

ANT-375000 / Spotbox3 API guide for firmware 3.2.x

Contents

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

For further help and advice please contact Antrica on:
Email: support@antrica.com
Phone: +44 1628 626098

Document info

Version	date	author	Comments
0.1	15-Feb-23	David M	Initial draft

Spotbox API Version 3.2.x

Table of Contents

Document info.....	2
1. Introduction.....	8
2. Changes.....	9
2.1. V.1.1.1.....	9
2.2. V.2.0.x.....	9
2.3. V.3.1.x.....	9
2.4. V.3.2.x.....	9
3. POST /api/login.....	10
3.1. request (<i>json object</i>):.....	10
3.2. response (<i>json object</i>):.....	10
3.3. Example:.....	10
3.3.1. request:.....	10
3.3.2. response:.....	10
4. POST /api/logout.....	11
4.1. response (<i>json object</i>):.....	11
4.2. Example:.....	11
4.2.1. request:.....	11
4.2.2. response:.....	11
5. GET /api/view.....	12
5.1. response (<i>json object</i>):.....	12
5.2. Example:.....	12
5.2.1. request:.....	12
5.2.2. response:.....	12
6. GET /api/views.....	13
6.1. response (<i>json object</i>):.....	13
6.2. Example:.....	13
6.2.1. request:.....	13
6.2.2. response:.....	13
7. GET /api/view/mode.....	14
7.1. response (<i>json object</i>):.....	14
7.2. Example:.....	14
7.2.1. request:.....	14
7.2.2. response:.....	14
8. PUT /api/view/mode.....	15
8.1. request (<i>json object</i>):.....	15
8.2. response (<i>json object</i>):.....	15
8.3. Example:.....	15
8.3.1. request:.....	15
8.3.2. response:.....	15
9. GET /api/view/default.....	16
9.1. response (<i>json object</i>):.....	16
9.2. Example:.....	16
9.2.1. request:.....	16
9.2.2. response:.....	16
10. PUT /api/view/default.....	17

10.1. request (<i>json object</i>):	17
10.2. response (<i>json object</i>):	17
10.3. Example:.....	17
10.3.1. request:	17
10.3.2. response:	17
11. GET /api/view/status	18
11.1. response (<i>json object</i>):	18
11.2. Example:.....	18
11.2.1. request:	18
11.2.2. response:	18
12. GET /api/view/<v:int>	19
12.1. response (<i>json object</i>):	19
12.2. Example:.....	19
12.2.1. request:	19
12.2.2. response:	19
13. PUT /api/view/<v:int>/play	20
13.1. request (<i>json object</i>):	20
13.2. response (<i>json object</i>):	20
13.3. Example:.....	20
13.3.1. request:	20
13.3.2. response:	20
14. GET /api/cameras	21
14.1. response (<i>json object</i>):	21
14.2. Example:.....	21
14.2.1. request:	21
14.2.2. response:	21
15. PUT /api/cameras	22
15.1. request (<i>json object</i>):	22
15.2. response (<i>json object</i>):	22
15.3. Example:.....	22
15.3.1. request:	22
15.3.2. response:	22
16. DELETE /api/cameras	23
16.1. response (<i>json object</i>):	23
16.2. Example:.....	23
16.2.1. request:	23
16.2.2. response:	23
17. PUT /api/cameras/restart	24
17.1. response (<i>json object</i>):	24
17.2. Example:.....	24
17.2.1. request:	24
17.2.2. response:	24
18. PUT /api/camera	25
18.1. request (<i>json object</i>):	25
18.2. response (<i>json object</i>):	25
18.3. Example:.....	25
18.3.1. request:	25
18.3.2. response:	25
19. GET /api/camera/<n:int>	26
19.1. response (<i>json object</i>):	26
19.2. Example:.....	26
19.2.1. request:	26
19.2.2. response:	26

20.	PUT /api/camera/<n:int>	27
20.1.	request (<i>json object</i>):	27
20.2.	response (<i>json object</i>):	27
20.3.	Example:	27
20.3.1.	request:	27
20.3.2.	response:	27
21.	DELETE /api/camera/<n:int>	28
21.1.	response (<i>json object</i>):	28
21.2.	Example:	28
21.2.1.	request:	28
21.2.2.	response:	28
22.	GET /api/display	29
22.1.	response (<i>json object</i>):	29
22.2.	Example:	29
22.2.1.	request:	29
22.2.2.	response:	29
23.	GET /api/display/modes	30
23.1.	response (<i>json object</i>):	30
23.2.	Example:	30
23.2.1.	request:	30
23.2.2.	response:	30
24.	GET /api/onvif/cameras	31
24.1.	response (<i>json object</i>):	31
24.2.	Example:	31
24.2.1.	request:	31
24.2.2.	response:	31
25.	GET /api/onvif/profiles	32
25.1.	params:	32
25.2.	response (<i>json object</i>):	32
25.3.	Example:	32
25.3.1.	request:	32
25.3.2.	response:	32
26.	GET /api/network/<iface>	33
26.1.	response (<i>json object</i>):	33
26.2.	Example:	33
26.2.1.	request:	33
26.2.2.	response:	33
27.	PUT /api/network/<iface>	34
27.1.	request (<i>json object</i>):	34
27.2.	response (<i>json object</i>):	34
27.3.	Example:	34
27.3.1.	request:	34
27.3.2.	response:	34
28.	PUT /api/network/restart/<iface>	35
28.1.	response (<i>json object</i>):	35
28.2.	Example:	35
28.2.1.	request:	35
28.2.2.	response:	35
29.	PUT /api/network/wifi/connect/<iface>	36
29.1.	request (<i>json object</i>):	36
29.2.	response (<i>json object</i>):	36
29.3.	Example:	36

29.3.1. request:	36
29.3.2. response:	36
30. GET /api/network/wifi/scan.....	37
30.1. response (<i>json object</i>):	37
30.2. Example:.....	37
30.2.1. request:	37
30.2.2. response:	37
31. GET /api/system/info	38
31.1. response (<i>json object</i>):	38
31.2. Example:.....	38
31.2.1. request:	38
31.2.2. response:	38
32. PUT /api/system/reboot.....	39
32.1. response (<i>json object</i>):	39
32.2. Example:.....	39
32.2.1. request:	39
32.2.2. response:	39
33. PUT /api/system/restart	40
33.1. response (<i>json object</i>):	40
33.2. Example:.....	40
33.2.1. request:	40
33.2.2. response:	40
34. PUT /api/system/factoryreset.....	41
34.1. response (<i>json object</i>):	41
34.2. Example:.....	41
34.2.1. request:	41
34.2.2. response:	41
35. PUT /api/system/user.....	42
35.1. request (<i>json object</i>):	42
35.2. response (<i>json object</i>):	42
35.3. Example:.....	42
35.3.1. request:	42
35.3.2. response:	42
36. GET /api/system/config	43
36.1. response (<i>json object</i>):	43
36.2. Example:.....	43
36.2.1. request:	43
36.2.2. response:	43
37. PUT /api/system/config	44
37.1. request (<i>json object</i>):	44
37.2. response (<i>json object</i>):	44
37.3. Example:.....	44
37.3.1. request:	44
37.3.2. response:	44
38. POST /api/system/config	45
38.1. request (Multipart-Encoded JSON File):	45
38.2. response (<i>json object</i>):	45
38.3. Example:.....	45
38.3.1. request:	45
38.3.2. response:	45
39. DELETE /api/system/ssl.....	46
39.1. response (<i>json object</i>):	46

39.2. Example:.....	46
39.2.1. request:.....	46
39.2.2. response:.....	46
40. POST /api/system/ssl/key.....	47
40.1. request (Multipart-Encoded File):	47
40.2. response (<i>json object</i>):	47
40.3. Example:.....	47
40.3.1. request:.....	47
40.3.2. response:.....	47
41. POST /api/system/ssl/cert.....	48
41.1. request (Multipart-Encoded File):	48
41.2. response (<i>json object</i>):	48
41.3. Example:.....	48
41.3.1. request:.....	48
41.3.2. response:.....	48
42. Definitions	49
42.1. profile	49

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. Changes

2.1. 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
```

2.2. V.2.0.x

The following routes have been added:

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

2.3. 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.

2.4. V.3.2.x

Addition of `rtsp_src` to system configuration when supported by the device

3. POST /api/login

login to session

3.1. request (json object):

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

3.2. response (json object):

- *result*: OK.

3.3. Example:

3.3.1. request:

POST /api/login

```
{"user": "admin", "password": "1234"}
```

3.3.2. response:

200 OK

Content-Type: application/json

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

4. POST /api/logout

Logout of a session

4.1. response (*json object*):

- *result*: OK.

4.2. Example:

4.2.1. request:

POST /api/logout

4.2.2. response:

200 OK

Content-Type: application/json

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

5. GET /api/view

List of possible view modes

5.1. response (*json object*):

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

5.2. Example:

5.2.1. request:

GET /api/view

5.2.2. response:

200 OK

Content-Type: application/json

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

6. GET /api/views

Get list of cameras in each view

6.1. response (*json object*):

- *result* (*array*)
 - *Items* (*array*)
 - *Items* (*object*): Cannot contain additional properties.
 - *cnum* (*integer*)
 - *url* (*['string', 'null']*)
 - *profile* (*['integer', 'null']*)

6.2. Example:

6.2.1. request:

```
GET /api/views
```

6.2.2. response:

```
200 OK
Content-Type: application/json
{"result": [[{"cnum": 0, "url": "rtsp://user:a1b2c3d4@192.168.86.10", "profile": null}]]}
```

7. GET /api/view/mode

Get current view mode

7.1. response (*json object*):

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

7.2. Example:

7.2.1. request:

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

7.2.2. response:

```
200 OK
```

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

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

8. PUT /api/view/mode

Change current view mode

8.1. request (json object):

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

8.2. response (json object):

- *result*: `OK`.

8.3. Example:

8.3.1. request:

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

8.3.2. response:

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

9. GET /api/view/default

Get default view mode

9.1. response (*json object*):

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

9.2. Example:

9.2.1. request:

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

9.2.2. response:

```
200 OK
```

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

```
{"result": {"mode": "grid_1x1", "camera": 0}}
```


10.PUT /api/view/default

Set default view mode

10.1. request (*json object*):

10.2. response (*json object*):

- *result: OK.*

10.3. Example:

10.3.1. request:

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

10.3.2. response:

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

11.GET /api/view/status

Return current status of cameras displayed

11.1. response (*json object*):

- *result* (*array*)
 - **Items**
 - **Any of**
 - *object*: Cannot contain additional properties.
 - *status* (*integer*)
 - *status_exit* (*integer*)
 - *time_start* (*number*)
 - *time_duration* (*number*)
 - *time_update* (*number*)
 - *uri* (*['string', 'null']*)
 - *null*

11.2. Example:

11.2.1. request:

GET /api/view/status

11.2.2. response:

200 OK

Content-Type: application/json

```
{
  "result": [
    {
      "status": 4,
      "status_exit": 0,
      "time_start": 1663261703,
      "time_duration": 6,
      "time_update": 1663261709,
      "uri": "rtsp://user:a1b2c3d4@192.168.86.10"
    }
  ]
}
```

12.GET /api/view/<v:int>

Get list of cameras playing on view v

12.1. response (*json object*):

- *result* (*array*)
 - *Items*
 - **Any of**
 - *object*: Cannot contain additional properties.
 - *cnum* (*integer*)
 - *url* (['string', 'null'])
 - *profile*
 - **Any of**
 - : Refer to *#/definitions/profile*.
 - *null*
 - *null*

12.2. Example:

12.2.1. request:

```
GET /api/view/0
```

12.2.2. response:

```
200 OK
```

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

```
{"result": [{"cnum": 0, "url": "rtsp://user:a1b2c3d4@192.168.86.10", "profile": null}]}
```

13.PUT /api/view/<v:int>/play

Play list of cameras on specific view *v* .

13.1. request (*json object*):

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

13.2. response (*json object*):

- *result*: OK.

13.3. Example:

13.3.1. request:

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

13.3.2. response:

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

14.GET /api/cameras

Get list of camera settings

14.1. response (*json object*):

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

14.2. Example:

14.2.1. request:

GET /api/cameras

14.2.2. response:

200 OK

Content-Type: application/json

```
{
  "result": [
    {
      "username": "user",
      "password": "a1b2c3d4",
      "label": "C1",
      "uri": "rtsp://192.168.86.10",
      "period": 0,
      "restart": 0,
      "lab_position": 0,
      "zoom": 0,
      "props": {
        "latency": 0,
        "timeout": 0,
        "protocols": 0,
        "audio": 0
      },
      "profiles": [],
      "grids": {
        "grid_1x1": 0,
        "grid_2x2": 0,
        "grid_3x3": null,
        "grid_4x4": null
      },
      "idx": 0
    },
    {
      "username": "user",
      "password": "a1b2c3d4",
      "label": "C2",
      "uri": "rtsp://192.168.86.11:554/Streaming/Channels/101",
      "period": 0,
      "restart": 0,
      "lab_position": 0,
      "zoom": 0,
      "props": {
        "latency": 0,
        "timeout": 0,
        "protocols": 0,
        "audio": 0
      },
      "profiles": [],
      "grids": {
        "grid_1x1": null,
        "grid_2x2": 1,
        "grid_3x3": null,
        "grid_4x4": null
      },
      "idx": 1
    },
    {
      "username": "user",
      "password": "a1b2c3d4",
      "label": "C3",
      "uri": "rtsp://192.168.86.11:554/Streaming/Channels/101",
      "period": 0,
      "restart": 0,
      "lab_position": 0,
      "zoom": 0,
      "props": {
        "latency": 0,
        "timeout": 0,
        "protocols": 0,
        "audio": 0
      },
      "profiles": [],
      "grids": {
        "grid_1x1": null,
        "grid_2x2": 3,
        "grid_3x3": null,
        "grid_4x4": null
      },
      "idx": 2
    }
  ]
}
```

15.PUT /api/cameras

Set new list of cameras to be displayed

15.1. request (json object):

- *cameras* (array)
 - Items: Refer to *#/definitions/camera*.

15.2. response (json object):

- *result*: OK.

15.3. Example:

15.3.1. request:

PUT /api/cameras

```
{"cameras": [{"label": "C1", "grids": {"grid_1x1": 0, "grid_2x2": 0}, "props": {"audio": 0}, "uri": "rtsp://192.168.86.10", "username": "user", "password": "a1b2c3d4"}, {"label": "C2", "grids": {"grid_2x2": 1}, "uri": "rtsp://192.168.86.11:554/Streaming/Channels/101", "username": "user", "password": "a1b2c3d4"}]}
```

15.3.2. response:

200 OK

Content-Type: application/json

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

16.DELETE /api/cameras

Delete existing list of cameras

16.1. response (*json object*):

- *result*: 0.

16.2. Example:

16.2.1. request:

DELETE /api/cameras

16.2.2. response:

200 OK

Content-Type: application/json

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

17.PUT /api/cameras/restart

restart playing cameras in a view

17.1. response (*json object*):

- *result*: OK.

17.2. Example:

17.2.1. request:

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

17.2.2. response:

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


18.PUT /api/camera

Append new camera to camera list, return camera number

18.1. request (*json object*):

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

18.2. response (*json object*):

- *result* (*integer*)

18.3. Example:

18.3.1. request:

PUT /api/camera

```
{"camera": {"label": "C3", "grids": {"grid_2x2": 3}, "uri": "rtsp://192.168.86.11:554/Streaming/Channels/101", "username": "user", "password": "a1b2c3d4"}}
```

18.3.2. response:

200 OK

Content-Type: application/json

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

19.GET /api/camera/<n:int>

Get camera definition for camera *n*

19.1. response (*json object*):

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

19.2. Example:

19.2.1. request:

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

19.2.2. response:

```
200 OK
```

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

```
{"result": {"username": "user", "password": "a1b2c3d4", "label": "C1", "uri":  
"rtsp://192.168.86.10", "period": 0, "restart": 0, "lab_position": 0, "zoom": 0, "  
props": {"latency": 0, "timeout": 0, "protocols": 0, "audio": 0}, "profiles": [],  
"grids": {"grid_1x1": 0, "grid_2x2": 0, "grid_3x3": null, "grid_4x4": null}, "idx"  
: 0}}
```

20.PUT /api/camera/<n:int>

Update camera definition for camera *n*

20.1. request (json object):

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

20.2. response (json object):

- *result* (integer)

20.3. Example:

20.3.1. request:

PUT /api/camera/2

```
{"camera": {"label": "NEW LABEL", "grids": {"grid_2x2": 2}, "uri": "rtsp://192.168.86.10", "username": "user", "password": "a1b2c3d4"}}
```

20.3.2. response:

200 OK

Content-Type: application/json

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

21.DELETE /api/camera/<n:int>

Delete camera *n*

21.1. response (*json object*):

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

21.2. Example:

21.2.1. request:

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

21.2.2. response:

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

22.GET /api/display

Get current display resolution

22.1. response (*json object*):

- *result* (*string*)

22.2. Example:

22.2.1. request:

GET /api/display

22.2.2. response:

200 OK

Content-Type: application/json

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

23.GET /api/display/modes

Get current resolutions supported by display

23.1. response (*json object*):

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

23.2. Example:

23.2.1. request:

GET /api/display/modes

23.2.2. response:

200 OK

Content-Type: application/json

```
{
  "result": [
    "640x480@73", "640x480@67", "640x480@60", "640x480@75", "720x480@60",
    "720x400@70", "720x576@50", "800x600@60", "800x600@75", "800x600@72", "800x600@56",
    "832x624@75", "1024x768@70", "1024x768@75", "1024x768@60", "1280x720@50", "1280x1024@75",
    "1280x960@60", "1280x720@60", "1280x1024@60", "1440x900@60", "1680x1050@60",
    "1920x1080@50", "1920x2160@60", "1920x1080@60", "2560x1440@60", "3840x2160@60",
    "3840x2160@24", "3840x2160@25", "3840x2160@30", "3840x2160@50"
  ]
}
```

24.GET /api/onvif/cameras

Search network for available onvif cameras

24.1. response (*json object*):

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

24.2. Example:

24.2.1. request:

GET /api/onvif/cameras

24.2.2. response:

200 OK

Content-Type: application/json

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

25.GET /api/onvif/profiles

Query onvif cameras for available profiles

25.1. params:

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

25.2. response (json object):

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

25.3. Example:

25.3.1. request:

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

25.3.2. 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=mcast&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=mcast&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=unicast&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=unicast&profile=Profile_2"
    }
  ]
}
```


26.GET /api/network/<iface>

Get network configuration for *iface*

26.1. response (*json object*):

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

26.2. Example:

26.2.1. request:

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

26.2.2. response:

```
200 OK
```

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

```
{"result": {"Enabled": true, "Type": "Wired", "DNS": "192.168.86.1", "NTP": "192.168.86.1", "IPv4": {"Netmask": "255.255.255.0", "Address": "192.168.86.221", "Gateway": "192.168.86.1", "Method": "static"}, "State": "ready", "Address": "CA:33:6D:C6:FD:AD"}}
```

27.PUT /api/network/<iface>

Set network configuration for *iface*

27.1. request (json object):

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

27.2. response (json object):

- *result*: `OK`.

27.3. Example:

27.3.1. request:

```
PUT /api/network/wlan0
{"config":{"Enabled": true, "Connect": null, "AgentParams": [{"ssid":"localwif
i", "passphrase":"password"}]}}
```

27.3.2. response:

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

28.PUT /api/network/restart/<iface>

Restart network interface *iface*

28.1. response (json object):

- *result*: OK.

28.2. Example:

28.2.1. request:

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

28.2.2. response:

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

29.PUT /api/network/wifi/connect/<iface>

Attempt connection for interface *iface*

29.1. request (json object):

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

29.2. response (json object):

- `result`: OK.

29.3. Example:

29.3.1. request:

```
PUT /api/network/wifi/connect/wlan0
{"connect": "localwifi"}
```

29.3.2. response:

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

30.GET /api/network/wifi/scan

Return available wifi networks

30.1. response (*json object*):

- *result*
 - Any of
 - *object*: Cannot contain additional properties.
 - *wlan0* (*array*)
 - *Items* (*object*): Cannot contain additional properties.
 - *Name* (*string*)
 - *State*: Refer to [#/definitions/net_state](#).
 - *Strength* (*integer*)
 - *Security* (*array*)
 - *Items*: One of: ['none', 'wep', 'psk', 'ieee8021x', 'wps'].
- *null*

30.2. Example:

30.2.1. request:

```
GET /api/network/wifi/scan
```

30.2.2. response:

```
200 OK
Content-Type: application/json
{"result": {"wlan0": [{"Name": "localwifi", "State": "ready", "Strength": 49, "Security": ["psk"]}]}}
```

31.GET /api/system/info

Get sytem info

31.1. response (*json object*):

- *result* (*object*): Can contain additional properties.
 - *cpu* (*object*): Cannot contain additional properties.
 - *temp* (*number*)
 - *date* (*string*)
 - *version* (*object*): Cannot contain additional properties.
 - *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 3.1.3",
      "IMAGE_BUILD_DATE": "Wed 3 Aug 23:25:20 BST 2022",
      "HW_REVISION": "spotbox rk3566-pc",
      "sn": "",
      "sys": {
        "cpu_temp": 37.777,
        "gpu_temp": 38.888,
        "boot_time": "15-09-2022 09:11:31",
        "cpu_freq": 1608.0,
        "mem": 35.1,
        "cpu_times": 9.6,
        "date": "Thu Sep 15 17:08:47 2022"
      }
    }
  }
}
```

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.PUT /api/system/factoryreset

Reset device to factory defaults

34.1. response (*json object*):

- *result*: OK.

34.2. Example:

34.2.1. request:

PUT /api/system/factoryreset

34.2.2. response:

200 OK

Content-Type: application/json

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

35.PUT /api/system/user

Update login password for admin

35.1. request (*json object*):

- *user (object)*: Cannot contain additional properties.
 - *username (string)*
 - *password (string)*

35.2. response (*json object*):

- *result: OK.*

35.3. Example:

35.3.1. request:

PUT /api/system/user

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

35.3.2. response:

200 OK

Content-Type: application/json

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

36.GET /api/system/config

Get current system configuration

36.1. response (json object):

- **result** (object): Cannot contain additional properties.
 - **cameras** (array)
 - **Items**: Refer to `#/definitions/camera`.
 - **network** (object): Cannot contain additional properties.
 - **eth0**: Refer to `#/definitions/network_interface`.
 - **wlan0**: Refer to `#/definitions/network_interface`.
 - **lo0**: Refer to `#/definitions/network_interface`.
 - **display** (string)
 - **default_view**: Refer to `#/definitions/default_view`.
 - **grid_font**: Refer to `#/definitions/grid_font`.
 - **https_enable** (boolean)
 - **user_config** (object)
 - **rtsp_svr**: Refer to `#/definitions/rtsp_svr`.
 - **web_ui**: Refer to `#/definitions/web_ui`.

36.2. Example:

36.2.1. request:

GET /api/system/config

36.2.2. response:

200 OK

Content-Type: application/json

```
{
  "result": {
    "cameras": [
      {
        "username": "user",
        "password": "a1b2c3d4",
        "label": "C1",
        "uri": "rtsp://192.168.86.10",
        "period": 0,
        "restart": 0,
        "lab_position": 0,
        "zoom": 0,
        "props": {
          "latency": 0,
          "timeout": 0,
          "protocols": 0,
          "audio": 0
        },
        "profiles": [],
        "grids": {
          "grid_1x1": 0,
          "grid_2x2": 0,
          "grid_3x3": null,
          "grid_4x4": null
        },
        "idx": 0,
        "username": "user",
        "password": "a1b2c3d4",
        "label": "C2",
        "uri": "rtsp://192.168.86.11:554/Streaming/Channels/101",
        "period": 0,
        "restart": 0,
        "lab_position": 0,
        "zoom": 0,
        "props": {
          "latency": 0,
          "timeout": 0,
          "protocols": 0,
          "audio": 0
        },
        "profiles": [],
        "grids": {
          "grid_1x1": null,
          "grid_2x2": 1,
          "grid_3x3": null,
          "grid_4x4": null
        },
        "idx": 1
      }
    ],
    "network": {
      "eth0": {
        "Enabled": true,
        "Type": "Wired",
        "DNS": "192.168.86.1",
        "NTP": "192.168.86.1",
        "IPv4": {
          "Method": "static",
          "Address": "192.168.86.221",
          "Gateway": "192.168.86.1",
          "Netmask": "255.255.255.0"
        }
      },
      "wlan0": {
        "Enabled": true,
        "Type": "WiFi",
        "DNS": "192.168.0.1",
        "NTP": "192.168.0.1",
        "IPv4": {
          "Method": "dhcp"
        },
        "AgentParams": [
          {
            "ssid": "localwifi",
            "passphrase": "password"
          }
        ],
        "Connect": null,
        "lo0": {
          "Type": "Loopback"
        }
      }
    },
    "display": "1920x1080",
    "default_view": {
      "mode": "grid_1x1"
    },
    "grid_font": {
      "fullscreen": "s",
      "grid_1x1": "s",
      "grid_2x2": "s",
      "grid_3x3": "s",
      "grid_4x4": "s"
    },
    "user_config": {
      "location": "Bristol"
    },
    "web_ui": {
      "iface": "eth0",
      "host": "0.0.0.0",
      "https_enable": false,
      "https_port": 443,
      "http_port": 80
    }
  }
}
```

37.PUT /api/system/config

Update current system configuration (requires restart to apply)

37.1. request (json object):

- **config** (object): Cannot contain additional properties.
 - **cameras** (array)
 - **Items**: Refer to `#/definitions/camera`.
 - **network** (object): Cannot contain additional properties.
 - **eth0**: Refer to `#/definitions/network_interface`.
 - **wlan0**: Refer to `#/definitions/network_interface`.
 - **Lo0**: Refer to `#/definitions/network_interface`.
 - **display** (string)
 - **default_view**: Refer to `#/definitions/default_view`.
 - **grid_font**: Refer to `#/definitions/grid_font`.
 - **https_enable** (boolean)
 - **user_config** (object)
 - **rtsp_svr**: Refer to `#/definitions/rtsp_svr`.
 - **web_ui**: Refer to `#/definitions/web_ui`.

37.2. response (json object):

- **result**: OK.

37.3. Example:

37.3.1. request:

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

37.3.2. response:

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

38.POST /api/system/config

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

38.1. request (Multipart-Encoded JSON File):

- *config* (object): Cannot contain additional properties.
 - *cameras* (array)
 - **Items**: Refer to `#/definitions/camera`.
 - *network* (object): Cannot contain additional properties.
 - **eth0**: Refer to `#/definitions/network_interface`.
 - **wlan0**: Refer to `#/definitions/network_interface`.
 - **Lo0**: Refer to `#/definitions/network_interface`.
 - *display* (string)
 - *default_view*: Refer to `#/definitions/default_view`.
 - *grid_font*: Refer to `#/definitions/grid_font`.
 - *https_enable* (boolean)
 - *user_config* (object)
 - *rtsp_svr*: Refer to `#/definitions/rtsp_svr`.
 - *web_ui*: Refer to `#/definitions/web_ui`.

38.2. response (json object):

- *result*: OK.

38.3. Example:

38.3.1. request:

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

38.3.2. response:

200 OK

Content-Type: application/json

39.DELETE /api/system/ssl

Delete current ssl certificate and key

39.1. response (*json object*):

- *result*: OK.

39.2. Example:

39.2.1. request:

DELETE /api/system/ssl

39.2.2. response:

200 OK

Content-Type: application/json

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

40. POST /api/system/ssl/key

Load new ssl key from file

40.1. request (Multipart-Encoded File):

40.2. response (json object):

- *result: OK.*

40.3. Example:

40.3.1. request:

POST /api/system/ssl/key

40.3.2. response:

200 OK

Content-Type: application/json

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

41. POST /api/system/ssl/cert

Load new ssl certificate from file

41.1. request (Multipart-Encoded File):

41.2. response (json object):

- *result*: OK.

41.3. Example:

41.3.1. request:

POST /api/system/ssl/cert

41.3.2. response:

200 OK

Content-Type: application/json

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


42. Definitions

42.1. profile

- *object*: Cannot contain additional properties.
 - *uri* (*string*)
 - *name* (*string*)
 - *codec* (*string*)
 - *res_width* (*string*)
 - *res_height* (*string*)
 - *fps* (*string*)
 - *enabled* (*boolean*) ## view
- *object*: Cannot contain additional properties.
 - *grid* (*string*)
 - *cameras* (*array*)
 - **Items** (*object*): Cannot contain additional properties.
 - *camera* (*integer*)
 - *profile* (*integer*, 'null') ## camera
- *object*: Cannot contain additional properties.
 - *grids* (*object*): Cannot contain additional properties.
 - *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')
 - *props* (*object*): Cannot contain additional properties.
 - *audio* (*integer*)
 - *latency* (*integer*)
 - *timeout* (*integer*)
 - *protocols* (*integer*)
 - *drop_on_latency* (*integer*)
 - *debug* (*integer*)
 - *zoom* (*integer*)
 - *onvif* (*boolean*)
 - *profiles* (*array*)
 - **Items**: Refer to [#/definitions/profile](#).
 - *idx* (*integer*): Minimum: 0. ## net_state

- : One of: ['idle', 'failure', 'association', 'configuration', 'ready', 'disconnect', 'offline', 'online']. ## network_interface
- *object*: Cannot contain additional properties.
 - **Available** (boolean)
 - **Enabled** (boolean)
 - **Type**: One of: ['Wired', 'Loopback', 'WiFi'].
 - **State**: Refer to #/definitions/net_state.
 - **AgentParams** (array)
 - **Items** (*object*): Cannot contain additional properties.
 - **name** (['string', 'null'])
 - **ssid** (['string', 'null'])
 - **identity** (['string', 'null'])
 - **passphrase** (['string', 'null'])
 - **wpspin** (['string', 'null'])
 - **Connect** (['string', 'null'])
 - **Address** (['string', 'null'])
 - **DNS**
 - **Any of**
 - *string*: ipv4
 - *string*: Maxlength: 0
 - **NTP**
 - **Any of**
 - *string*: ipv4
 - *string*: Maxlength: 0
 - **IPv4** (*object*): Cannot contain additional properties.
 - **Method**: One of: ['manual', 'dhcp', 'static'].
 - **Netmask**
 - **Any of**
 - *string*: ipv4
 - *string*: Maxlength: 0
 - **Address**
 - **Any of**
 - *string*: ipv4
 - *string*: Maxlength: 0
 - **Gateway**
 - **Any of**
 - *string*: ipv4
 - *string*: Maxlength: 0
 - **Method** (*string*): *Deprecated*
 - **Netmask** (*string*): *Deprecated*
 - **Gateway** (*string*): *Deprecated* ## grid_modes
- : One of: ['fullscreen', 'grid_1x1', 'grid_2x2', 'grid_3x3', 'grid_4x4']. ## grid_font
- *object*: Cannot contain additional properties.

- **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']. ## default_view
- **object**: Cannot contain additional properties.
 - **mode**: Refer to #/definitions/grid_modes.
 - **camera** (integer) ## web_ui
- **object**: Cannot contain additional properties.
 - **iface**: eth0.
 - **host**
 - **Any of**
 - string: ipv4
 - string: Maxlength: 0
 - **https_enable** (boolean)
 - **https_port** (integer): Minimum: 0.
 - **http_port** (integer): Minimum: 0. ## rtsp_stream
- **object**: Cannot contain additional properties.
 - **name** (string)
 - **enc_props** (object): Cannot contain additional properties.
 - **bps** (integer)
 - **profile**: One of: ['baseline', 'main', 'high'].
 - **width** (integer)
 - **height** (integer)
 - **gop** (integer)
 - **rc-mode**: One of: ['cbr', 'vbr'].
 - **framerate** (integer)
 - **transport**: One of: ['UNICAST', 'MCAST', 'UDP', 'TCP', 'ANY'].
 - **stream_address** (object): Cannot contain additional properties.
 - **address**
 - **Any of**
 - string: ipv4
 - string: Maxlength: 0
 - **port** (integer)
 - **tTL** (integer) ## rtsp_svr
- **object**: Cannot contain additional properties.
 - **available** (boolean)
 - **enabled** (boolean)
 - **onvif** (boolean)
 - **port** (integer)
 - **username** (['string', 'null'])
 - **password** (['string', 'null'])
 - **streams** (array)

- **Items:** Refer to `#/definitions/rtsp_stream`.