Hello!

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

This tutorial will show bOS RPC service interacting with IFTTT 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 offers HTTP basic authentication with username, and password.

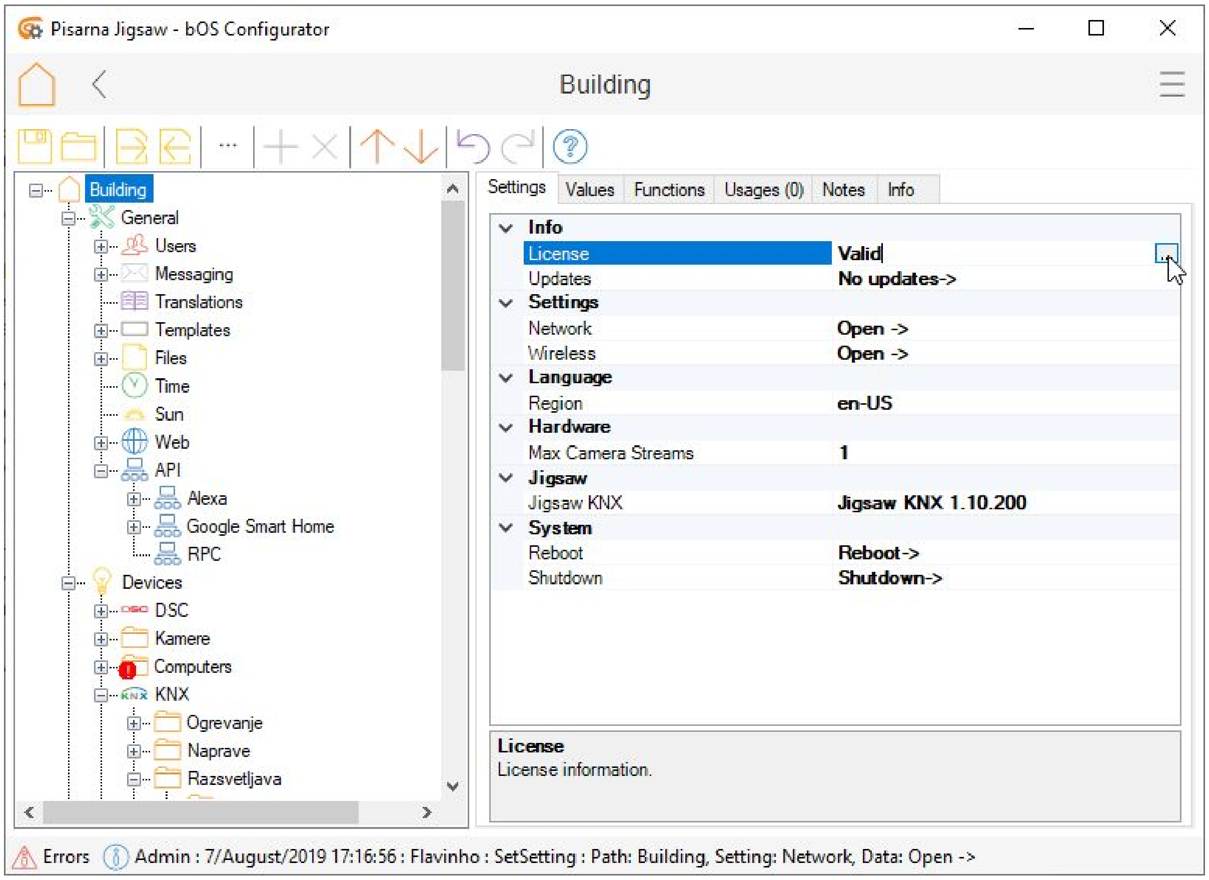
In this example, we'll be using IFTTT to turn a KNX light ON based on your location, so a **geofencing scenario** that could be changed and used to open a door, turn the music on, etc.
The possibilities are endless.

Step 1: Port forwarding

A. Since the command will be coming from IFTTT cloud, your server must be set to receive commands from external networks by having port forwarding enabled on your router.
You can check your port forwarding on this site: https://portchecker.co/. If you have port forwarding already enabled, you can skip to Step 2

B. A static WAN IP address is needed to access your network and should be obtained
    a. from your internet provider. If you don't have a static WAN IP address, use a dynamic DNS
    b. service. On your router setup page enable port forwarding (NAT) to the local IP address of your controller. It is recommended to only forward the port 443 for secure https connection.

C. Now open bOS Configurator, under Building, open the menu from the License section:

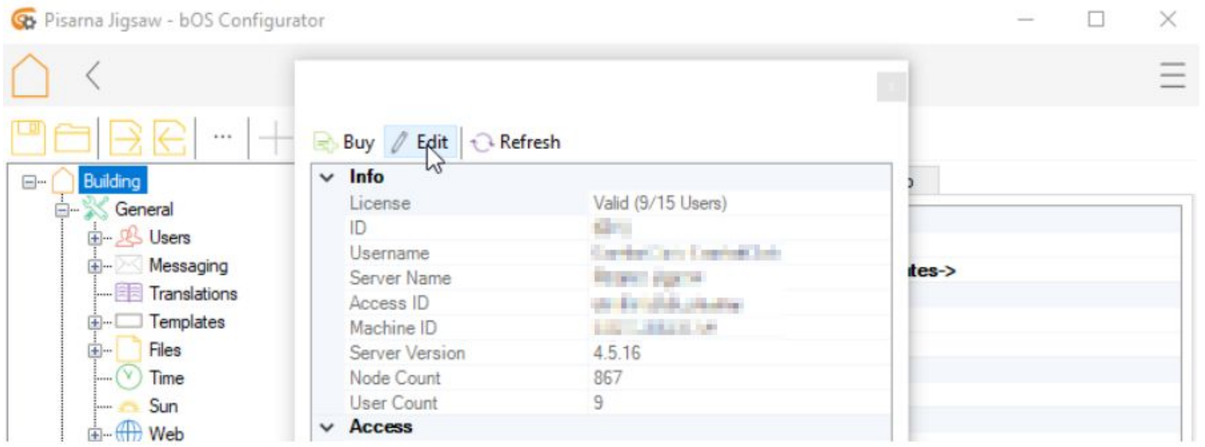


```
Info
  License               Valid
  Updates               No updates->
Settings
  Network               Open ->
  Wireless              Open ->
Language
  Region                en-US
Hardware
  Max Camera Streams    1
Jigsaw
  Jigsaw KNX            Jigsaw KNX 1.10.200
System
  Reboot                Reboot->
  Shutdown              Shutdown->
```

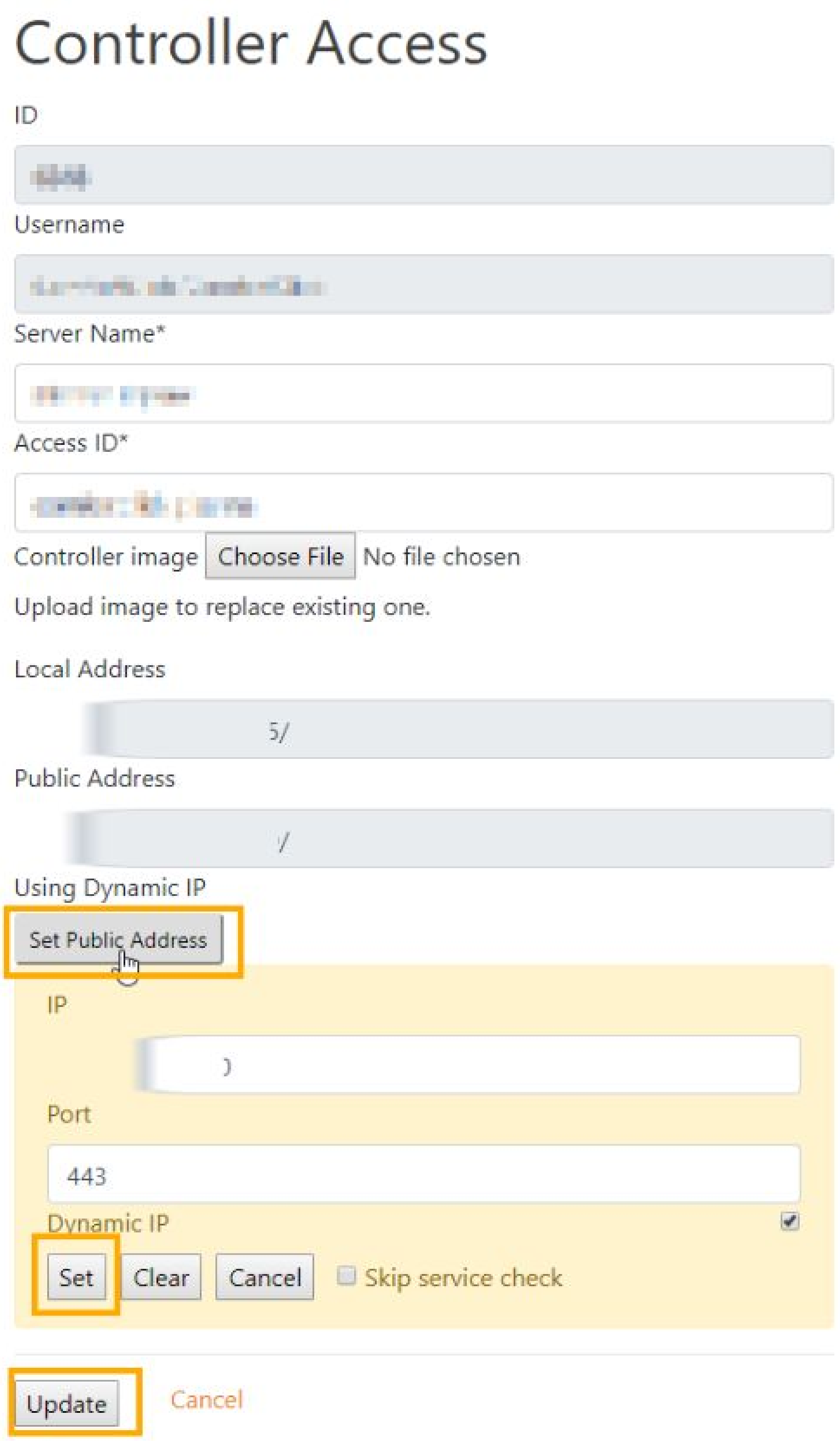
License
License information.

Errors Admin : 7/August/2019 17:16:56 : Flavinho : SetSetting : Path: Building, Setting: Network, Data: Open ->

Click "Edit" and on the next page set the public address and click "Update".



```
Info
  License          Valid (9/15 Users)
  ID               [redacted]
  Username         [redacted]
  Server Name      [redacted]
  Access ID        [redacted]
  Machine ID       [redacted]
  Server Version   4.5.16
  Node Count       867
  User Count       9
Access
```



Controller Access

ID

[redacted]

Username

[redacted]

Server Name*

[redacted]

Access ID*

[redacted]

Controller image Choose File No file chosen

Upload image to replace existing one.

Local Address

[redacted]

Public Address

[redacted]

Using Dynamic IP

Set Public Address

IP

[redacted]

Port

443

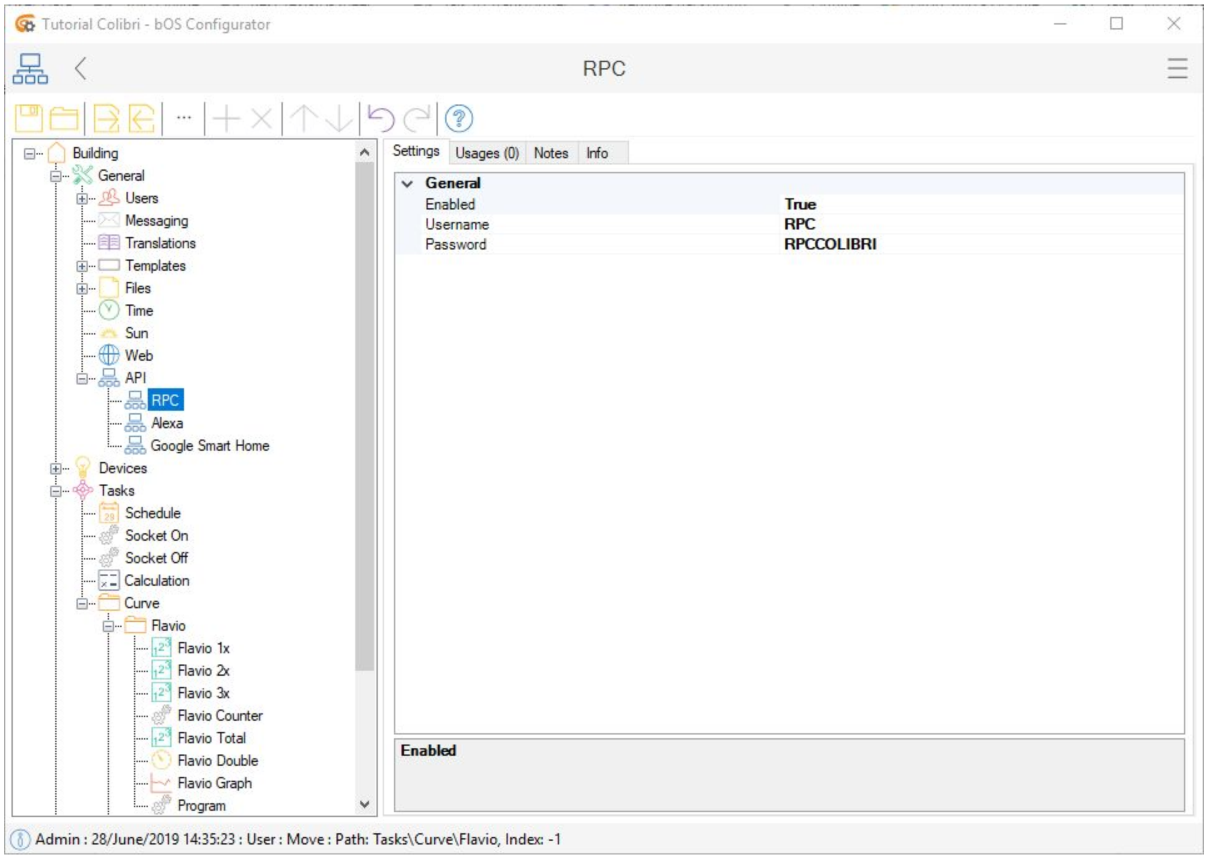
Dynamic IP ✓

Set Clear Cancel ☐ Skip service check

Update Cancel

Step 2: Configure your RPC service

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



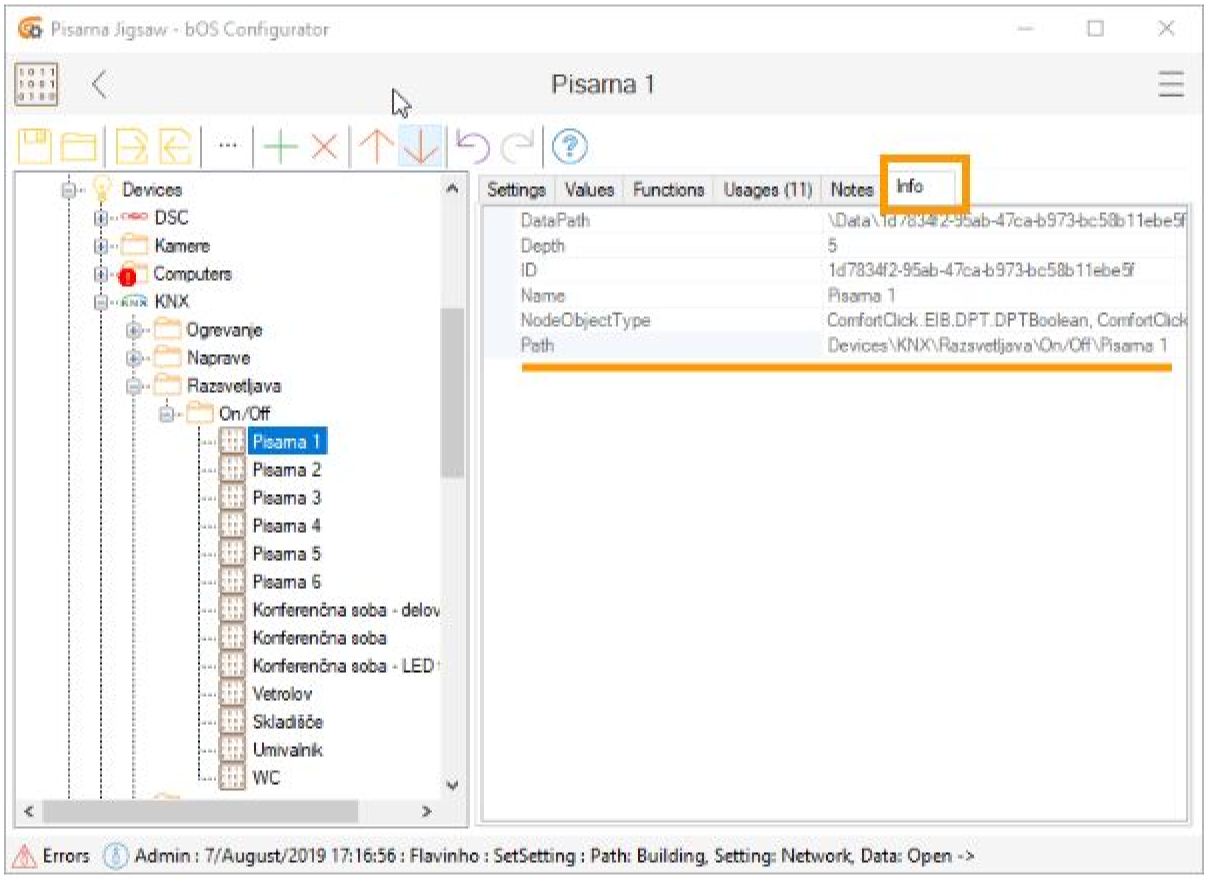
Enabled	True
Username	RPC
Password	RPCCOLIBRI

Enabled

Admin : 28/June/2019 14:35:23 : User : Move : Path: Tasks\Curve\Flavio, Index: -1

Step 3: Find your Device Path

- In this example, the *Info* tab from the KNX light will show us the path: Devices\KNX\Razsvetljava\On/Off\Pisarna 1



Pisarna Jigsaw - bOS Configurator

Pisarna 1

DataPath	\Data\1d7834f2-95ab-47ca-b973-bc58b11ebe5f
Depth	5
ID	1d7834f2-95ab-47ca-b973-bc58b11ebe5f
Name	Pisarna 1
NodeObjectType	ComfortClick.EIB.DPT.DPTBoolean, ComfortClick
Path	Devices\KNX\Razsvetljava\On/Off\Pisarna 1

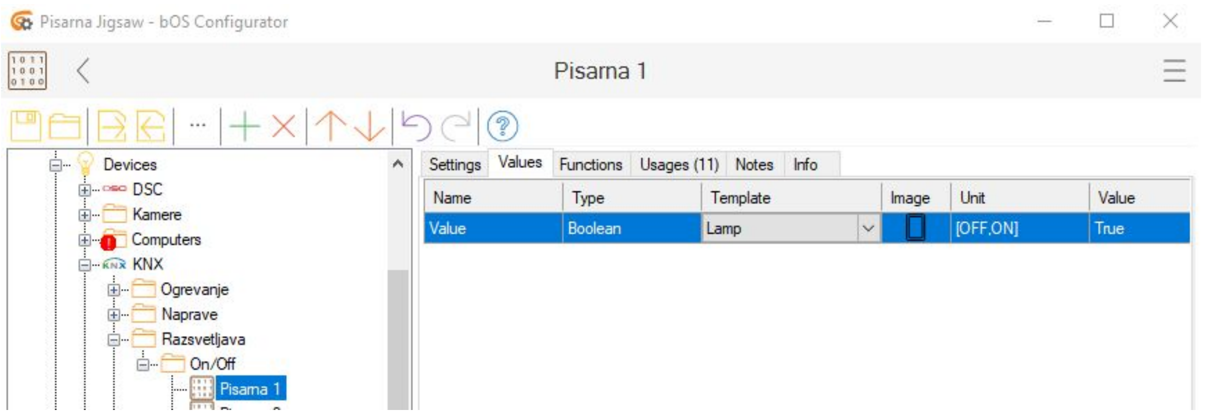
Errors   Admin : 7/August/2019 17:16:56 : Flavinho : SetSetting : Path: Building, Setting: Network, Data: Open ->

Now we need to form the JSON body to POST on IFTTT.
The Path we have is: **Devices\KNX\Razsvetljava\On/Off\Pisarna 1**

We need to double the backslashes on the Path to form the JSON body to POST on IFTTT service, so now we have: **Devices\\KNX\\Razsvetljava\\On/Off\\Pisarna 1**

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 at the end of this tutorial.

For our valueName, we have"*Value*":


Pisarna Jigsaw - bOS Configurator

Pisarna 1

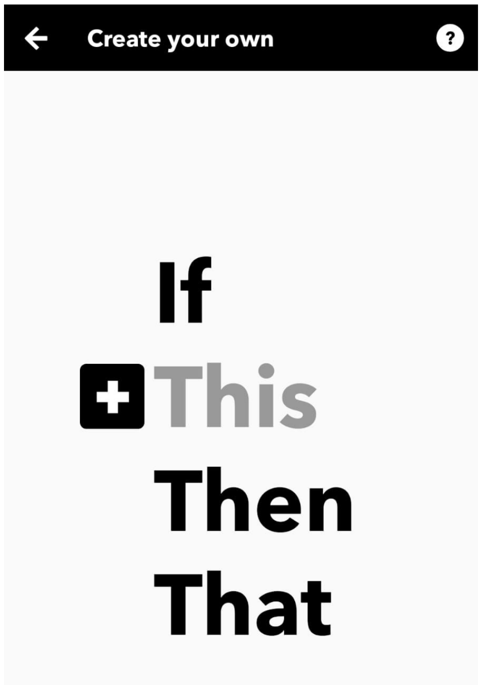
Name	Type	Template	Image	Unit	Value
Value	Boolean	Lamp		[OFF,ON]	True

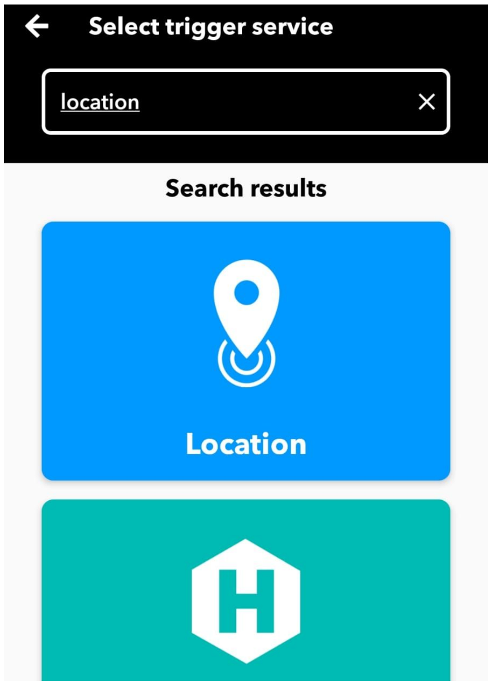
To turn the Light ON, our body post is:
{"objectName":"Devices\\KNX\\Razsvetljava\\On/Off\\Pisarna 1","valueName":"Value","value":"true"}

*the value, in this case "true", must be written in lowercase.

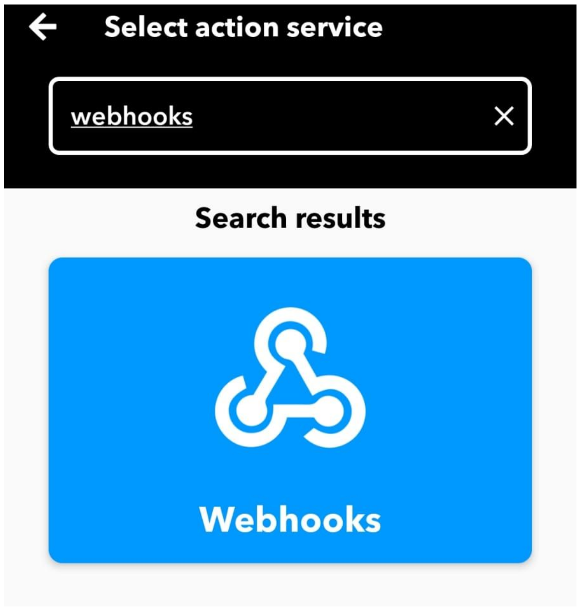
Step 4: IFTTT

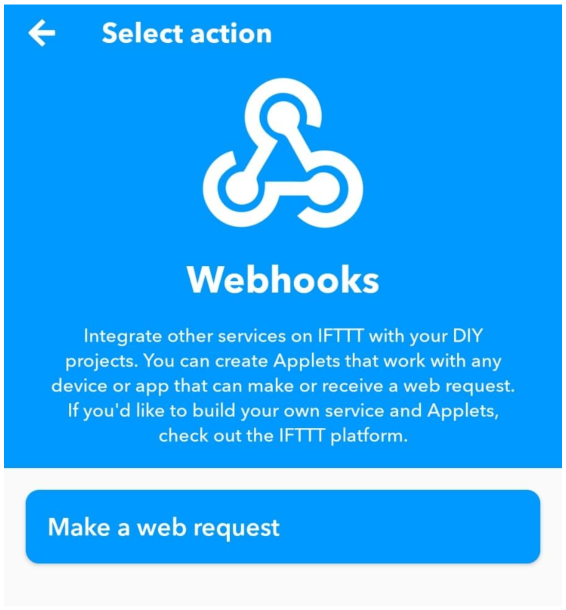
- Create a new Applets from scratch, click on "This", select the "location" trigger and set your address area.





- Click on “That”, select the “Webhooks” trigger and click on “Make a web request”:





Integrate other services on IFTTT with your DIY projects. You can create Applets that work with any device or app that can make or receive a web request. If you'd like to build your own service and Applets, check out the IFTTT platform.

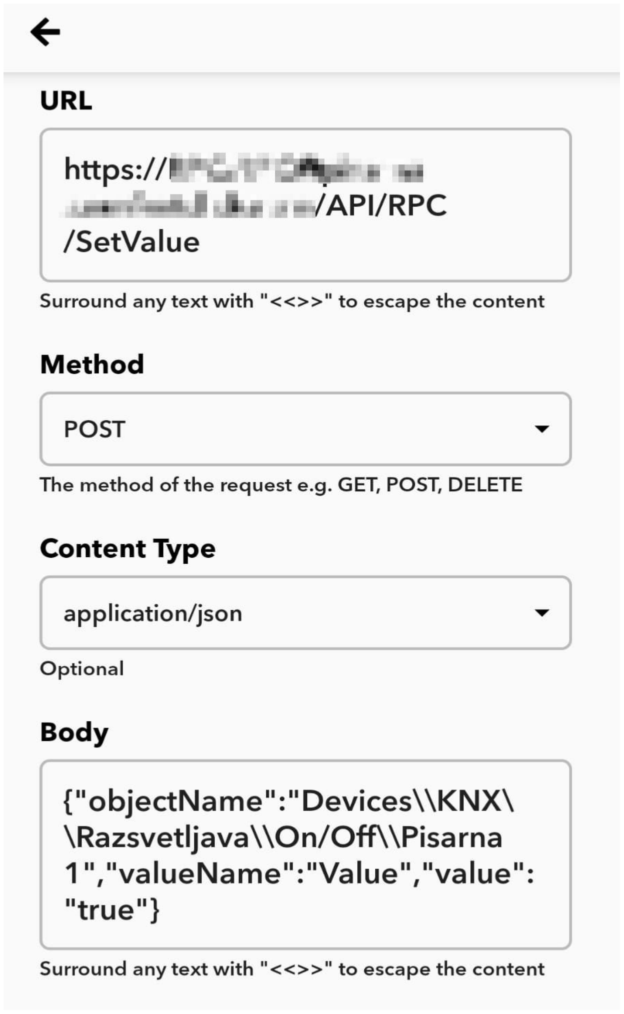
- Web request form:
  URL
  https://RPCUserName:RPCPassword@your_ip_address_or_dynamic_dns_address/API/RPC/SetValue

  in our example:
  https://RPC:RPCCOLIBRI@*ip_address_or_dynamic_dns_address*/API/RPC/SetValue

  Method
  POST

  Content Type
  application/json

  Body
  {"objectName":"Devices\\KNX\\Razsvetljava\\On/Off\\Pisarna 1","valueName":"Value","value":"true"}



URL

https://[redacted]/API/RPC/SetValue

Surround any text with "<<>>" to escape the content

Method

POST

The method of the request e.g. GET, POST, DELETE

Content Type

application/json

Optional

Body

{"objectName":"Devices\\KNX\\Razsvetljava\\On/Off\\Pisarna 1","valueName":"Value","value":"true"}

Surround any text with "<<>>" to escape the content

Save and finish your Applet.

Done! Now every time we enter the specified area the KNX light will turn on.

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
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.