

# ANT-37100 / Spotbox2 API guide for firmware 2.0.x

## Contents

ANT-37100 / Spotbox2 API guide for firmware 2.0.x .....	1
Document info.....	1
Extended Table of Contents.....	2
Introduction .....	5
1. Changes .....	6
2. POST /api/login .....	6
3. POST /api/logout .....	7
4. GET /api/view .....	8
5. GET /api/views.....	9
6. GET /api/view/mode .....	10
7. PUT /api/view/mode .....	11
8. GET /api/view/default .....	12
9. PUT /api/view/default.....	13
10. GET /api/view/status .....	14
11. GET /api/view/<v:int> .....	15
12. PUT /api/view/<v:int>/play .....	16
13. GET /api/cameras .....	17
14. PUT /api/cameras .....	19
15. DELETE /api/cameras.....	21
16. PUT /api/cameras/restart .....	22
17. PUT /api/camera .....	23
18. GET /api/camera/<n:int> .....	25
19. PUT /api/camera/<n:int> .....	27
20. DELETE /api/camera/<n:int> .....	29
21. GET /api/display .....	30
22. GET /api/display/modes .....	31
23. GET /api/onvif/cameras.....	32
24. GET /api/onvif/profiles .....	33
25. GET /api/network/<iface> .....	34
26. GET /api/system/info .....	35
27. PUT /api/system/reboot.....	36
28. PUT /api/system/restart .....	37
29. PUT /api/system/factoryreset.....	38
30. PUT /api/system/user.....	39
31. GET /api/system/config .....	40
32. PUT /api/system/config .....	42
33. POST /api/system/config .....	43
34. DELETE /api/system/ssl.....	44
35. POST /api/system/ssl/key.....	45
36. POST /api/system/ssl/cert .....	46

## Document info

For further help and advice please contact Antrica on:

Email: [support@antrica.com](mailto:support@antrica.com)

Phone: +44 1628 626098 ext 3

Version	date	author	Comments
1.0	9-Sep-22	David M	Updated for firmware 2.0.x

# Extended Table of Contents

ANT-37100 / Spotbox2 API guide for firmware 2.0.x .....	1
Document info.....	1
Extended Table of Contents.....	2
Introduction .....	5
1. Changes.....	6
1.1. V.1.1.1.....	6
1.2. V.2.0.x.....	6
2. POST /api/login .....	6
2.1. request ( <i>json object</i> ): .....	6
2.2. response ( <i>json object</i> ): .....	6
2.3. Example:.....	6
3. POST /api/logout .....	7
3.1. response ( <i>json object</i> ): .....	7
3.2. Example:.....	7
4. GET /api/view .....	8
4.1. response ( <i>json object</i> ): .....	8
4.2. Example:.....	8
5. GET /api/views.....	9
5.1. response ( <i>json object</i> ): .....	9
5.2. Example:.....	9
6. GET /api/view/mode .....	10
6.1. response ( <i>json object</i> ): .....	10
6.2. Example:.....	10
7. PUT /api/view/mode .....	11
7.1. request ( <i>json object</i> ): .....	11
7.2. response ( <i>json object</i> ): .....	11
7.3. Example:.....	11
8. GET /api/view/default .....	12
8.1. response ( <i>json object</i> ): .....	12
8.2. Example:.....	12
9. PUT /api/view/default.....	13
9.1. request ( <i>json object</i> ): .....	13
9.2. response ( <i>json object</i> ): .....	13
9.3. Example:.....	13
10. GET /api/view/status .....	14
10.1. response ( <i>json object</i> ): .....	14
10.2. Example:.....	14
11. GET /api/view/<v:int> .....	15
11.1. response ( <i>json object</i> ): .....	15
11.2. Example:.....	15
12. PUT /api/view/<v:int>/play .....	16
12.1. request ( <i>json object</i> ): .....	16
12.2. response ( <i>json object</i> ): .....	16
12.3. Example:.....	16
13. GET /api/cameras .....	17
13.1. response ( <i>json object</i> ): .....	17
13.2. Example:.....	17
14. PUT /api/cameras .....	19
14.1. request ( <i>json object</i> ): .....	19

14.2.	response ( <i>json object</i> ): .....	19
14.3.	Example: .....	19
15.	DELETE /api/cameras .....	21
15.1.	response ( <i>json object</i> ): .....	21
15.2.	Example: .....	21
16.	PUT /api/cameras/restart .....	22
16.1.	response ( <i>json object</i> ): .....	22
16.2.	Example: .....	22
17.	PUT /api/camera .....	23
17.1.	request ( <i>json object</i> ): .....	23
17.2.	response ( <i>json object</i> ): .....	23
17.3.	Example: .....	23
18.	GET /api/camera/<n:int> .....	25
18.1.	response ( <i>json object</i> ): .....	25
18.2.	Example: .....	25
19.	PUT /api/camera/<n:int> .....	27
19.1.	request ( <i>json object</i> ): .....	27
19.2.	response ( <i>json object</i> ): .....	27
19.3.	Example: .....	27
20.	DELETE /api/camera/<n:int> .....	29
20.1.	response ( <i>json object</i> ): .....	29
20.2.	Example: .....	29
21.	GET /api/display .....	30
21.1.	response ( <i>json object</i> ): .....	30
21.2.	Example: .....	30
22.	GET /api/display/modes .....	31
22.1.	response ( <i>json object</i> ): .....	31
22.2.	Example: .....	31
23.	GET /api/onvif/cameras .....	32
23.1.	response ( <i>json object</i> ): .....	32
23.2.	Example: .....	32
24.	GET /api/onvif/profiles .....	33
24.1.	params: .....	33
24.2.	response ( <i>json object</i> ): .....	33
24.3.	Example: .....	33
25.	GET /api/network/<iface> .....	34
25.1.	response ( <i>json object</i> ): .....	34
25.2.	Example: .....	34
26.	GET /api/system/info .....	35
26.1.	response ( <i>json object</i> ): .....	35
26.2.	Example: .....	35
27.	PUT /api/system/reboot .....	36
27.1.	response ( <i>json object</i> ): .....	36
27.2.	Example: .....	36
28.	PUT /api/system/restart .....	37
28.1.	response ( <i>json object</i> ): .....	37
28.2.	Example: .....	37
29.	PUT /api/system/factoryreset .....	38
29.1.	response ( <i>json object</i> ): .....	38
29.2.	Example: .....	38
30.	PUT /api/system/user .....	39

---

- 30.1. request (*json object*): ..... 39
- 30.2. response (*json object*): ..... 39
- 30.3. Example:..... 39
- 31. GET /api/system/config ..... 40
  - 31.1. response (*json object*): ..... 40
  - 31.2. Example:..... 41
- 32. PUT /api/system/config ..... 42
  - 32.1. response (*json object*): ..... 42
  - 32.2. Example:..... 42
- 33. POST /api/system/config ..... 43
  - 33.1. request (Multipart-Encoded File): ..... 43
  - 33.2. response (*json object*): ..... 43
  - 33.3. Example:..... 43
- 34. DELETE /api/system/ssl..... 44
  - 34.1. response (*json object*): ..... 44
  - 34.2. Example:..... 44
- 35. POST /api/system/ssl/key..... 45
  - 35.1. request (Multipart-Encoded File): ..... 45
  - 35.2. response (*json object*): ..... 45
  - 35.3. Example:..... 45
- 36. POST /api/system/ssl/cert ..... 46
  - 36.1. request (Multipart-Encoded File): ..... 46
  - 36.2. response (*json object*): ..... 46
  - 36.3. Example:..... 46

## 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.

## 1. Changes

### 1.1. 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
```

### 1.2. V.2.0.x

The following routes have been added:

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

## 2. POST /api/login

login to session

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

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

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

- *result*: OK.

### 2.3. Example:

#### 2.3.1. request:

```
POST /api/login
{"user": "admin", "password": "1234"}
```

#### 2.3.2. response:

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

### 3. POST /api/logout

Logout of a session

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

- *result*: OK.

#### 3.2. Example:

##### 3.2.1. request:

POST /api/logout

##### 3.2.2. response:

200 OK

Content-Type: application/json

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

## 4. GET /api/view

List of possible view modes

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

- *result* (array)
  - Items: One of: ['fullscreen', 'grid\_1x1', 'grid\_2x2', 'grid\_3x3', 'grid\_4x4'].

### 4.2. Example:

#### 4.2.1. request:

```
GET /api/view
```

#### 4.2.2. response:

```
200 OK
```

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

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



## 5. GET /api/views

Get list of cameras in each view

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

- *result* (array)
  - *Items* (array)
    - *Items* (object)
      - *cnum* (integer)
      - *url* (string)
      - *profile* (integer)

### 5.2. Example:

#### 5.2.1. request:

**GET /api/views**

#### 5.2.2. response:

**200 OK**

**Content-Type: application/json**

```

{"result": [[{"profile": null, "url": "rtsp://192.168.1.101", "cnum": 0}], [{"profile": null, "url": "rtsp://192.168.1.102", "cnum": 1}], [], []]}
  
```

## 6. GET /api/view/mode

Get current view mode

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

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

### 6.2. Example:

#### 6.2.1. request:

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

#### 6.2.2. response:

```
200 OK
```

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

```
{"result": "grid_2x2"}
```

## 7. PUT /api/view/mode

Change current view mode

### 7.1. request (json object):

- *mode*: One of: ['fullscreen', 'grid\_1x1', 'grid\_2x2', 'grid\_3x3', 'grid\_4x4'].
- *camera* (integer)

### 7.2. response (json object):

- *result*: OK.

### 7.3. Example:

#### 7.3.1. request:

```
PUT /api/view/mode
{"mode": "grid_2x2"}
```

#### 7.3.2. response:

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

## 8. GET /api/view/default

Get default view mode

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

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

### 8.2. Example:

#### 8.2.1. request:

```
GET /api/view/default
```

#### 8.2.2. response:

```
200 OK
```

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

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

## 9. PUT /api/view/default

Set default view mode

### 9.1. request (json object):

- *mode*: One of: ['fullscreen', 'grid\_1x1', 'grid\_2x2', 'grid\_3x3', 'grid\_4x4'].
- *camera* (integer)

### 9.2. response (json object):

- *result*: OK.

### 9.3. Example:

#### 9.3.1. request:

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

#### 9.3.2. response:

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

## 10.GET /api/view/status

Return current status of cameras displayed

### 10.1. response (json object):

- *result* (array)
  - *Items* (object)
    - *status* (integer)
    - *status\_exit* (integer)
    - *time\_start* (number)
    - *time\_duration* (number)
    - *time\_update* (number)
    - *uri* (string)

### 10.2. Example:

#### 10.2.1. request:

**GET /api/view/status**

#### 10.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{
  "result": [
    {
      "status": "PLAYING",
      "time_start": 1661797329,
      "status_exit": "NONE",
      "time_duration": 0,
      "uri": "rtsp://192.168.1.101",
      "time_update": 1661797329
    },
    {
      "status": "PLAYING",
      "time_start": 1661797329,
      "status_exit": "NONE",
      "time_duration": 0,
      "uri": "rtsp://192.168.1.102",
      "time_update": 1661797329
    },
    {
      "status": "STOPPED",
      "time_start": 1661786211,
      "status_exit": "OK",
      "time_duration": 3631,
      "uri": "rtsp://192.168.1.103",
      "time_update": 1661797329
    },
    null
  ]
}
```

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

Get list of cameras playing on view v

### 11.1. response (json object):

- *result* (array)
  - *Items* (object)
    - *cnum* (integer)
    - *url* (string)
    - *profile*
      - **One of**
        - *object*
          - *uri* (string)
          - *name* (string)
          - *codec* (string)
          - *res\_width* (string)
          - *res\_height* (string)
          - *fps* (string)
        - *null*

### 11.2. Example:

#### 11.2.1. request:

**GET /api/view/0**

#### 11.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": [{"profile": null, "url": "rtsp://192.168.1.101", "cnum": 0}]}
```

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

Play list of cameras on specific view v .

### 12.1. request (json object):

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

### 12.2. response (json object):

- *result*: OK.

### 12.3. Example:

#### 12.3.1. request:

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

#### 12.3.2. response:

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



## 13.GET /api/cameras

Get list of camera settings

### 13.1. response (json object):

- *result* (array)
  - *Items* (object)
    - *grids* (object)
      - *fullScreen* (['integer', 'null'])
      - *grid\_1x1* (['integer', 'null'])
      - *grid\_2x2* (['integer', 'null'])
      - *grid\_3x3* (['integer', 'null'])
      - *grid\_4x4* (['integer', 'null'])
    - *Label* (string)
    - *username* (string)
    - *password* (string)
    - *uri* (string)
    - *period* (integer)
    - *restart* (integer)
    - *Lab\_position* (integer)
    - *view* (['integer', 'null']): Minimum: 0. Maximum: 15.
    - *props* (object)
      - *Latency* (integer)
      - *timeout* (integer)
      - *protocols* (integer)
      - *drop\_on\_Latency* (integer)
      - *debug* (integer)
    - *zoom* (integer)
    - *onvif* (boolean)
    - *profiles* (array)
      - *Items* (object)
        - *uri* (string)
        - *name* (string)
        - *codec* (string)
        - *res\_width* (string)
        - *res\_height* (string)
        - *fps* (string)
    - *idx* (integer): Minimum: 0.

### 13.2. Example:

#### 13.2.1. request:

**GET /api/cameras**

**13.2.2. response:****200 OK****Content-Type: application/json**

```
{"result": [{"username": "", "idx": 0, "grids": {"grid_1x1": 0, "grid_2x2": 0, "grid_4x4": null, "grid_3x3": null}, "uri": "rtsp://192.168.1.101", "profiles": [], "label": "C1", "period": 0, "lab_position": 0, "props": {"latency": 0, "audio": 0, "timeout": 0, "protocols": 0}, "password": "", "restart": 0, "zoom": 0}, {"username": "", "idx": 1, "grids": {"grid_1x1": null, "grid_2x2": 1, "grid_4x4": null, "grid_3x3": null}, "uri": "rtsp://192.168.1.102", "profiles": [], "label": "C2", "period": 0, "lab_position": 0, "props": {"latency": 0, "audio": 1, "timeout": 0, "protocols": 0}, "password": "", "restart": 0, "zoom": 0}, {"username": "", "idx": 2, "grids": {"grid_1x1": null, "grid_2x2": 2, "grid_4x4": null, "grid_3x3": null}, "uri": "rtsp://192.168.1.103", "profiles": [], "label": "C3", "period": 0, "lab_position": 0, "props": {"latency": 0, "audio": 1, "timeout": 0, "protocols": 0}, "password": "", "restart": 0, "zoom": 0}]}
```

## 14.PUT /api/cameras

Set new list of cameras to be displayed

### 14.1. request (json object):

- *cameras* (array)
  - *Items* (object)
    - *grids* (object)
      - *fullScreen* (['integer', 'null'])
      - *grid\_1x1* (['integer', 'null'])
      - *grid\_2x2* (['integer', 'null'])
      - *grid\_3x3* (['integer', 'null'])
      - *grid\_4x4* (['integer', 'null'])
    - *Label* (string)
    - *username* (string)
    - *password* (string)
    - *uri* (string)
    - *period* (integer)
    - *restart* (integer)
    - *Lab\_position* (integer)
    - *view* (['integer', 'null']): Minimum: 0. Maximum: 15.
    - *props* (object)
      - *Latency* (integer)
      - *timeout* (integer)
      - *protocols* (integer)
      - *drop\_on\_Latency* (integer)
      - *debug* (integer)
    - *zoom* (integer)
    - *onvif* (boolean)
    - *profiles* (array)
      - *Items* (object)
        - *uri* (string)
        - *name* (string)
        - *codec* (string)
        - *res\_width* (string)
        - *res\_height* (string)
        - *fps* (string)
    - *idx* (integer): Minimum: 0.

### 14.2. response (json object):

- *result*: OK.

### 14.3. Example:

#### 14.3.1. request:

**PUT /api/cameras**

```
{"cameras": [{"label": "C1", "grids": {"grid_1x1": 0, "grid_2x2": 0}, "props":
```

```
{"audio": 0}, "uri": "rtsp://192.168.1.101"}, {"label": "C2", "grids": {"grid_2x2": 1}, "props": {"audio": 1}, "uri": "rtsp://192.168.1.102"}]}
```

**14.3.2. response:**

**200 OK**

**Content-Type: application/json**

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

## 15.DELETE /api/cameras

Delete existing list of cameras

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

- *result*: 0.

### 15.2. Example:

#### 15.2.1. request:

**DELETE /api/cameras**

#### 15.2.2. response:

**200 OK**

**Content-Type: application/json**

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

## 16.PUT /api/cameras/restart

restart playing cameras in a view

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

- *result*: OK.

### 16.2. Example:

#### 16.2.1. request:

```
PUT /api/cameras/restart
```

#### 16.2.2. response:

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

## 17.PUT /api/camera

Append new camera to camera list, return camera number

### 17.1. request (json object):

- *camera* (object)
  - *grids* (object)
    - *fullscreen* (['integer', 'null'])
    - *grid\_1x1* (['integer', 'null'])
    - *grid\_2x2* (['integer', 'null'])
    - *grid\_3x3* (['integer', 'null'])
    - *grid\_4x4* (['integer', 'null'])
  - *label* (string)
  - *username* (string)
  - *password* (string)
  - *uri* (string)
  - *period* (integer)
  - *restart* (integer)
  - *lab\_position* (integer)
  - *view* (['integer', 'null']): Minimum: 0. Maximum: 15.
  - *props* (object)
    - *latency* (integer)
    - *timeout* (integer)
    - *protocols* (integer)
    - *drop\_on\_latency* (integer)
    - *debug* (integer)
  - *zoom* (integer)
  - *onvif* (boolean)
  - *profiles* (array)
    - *Items* (object)
      - *uri* (string)
      - *name* (string)
      - *codec* (string)
      - *res\_width* (string)
      - *res\_height* (string)
      - *fps* (string)
  - *idx* (integer): Minimum: 0.

### 17.2. response (json object):

- *result* (integer)

### 17.3. Example:

#### 17.3.1. request:

```
PUT /api/camera
{"camera": {"label": "C3", "grids": {"grid_2x2": 2}, "props": {"audio": 1}, "u
```

```
ri": "rtsp://192.168.1.103"}}}
```

**17.3.2. response:**

**200 OK**

**Content-Type: application/json**

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



## 18.GET /api/camera/<n:int>

Get camera definition for camera *n*

### 18.1. response (json object):

- *result* (object)
  - *grids* (object)
    - *fullscreen* (['integer', 'null'])
    - *grid\_1x1* (['integer', 'null'])
    - *grid\_2x2* (['integer', 'null'])
    - *grid\_3x3* (['integer', 'null'])
    - *grid\_4x4* (['integer', 'null'])
  - *label* (string)
  - *username* (string)
  - *password* (string)
  - *uri* (string)
  - *period* (integer)
  - *restart* (integer)
  - *lab\_position* (integer)
  - *view* (['integer', 'null']): Minimum: 0. Maximum: 15.
  - *props* (object)
    - *latency* (integer)
    - *timeout* (integer)
    - *protocols* (integer)
    - *drop\_on\_latency* (integer)
    - *debug* (integer)
  - *zoom* (integer)
  - *onvif* (boolean)
  - *profiles* (array)
    - *Items* (object)
      - *uri* (string)
      - *name* (string)
      - *codec* (string)
      - *res\_width* (string)
      - *res\_height* (string)
      - *fps* (string)
  - *idx* (integer): Minimum: 0.

### 18.2. Example:

#### 18.2.1. request:

GET /api/camera/0

#### 18.2.2. response:

200 OK

Content-Type: application/json

```
{"result": {"username": "", "idx": 0, "grids": {"grid_1x1": 0, "grid_2x2": 0, "grid_4x4": null, "grid_3x3": null}, "uri": "rtsp://192.168.1.101", "profiles": [], "label": "C1", "period": 0, "lab_position": 0, "props": {"latency": 0, "audio": 0, "timeout": 0, "protocols": 0}, "password": "", "restart": 0, "zoom": 0}}
```

## 19.PUT /api/camera/<n:int>

Update camera definition for camera *n*

### 19.1. request (json object):

- *camera* (object)
  - *grids* (object)
    - *fullscreen* (['integer', 'null'])
    - *grid\_1x1* (['integer', 'null'])
    - *grid\_2x2* (['integer', 'null'])
    - *grid\_3x3* (['integer', 'null'])
    - *grid\_4x4* (['integer', 'null'])
  - *label* (string)
  - *username* (string)
  - *password* (string)
  - *uri* (string)
  - *period* (integer)
  - *restart* (integer)
  - *lab\_position* (integer)
  - *view* (['integer', 'null']): Minimum: 0. Maximum: 15.
  - *props* (object)
    - *latency* (integer)
    - *timeout* (integer)
    - *protocols* (integer)
    - *drop\_on\_latency* (integer)
    - *debug* (integer)
  - *zoom* (integer)
  - *onvif* (boolean)
  - *profiles* (array)
    - *Items* (object)
      - *uri* (string)
      - *name* (string)
      - *codec* (string)
      - *res\_width* (string)
      - *res\_height* (string)
      - *fps* (string)
  - *idx* (integer): Minimum: 0.

### 19.2. response (json object):

- *result* (integer)

### 19.3. Example:

#### 19.3.1. request:

```
PUT /api/camera/2
{"camera": {"label": "NEW LABEL", "grids": {"grid_2x2": 2}, "props": {"audio":
```

```
1}, "uri": "rtsp://192.168.1.103"}}
```

**19.3.2. response:**

**200 OK**

**Content-Type: application/json**

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

## 20.DELETE /api/camera/<n:int>

Delete camera *n*

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

- *result*
  - One of
    - *integer*
    - *null*

### 20.2. Example:

#### 20.2.1. request:

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

#### 20.2.2. response:

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

## 21.GET /api/display

Get current display resolution

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

- *result* (*string*)

### 21.2. Example:

#### 21.2.1. request:

**GET /api/display**

#### 21.2.2. response:

**200 OK**

**Content-Type: application/json**

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

## 22.GET /api/display/modes

Get current resolutions supported by display

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

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

### 22.2. Example:

#### 22.2.1. request:

**GET /api/display/modes**

#### 22.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": ["640x480-59", "640x480-60", "720x576-50", "720x480-59", "720x480-60", "1280x720-50", "1280x720-59", "1280x720-60", "1920x1080-50", "1920x1080-59", "1920x1080-60", "3840x2160-25", "3840x2160-30"]}
```

## 23.GET /api/onvif/cameras

Search network for available onvif cameras

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

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

### 23.2. Example:

#### 23.2.1. request:

GET /api/onvif/cameras

#### 23.2.2. response:

200 OK

Content-Type: application/json

```
{"result": ["192.168.86.10", "192.168.86.11"]}
```



## 24.GET /api/onvif/profiles

Query onvif cameras for available profiles

### 24.1. params:

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

### 24.2. response (json object):

- *result* (object)
  - *uri* (string)
  - *name* (string)
  - *codec* (string)
  - *res\_width* (string)
  - *res\_height* (string)
  - *fps* (string)

### 24.3. Example:

#### 24.3.1. request:

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

#### 24.3.2. response:

**200 OK**

**Content-Type: application/json**

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

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

Get network configuration for *iface*

### 25.1. response (json object):

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

### 25.2. Example:

#### 25.2.1. request:

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

#### 25.2.2. response:

```
200 OK
```

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

```
{"result": {"NTP": "80.86.38.193", "MAC": "00:16:2A:80:00:00", "Gateway": "192.168.86.1", "Netmask": "255.255.255.0", "DNS": "192.168.0.1", "Address": "192.168.86.8", "Method": "manual"}}
```

## 26.GET /api/system/info

Get sytem info

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

- *result* (*object*)
  - *cpu* (*object*)
    - *temp* (*number*)
  - *date* (*string*)
  - *version* (*object*)
    - *SW\_VERSION* (*string*)
    - *IMAGE\_BUILD\_DATE* (*string*)
    - *HW\_REVISION* (*string*)

### 26.2. Example:

#### 26.2.1. request:

**GET /api/system/info**

#### 26.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{"result": {"date": "Mon Aug 29 18:22:17 2022", "version": {"SW_VERSION": "spottbox 1.1.1", "IMAGE_BUILD_DATE": "Fri Sep 25 11:23:18 BST 2020", "HW_REVISION": "nano2 1.0"}, "cpu": {"temp": "67"}}
```

## 27.PUT /api/system/reboot

reboot device

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

- *result*: OK.

### 27.2. Example:

#### 27.2.1. request:

**PUT /api/system/reboot**

#### 27.2.2. response:

**200 OK**

**Content-Type: application/json**

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

## 28.PUT /api/system/restart

restart spotbox application

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

- *result*: OK.

### 28.2. Example:

#### 28.2.1. request:

PUT /api/system/restart

#### 28.2.2. response:

200 OK

Content-Type: application/json

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

## 29.PUT /api/system/factoryreset

Reset device to factory defaults

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

- *result*: OK.

### 29.2. Example:

#### 29.2.1. request:

PUT /api/system/factoryreset

#### 29.2.2. response:

200 OK

Content-Type: application/json

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

## 30.PUT /api/system/user

Update login password for admin

### 30.1. request (json object):

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

### 30.2. response (json object):

- *result*: OK.

### 30.3. Example:

#### 30.3.1. request:

**PUT /api/system/user**

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

#### 30.3.2. response:

**200 OK**

**Content-Type: application/json**

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

## 31.GET /api/system/config

Get current system configuration

### 31.1. response (json object):

- *result* (object)
  - *cameras* (array)
    - *Items* (object)
      - *grids* (object)
        - *fullScreen* (['integer', 'null'])
        - *grid\_1x1* (['integer', 'null'])
        - *grid\_2x2* (['integer', 'null'])
        - *grid\_3x3* (['integer', 'null'])
        - *grid\_4x4* (['integer', 'null'])
      - *Label* (string)
      - *username* (string)
      - *password* (string)
      - *uri* (string)
      - *period* (integer)
      - *restart* (integer)
      - *Lab\_position* (integer)
      - *view* (['integer', 'null']): Minimum: 0. Maximum: 15.
      - *props* (object)
        - *Latency* (integer)
        - *timeout* (integer)
        - *protocols* (integer)
        - *drop\_on\_Latency* (integer)
        - *debug* (integer)
      - *zoom* (integer)
      - *onvif* (boolean)
      - *profiles* (array)
        - *Items* (object)
          - *uri* (string)
          - *name* (string)
          - *codec* (string)
          - *res\_width* (string)
          - *res\_height* (string)
          - *fps* (string)
      - *idx* (integer): Minimum: 0.
    - *network* (object)
      - *eth0* (object)
        - *Method*: One of: [ 'manual', 'dhcp' ].
        - *Netmask* (string)
        - *Address* (string)
        - *Gateway* (string)



- *DNS* (string)
- *NTP* (string)
- *display* (string)
- *default\_view* (object)
  - *mode*: One of: ['fullscreen', 'grid\_1x1', 'grid\_2x2', 'grid\_3x3', 'grid\_4x4'].
  - *camera* (integer)
- *grid\_font* (object)
  - *fullscreen*: One of: ['s', 'm', 'l'].
  - *grid\_1x1*: One of: ['s', 'm', 'l'].
  - *grid\_2x2*: One of: ['s', 'm', 'l'].
  - *grid\_3x3*: One of: ['s', 'm', 'l'].
  - *grid\_4x4*: One of: ['s', 'm', 'l'].
- *https\_enable* (boolean)
- *user\_config* (object)

## 31.2. Example:

### 31.2.1. request:

**GET /api/system/config**

### 31.2.2. response:

**200 OK**

**Content-Type: application/json**

```
{
  "result": {
    "default_view": {
      "mode": "grid_1x1"
    },
    "network": {
      "lo0": {
        "Method": "static"
      },
      "eth0": {
        "NTP": "80.86.38.193",
        "Gateway": "192.168.86.1",
        "Netmask": "255.255.255.0",
        "DNS": "192.168.0.1",
        "Address": "192.168.86.8",
        "Method": "manual"
      }
    },
    "cameras": [
      {
        "username": "",
        "idx": 0,
        "grids": {
          "grid_1x1": 0,
          "grid_2x2": 0,
          "grid_4x4": null,
          "grid_3x3": null
        },
        "uri": "rtsp://192.168.1.101",
        "profiles": [],
        "label": "C1",
        "period": 0,
        "lab_position": 0,
        "props": {
          "latency": 0,
          "audio": 0,
          "timeout": 0,
          "protocols": 0
        },
        "password": "",
        "restart": 0,
        "zoom": 0
      },
      {
        "username": "",
        "idx": 1,
        "grids": {
          "grid_1x1": null,
          "grid_2x2": 1,
          "grid_4x4": null,
          "grid_3x3": null
        },
        "uri": "rtsp://192.168.1.102",
        "profiles": [],
        "label": "C2",
        "period": 0,
        "lab_position": 0,
        "props": {
          "latency": 0,
          "audio": 1,
          "timeout": 0,
          "protocols": 0
        },
        "password": "",
        "restart": 0,
        "zoom": 0
      }
    ],
    "display": "1920x1080-60",
    "grid_font": {
      "grid_3x3": "l",
      "fullscreen": "s",
      "grid_4x4": "m",
      "grid_1x1": "s",
      "grid_2x2": "s"
    },
    "https_enable": false,
    "user_config": {
      "location": "Bristol"
    }
  }
}
```

## 32.PUT /api/system/config

Update current system configuration (requires restart to apply)

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

- *result* (*object*)

### 32.2. Example:

#### 32.2.1. request:

**PUT /api/system/config**

```
{"config": {"user_config": {"location": "Bristol"}}
```

#### 32.2.2. response:

**200 OK**

**Content-Type: application/json**

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

## 33. POST /api/system/config

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

### 33.1. request (Multipart-Encoded File):

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

- *result (object)*

### 33.3. Example:

#### 33.3.1. request:

```
{ "files" : { "file": "<binary...data>"}}
```

#### 33.3.2. response:

200 OK

Content-Type: application/json

## 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/key

Load new ssl key from file

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

### 35.2. response (json object):

- *result: OK.*

### 35.3. Example:

#### 35.3.1. request:

```
{ "files" : { "file": "<binary...data>"}}
```

#### 35.3.2. response:

200 OK

Content-Type: application/json

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

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

Load new ssl certificate from file

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

### 36.2. response (json object):

- *result: OK.*

### 36.3. Example:

#### 36.3.1. request:

```
{ "files" : { "file": "<binary...data>"}}
```

#### 36.3.2. response:

200 OK

Content-Type: application/json

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