

Be Smart Home and Professional Server



Application Programming Interface

API version 1.1

Table of contents

Introduction.....4

Our mission.....4

API Documentation.....4

 Introduction4

 Authorization5

 Raw GPIO management.....5

 Command syntax.....5

 Examples.....5

 Get state of port number 15

 Get state of all GPIO ports6

 Set state of port 3 to high6

 State state of all port to low:.....6

 Lights management.....6

 Get State of all defined lights :.....7

 Get State of lights by name:.....7

 Get detailed information of light by name:.....7

 Get light specified light parameter.....8

 Define new light source at GPIO 5.....8

 Change light parameter "reversed".....8

 Change light name.....8

 Change light group assignment.....8

 Doors management.....8

 2 state door locks (open/close)8

 Time unlocking gateway (unlock interval).....8

 Video source management.....9

 Triggers management.....9

 Dashboard management.....9

Introduction

This documents describes official API for both Be Smart Home and Be Smart Professional Server software provided you by Smartphone Solutions company. It allows you to control most of the devices inside and outside your house.

Our mission

Our mission is to provide the latest and professional technology for controlling buildings using mobile devices such as smartphones, laptops and tablets. Our products are solutions that do not require the high cost of installation and operation. We use a public and widely used solutions based on open source servers such as Apache, Linux and other systems. We strive to meet the expectations of our customers, so that each of them was satisfied.

If you need more information about our mission and services please visit:

http://smartphonesolutions.eu/our-mission_f_2_con_mission

API Documentation

Introduction

We have designed RESTFul API provided by PHP language service runned on Raspberry PI with Apache Web server. More information about server software could be reached here:

http://smartphonesolutions.eu/What-is-the-BeSmart-Server-_f_3_con_20

http://smartphonesolutions.eu/web-management-console-pro_f_2_con_webmanagement

Authorization

We gives you simple stateless authorization by query string with token. If you have password authentication enabled to MUST attach to every call a ?p=PASSWORD query string. It doesn't matter what kind of method do you use.

Let's assume that you password token is **kittyPassword**

Here is some examples

```
GET  ./DASH?p=kittyPassword
GET  ./GPIO/1?p=kittyPassword
PUT  ./GPIO/1/on?p=kittyPassword
```

Remember that you have to attach this token to each call.

Raw GPIO management

Command syntax

If you don't want to use aliases like lights, doors or you have a device which do not belong to any of pre-definied groups you can you basic raw GPIO steering. Here is the complex URI RESOURCE syntax:

```
./GPIO/port_number/on|off
```

Notice that port are numbered from 1

Examples

Get state of port number 1

```
GET  ./GPIO/1
```

Result example

```
200 OK
{"1":"on"}
```

Get state of all GPIO ports

use only if very need - function may long time

```
GET ./GPIO
```

Result example

```
200 OK
{"1":"on","2":"off","3":"off"}
```

Set state of port 3 to high

```
PUT ./GPIO/3/off
```

State state of all port to low:

```
PUT ./GPIO/off
```

Lights management

Lights is controlled by simple 2 state relay with no time delay. High voltage state on the GPIO pin means that light is turned on and low means that it is turned off. If you have advanced steering module you may be interested in reverse low/high voltage states. It could be done by setting reverse state parameter. Here is detailed API description for the lights

controlling.

Get State of all defined lights :

```
GET ./Lights
```

Result example

```
200 OK
{"Kitchen":"on","Lobby":"off","Bathroom Floor":"off"}
```

Get State of lights by name:

```
GET ./Lights/Kitchen
```

Result example

```
200 OK
{"Kitchen":"on"}
```

Get detailed information of light by name:

```
GET ./Lights/Kitchen/define
```

Result example

```
200 OK
{"name":"Kitchen",
"GPIO":"1"
"reversed":"no",
"group":"Inside",
"extra":"","
"state":"on"}
"lastChange":"2014-03-20 12:00:00"}
```

Get light specified light parameter

```
GET ./Lights/NewRoom/define/reversed

200 OK
{"name":"Kitchen",
"reversed":"no"}
```

Define new light source at GPIO 5

```
PUT ./Lights/NewRoom/define/GPIO/5
```

Change light parameter "reversed"

```
PUT ./Lights/NewRoom/define/reversed/on
```

Change light name

```
PUT ./Lights/NewRoom/define/name/NewerRoom
```

Change light group assignment

```
PUT ./Lights/NewRoom/define/group/My new group
```

Doors management**2 state door locks (open/close)****Time unlocking gateway (unlock interval)**

Video source management

Triggers management

Dashboard management