

Hello!

Here we have a short tutorial on how to integrate bOS and TASKER using new bOS' RPC service. bOS offers a JSON RPC Service for integration with Tasker, IFTTT and other services.

This tutorial will show bOS RPC service interacting with TASKER on Android in a simple task, but the important thing here is to understand how to set-up the RPC service so you can do more complex integrations.

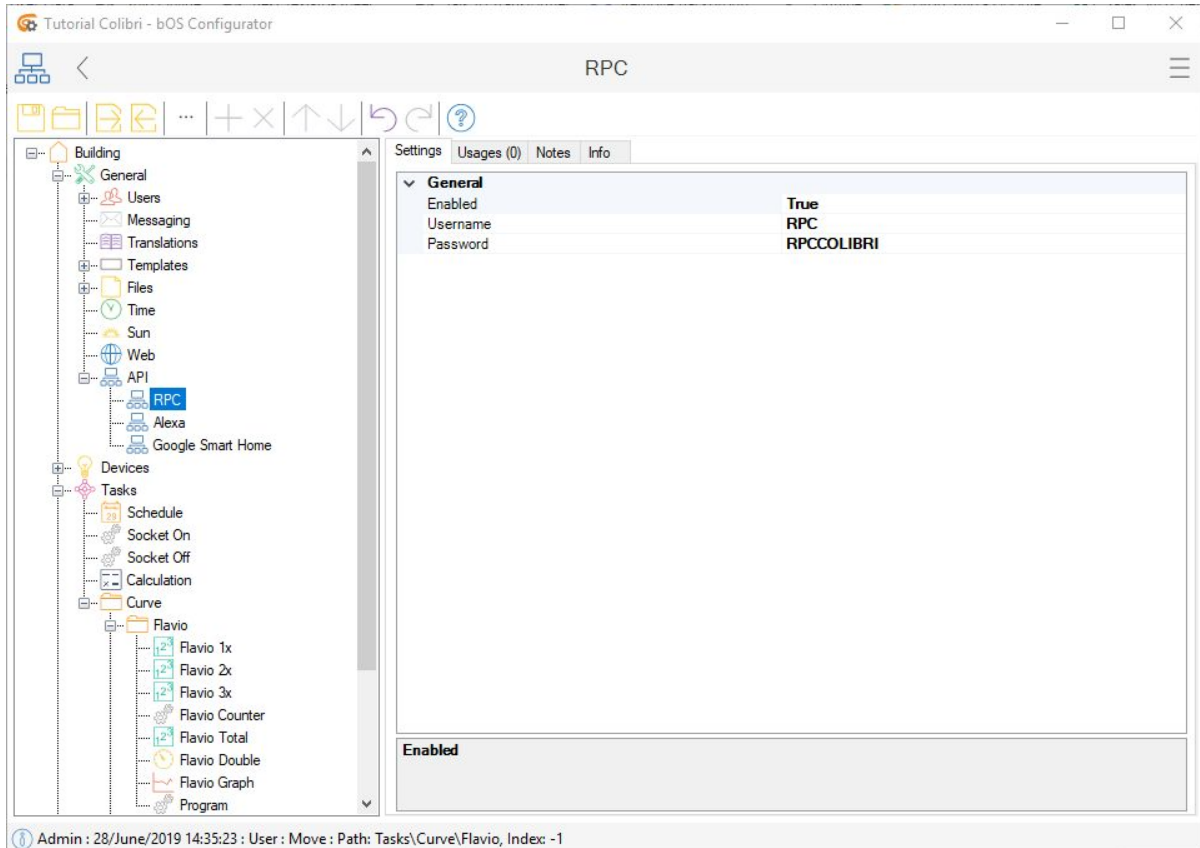
RPC service on the new bOS offers HTTP basic authentication with username, and password.

In this example, we'll be using Tasker to turn a Z-Wave Socket off.

Of course, this can be done easily within bOS Client app, but this is just for demo purposes. You can change and adapt the task to your needs, for example, "If my phone battery is lower than 80%, turn the socket on", or "If I've reached my house, turn front door light on". The possibilities are endless.

Step 1: Configure your RPC service

- In bOS Configurator, under *API*, enable *RPC* and define your *Username* and *Password*:



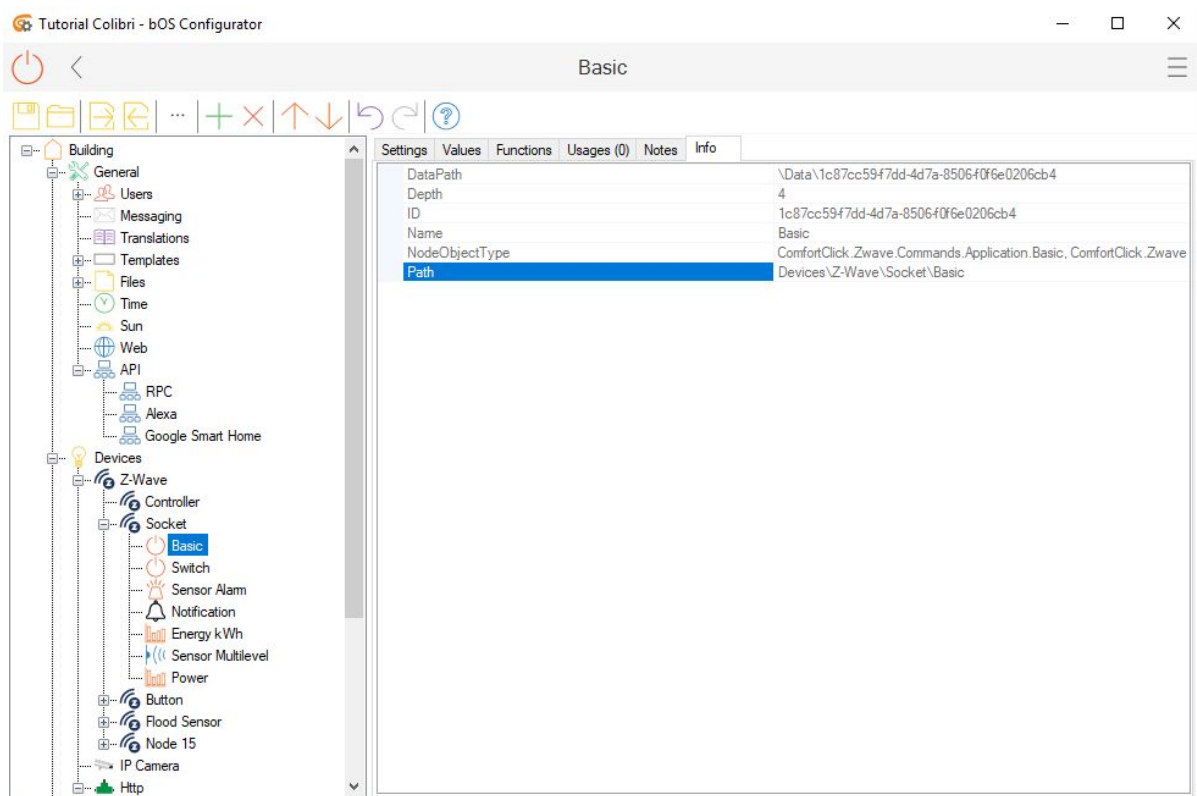
The screenshot shows the bOS Configurator application window titled "Tutorial Colibri - bOS Configurator". The interface is divided into a left sidebar with a tree view and a main right pane. The tree view on the left shows a hierarchy: Building > General > API > RPC. The "RPC" item is selected and highlighted in blue. The main pane on the right is titled "RPC" and contains a "Settings" tab. Under the "General" section, the following configuration is visible:

Setting	Value
Enabled	True
Username	RPC
Password	RPCCOLIBRI

Below the settings table, there is a section labeled "Enabled" with a grey background. At the bottom of the window, a status bar displays the text: "Admin : 28/June/2019 14:35:23 : User : Move : Path: Tasks\Curve\Flavio, Index: -1".

Step 2: Find your Device Path

- In this example, the *Info* tab from the Z-Wave socket device will show us the path:
Devices\Z-Wave\Socket\Basic



Now we need to form the JSON body to POST on TASKER.

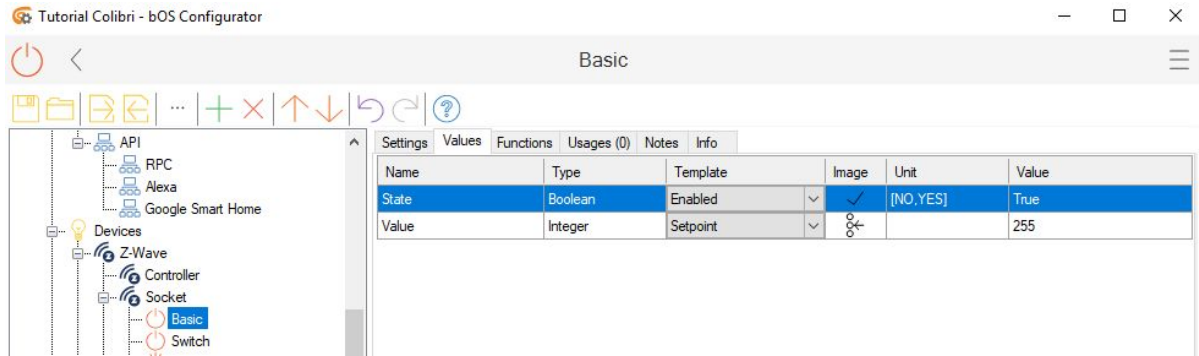
The Path we have is: **Devices\Z-Wave\Socket\Basic**

We need to double the backslashes on the Path to form the JSON body to POST on TASKER service, so now we have: **Devices\\Z-Wave\\Socket\\Basic**

The simple SetValue body post looks something like this:

{"objectName":"INSERT PATH HERE","valueName":"Value","value":"true"} where "value":"true" is to turn the value to true or 1. You can change to false or 0 or look for other RPC examples.

For our valueName, we have "State":

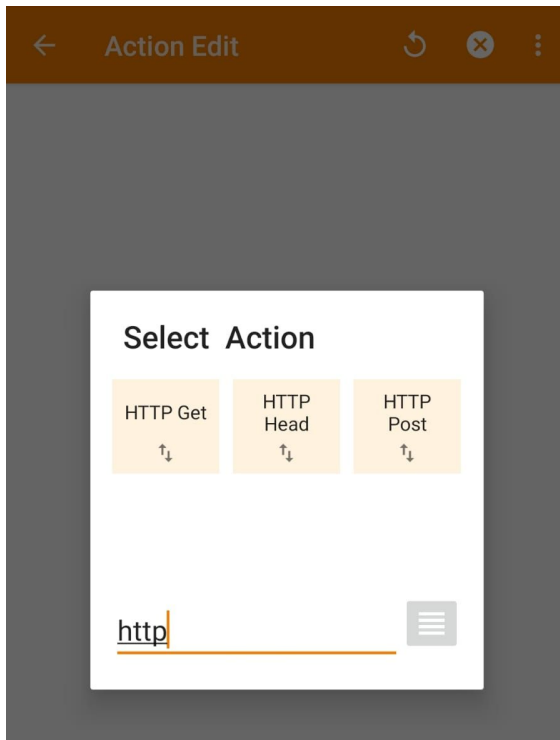


To turn the Socket off, our body post is:

```
{"objectName":"Devices\\Z-Wave\\Socket\\Basic","valueName":"State","value":"false"}
```

Step 3: Tasker

- Create a new task and include the “HTTP Post” action:



- In your *HTTP Post*:
Server:Port
`https://RPCUserName:RPCPassword@your_ip_address_or_dynamic_dns_address/API/RPC/SetValue`

in our example:

```
https://RPC:RPCCOLIBRI@ip_address_or_dynamic_dns_address/API/RPC/SetValue
```

Data / File (here we include our body post)

```
{"objectName":"Devices\\Z-Wave\\Socket\\Basic","valueName":"State","value":"false"}
```

Content Type

application/json; charset=utf-8

Trust Any Certificate

yes

Action Edit
1. HTTP Post

Server:Port

Path

Data / File

```
{"objectName":"Devices\\Z-Wave\\Socket\\Basic","valueName":"State","value":"false"}
```

Cookies

User Agent

User Agent

Optional

Timeout

10

Content Type

application/json; charset=utf-8

Output File

Optional

Trust Any Certificate ☒

Continue Task After Error ☐

If ☐

Label ☐

Done! Now every time we run this task our Socket will turn off.

This is a simple example for demo purposes, but with more complex commands you can do whatever you want... call scenes, set other values, etc.

To call an RPC Service an HTTP POST call must be performed with data content-type = "application/json";

Service offers the following commands:

SetValue

POST data example (Turn ON a light): {"objectName":"Devices\\KNX\\Light 1","valueName":"Value","value":"true"}

GetValue



Building Automation Software & Servers

POST data example (Get light status): {"objectName":"Devices\\KNX\\Light 1","valueName":"Value"}

CallFunction

POST data example (Send alert to user):

{"objectName":"Building\\General\\Users\\User","functionName":"SendAlert","value":["Hello"]}

I hope you've enjoyed the tutorial and please post here the examples and ideas you have using the new bOS' RPC API.