

Smart Home

Overview

This is the documentation for smart home web rest api by [reem alghamdi](#)

Version information

Version : 1.0

Contact information

Contact Email : reem.brain@gmail.com

License information

License : MIT

License URL : <https://opensource.org/licenses/MIT>

Terms of service : null

URI scheme

Host : gp.reem-codes.com

BasePath : /api

Schemes : HTTPS, HTTP

Tags

- **Command** : What user wants raspberry to do. It is a mapping between the time, the hardware and the configuration
- **Configuration** : All the possible configuration and status for a given hardware. eg: on, off, red, extended ...
- **Hardware** : Everything about the sensors and actuators connected to raspberry pi
- **RaspberryPi** : raspberry pi connected to users
- **Response** : What the raspberry wants the user to know. It is a mapping between the time of execution, the command triggered and whether the action was done or not
- **Schedule** : scheduling information for command if any. Specifys the days and time of day the user would like a command to be triggered
- **User** : saving user information to correctly connect them with their respectful raspberry pi

Security

bearerAuth

Type : apiKey

Name : Authorization

In : HEADER

Paths

Add a new command

POST /command

Parameters

Type	Name	Description	Schema
Body	command <i>optional</i>	The command to create.	command

command

Name	Description	Schema
configurationId <i>required</i>	the id of the configuration to apply to the hardware Example : 3	integer
hardwareId <i>required</i>	the id of the hardware this command targets Example : 1	integer
scheduleId <i>optional</i>	the id of the schedule if any Example : 453	integer

Responses

HTTP Code	Description	Schema
201	Created	Command
405	Invalid input	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- Command

get all commands

```
GET /command
```

Responses

HTTP Code	Description	Schema
200	return an array of command objects	< Command > array

Produces

- `application/json`

Tags

- Command

get a command by id

```
GET /command/{commandId}
```

Parameters

Type	Name	Schema
Path	commandId <i>required</i>	integer

Responses

HTTP Code	Description	Schema
200	get the command	Command
404	Not found	No Content

Produces

- `application/json`

Tags

- Command

edit an existing command given its id

```
PUT /command/{commandId}
```

Parameters

Type	Name	Description	Schema
Path	commandId <i>required</i>		integer
Body	command <i>optional</i>	The command to edit.	command

command

Name	Description	Schema
configurationId <i>required</i>	the id of the configuration to apply to the hardware Example : 3	integer
hardwareId <i>required</i>	the id of the hardware this command targets Example : 1	integer
scheduleId <i>optional</i>	the id of the schedule if any Example : 453	integer

Responses

HTTP Code	Description	Schema
200	Edited	Command
404	Not found	No Content
405	Invalid input	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- Command

delete a command by id

```
DELETE /command/{commandId}
```

Parameters

Type	Name	Schema
Path	commandId <i>required</i>	integer

Responses

HTTP Code	Description	Schema
204	deleted	No Content
404	Not found	No Content

Tags

- Command

Add a new configuration to a hardware

POST /configuration

Parameters

Type	Name	Description	Schema
Body	configuration <i>optional</i>	The configuration to create.	configuration

configuration

Name	Description	Schema
description <i>optional</i>	information about the configuration Example : "EXTEND means that it will become longer by 3cm"	string
hardwareId <i>required</i>	the hardware this configuration belongs to Example : "ON belongs to the hardwareID 1, which is RED LED"	integer
name <i>required</i>	the configuration name Example : "ON"	string

Responses

HTTP Code	Description	Schema
201	Created	Configuration
405	Invalid input	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- Configuration

get a configuration by id

GET /configuration/{configurationId}

Parameters

Type	Name	Schema
Path	configurationId <i>required</i>	integer

Responses

HTTP Code	Description	Schema
200	get the configuration	Configuration
404	Not found	No Content

Produces

- `application/json`

Tags

- Configuration

edit an existing configuration given its id

PUT /configuration/{configurationId}

Parameters

Type	Name	Description	Schema
Path	configurationId <i>required</i>		integer
Body	configuration <i>optional</i>	The configuration to edit.	configuration

configuration

Name	Description	Schema
description <i>optional</i>	information about the configuration Example : "EXTEND means that it will become longer by 3cm"	string
hardwareId <i>optional</i>	the hardware this configuration belongs to Example : "ON belongs to the hardwareID 1, which is RED LED"	integer
name <i>optional</i>	the configuration name Example : "ON"	string

Responses

HTTP Code	Description	Schema
200	Edited	Configuration
404	Not found	No Content
405	Invalid input	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- Configuration

delete a configuration by id

```
DELETE /configuration/{configurationId}
```

Parameters

Type	Name	Schema
Path	configurationId <i>required</i>	integer

Responses

HTTP Code	Description	Schema
204	deleted	No Content
404	Not found	No Content

Tags

- Configuration

connect a user to raspberry pi

POST /connect

Parameters

Type	Name	Description	Schema
Body	user_raspberry <i>optional</i>	The user and raspberry data	user_raspberry

user_raspberry

Name	Description	Schema
raspberry_id <i>required</i>	the raspberry id that needs to be connected to the logged in user Example : 3	integer

Responses

HTTP Code	Description	Schema
200	return an array of raspberry pi objects connected	< RaspberryPi > array
405	Invalid input	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- RaspberryPi
- User

disconnect a user from a raspberry pi

POST /disconnect

Parameters

Type	Name	Description	Schema
Body	user_raspberry <i>optional</i>	The user and raspberry data	user_raspberry

user_raspberry

Name	Description	Schema
raspberry_id <i>required</i>	the raspberry id that needs to be disconnected from the logged in user Example : 4	integer

Responses

HTTP Code	Description	Schema
200	return an array of raspberry pi objects connected	< RaspberryPi > array
405	Invalid input	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- RaspberryPi
- User

Add a new Hardware to the system

POST /hardware

Parameters

Type	Name	Description	Schema
Body	hardware <i>optional</i>	The hardware to create.	hardware

hardware

Name	Description	Schema
description <i>optional</i>	additional info regarding the hardware Example : "LED is a small electrical component that can emit light"	string
gpio <i>required</i>	the gpio pin the hardware is installed at Example : 11	integer
icon <i>optional</i>	URL image to desired icon in client Example : "https://image.flaticon.com/icons/png/512/32/32750.png"	string
name <i>required</i>	the hardware name Example : "RGB LED"	string

Responses

HTTP Code	Description	Schema
201	Created	Hardware
405	Invalid input	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- Hardware

get all hardwares connected to raspberry pi

GET /hardware

Responses

HTTP Code	Description	Schema
200	return an array of hardware objects	< Hardware > array

Produces

- `application/json`

Tags

- Hardware

get a hardware by id

GET /hardware/{hardwareId}

Parameters

Type	Name	Schema
Path	hardwareId <i>required</i>	integer

Responses

HTTP Code	Description	Schema
200	get the hardware	Hardware
404	Not found	No Content

Produces

- `application/json`

Tags

- Hardware

edit an existing hardware given its id

```
PUT /hardware/{hardwareId}
```

Parameters

Type	Name	Description	Schema
Path	hardwareId <i>required</i>		integer
Body	hardware <i>optional</i>	The hardware to edit.	hardware

hardware

Name	Description	Schema
description <i>optional</i>	additional info regarding the hardware Example : "LED is a small electrical component that can emit light"	string
gpio <i>required</i>	the gpio pin the hardware is installed at Example : 11	integer
icon <i>optional</i>	URL image to desired icon in client Example "https://image.flaticon.com/icons/png/512/32/32750.png"	: string

Name	Description	Schema
name <i>required</i>	the hardware name Example : "RGB LED"	string

Responses

HTTP Code	Description	Schema
200	Edited	Hardware
404	Not found	No Content
405	Invalid input	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- Hardware

delete a hardware by id

```
DELETE /hardware/{hardwareId}
```

Parameters

Type	Name	Schema
Path	hardwareId <i>required</i>	integer

Responses

HTTP Code	Description	Schema
204	deleted	No Content

HTTP Code	Description	Schema
404	Not found	No Content

Tags

- Hardware

get all commands given a hardwareID

```
GET /hardware/{hardwareId}/command
```

Parameters

Type	Name	Schema
Path	hardwareId <i>required</i>	integer

Responses

HTTP Code	Description	Schema
200	return an array of command objects	< Command > array

Produces

- [application/json](#)

Tags

- Command
- Hardware

get all configuration for a given hardware

```
GET /hardware/{hardwareId}/configuration
```

Parameters

Type	Name	Schema
Path	hardwareId <i>required</i>	integer

Responses

HTTP Code	Description	Schema
200	return an array of configuration objects	< Configuration > array

Produces

- [application/json](#)

Tags

- Configuration
- Hardware

user logs in

POST /login

Parameters

Type	Name	Description	Schema
Body	user <i>optional</i>	The user credentials.	user

user

Name	Description	Schema
email <i>required</i>	the user email used in login Example : " reem.brain@gmail.com "	string (email)
password <i>required</i>	user password for verification Example : " Big_s3cret "	string

Responses

HTTP Code	Description	Schema
200	Created	Response 200
405	Invalid input	No Content

Response 200

Name	Schema
token <i>optional</i>	string (jwt bearer token)

Consumes

- `application/json`

Produces

- `application/json`

Tags

- User

Add a new raspberry pi to the system

POST /raspberry

Parameters

Type	Name	Description	Schema
Body	raspberry <i>optional</i>	The raspberry to create.	raspberry

raspberry

Name	Description	Schema
id <i>required</i>	the raspberry pi id Example : 14	integer

Responses

HTTP Code	Description	Schema
201	Created	RaspberryPi
405	Invalid input	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- RaspberryPi

get all raspberry pis connected to a user

GET /raspberry

Responses

HTTP Code	Description	Schema
200	return an array of raspberry pi objects	< RaspberryPi > array

Produces

- `application/json`

Tags

- RaspberryPi

user register

POST /register

Parameters

Type	Name	Description	Schema
Body	user <i>optional</i>	The user credentials.	user

user

Name	Description	Schema
email <i>required</i>	the user email wanted Example : <code>"reem.brain@gmail.com"</code>	string (email)
password <i>required</i>	user wanted password Example : <code>"Big_s3cret"</code>	string

Responses

HTTP Code	Description	Schema
200	Created	Response 200
405	Invalid input	No Content

Response 200

Name	Schema
token <i>optional</i>	string (jwt bearer token)

Consumes

- `application/json`

Produces

- `application/json`

Tags

- User

Add a new response

POST /response

Parameters

Type	Name	Description	Schema
Body	response <i>optional</i>	The response to create.	response

response

Name	Description	Schema
commandId <i>required</i>	the command id the response is for Example : 23	integer
executionTime <i>optional</i>	the actual time raspberry pi executed the command	string (date-time)
isDone <i>required</i>	whether the command has been successfully done or not Example : true	boolean
isRead <i>optional</i>	whether the response has been read by the android client Example : false	boolean
message <i>optional</i>	optional message regarding the action Example : "the command 243 was successfully executed!"	string

Responses

HTTP Code	Description	Schema
201	Created	Response
405	Invalid input	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- Response

get all responses

GET /response

Responses

HTTP Code	Description	Schema
200	return an array of response objects	< Response array >

Produces

- `application/json`

Tags

- Response

get a response by id

GET /response/{responseId}

Parameters

Type	Name	Schema
Path	responseId <i>required</i>	integer

Responses

HTTP Code	Description	Schema
200	get the response	Response

HTTP Code	Description	Schema
404	Not found	No Content

Produces

- `application/json`

Tags

- Response

edit an existing response given its id

```
PUT /response/{responseId}
```

Parameters

Type	Name	Description	Schema
Path	responseId <i>required</i>		integer
Body	response <i>optional</i>	The response to edit.	response

response

Name	Description	Schema
commandId <i>optional</i>	the command id the response is for Example : <code>23</code>	integer
executionTime <i>optional</i>	the actual time raspberry pi executed the command	string (date-time)
isDone <i>optional</i>	whether the command has been successfully done or not Example : <code>false</code>	boolean
isRead <i>optional</i>	whether the response has been read by the android client Example : <code>true</code>	boolean

Name	Description	Schema
message <i>optional</i>	optional message regarding the action Example : "the command 243 was successfully executed!"	string

Responses

HTTP Code	Description	Schema
200	Edited	Response
404	Not found	No Content
405	Invalid input	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- Response

delete a response by id

```
DELETE /response/{responseId}
```

Parameters

Type	Name	Schema
Path	responseId <i>required</i>	integer

Responses

HTTP Code	Description	Schema
204	deleted	No Content

HTTP Code	Description	Schema
404	Not found	No Content

Tags

- Response

Add a new schedule

POST /schedule

Parameters

Type	Name	Description	Schema
Body	schedule <i>optional</i>	The schedule to create.	schedule

schedule

Name	Description	Schema
commandId <i>required</i>	the command id this schedule belongs to Example : 231	integer
days <i>optional</i>	value between 0 and 127, representing 7 bits each bit correspond to a day in the order: sun mon tues web thurs fri sat Minimum value : 0 Maximum value : 127 Example : 120	integer (int64)
time <i>optional</i>	the time of day the command shall be executed Example : "13:50"	string (time)

Responses

HTTP Code	Description	Schema
201	Created	Schedule
405	Invalid input	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- Schedule

get all schedules

GET /schedule

Responses

HTTP Code	Description	Schema
200	return an array of schedule objects	< Schedule > array

Produces

- `application/json`

Tags

- Schedule

get a schedule by id

GET /schedule/{scheduleId}

Parameters

Type	Name	Schema
Path	scheduleId <i>required</i>	integer

Responses

HTTP Code	Description	Schema
200	get the schedule	Schedule
404	Not found	No Content

Produces

- `application/json`

Tags

- Schedule

edit an existing schedule given its id

```
PUT /schedule/{scheduleId}
```

Parameters

Type	Name	Description	Schema
Path	scheduleId <i>required</i>		integer
Body	schedule <i>optional</i>	The schedule to edit.	schedule

schedule

Name	Description	Schema
commandId <i>required</i>	the command id this schedule belongs to Example : <code>231</code>	integer
days <i>optional</i>	value between 0 and 127, representing 7 bits each bit correspond to a day in the order: sun mon tues web thurs fri sat Minimum value : <code>0</code> Maximum value : <code>127</code> Example : <code>120</code>	integer (int64)
time <i>optional</i>	the time of day the command shall be executed Example : <code>"13:50"</code>	string (time)

Responses

HTTP Code	Description	Schema
200	Edited	Schedule
404	Not found	No Content
405	Invalid input	No Content

Consumes

- `application/json`

Produces

- `application/json`

Tags

- Schedule

delete a schedule by id

```
DELETE /schedule/{scheduleId}
```

Parameters

Type	Name	Schema
Path	scheduleId <i>required</i>	integer

Responses

HTTP Code	Description	Schema
204	deleted	No Content
404	Not found	No Content

Tags

- Schedule

Definitions

Command

Name	Description	Schema
configuration <i>required</i>	the configuration desired for the hardware	Configuration
hardware <i>required</i>	the hardware this command is issued for	Hardware
id <i>required</i>	the identifier for the command in the database Example : 456	integer (int64)
schedule <i>optional</i>	the schedule for this command. If none this command is immediate	Schedule
updateAt <i>optional</i>	time of creation/last updating	string (date-time)

Configuration

Name	Description	Schema
description <i>optional</i>	information about the configuration Example : "EXTEND means that it will become longer by 3cm"	string
hardware <i>required</i>	the hardware this configuration belongs to	Hardware
id <i>required</i>	the identifier for the configuration in the database Example : 3	integer (int64)
name <i>required</i>	name of the configuration Example : "ON"	string
updateAt <i>optional</i>	time of creation/last updating	string (date-time)

Hardware

Name	Description	Schema
description <i>optional</i>	additional info regarding the hardware Example : "LED is a small electrical component that can emit light"	string
gpio <i>required</i>	the gpio pin the hardware is installed at Example : 11	integer
icon <i>optional</i>	URL image to desired icon in client Example : "https://image.flaticon.com/icons/png/512/32/32750.png"	string
id <i>required</i>	the identifier for the configuration in the database Example : 1	integer (int64)
name <i>required</i>	the hardware name Example : "RGB LED"	string
raspberrypi <i>optional</i>	the raspberrypi this hardware is connected to	Raspberrypi
status <i>optional</i>	the current configuration for the hardware	Configuration
updatedAt <i>optional</i>	time of creation/last updating	string (date-time)

Raspberrypi

Name	Description	Schema
id <i>required</i>	the identifier for the user in the database Example : 1	integer (int64)
updatedAt <i>optional</i>	time of creation/last updating	string (date-time)

Response

Name	Description	Schema
command <i>required</i>	the command executed resulting in this response	Command

Name	Description	Schema
executionTime <i>optional</i>	the actual time raspberry pi executed the command	string (date-time)
id <i>required</i>	the identifier for the response in the database Example : 865	integer (int64)
isDone <i>required</i>	whether the command has been successfully done or not Example : false	boolean
isRead <i>optional</i>	whether the response has been read by the android client Example : true	boolean
message <i>optional</i>	optional message regarding the action Example : "electrical error: hardware is not connected to the circuit"	string
updateAt <i>optional</i>	time of creation/last updating	string (date-time)

Schedule

Name	Description	Schema
command <i>required</i>	the command where this scheduling info is for	Command
days <i>optional</i>	value between 0 and 127, representing 7 bits each bit correspond to a day in the order: sun mon tues web thurs fri sat Minimum value : 0 Maximum value : 127 Example : 120	integer (int64)
id <i>required</i>	the identifier for the schedule in the database Example : 75	integer (int64)
time <i>optional</i>	the time of day the command shall be executed Example : "13:50"	string (time)
updateAt <i>optional</i>	time of creation/last updating	string (date-time)

User

Name	Description	Schema
email <i>required</i>	the user email used in login Example : "reem.brain@gmail.com"	string (email)
id <i>required</i>	the identifier for the user in the database Example : 1	integer (int64)
password <i>required</i>	user password for verification Example : "Big_s3cret"	string
updateAt <i>optional</i>	time of creation/last updating	string (date-time)