
Magicblocks Documentation

Release latest

Aug 11, 2020

Contents

1	Getting started	3
1.1	Setting up magicblocks	3
1.2	Setting up a device	5
1.3	Go to the playground!	7
2	Examples	9
2.1	Digital Out	9

The Internet of things is nothing new. It is an evolutionary next step in the machine to machine communication paradigm. With an ever increasing number of ‘things’ at our disposal the need for connected ‘things’ has grown rapidly. However with increasing options comes added complexity which introduces steep learning curves and more importantly more confusion. What devices should I use? What cloud platform will best server my interest? I am new to IoT where should I start? Those are the question that would linger in any one who would dare to wonder in to the internet of Things. We, at magicblocks.io have put some magic to work to make the life of both the newcomer and the professional one step easier.

magicblocks.io is a launchpad for learning and prototyping your internet of things. It consists of

- **Hardware suite made up of**
 - Development boards
 - Prototyping kits for sensors & actuators
- **Cloud platform made up of**
 - Drag and drop editor to easily cook up your solutions
 - Dashboards to visualize your data
 - Data storage

Everything has been designed to make the learning curve as shallow as possible for the newcomer and as flexible as possible for the advanced user.

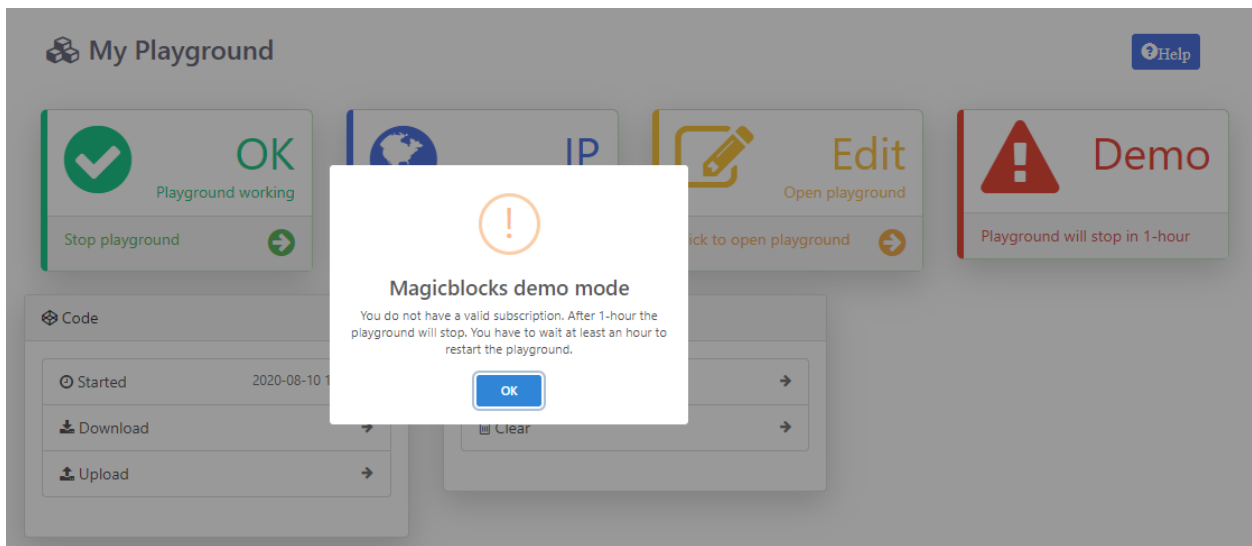
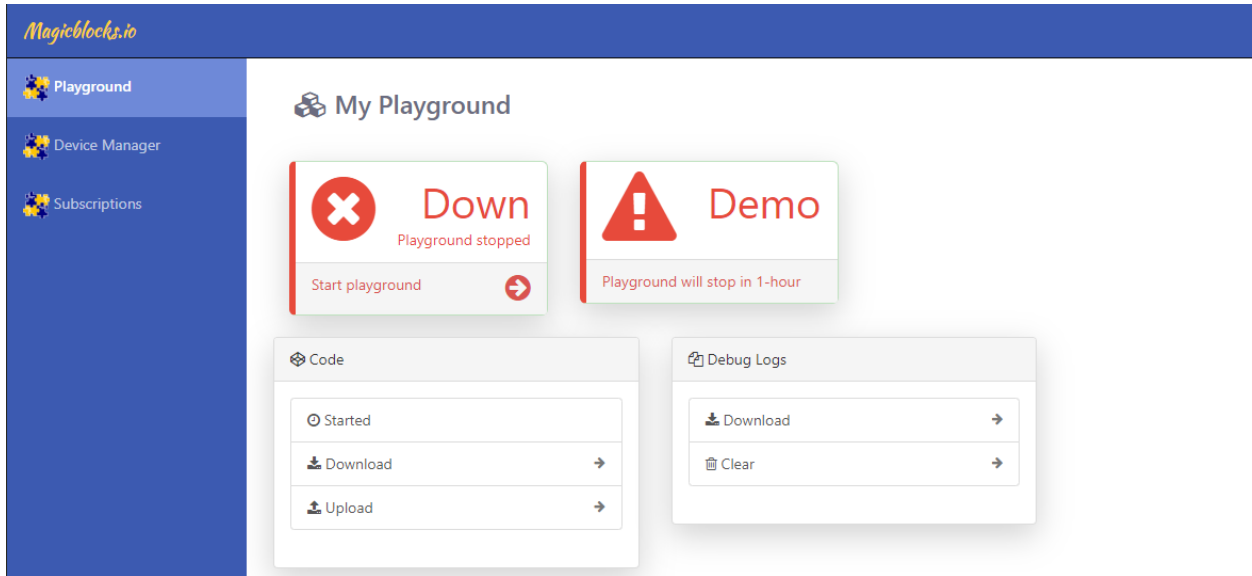
1.1 Setting up magicblocks

1.1.1 Sign Up & Login

- Visit magicblocks.io and click signup at the top.
- After filling the details and submitting you will get the activation link via email. Click on the link to activate the account.
- Once done, you can login to your account from magicblocks.io

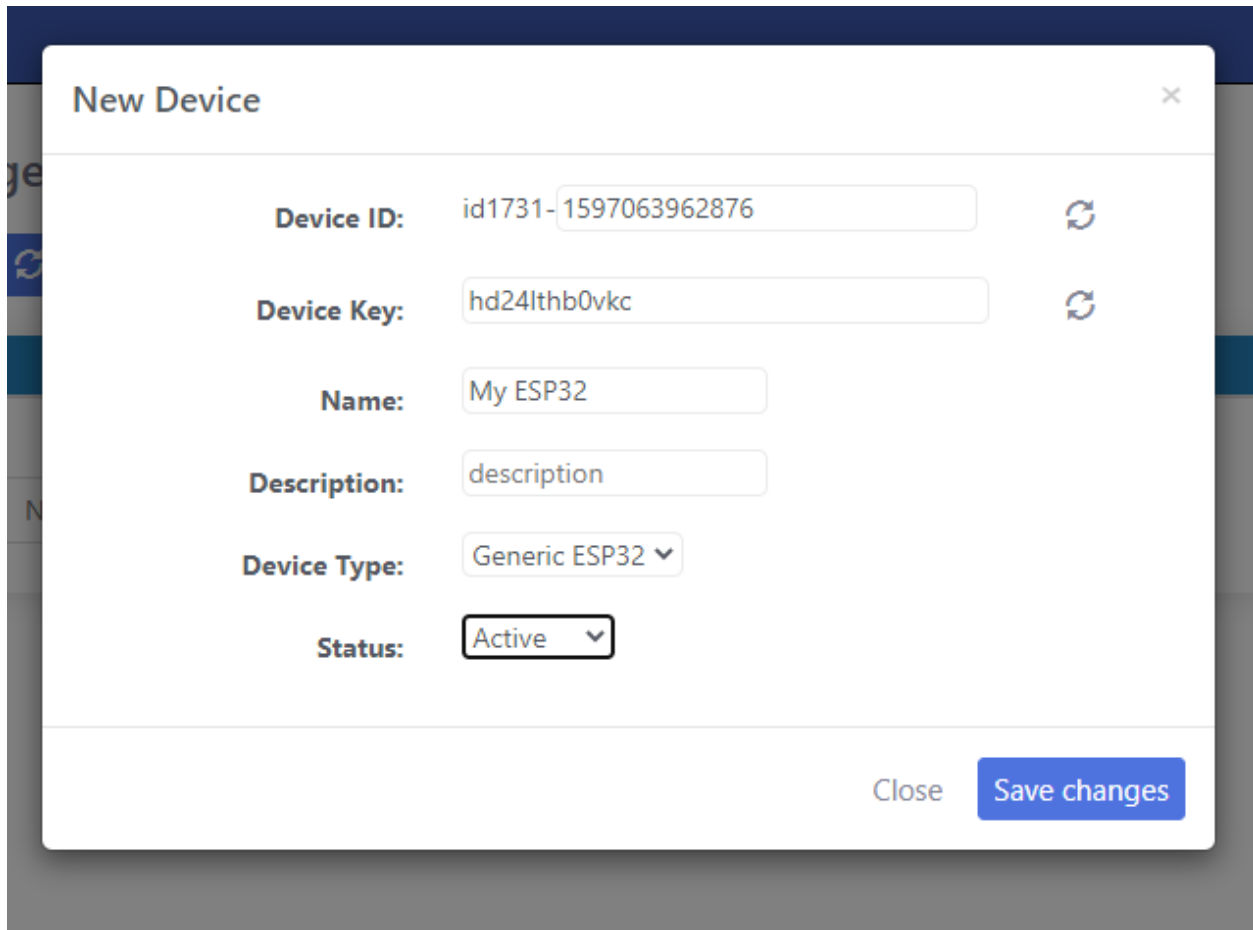
1.1.2 Start the Playground

When you login for the first time your playground will not be running. Playground is the visual programming environment based on Node-Red that has been customized for seamless integration with hardware devices to enable IoT. If you do not have a valid subscription, you will be allowed to run the playground only for 1 hour continuously before it is automatically stopped. You will need to restart the playground manually after this 1 hour period



1.1.3 Create a new device

Go to Device Manager and add a new generic ESP32 device



New Device ✕

Device ID: id1731-1597063962876 ↻

Device Key: hd24lthb0vkc ↻

Name: My ESP32

Description: description

Device Type: Generic ESP32 ▾

Status: Active ▾

Close Save changes

Keep this browser tab open since you will need to copy the device Id and the key to setup the device in the next step.

1.2 Setting up a device

1.2.1 ESP32 based development boards

- Install Arduino core for ESP32. Follow the instructions on [official arduino core for ESP3](#)
- Clone the [ESP32-Magicblocks](#) library to your Arduino libraries folder
- Create a new project and past the code below in the .ino file:

```
#include "ESP32_MB_Core.h"

ESP32_MB_Core device;

void setup() {
  device.init();
}

void onCustomPayload(char* payload, int length) {
  Serial.println("Payload received!");
  Serial.println(payload);
}
```

(continues on next page)

(continued from previous page)

```
void loop() {  
  device.loop();  
}
```

This is the bare minimum sketch that should be uploaded for the device to connect to the platform

- Upload to your ESP32 board
- Reset the ESP32 board and wait for search for its configuration WiFi access point to come up. It would be named “ESP32-MB” in the default configuration
- Connect to this WiFi access point and go to <http://192.168.4.1>

Setup Admin

Magicblocks Settings

Status: **Connected**

Name:

DeviceId:

Key:

WiFi Settings

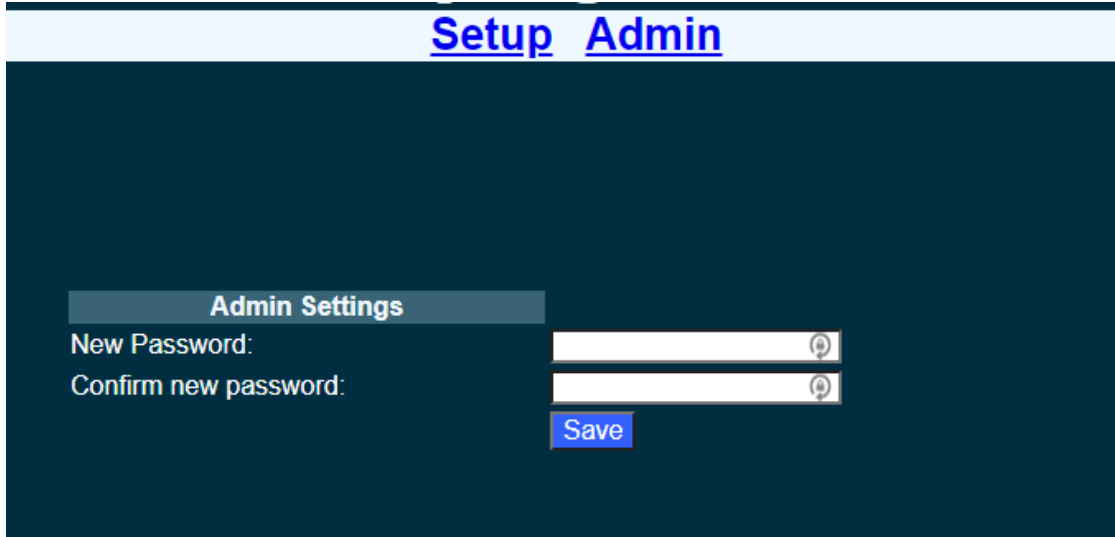
Status: **Connected to Aka Home**

SSID:

Password:

Save

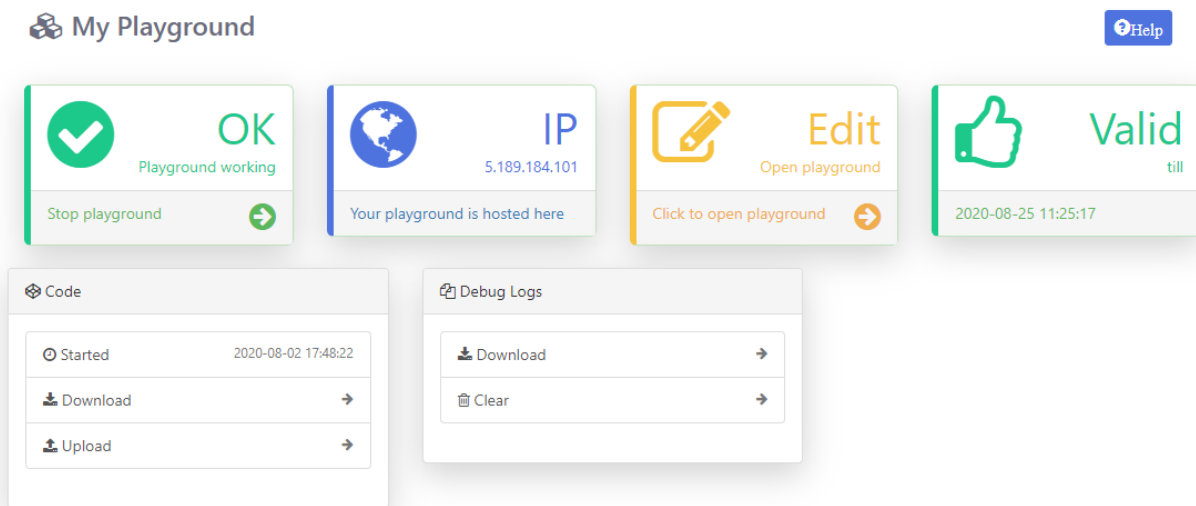
- Fill in the device Id and the key that was generated in the previous step. You can change the **Name** to any name you like up to 32 characters. Select the WiFi SSID from the drop down and enter the WiFi password. If your WiFi is not listed here make sure the WiFi AP is active and reset the ESP32 board and retry.
- Save the configuration. The default admin password is 12345. You can change this from the **Admin** tab



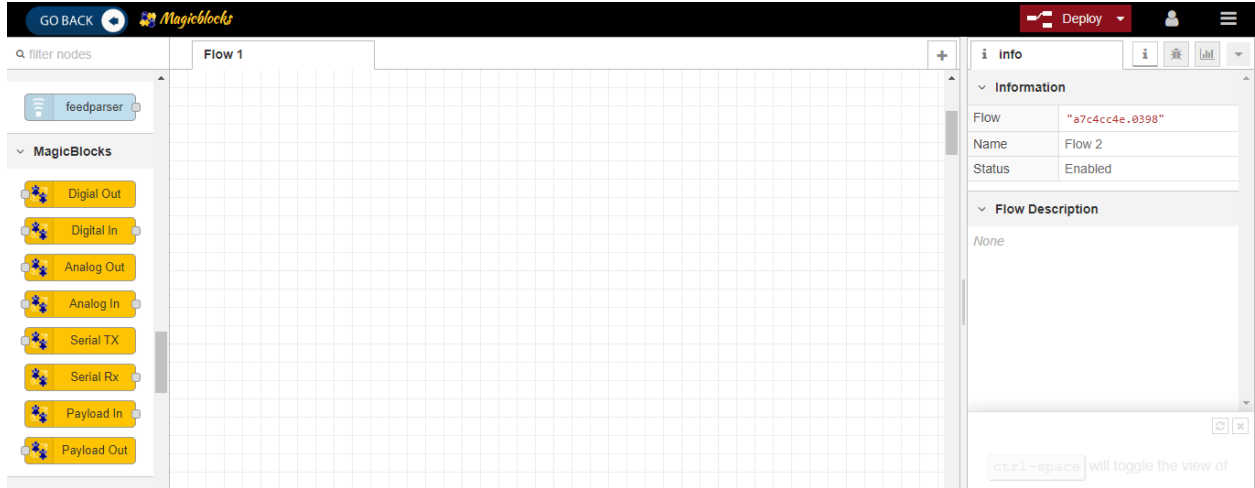
- Once saved, the ESP32 will restart and try to connect to the WiFi and then to magicblocks. You can connect to the ESP32's configuration WiFi AP and navigate to <http://192.168.4.1> to check the status of the connection.
- Login to magicblocks, navigate to **Device Manager** and check if the **Connection** column comes up as a blinking green indicator. If yes, you can proceed to the next step

1.3 Go to the playground!

- Login to magicblocks and click on **Edit** to open the playground.



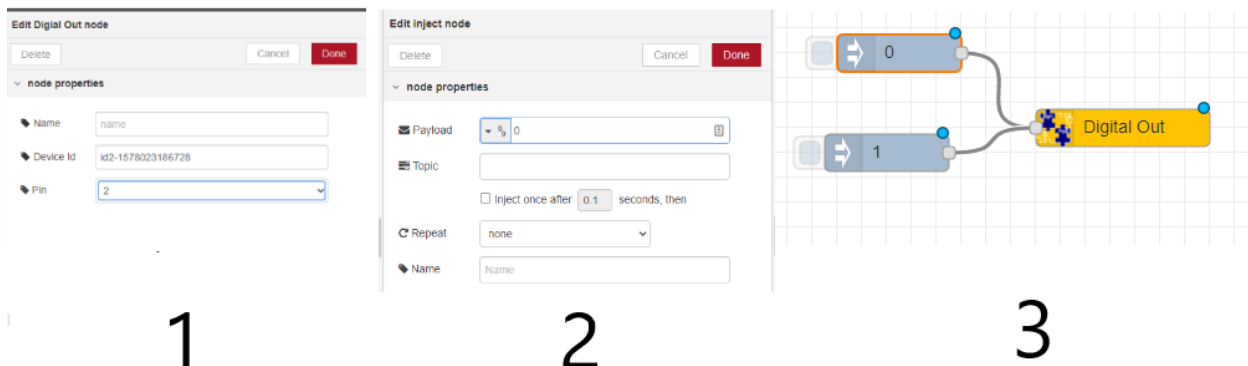
- The playground is a hosted instance of a customized version of the open source Node-Red application. On the left you will find the palette where all the blocks will reside. You can drag and drop the blocks from the palette to the canvas and start rolling very quickly. Once done click the **Deploy** button on the top right corner and all your changes will be saved. In the next section we will go through some examples covering all the relevant blocks



2.1 Digital Out

2.1.1 Using inject nodes

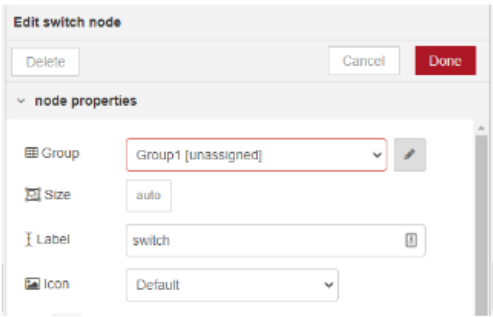
Controlling a pin in digital out mode is the extremely easy. - Open the Device Manager in your magicblocks account - Turn on your ESP32 board that was setup in the previous section and wait for it to connect - Copy the device Id of this ESP32 board - Open the playground - Drag and drop the **Digital Out** block under the category **Magicblocks** - Double click on the block and paste the device Id - Select the pin number you want to control. eg: pin 2 for inbuilt LED - Drag and drop 2 **inject** blocks under the **input** category and connect them to the **Digital Out** block - In one **inject** node set the value you want to inject to 1 and set the value to 0 in the other one - Click deploy - Now the you can toggle the pin by clicking the inject nodes.



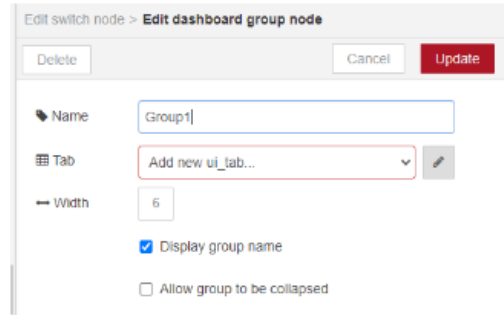
2.1.2 Using UI widgets

To make things more interesting you can control the inbuilt LED via the web browser. Thanks to the [node-red-dASHBOARD](#) extension this is just a matter of adding a new block.

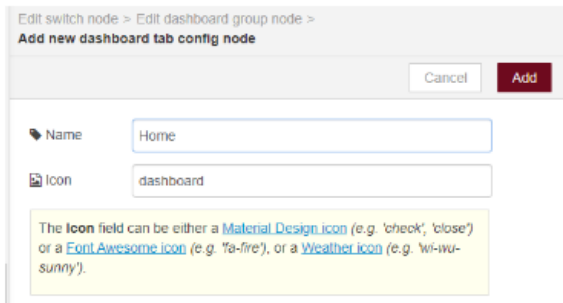
- Drag and drop a **switch** block from the **dashboard** category.
- Create a new UI Group
- Create a new UI tab in the group
- Connect the switch block to the Digital Out block
- Click deploy
- Open the link to the web UI
- Click on the switch and the LED should turn on and off



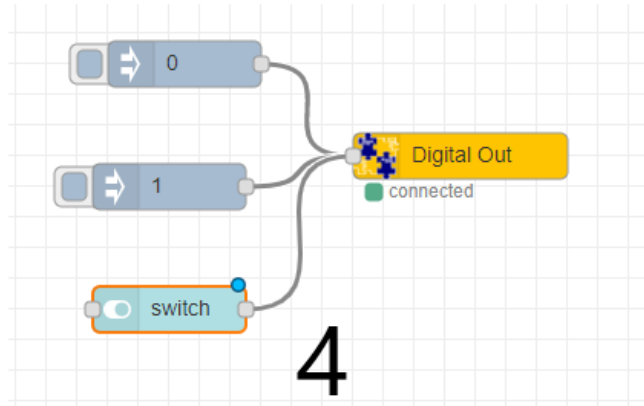
1



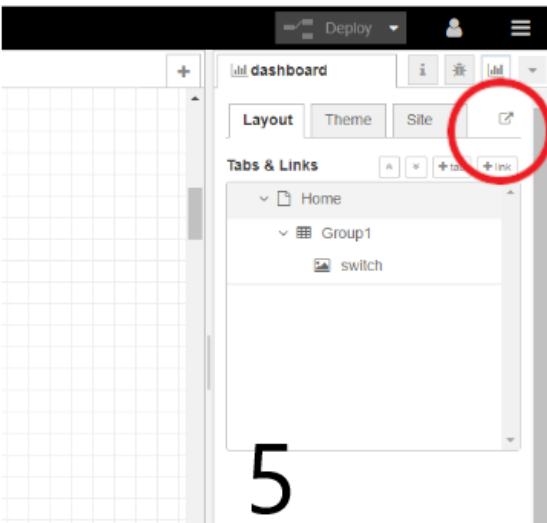
2



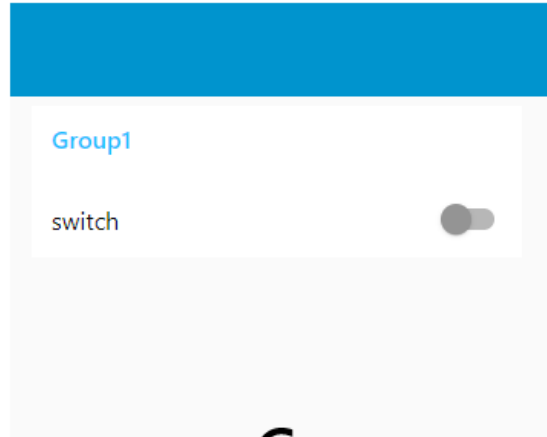
3



4



5



6