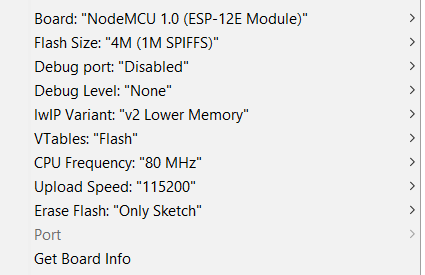
ESP-NFC-FIREBASE INTERFACE

# Pre-requisites:

1. Arduino
2. Arduino packages for ESP
3. NFC PN532
4. FireBase Database System

# Arduino Installation for ESP8266:

1. Install the latest Arduino from the [Arduino website](https://www.arduino.cc/en/main/software)
2. Start Arduino and open *Preferences* window.
3. Enter https://arduino.esp8266.com/stable/package\_esp8266com\_index.json into *Additional Board Manager URLs* field. You can add multiple URLs, separating them with commas.
4. Open *Boards Manager* from *Tools > Board* menu and install *esp8266* platform (and select your ESP8266 board from *Tools > Board* menu after installation).
5. This application uses ESP8266 *version 1.0* board. The *Board* in *Tools* menu should look like the figure below.

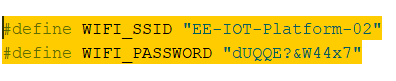


**Important ESP libraries:**

1. ESP8266WiFi.h
2. WiFiUdp.h
3. WiFiClient.h
4. WiFiManager.h //https://github.com/tzapu/WiFiManager

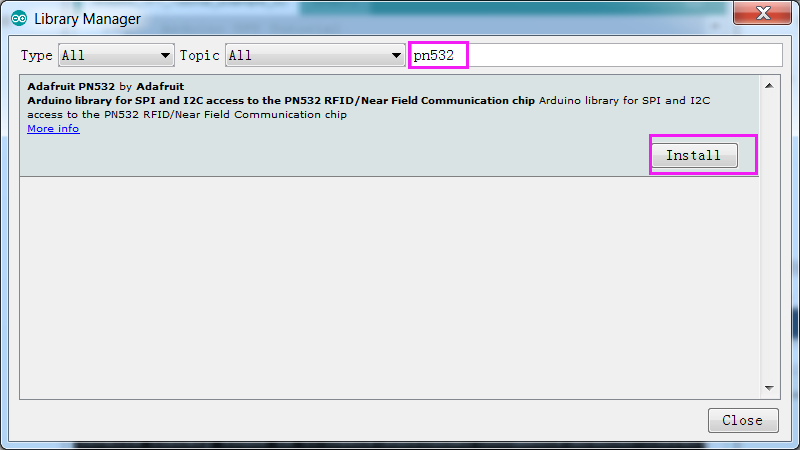
**Important changes to be done to the code:**

ESP connects to WiFi, thus the SSID and Password of the WiFi connection needs to be defined. Make sure to change these below 2 fields when the WiFi changes.



NFC PN532:

In Arduino IDE, select *Sketch > Including Library > Manage Libraries*, type in *PN532* to search. Then we can see *Adafruit PN532* searched out, and click **INSTALL** at the right side to start the installation.



Download the libraries for PN532 from the github *InventionworksIOT.* The libraries needed specifically for this application are PN532, PN532\_SPI. Put these folders into the *libraries* folder inside *Arduino* folder.

**Folder structure:**

*Arduino > libraries > PN532*

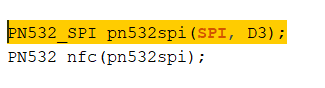
*PN532\_SPI*

<https://github.com/elechouse/PN532>

<https://github.com/adafruit/Adafruit-PN532>

**PN532 hardware connection to ESP8266:**

The PN532 board uses SPI connection to connect to ESP8266. SPI contains 4 pins: SCK, MOSI, MISO, SS. The first 3 pins are fixed, while the SS pin can be any GPIO pin as chosen in the code.



Make the connections as follows:

**PN532 NodeMCU**

GND GND

3.3V 3.3V

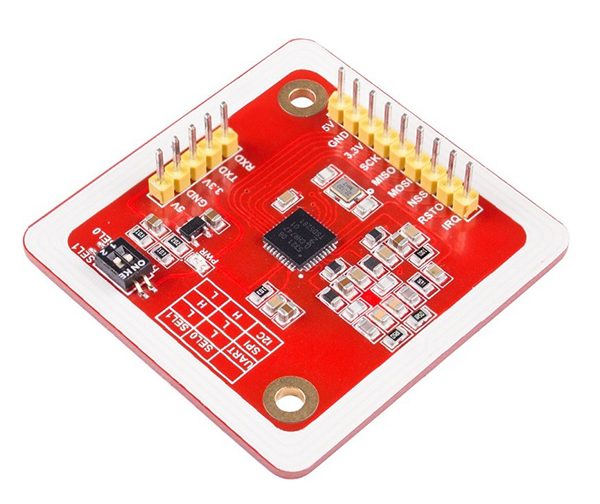
SCK D5

MISO D6

MOSI D7

SS D3

**NFC PN532 board and RFID tags**



**PN532 board RFID tags**

**Write tag:**

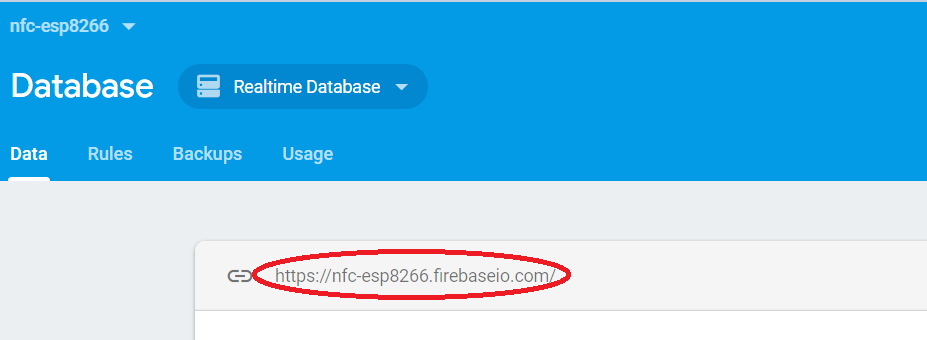
Use the Arduino project *nfc\_esp8266\_writetag* to write to the RFID tags. Each of the RFID tags are given names for example, Mark, Akhila, Sarnab, and Yash. This needs to be done only once (as this behaves as identity of the card and the person using the card, for the time logging system).

Firebase Database Management System:

1. Download and install [firebase-arduino-master](https://drive.google.com/file/d/0B4tww7CJlZPyUTd1enFxNklKOE0/view) library in Arduino IDE
2. Create Firebase project by going to Firebase Console (need gmail account to create project)
3. Click on Database from the *Firebase Develop* menu on the left and you will see the host name as shown in the image

Copy that host name and paste it in the Arduino code at line

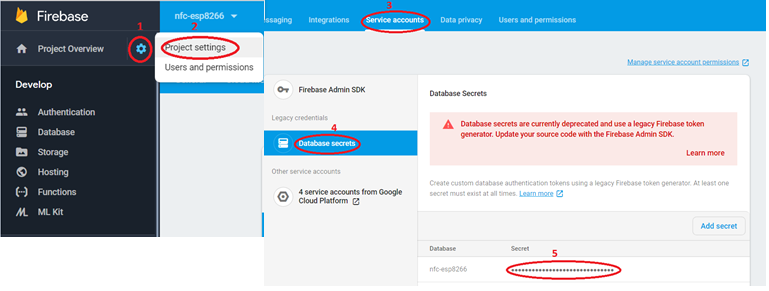
*#define FIREBASE\_HOST "nfc-esp8266.firebaseio.com"*



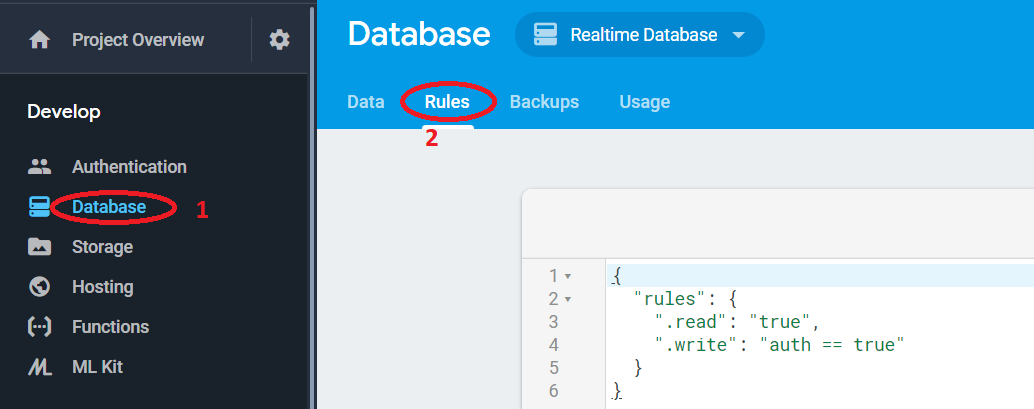
1. Go to *Settings > Project Settings > Service Accounts > Database Secrets*

Copy “*Database Secrets”* as shown and paste it at line

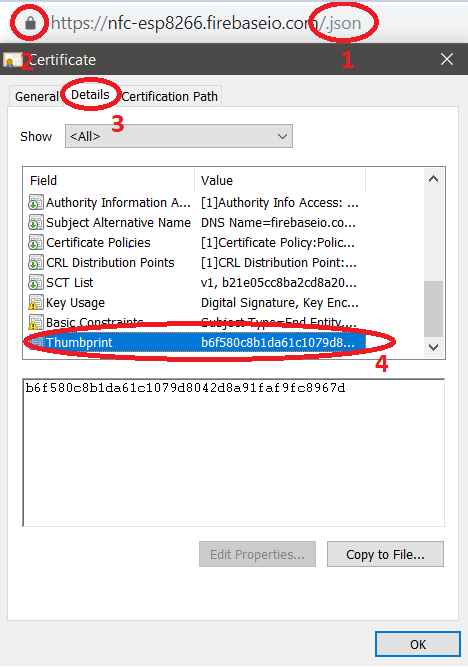
*#define FIREBASE\_AUTH "7Hexampleexampleexample…”*



1. Go to *Database* in Firebase Develop menu > *Rules.* Make sure the rules are set as shown in the below figure



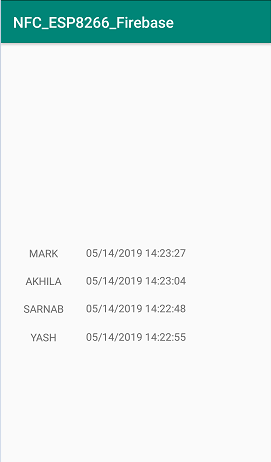
1. Just by doing the above steps and running the ESP code will result in lots of "pushing failed:..." This is due to a **Firebase certificate fingerprint** that had changed. We can find this latter by typing our project URL appended by /.json in a browser and hit the lock icon to access certificate details. On the other side, in our FirebaseArduino library folder firebase-arduino-master/src, there is a file called FirebaseHttpClient.h. Inside it, there is a line where the fingerprint should be declared (hexadecimal digits by pairs). It has to match the one seen in the certificate details. That was no longer the case!





Android Studio:

The AndroidStudio project is in folder *AndroidStudioProjects.* Download this project from github and build this to form the app which is as shown below



Put it all together:

1. Download the Arduino project *nfc\_esp8266.ino* from github *InventionworksIOT > ESP\_NFC > src*
2. Program the ESP
3. Open *Tools > Serial Monitor.* Make sure to select the appropriate baud rate (the code in github uses 115200)
4. The *Serial Monitor* shows all the serial prints used in code, and can be used for debugging the ESP and PN532
5. When the RFID tags are placed near the PN532, the serial monitor shows the name corresponding to each tag and Firebase gets automatically updated and this is read by the Android app and the latest time logged is shown in the app