                "name": "Multicast:Profile101",
                "codec": "H264",
                "uri": "rtsp://192.168.86.13:554/media/stream.sdp?profile=Profile101",
                "fps": "20",
                "res_height": "1080",
                "res_width": "1920"
              },
              {
                "enabled": true,
                "name": "Unicast:Profile100",
                "codec": "H264",
                "uri": "rtsp://192.168.86.13:80/media/stream.sdp?profile=Profile100",
                "fps": "20",
                "res_height": "1080",
                "res_width": "1920"
              },
              {
                "enabled": true,
                "name": "Unicast:Profile101",
                "codec": "H264",
                "uri": "rtsp://192.168.86.13:80/media/stream.sdp?profile=Profile101",
                "fps": "20",
                "res_height": "1080",
                "res_width": "1920"
              }
            ]
          }
        ]
      }
    ]
  }
}
```



## 25. POST /api/system/config

Update system configuration from a file (requires restart to apply)

### 25.1. request (**Multipart-Encoded JSON File**):

- *config* (object)
  - *cameras*: Refer to `#/definitions/cameras`.
  - *views*: Refer to `#/definitions/ViewsT`.
  - *network*: Refer to `#/definitions/network`.
  - *display*: Refer to `#/definitions/display`.
  - *default\_view*: Refer to `#/definitions/DefaultViewT`.
  - *grid\_font*: Refer to `#/definitions/GridFontT`.
  - *https\_enable*: Refer to `#/definitions/https_enable`.
  - *user\_config*: Refer to `#/definitions/user_config`.
  - *rtsp\_svr*: Refer to `#/definitions/RtspSvrT`.
  - *web\_ui*: Refer to `#/definitions/WebUIT`.

### 25.2. response (**json object**):

- *result*: OK.

### 25.3. Example:

#### 25.3.1. request:

**POST /api/system/config**

## 26.PUT /api/system/config

Update current system configuration (requires restart to apply)

### 26.1. request (json object):

- *config* (object)
  - *cameras*: Refer to `#/definitions/cameras`.
  - *views*: Refer to `#/definitions/ViewsT`.
  - *network*: Refer to `#/definitions/network`.
  - *display*: Refer to `#/definitions/display`.
  - *default\_view*: Refer to `#/definitions/DefaultViewT`.
  - *grid\_font*: Refer to `#/definitions/GridFontT`.
  - *https\_enable*: Refer to `#/definitions/https_enable`.
  - *user\_config*: Refer to `#/definitions/user_config`.
  - *rtsp\_svr*: Refer to `#/definitions/RtspSvrT`.
  - *web\_ui*: Refer to `#/definitions/WebUIT`.

### 26.2. response (json object):

- *result*: OK.

### 26.3. Example:

#### 26.3.1. request:

```
PUT /api/system/config
{"config": {"user_config": {"location": "Bristol"}}
```

#### 26.3.2. response:

```
200 OK
Content-Type: application/json
{"result": "OK"}
```

## 27.DELETE /api/system/config/<v>

Reset config value to factory default value (requires restart to apply)

### 27.1. response (*json object*):

- *result*: OK.

### 27.2. Example:

#### 27.2.1. request:

```
DELETE /api/system/config/views
```

#### 27.2.2. response:

```
200 OK  
Content-Type: application/json  
{"result": "OK"}
```

## 28.GET /api/system/config/<v>

Get specific config entry

### 28.1. response (json object):

- *result*
  - Any of
    - : Refer to *#/definitions/cameras*.
    - : Refer to *#/definitions/ViewsT*.
    - : Refer to *#/definitions/network*.
    - : Refer to *#/definitions/display*.
    - : Refer to *#/definitions/DefaultViewT*.
    - : Refer to *#/definitions/GridFontT*.
    - : Refer to *#/definitions/https\_enable*.
    - : Refer to *#/definitions/user\_config*.
    - : Refer to *#/definitions/RtspSvrT*.
    - : Refer to *#/definitions/WebUIT*.

### 28.2. Example:

#### 28.2.1. request:

**GET /api/system/config/views**

#### 28.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{
  "result": {
    "grid_1x1": [
      {
        "cameras": [
          {
            "tag": "C*",
            "period": 0,
            "profile": null
          }
        ],
        "period": 60
      }
    ],
    "grid_2x2": [
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      }
    ],
    "grid_3x3": [
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      }
    ],
    "grid_4x4": [
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      },
      {
        "cameras": [],
        "period": 0
      }
    ]
  }
}
```

## 29.PUT /api/system/config/views

Configure cameras to display in grid

### 29.1. Example:

#### 29.1.1. request:

**PUT /api/system/config/views**

```
{"config": {"grid_1x1": [{"period": 60, "cameras": [{"tag": "C*"}]}}}
```

#### 29.1.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": "OK"}
```

## 30.PUT /api/system/factoryreset

Reset device to factory defaults

### 30.1. response (*json object*):

- *result*: OK.

### 30.2. Example:

#### 30.2.1. request:

PUT /api/system/factoryreset

#### 30.2.2. response:

200 OK

Content-Type: application/json

```
{"result": "OK"}
```

## 31.GET /api/system/info

Get sytem info

### 31.1. response (*json object*):

- *result* (*object*): Can contain additional properties.
  - *cpu* (*object*)
    - *temp* (*number*)
  - *date* (*string*)
  - *version* (*object*)
    - *SW\_VERSION* (*string*)
    - *IMAGE\_BUILD\_DATE* (*string*)
    - *HW\_REVISION* (*string*)

### 31.2. Example:

#### 31.2.1. request:

**GET /api/system/info**

#### 31.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{
  "result": {
    "version": {
      "SW_VERSION": "spotbox 3.3.1",
      "IMAGE_BUILD_DATE": "Mon 27 Mar 17:16:11 BST 2023",
      "HW_REVISION": "spotbox rk3566-pc"
    },
    "sn": "1",
    "sys": {
      "cpu_temp": 37.777,
      "gpu_temp": 39.444,
      "boot_time": "27-03-2023 16:48:35",
      "cpu_freq": 1800.0,
      "mem": 26.3,
      "cpu_times": 17.4
    },
    "date": "Tue Mar 28 13:53:07 2023"
  }
}
```

## 32.PUT /api/system/reboot

reboot device

### 32.1. response (*json object*):

- *result*: OK.

### 32.2. Example:

#### 32.2.1. request:

PUT /api/system/reboot

#### 32.2.2. response:

200 OK

Content-Type: application/json

```
{"result": "OK"}
```

## 33.PUT /api/system/restart

restart spotbox application

### 33.1. response (*json object*):

- *result*: OK.

### 33.2. Example:

#### 33.2.1. request:

PUT /api/system/restart

#### 33.2.2. response:

200 OK

Content-Type: application/json

```
{"result": "OK"}
```

## 34.DELETE /api/system/ssl

Delete current ssl certificate and key

### 34.1. response (*json object*):

- *result*: OK.

### 34.2. Example:

#### 34.2.1. request:

**DELETE /api/system/ssl**

#### 34.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": "OK"}
```

## 35. POST /api/system/ssl/cert

Load new ssl certificate from file

### 35.1. request (*Multipart-Encoded File*):

### 35.2. response (*json object*):

- *result: OK.*

### 35.3. Example:

#### 35.3.1. request:

POST /api/system/ssl/cert

#### 35.3.2. response:

200 OK

Content-Type: application/json

```
{"result": "OK"}
```

## 36. POST /api/system/ssl/key

Load new ssl key from file

### 36.1. request (*Multipart-Encoded File*):

### 36.2. response (*json object*):

- *result: OK.*

### 36.3. Example:

#### 36.3.1. request:

POST /api/system/ssl/key

#### 36.3.2. response:

200 OK

Content-Type: application/json

```
{"result": "OK"}
```

## 37.PUT /api/system/user

Update login password for admin

### 37.1. request (json object):

- *user* (object)
  - *username* (string)
  - *password* (string)

### 37.2. response (json object):

- *result*: OK.

### 37.3. Example:

#### 37.3.1. request:

**PUT /api/system/user**

```
{"user": {"username": "admin", "password": "**password**"}}
```

#### 37.3.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": "OK"}
```

## 38.GET /api/view

List of possible view modes

### 38.1. response (*json object*):

- *result* (array)
  - Items: Refer to *#/definitions/GridModesT*.

### 38.2. Example:

#### 38.2.1. request:

**GET /api/view**

#### 38.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": ["fullscreen", "grid_1x1", "grid_2x2", "grid_3x3", "grid_4x4"]}
```

## 39.GET /api/view/<v:int>

Get list of cameras playing on view v

### 39.1. response (json object):

- *result* (array)
  - Items
    - Any of
      - object
        - *idx* (integer)
        - *tag* (string)
        - *url* (['string', 'null'])
        - *profile*
          - Any of
            - integer
            - null
        - *period* (integer)
      - null

### 39.2. Example:

#### 39.2.1. request:

**GET /api/view/0**

#### 39.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{
  "result": [
    {
      "idx": 0,
      "tag": "C1",
      "url": "rtsp://user:a1b2c3d4@192.168.86.10",
      "profile": 0,
      "period": 60
    },
    {
      "idx": 1,
      "tag": "C2",
      "url": "rtsp://192.168.86.11",
      "profile": 0,
      "period": 60
    },
    {
      "idx": 2,
      "tag": "C3",
      "url": "rtsp://192.168.86.13:554/media/stream.sdp?profile=Profile100"
    }
  ]
}
```

## 40.PUT /api/view/<v:int>/play

Play list of cameras on specific view *v* .

### 40.1. request (*json object*):

- *cameras* (*array*)
  - *Items* (*integer*): Minimum: 0.
- *period* (*integer*): Minimum: 0.

### 40.2. response (*json object*):

- *result*: OK.

### 40.3. Example:

#### 40.3.1. request:

```
PUT /api/view/0/play  
{ "cameras": [0, 1], "period": 10 }
```

#### 40.3.2. response:

```
200 OK  
Content-Type: application/json  
{ "result": "OK" }
```

## 41.GET /api/view/default

Get default view mode

### 41.1. response (*json object*):

- *result*: Refer to `#/definitions/DefaultViewT`.

### 41.2. Example:

#### 41.2.1. request:

`GET /api/view/default`

#### 41.2.2. response:

`200 OK`

`Content-Type: application/json`

```
{"result": {"camera": 0, "name": "", "mode": "grid_2x2"}}
```

## 42.PUT /api/view/default

Set default view mode

### 42.1. request (json object):

Refer to `#/definitions/DefaultViewT`. ## response (json object): - **result**: OK.

### 42.2. Example:

#### 42.2.1. request:

```
PUT /api/view/default
{"mode": "grid_1x1"}
```

#### 42.2.2. response:

```
200 OK
Content-Type: application/json
{"result": "OK"}
```

## 43.GET /api/view/mode

Get current view mode

### 43.1. response (*json object*):

- *result*: Refer to `#/definitions/GridModesT`.

### 43.2. Example:

#### 43.2.1. request:

```
GET /api/view/mode
```

#### 43.2.2. response:

```
200 OK
```

```
Content-Type: application/json
```

```
{"result": "grid_1x1"}
```

## 44.PUT /api/view/mode

Change current view mode

### 44.1. request (json object):

- *mode*: Refer to `#/definitions/GridModesT`.
- *camera* (integer)

### 44.2. response (json object):

- *result*: `OK`.

### 44.3. Example:

#### 44.3.1. request:

```
PUT /api/view/mode  
{  
  "mode": "grid_1x1"  
}
```

#### 44.3.2. response:

```
200 OK  
Content-Type: application/json  
{  
  "result": "OK"  
}
```

## 45.GET /api/view/status

Return current status of cameras displayed

### 45.1. response (*json object*):

- *result* (*array*)
  - *Items*
    - **Any of**
      - *object*
        - *status* (*integer*)
        - *status\_exit* (*integer*)
        - *time\_start* (*number*)
        - *time\_duration* (*number*)
        - *time\_update* (*number*)
        - *uri* (*['string', 'null']*)
      - *null*

### 45.2. Example:

#### 45.2.1. request:

**GET /api/view/status**

#### 45.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": [{"status": 0, "status_exit": 0, "time_start": 1680011602, "time_duration": 0, "time_update": 0, "uri": "rtsp://user:a1b2c3d4@192.168.86.10"}]}
```

## 46.GET /api/views

Get list of cameras in each view

### 46.1. response (*json object*):

- *result* (*array*)
  - *Items* (*array*)
    - *Items* (*object*)
      - *idx* (*integer*)
      - *tag* (*string*)
      - *url* (*['string', 'null']*)
      - *profile* (*['integer', 'null']*)
      - *period* (*integer*)
      - *time\_remaining* (*['integer', 'null']*)
      - *status* (*integer*)

### 46.2. Example:

#### 46.2.1. request:

**GET /api/views**

#### 46.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{
  "result": [
    [
      {
        "idx": 0,
        "tag": "C1",
        "url": "rtsp://user:a1b2c3d4@192.168.86.10",
        "profile": 0,
        "period": 60
      },
      {
        "idx": 1,
        "tag": "C2",
        "url": "rtsp://192.168.86.11",
        "profile": 0,
        "period": 60
      },
      {
        "idx": 2,
        "tag": "C3",
        "url": "rtsp://192.168.86.13:554/media/stream.sdp?profile=Profile100"
      }
    ]
  ]
}
```

## 47. Definitions

### 47.1. GridModesT

- : One of: ['fullscreen', 'grid\_1x1', 'grid\_2x2', 'grid\_3x3', 'grid\_4x4'].

### 47.2. ProfileT

- *object*
  - *enabled* (boolean)
  - *name* (string)
  - *codec* (string)
  - *uri* (string)
  - *fps*
    - Any of
      - integer
      - string
  - *res\_height*
    - Any of
      - integer
      - string
  - *res\_width*
    - Any of
      - integer
      - string

### 47.3. ViewListT

- *object*
  - *cameras* (array)
    - **Items:** Refer to [#/definitions/ViewCamT](#).
  - *period* (integer)

### 47.4. ViewCamT

- *object*
  - *tag* (string)
  - *period* (integer)
  - *profile*
    - One of
      - integer
      - null

### 47.5. ViewsT

- *object*
  - *grid\_1x1* (array)
    - **Items:** Refer to [#/definitions/ViewListT](#).
  - *grid\_2x2* (array)
    - **Items:** Refer to [#/definitions/ViewListT](#).
  - *grid\_3x3* (array)

- **Items:** Refer to [#/definitions/ViewListT](#).
- *grid\_4x4* (array)
  - **Items:** Refer to [#/definitions/ViewListT](#).

#### 47.6. CameraGridsT

- *object*
  - *grid\_1x1*
    - **One of**
      - *integer*
      - *null*
  - *grid\_2x2*
    - **One of**
      - *integer*
      - *null*
  - *grid\_3x3*
    - **One of**
      - *integer*
      - *null*
  - *grid\_4x4*
    - **One of**
      - *integer*
      - *null*

#### 47.7. CameraPropsT

- *object*
  - *audio* (*integer*)
  - *latency* (*integer*)
  - *protocols* (*integer*)
  - *timeout* (*integer*)

#### 47.8. CameraT

- *object*: Properties for Camera .
  - *tag* (*string*)
  - *uri* (*string*)
  - *label* (*string*)
  - *devinfo* (*string*)
  - *idx* (*integer*)
  - *lab\_position* (*integer*): One of: [*0*, *1*, *2*, *3*, *99*].
  - *password* (*string*)
  - *restart* (*integer*)
  - *username* (*string*)
  - *zoom* (*integer*)
  - *props*: Refer to [#/definitions/CameraPropsT](#).
  - *profiles* (array)
    - **Items:** Refer to [#/definitions/ProfileT](#).

- **grids**: Refer to `#/definitions/CameraGridsT`. *Deprecated*
- **period** (*integer*): Minimum: 0.

#### 47.9. NetStateT

- : One of: ['association', 'configuration', 'disconnect', 'failure', 'idle', 'offline', 'online', 'ready'].

#### 47.10. NetSecurity

- : One of: ['none', 'wep', 'psk', 'ieee8021x', 'wps'].

#### 47.11. NetAgentParam

- *object*
  - **identity** (*string*)
  - **name** (*string*)
  - **passphrase** (*string*)
  - **ssid** (*string*)
  - **wpspin** (*string*)

#### 47.12. NetIpv4

- *object*
  - **Address** (*string*)
  - **Gateway** (*string*)
  - **Method** (*string*): One of: ['dhcp', 'manual', 'static'].
  - **Netmask** (*string*)

#### 47.13. NetworkInterfaceT

- *object*
  - **Type** (*string*): One of: ['Loopback', 'WiFi', 'Wired'].
  - **AvaiLable** (*boolean*)
  - **EnabLed** (*boolean*)
  - **AgentParams** (*array*)
    - **Items**: Refer to `#/definitions/NetAgentParam`.
  - **Connect** (*string*)
  - **DNS** (*string*)
  - **IPv4**: Refer to `#/definitions/NetIpv4`.
  - **NTP** (*string*)
  - **State** (*string*): One of: ['association', 'configuration', 'disconnect', 'failure', 'idle', 'offline', 'online', 'ready'].
  - **Address** (*string*)
  - **Method** (*string*): *Deprecated*
  - **Netmask** (*string*): *Deprecated*
  - **Gateway** (*string*): *Deprecated*

#### 47.14. cameras

- *array*
  - **Items**: Refer to `#/definitions/CameraT`.

### 47.15. network

- *object*
  - *eth0*: Refer to `#/definitions/NetworkInterfaceT`.
  - *wlan0*: Refer to `#/definitions/NetworkInterfaceT`.
  - *lo0*: Refer to `#/definitions/NetworkInterfaceT`.

### 47.16. GridFontE

- : One of: ['l', 'm', 's'].

### 47.17. GridFontT

- *object*
  - *fullscreen* (*string*): One of: ['l', 'm', 's'].
  - *grid\_1x1* (*string*): One of: ['l', 'm', 's'].
  - *grid\_2x2* (*string*): One of: ['l', 'm', 's'].
  - *grid\_3x3* (*string*): One of: ['l', 'm', 's'].
  - *grid\_4x4* (*string*): One of: ['l', 'm', 's'].

### 47.18. DefaultViewT

- *object*
  - *camera* (*integer*)
  - *name* (*string*)
  - *mode* (*string*)

### 47.19. WebUIT

- *object*
  - *host* (*string*)
  - *http\_port* (*integer*)
  - *https\_enable* (*boolean*)
  - *https\_port* (*integer*)
  - *iface* (*string*): One of: ['eth0', 'wlan0'].

### 47.20. RtspStreamT

- *object*
  - *enc\_props*: Refer to `#/definitions/RtspEncProps`.
  - *codec* (*string*): One of: ['h264', 'h265'].
  - *framerate* (*integer*)
  - *name* (*string*)
  - *stream\_address*: Refer to `#/definitions/RtspStreamAddress`.
  - *transport* (*string*): One of: ['ANY', 'MCAST', 'TCP', 'UDP', 'UNICAST'].

### 47.21. RtspEncProps

- *object*
  - *bps* (*integer*)
  - *gop* (*integer*)
  - *height* (*integer*)
  - *width* (*integer*)

- *profile* (string): One of: ['baseline', 'high', 'main'].
- *rc\_mode* (string): One of: ['cbr', 'vbr'].

#### **47.22. RtspStreamAddress**

- *object*
  - *address* (string)
  - *port* (integer)
  - *tTL* (integer)

#### **47.23. RtspSvrT**

- *object*
  - *available* (boolean)
  - *enabled* (boolean)
  - *onvif* (boolean)
  - *password* (string)
  - *port* (integer)
  - *streams* (array)
    - **Items:** Refer to [#/definitions/RtspStreamT](#).
  - *username* (string)

#### **47.24. user\_config**

- *object*

#### **47.25. display**

- *string*

#### **47.26. https\_enable**

- *boolean*