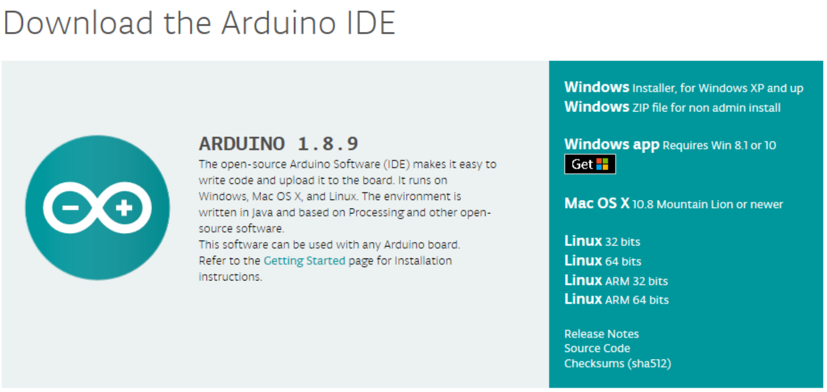
A - Pre-requisites for the IoT 101 Workshop

Welcome to IoT 101 workshop. Before getting to the actual training you have to setup your machine and install drivers, libraries and all necessary software.

1. Install Arduino IDE

All programming on the Node-MCU board is made using Arduino IDE. If you already have it installed, you may skip this section.

1. Go to www.arduino.cc then click Software/Downloads and Download Arduino IDE



Download the Arduino IDE

ARDUINO 1.8.9

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for Installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
Get

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Release Notes
Source Code
Checksums (sha512)

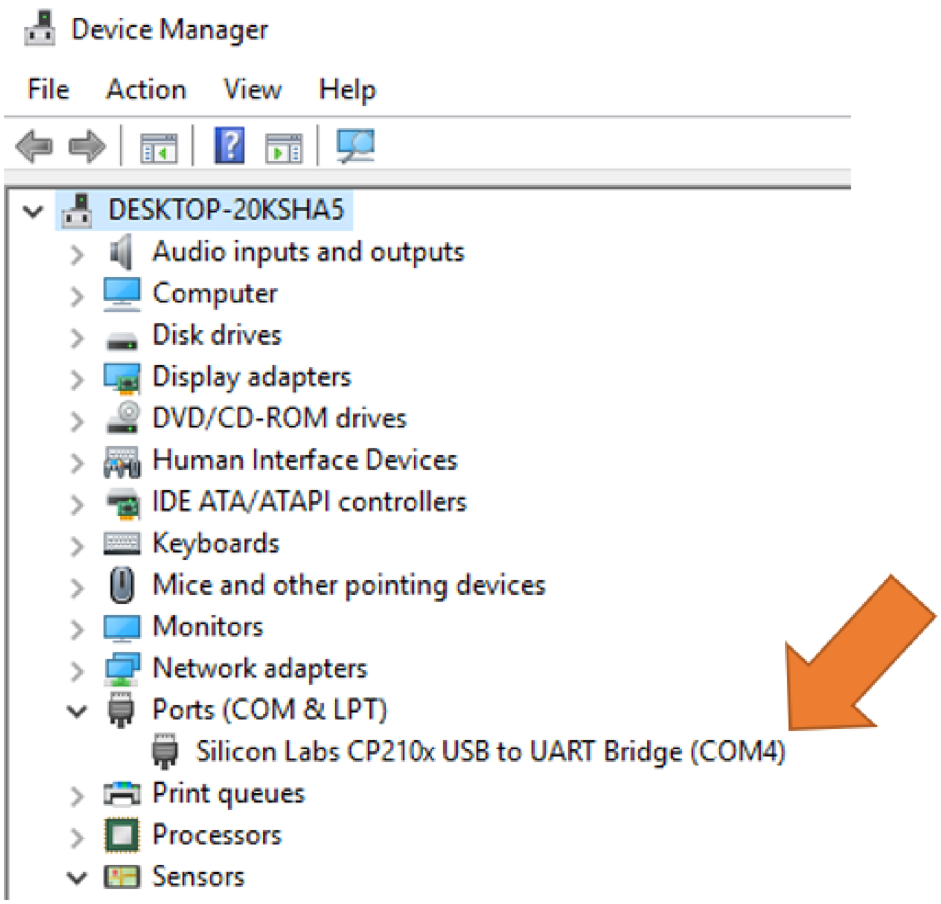
2. Execute and follow the instructions (basically agree and click next)
3. At the end of the installation, you'll be prompted to allow the installation of several drivers. Accept them all as these drivers are used to allow the serial communication between the computer and the board.
4. Once you're finished with Arduino setup, you may proceed to the next step.

2. Connect the board to your computer for a quick check

A. Connect the board to any available USB port

B. Open the Device manager (by typing “device manager” in the windows search box) or right-clicking the windows button and selecting “device manager”

C. Check if you see a **Silicon Labs CP210x USB to UART Bridge** device, as shown in the next picture.



DESKTOP-20KSHA5
- Audio inputs and outputs
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
  - Silicon Labs CP210x USB to UART Bridge (COM4)
- Print queues
- Processors
- Sensors

D. In the picture above, you can see the device got COM4 assigned to it. Make sure your device has a COM port showing as well.

E. If you can’t see any USB to UART Bridge in your device manager under Ports, please reboot your computer. Sometimes a reboot is required so Windows can recognize the driver you just installed.

F. In some rare cases, it is required to remove Arduino IDE and reinstall it to the latest version.

G. If you can’t still see it even after a reboot, please let us know so we can assist you.

Attention Mac users: Silicon Lab driver for the board is not installed on the default load. The driver can be found at: https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers.

3. Install support for the ESP8266 in the Arduino IDE

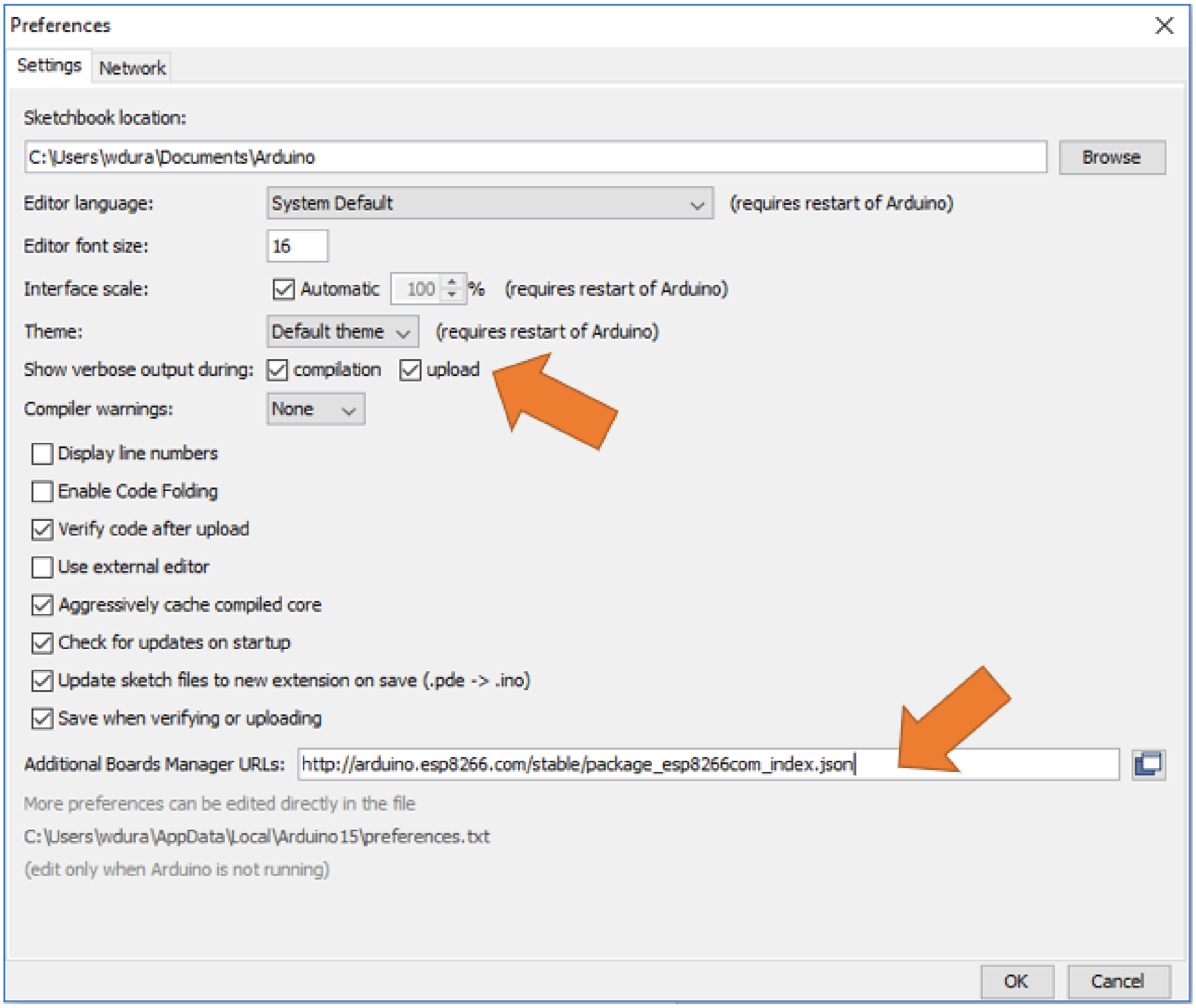
While the Arduino IDE supports most official and unofficial Arduino boards, the ESP8266 / node MCU boards are not supported by default. In this step you'll install the additional information to enable it.

Make sure you have internet connection before starting this process. You won't be able to install this board support with no Internet access.

1. Open Arduino IDE
2. Open File/Preferences
3. Find a text box called "Additional board manager URLs and type:

```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

4. Under the "Show verbose output during" configuration mark the checkboxes "compilation" and "upload". This is important as this will give us clear error messages, we can use to identify future problems.
5. Click OK



Sketchbook location: C:\Users\wdura\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 16

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: ☑ compilation ☑ upload

Compiler warnings: None

☐ Display line numbers

☐ Enable Code Folding

☑ Verify code after upload

☐ Use external editor

☑ Aggressively cache compiled core

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

☑ Save when verifying or uploading

Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

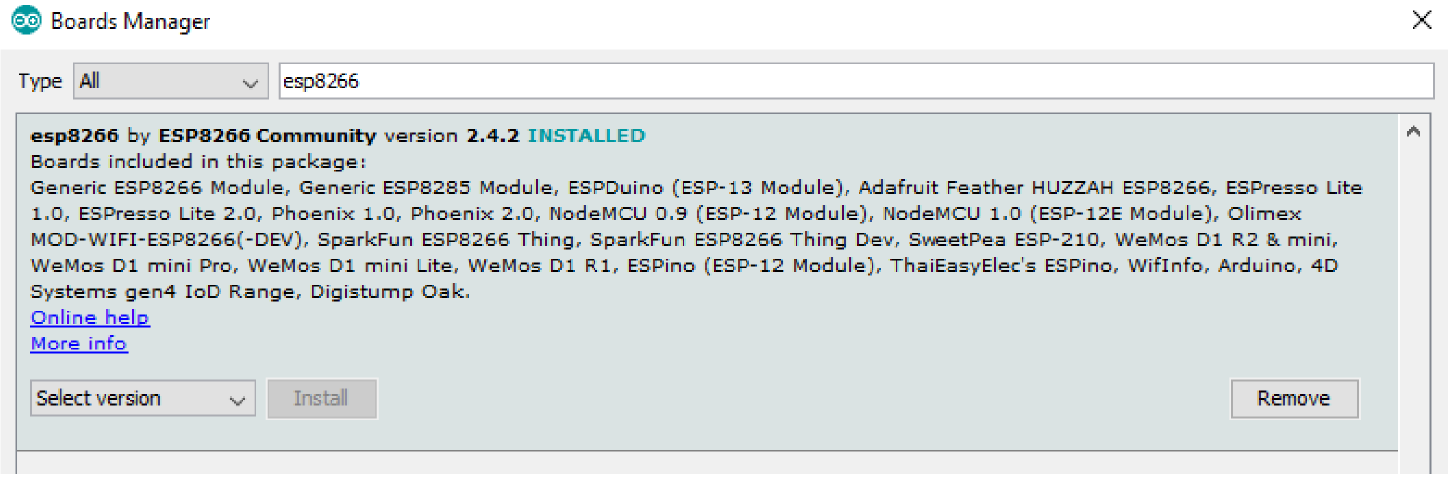
More preferences can be edited directly in the file

C:\Users\wdura\AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)

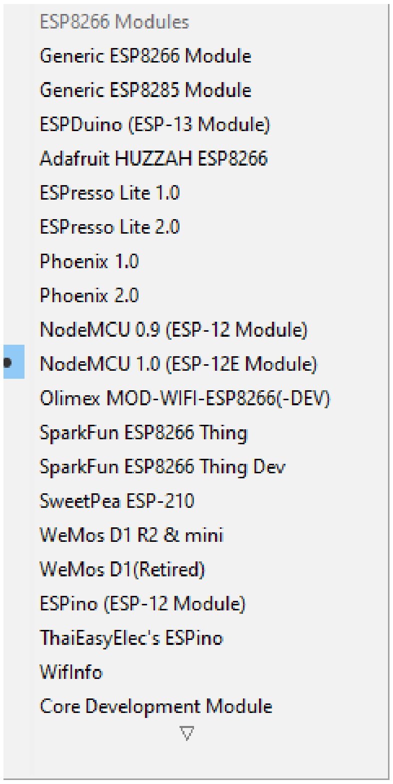
6. Now go to menu Tools/Board/Boards Manager (as the picture below shows)

7. Type ESP8266 in the search box and wait until the board manager finds “esp8266 by ESP8266 Community”
8. Click on it
9. Select the latest version (as of October 2018, latest is V2.4.2)
10. Click “Install”
11. Be patient, this process can take several minutes underower connections



esp8266 by **ESP8266 Community** version **2.4.2** **INSTALLED**
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, WeMos D1 R2 & mini, WeMos D1 mini Pro, WeMos D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak.
Online help
More info

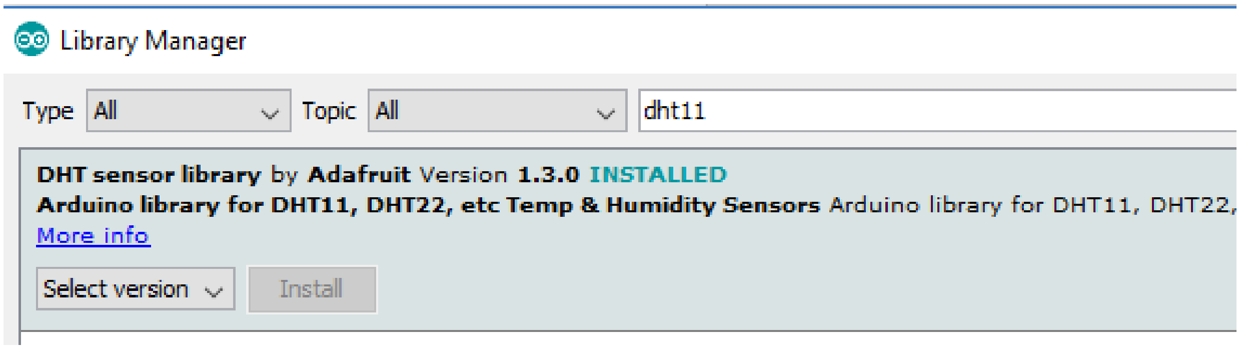
12. Once this process is complete, make sure you have the boards installed by clicking Tools/Board/ and see if you can find Generic ESP8266 and NodeMCU 1.0 (ESP-12E Module). You might have to scroll down the list to find it.
13. If you find NodeMCU 1.0 (ESP-12E Module), please go ahead and select it. This is the board we’ll be using throughout this workshop.



4. Install Arduino Libraries

This workshop will require several libraries that are not native to Arduino IDE, so we'll have to install them.

1. On the Arduino IDE go to menu Sketch/Include Library/Manage Libraries
2. Type **DHT11**, wait to see the results, click on **DHT Sensor Library by Adafruit**, select the highest version in the "Select Version" dropdown list, and then click Install.



DHT sensor library by **Adafruit** Version **1.3.0** INSTALLED
Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22,
More info

3. Follow the same procedure to install 3 additional libraries:

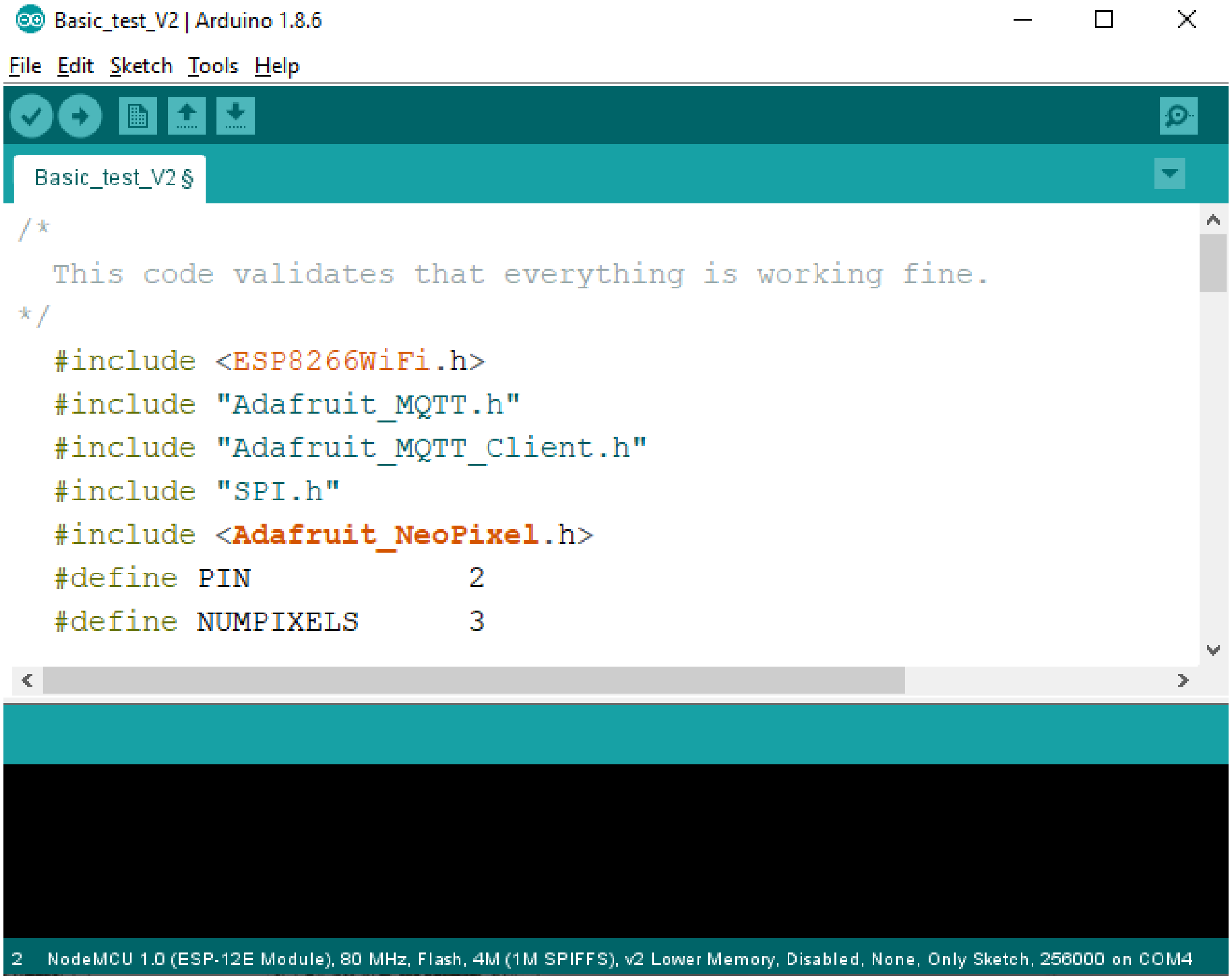
What to type	Library name
MQTT	Adafruit MQTT Library by Adafruit
Neopixel	Adafruit Neopixel by Adafruit
Adafruit unified sensor	Adafruit unified sensor by Adafruit

4. Click Close.

5. Basic Test

It is time now to test the board connectivity to your computer:

1. Navigate to https://github.com/oneweekiot/iot101 select Code, Basic_test_V2, Basic_test_V2.ino, and copy its content to an empty file on Arduino IDE.
2. Open a new Sketch in the Arduino IDE and paste the code you just copied.
3. Make sure NodeMCU 1.0 board and the correct COM port are showing in the bottom right part of the Arduino IDE (as the picture below shows)



```
/*
  This code validates that everything is working fine.
*/
  #include <ESP8266WiFi.h>
  #include "Adafruit_MQTT.h"
  #include "Adafruit_MQTT_Client.h"
  #include "SPI.h"
  #include <Adafruit_NeoPixel.h>
  #define PIN             2
  #define NUMPIXELS       3
```

2   NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 256000 on COM4

4. Select menu Sketch/Upload or simply press CTRL+U.
5. Wait for the compilation.
6. Last message we expect to see after compiling and uploading successfully is “flush complete”.
7. Press button 1 on the OneWeekIoT shield, and the 3 NeoPixels on the board should light up and sounds will play on the speaker. Press button 2 for a different tone and different colors on the NeoPixels.
8. If this was the case, congratulations, you got all libraries needed for this workshop and now you’re all set.

6. Creating an Adafruit Account

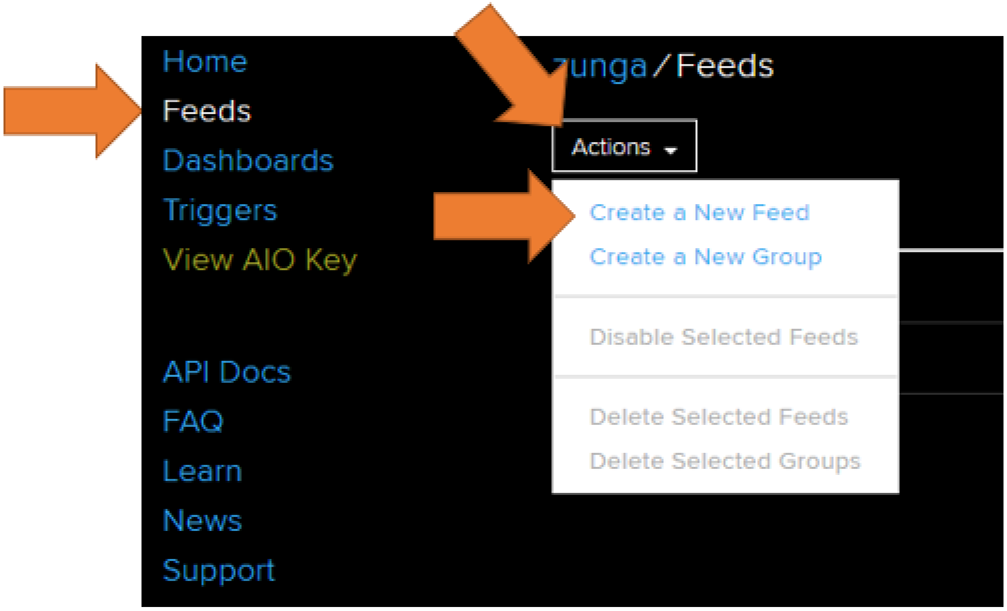
This workshop requires an online MQTT broker. While several professional options exist, we'll be using Adafruit IO given its simplicity and continuous support of the maker community.

1. Visit http://io.adafruit.com and click "Get Started for free" – don't worry, it's REALLY free.
2. The free account provides you with (information as of Oct 2018):
   a. Up to 10 feeds
   b. Up to 5 Dashboards
   c. Storage for 30 days
   d. Rate: 30 transactions per minute
3. **IMPORTANT**: after creating your account, log out, visit https://io.adafruit.com again so you'll be able to get to the Dashboard.

7. Creating Feeds and the Dashboard

We need to create 5 new feeds:

1. To create them click feeds, then click in the Action menu and select New Feed. All you need to provide is a name and an [optional] description.

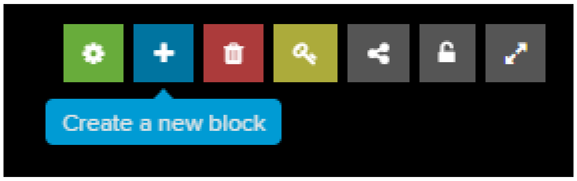


2. Feeds to create:
    - **Temperature**
    - **Humidity**
    - **Light**
    - **Neopixel**
    - **Slider**

3. Once you have the feeds, you need to create a Dashboard to show them. Click in the Dashboards (left menu), then Actions, create a new dashboard.
4. Name your dashboard **IoT-101**
5. Select your new dashboard IoT-101 to navigate to it.

8. Configuring the Dashboard

You'll get a blank dashboard. Now you will need add new blocks. Click the blue button with a plus sign.



We're creating 6 new blocks:

1. **Stream**: for this block, select feed named "temperature", click "next step", set the block title as "Temperature", then click "Create block"

2. **Stream**: for this block, select the feed named "humidity", click "next step", set the block title as "Humidity", then click "Create block"

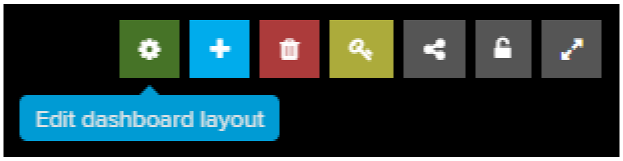
3. **Stream**: for this block, select the feed named "light", click "next step", set the block title as "Light", then click "Create block"

4. **Line chart**: for this block, select the feed named "temperature", then select another feed named "humidity", click "next step", set the block title as "Temperature and Humidity" then click "Create block"

5. **Slider**: for this block, select the feed named "slider", click "next step", set slider max value to "1000", set the block title as "Slider", then click "Create block"

6. **Color Picker**: for this block, select the feed named "neopixel", click "next step", set the block title as "NeoPixel", then click "Create block"

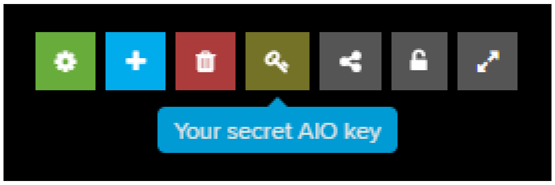
While creating dashboards, blocks are placed randomly. Once you have created all block, you can rearrange blocks, resize, reconfigure and delete them.



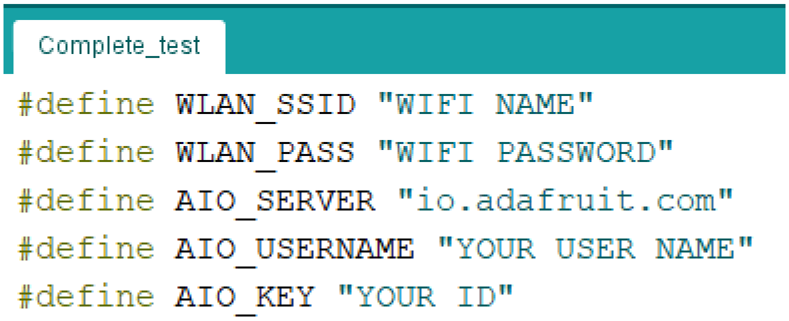
9. Updating credentials

In this step we'll get and store MQTT credential in the code, so the device can authenticate and connect.

1. In the top menu, click in the yellow button with a key:



2. Copy both username and the key.
3. Navigate to https://github.com/oneweekiot/iot101 select Code, Complete_test, Complete_test.ino, and copy its content to an empty file on Arduino IDE.
4. Find the #define declarations and update both AIO_USERNAME and AIO_KEY with the values from your Adafruit IO account.



```
#define WLAN_SSID "WIFI NAME"
#define WLAN_PASS "WIFI PASSWORD"
#define AIO_SERVER "io.adafruit.com"
#define AIO_USERNAME "YOUR USER NAME"
#define AIO_KEY "YOUR ID"
```

5. Update WLAN_SSID with your Wi-Fi network name and WLAN_PASS with your password.
6. Make sure NodeMCU board and the correct COM port are showing in the bottom right part of the Arduino IDE.
7. Select menu Sketch/Upload or simply press CTRL+U.
8. Wait for the compilation.
9. Last message we expect to see after compiling and uploading successfully is "flush complete".
10. Board will connect to the Wi-Fi, sound a beep, connect to the MQTT service and sound another beep.
11. If this was the case, watch for data showing up in your dashboard.