

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

Contents

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

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
0.2	15-Feb-23	David M	Install python section - from ANT-37000A v1.0.6

Spotbox API Version 3.2.x

Table of Contents

Document info.....	2
1. Introduction.....	8
2. EXAMPLE PYTHON CODE (v2.7.13)	8
2.1. Installing on Windows	8
3. Changes.....	11
3.1. V.1.1.1.....	11
3.2. V.2.0.x.....	11
3.3. V.3.1.x.....	11
3.4. V.3.2.x.....	11
4. POST /api/login	12
4.1. request (<i>json object</i>):	12
4.2. response (<i>json object</i>):	12
4.3. Example:.....	12
4.3.1. request:	12
4.3.2. response:	12
5. POST /api/logout	13
5.1. response (<i>json object</i>):	13
5.2. Example:.....	13
5.2.1. request:	13
5.2.2. response:	13
6. GET /api/view	14
6.1. response (<i>json object</i>):	14
6.2. Example:.....	14
6.2.1. request:	14
6.2.2. response:	14
7. GET /api/views.....	15
7.1. response (<i>json object</i>):	15
7.2. Example:.....	15
7.2.1. request:	15
7.2.2. response:	15
8. GET /api/view/mode	16
8.1. response (<i>json object</i>):	16
8.2. Example:.....	16
8.2.1. request:	16
8.2.2. response:	16
9. PUT /api/view/mode	17
9.1. request (<i>json object</i>):	17
9.2. response (<i>json object</i>):	17
9.3. Example:.....	17
9.3.1. request:	17
9.3.2. response:	17
10. GET /api/view/default	18
10.1. response (<i>json object</i>):	18
10.2. Example:.....	18
10.2.1. request:	18

10.2.2. response:	18
11. PUT /api/view/default.....	19
11.1. request (<i>json object</i>):	19
11.2. response (<i>json object</i>):	19
11.3. Example:.....	19
11.3.1. request:	19
11.3.2. response:	19
12. GET /api/view/status	20
12.1. response (<i>json object</i>):	20
12.2. Example:.....	20
12.2.1. request:	20
12.2.2. response:	20
13. GET /api/view/<v:int>	21
13.1. response (<i>json object</i>):	21
13.2. Example:.....	21
13.2.1. request:	21
13.2.2. response:	21
14. PUT /api/view/<v:int>/play	22
14.1. request (<i>json object</i>):	22
14.2. response (<i>json object</i>):	22
14.3. Example:.....	22
14.3.1. request:	22
14.3.2. response:	22
15. GET /api/cameras	23
15.1. response (<i>json object</i>):	23
15.2. Example:.....	23
15.2.1. request:	23
15.2.2. response:	23
16. PUT /api/cameras	24
16.1. request (<i>json object</i>):	24
16.2. response (<i>json object</i>):	24
16.3. Example:.....	24
16.3.1. request:	24
16.3.2. response:	24
17. DELETE /api/cameras	25
17.1. response (<i>json object</i>):	25
17.2. Example:.....	25
17.2.1. request:	25
17.2.2. response:	25
18. PUT /api/cameras/restart	26
18.1. response (<i>json object</i>):	26
18.2. Example:.....	26
18.2.1. request:	26
18.2.2. response:	26
19. PUT /api/camera	27
19.1. request (<i>json object</i>):	27
19.2. response (<i>json object</i>):	27
19.3. Example:.....	27
19.3.1. request:	27
19.3.2. response:	27
20. GET /api/camera/<n:int>	28
20.1. response (<i>json object</i>):	28

20.2. Example:	28
20.2.1. request:	28
20.2.2. response:	28
21. PUT /api/camera/<n:int>	29
21.1. request (<i>json object</i>):	29
21.2. response (<i>json object</i>):	29
21.3. Example:	29
21.3.1. request:	29
21.3.2. response:	29
22. DELETE /api/camera/<n:int>	30
22.1. response (<i>json object</i>):	30
22.2. Example:	30
22.2.1. request:	30
22.2.2. response:	30
23. GET /api/display	31
23.1. response (<i>json object</i>):	31
23.2. Example:	31
23.2.1. request:	31
23.2.2. response:	31
24. GET /api/display/modes	32
24.1. response (<i>json object</i>):	32
24.2. Example:	32
24.2.1. request:	32
24.2.2. response:	32
25. GET /api/onvif/cameras	33
25.1. response (<i>json object</i>):	33
25.2. Example:	33
25.2.1. request:	33
25.2.2. response:	33
26. GET /api/onvif/profiles	34
26.1. params:	34
26.2. response (<i>json object</i>):	34
26.3. Example:	34
26.3.1. request:	34
26.3.2. response:	34
27. GET /api/network/<iface>	35
27.1. response (<i>json object</i>):	35
27.2. Example:	35
27.2.1. request:	35
27.2.2. response:	35
28. PUT /api/network/<iface>	36
28.1. request (<i>json object</i>):	36
28.2. response (<i>json object</i>):	36
28.3. Example:	36
28.3.1. request:	36
28.3.2. response:	36
29. PUT /api/network/restart/<iface>	37
29.1. response (<i>json object</i>):	37
29.2. Example:	37
29.2.1. request:	37
29.2.2. response:	37
30. PUT /api/network/wifi/connect/<iface>	38

30.1. request (<i>json object</i>):	38
30.2. response (<i>json object</i>):	38
30.3. Example:.....	38
30.3.1. request:	38
30.3.2. response:	38
31. GET /api/network/wifi/scan.....	39
31.1. response (<i>json object</i>):	39
31.2. Example:.....	39
31.2.1. request:	39
31.2.2. response:	39
32. GET /api/system/info	40
32.1. response (<i>json object</i>):	40
32.2. Example:.....	40
32.2.1. request:	40
32.2.2. response:	40
33. PUT /api/system/reboot.....	41
33.1. response (<i>json object</i>):	41
33.2. Example:.....	41
33.2.1. request:	41
33.2.2. response:	41
34. PUT /api/system/restart	42
34.1. response (<i>json object</i>):	42
34.2. Example:.....	42
34.2.1. request:	42
34.2.2. response:	42
35. PUT /api/system/factoryreset.....	43
35.1. response (<i>json object</i>):	43
35.2. Example:.....	43
35.2.1. request:	43
35.2.2. response:	43
36. PUT /api/system/user.....	44
36.1. request (<i>json object</i>):	44
36.2. response (<i>json object</i>):	44
36.3. Example:.....	44
36.3.1. request:	44
36.3.2. response:	44
37. GET /api/system/config	45
37.1. response (<i>json object</i>):	45
37.2. Example:.....	45
37.2.1. request:	45
37.2.2. response:	45
38. PUT /api/system/config	46
38.1. request (<i>json object</i>):	46
38.2. response (<i>json object</i>):	46
38.3. Example:.....	46
38.3.1. request:	46
38.3.2. response:	46
39. POST /api/system/config	47
39.1. request (Multipart-Encoded JSON File):	47
39.2. response (<i>json object</i>):	47
39.3. Example:.....	47
39.3.1. request:	47
39.3.2. response:	47

40.	DELETE /api/system/ssl.....	48
40.1.	response (<i>json object</i>):	48
40.2.	Example:.....	48
40.2.1.	request:	48
40.2.2.	response:	48
41.	POST /api/system/ssl/key.....	49
41.1.	request (Multipart-Encoded File):	49
41.2.	response (<i>json object</i>):	49
41.3.	Example:.....	49
41.3.1.	request:	49
41.3.2.	response:	49
42.	POST /api/system/ssl/cert	50
42.1.	request (Multipart-Encoded File):	50
42.2.	response (<i>json object</i>):	50
42.3.	Example:.....	50
42.3.1.	request:	50
42.3.2.	response:	50
43.	Definitions	51
43.1.	profile	51

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.3.2.x

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

4. POST /api/login

login to session

4.1. request (*json object*):

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

4.2. response (*json object*):

- *result*: OK.

4.3. Example:

4.3.1. request:

POST /api/login

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

4.3.2. response:

200 OK

Content-Type: application/json

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

5. POST /api/logout

Logout of a session

5.1. response (*json object*):

- *result*: OK.

5.2. Example:

5.2.1. request:

POST /api/logout

5.2.2. response:

200 OK

Content-Type: application/json

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

6. GET /api/view

List of possible view modes

6.1. response (*json object*):

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

6.2. Example:

6.2.1. request:

GET /api/view

6.2.2. response:

200 OK

Content-Type: application/json

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

7. GET /api/views

Get list of cameras in each view

7.1. response (*json object*):

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

7.2. Example:

7.2.1. request:

```
GET /api/views
```

7.2.2. response:

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

8. GET /api/view/mode

Get current view mode

8.1. response (*json object*):

- *result*: Refer to [#/definitions/grid_modes](#).

8.2. Example:

8.2.1. request:

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

8.2.2. response:

```
200 OK
```

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

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

9. PUT /api/view/mode

Change current view mode

9.1. request (json object):

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

9.2. response (json object):

- *result*: `OK`.

9.3. Example:

9.3.1. request:

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

9.3.2. response:

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

10.GET /api/view/default

Get default view mode

10.1. response (*json object*):

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

10.2. Example:

10.2.1. request:

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

10.2.2. response:

```
200 OK
```

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

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

11.PUT /api/view/default

Set default view mode

11.1. request (*json object*):

11.2. response (*json object*):

- *result: OK.*

11.3. Example:

11.3.1. request:

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

11.3.2. response:

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

12.GET /api/view/status

Return current status of cameras displayed

12.1. 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*

12.2. Example:

12.2.1. request:

GET /api/view/status

12.2.2. 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"
    }
  ]
}
```

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

Get list of cameras playing on view v

13.1. 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*

13.2. Example:

13.2.1. request:

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

13.2.2. response:

```
200 OK
```

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

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

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

Play list of cameras on specific view *v* .

14.1. request (*json object*):

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

14.2. response (*json object*):

- *result*: OK.

14.3. Example:

14.3.1. request:

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

14.3.2. response:

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

15.GET /api/cameras

Get list of camera settings

15.1. response (json object):

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

15.2. Example:

15.2.1. request:

GET /api/cameras

15.2.2. 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}]}
```

16.PUT /api/cameras

Set new list of cameras to be displayed

16.1. request (json object):

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

16.2. response (json object):

- *result*: OK.

16.3. Example:

16.3.1. 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"}]}
```

16.3.2. response:

200 OK

Content-Type: application/json

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

17.DELETE /api/cameras

Delete existing list of cameras

17.1. response (*json object*):

- *result*: 0.

17.2. Example:

17.2.1. request:

DELETE /api/cameras

17.2.2. response:

200 OK

Content-Type: application/json

{"result": 0}

18.PUT /api/cameras/restart

restart playing cameras in a view

18.1. response (*json object*):

- *result*: OK.

18.2. Example:

18.2.1. request:

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

18.2.2. response:

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

19.PUT /api/camera

Append new camera to camera list, return camera number

19.1. request (json object):

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

19.2. response (json object):

- *result* (integer)

19.3. Example:

19.3.1. 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"}}
```

19.3.2. response:

200 OK

Content-Type: application/json

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

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

Get camera definition for camera *n*

20.1. response (*json object*):

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

20.2. Example:

20.2.1. request:

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

20.2.2. 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}}
```

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

Update camera definition for camera *n*

21.1. request (json object):

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

21.2. response (json object):

- *result* (integer)

21.3. Example:

21.3.1. request:

PUT /api/camera/2

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

21.3.2. response:

200 OK

Content-Type: application/json

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

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

Delete camera *n*

22.1. response (*json object*):

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

22.2. Example:

22.2.1. request:

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

22.2.2. response:

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

23.GET /api/display

Get current display resolution

23.1. response (*json object*):

- *result* (*string*)

23.2. Example:

23.2.1. request:

GET /api/display

23.2.2. response:

200 OK

Content-Type: application/json

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

24.GET /api/display/modes

Get current resolutions supported by display

24.1. response (*json object*):

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

24.2. Example:

24.2.1. request:

GET /api/display/modes

24.2.2. 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"
  ]
}
```

25.GET /api/onvif/cameras

Search network for available onvif cameras

25.1. response (*json object*):

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

25.2. Example:

25.2.1. request:

GET /api/onvif/cameras

25.2.2. response:

200 OK

Content-Type: application/json

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

26.GET /api/onvif/profiles

Query onvif cameras for available profiles

26.1. params:

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

26.2. response (json object):

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

26.3. Example:

26.3.1. request:

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

26.3.2. 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"
    }
  ]
}
```

27.GET /api/network/<iface>

Get network configuration for *iface*

27.1. response (json object):

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

27.2. Example:

27.2.1. request:

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

27.2.2. 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"}}
```

28.PUT /api/network/<iface>

Set network configuration for *iface*

28.1. request (json object):

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

28.2. response (json object):

- *result*: `OK`.

28.3. Example:

28.3.1. request:

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

28.3.2. response:

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

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

Restart network interface *iface*

29.1. response (json object):

- *result*: OK.

29.2. Example:

29.2.1. request:

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

29.2.2. response:

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

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

Attempt connection for interface *iface*

30.1. request (json object):

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

30.2. response (json object):

- *result*: OK.

30.3. Example:

30.3.1. request:

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

30.3.2. response:

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

31.GET /api/network/wifi/scan

Return available wifi networks

31.1. 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*

31.2. Example:

31.2.1. request:

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

31.2.2. response:

```
200 OK
```

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

```
{"result": {"wlan0": [{"Name": "localwifi", "State": "ready", "Strength": 49, "Security": ["psk"]}]}
```

32.GET /api/system/info

Get sytem info

32.1. 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*)

32.2. Example:

32.2.1. request:

GET /api/system/info

32.2.2. 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"
      }
    }
  }
}
```

33.PUT /api/system/reboot

reboot device

33.1. response (*json object*):

- *result*: OK.

33.2. Example:

33.2.1. request:

PUT /api/system/reboot

33.2.2. response:

200 OK

Content-Type: application/json

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

34.PUT /api/system/restart

restart spotbox application

34.1. response (*json object*):

- *result*: OK.

34.2. Example:

34.2.1. request:

PUT /api/system/restart

34.2.2. response:

200 OK

Content-Type: application/json

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

35.PUT /api/system/factoryreset

Reset device to factory defaults

35.1. response (*json object*):

- *result*: OK.

35.2. Example:

35.2.1. request:

PUT /api/system/factoryreset

35.2.2. response:

200 OK

Content-Type: application/json

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

36.PUT /api/system/user

Update login password for admin

36.1. request (json object):

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

36.2. response (json object):

- *result*: OK.

36.3. Example:

36.3.1. request:

PUT /api/system/user

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

36.3.2. response:

200 OK

Content-Type: application/json

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

37.GET /api/system/config

Get current system configuration

37.1. 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`.

37.2. Example:

37.2.1. request:

GET /api/system/config

37.2.2. 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
    }
  }
}
```

38.PUT /api/system/config

Update current system configuration (requires restart to apply)

38.1. 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`.

38.2. response (json object):

- *result*: OK.

38.3. Example:

38.3.1. request:

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

38.3.2. response:

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

39.POST /api/system/config

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

39.1. 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`.

39.2. response (json object):

- *result*: OK.

39.3. Example:

39.3.1. request:

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

39.3.2. response:

200 OK

Content-Type: application/json

40.DELETE /api/system/ssl

Delete current ssl certificate and key

40.1. response (*json object*):

- *result: OK.*

40.2. Example:

40.2.1. request:

DELETE /api/system/ssl

40.2.2. response:

200 OK

Content-Type: application/json

{"result": "OK"}

41. POST /api/system/ssl/key

Load new ssl key 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/key

41.3.2. response:

200 OK

Content-Type: application/json

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

42. POST /api/system/ssl/cert

Load new ssl certificate from file

42.1. request (Multipart-Encoded File):

42.2. response (json object):

- *result*: OK.

42.3. Example:

42.3.1. request:

POST /api/system/ssl/cert

42.3.2. response:

200 OK

Content-Type: application/json

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

43. Definitions

43.1. 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`.