DAY 2

# Internet Of Things

# Hands on Workshop

# IoT Platforms

- Blynk IoT
- ThingSpeak API
- Arduino IoT Cloud

# Access all Code (IoT Platforms)

shorturl.at/rvyR0

# Arduino IoT Cloud

Arduino IoT Cloud is an application that helps makers build connected objects in a quick, easy and secure way. You can connect multiple devices to each other and allow them to exchange real-time data. You can also monitor them from anywhere using a simple user interface.

Arduino has taken a step forward to keep IoT as simple and reachable as possible. Initially, IoT Cloud was available only to the Arduino Boards, but now, it even provides Web Sketches for other third-party boards like ESP8266 / 32, LoRaWAN as well.

# Arduino IoT Cloud

https://docs.arduino.cc/cloud/iot-cloud/tutorials/esp-32-cloud

# What is an API?

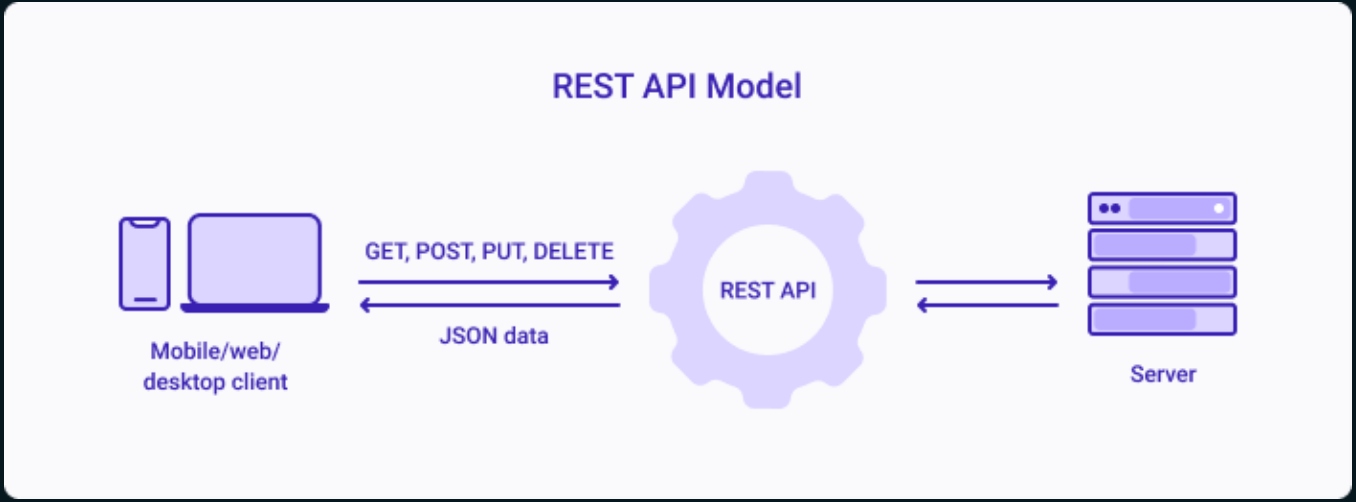
# Application Programming Interface

A RESTful API takes the REST architectural style and guides in interaction with RESTful web services. It defines a set of constraints, for how, the architecture of Web will behave.

Here, REST - Representational State Transfer

Referred to as a contract between an information provider and an information user — establishing the content required from the consumer (request) and the content required by the producer (response)

# Application Programming Interface




**HTTP Methods – GET, POST, PUT, HEAD, DELETE, PATCH, OPTIONS**
**Status Codes – 1xx, 2xx, 3xx, 4xx, 5xx**

# Application Programming Interface

| 1XX Informational | |
|---|---|
| 100 | Continue |
| 101 | Switching Protocols |
| 102 | Processing |

| 2XX Success | |
|---|---|
| 200 | OK |
| 201 | Created |
| 202 | Accepted |
| 203 | Non-authoritative Information |
| 204 | No Content |
| 205 | Reset Content |
| 206 | Partial Content |
| 207 | Multi-Status |
| 208 | Already Reported |
| 226 | IM Used |

| 3XX Redirectional | |
|---|---|
| 300 | Multiple Choices |
| 301 | Moved Permanently |
| 302 | Found |
| 303 | See Other |
| 304 | Not Modified |
| 305 | Use Proxy |
| 307 | Temporary Redirect |
| 308 | Permanent Redirect |

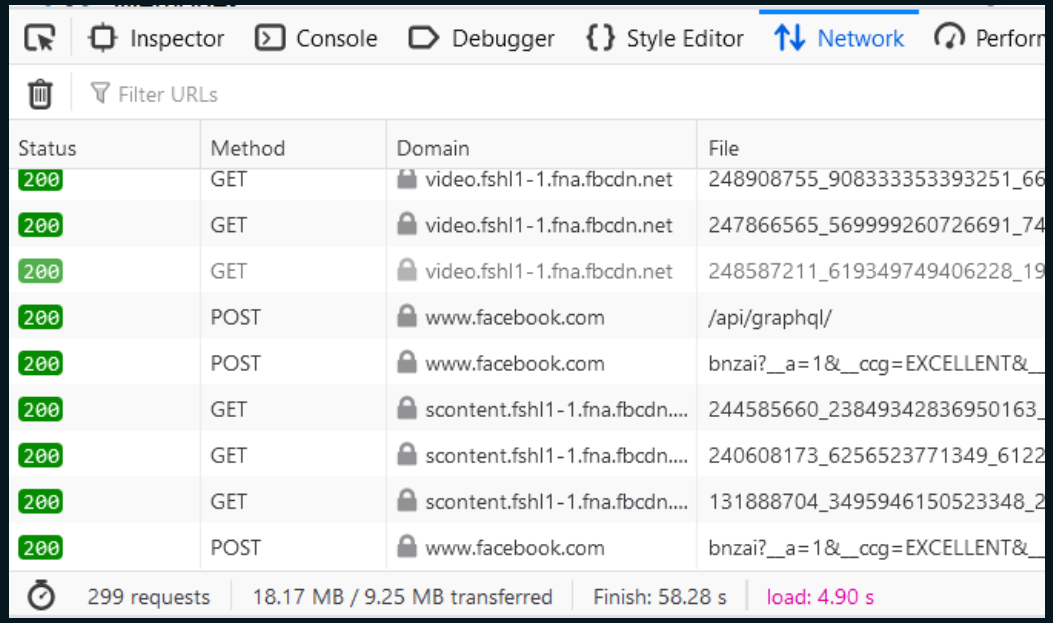
| 4XX Client Error | |
|---|---|
| 400 | Bad Request |
| 401 | Unauthorized |
| 402 | Payment Required |
| 403 | Forbidden |
| 404 | Not Found |
| 405 | Method Not Allowed |
| 406 | Not Acceptable |
| 407 | Proxy Authentication Required |
| 408 | Request Timeout |

| 4XX Client Error Continued | |
|---|---|
| 409 | Conflict |
| 410 | Gone |
| 411 | Length Required |
| 412 | Precondition Failed |
| 413 | Payload Too Large |
| 414 | Request-URI Too Long |
| 415 | Unsupported Media Type |
| 416 | Requested Range Not Satisfiable |
| 417 | Expectation Failed |
| 418 | I'm a teapot |
| 421 | Misdirected Request |
| 422 | Unprocessable Entity |
| 423 | Locked |
| 424 | Failed Dependency |
| 426 | Upgrade Required |
| 428 | Precondition Required |
| 429 | Too Many Requests |
| 431 | Request Header Fields Too Large |
| 444 | Connection Closed Without Response |
| 451 | Unavailable For Legal Reasons |
| 499 | Client Closed Request |

| 5XX Server Error | |
|---|---|
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 502 | Bad Gateway |
| 503 | Service Unavailable |
| 504 | Gateway Timeout |
| 505 | HTTP Version Not Supported |
| 506 | Variant Also Negotiates |
| 507 | Insufficient Storage |
| 508 | Loop Detected |
| 510 | Not Extended |
| 511 | Network Authentication Required |
| 599 | Network Connect Timeout Error |

# Application Programming Interface

**Open any Website > Right Click and select 'Inspect' > Go to Networks**



# Storage of IoT Data

So far, we've only Controlled device and Gathered data. We don't yet have a Perfect IoT System.

We shall now Set Up a system so that you are able to store data indefinite, and for NO COST of a database.

By the end, we will also build a REST API to access data from.

# How?

Using Google Sheet to store a Table of Data. We can use the data or view it from anywhere in the world.

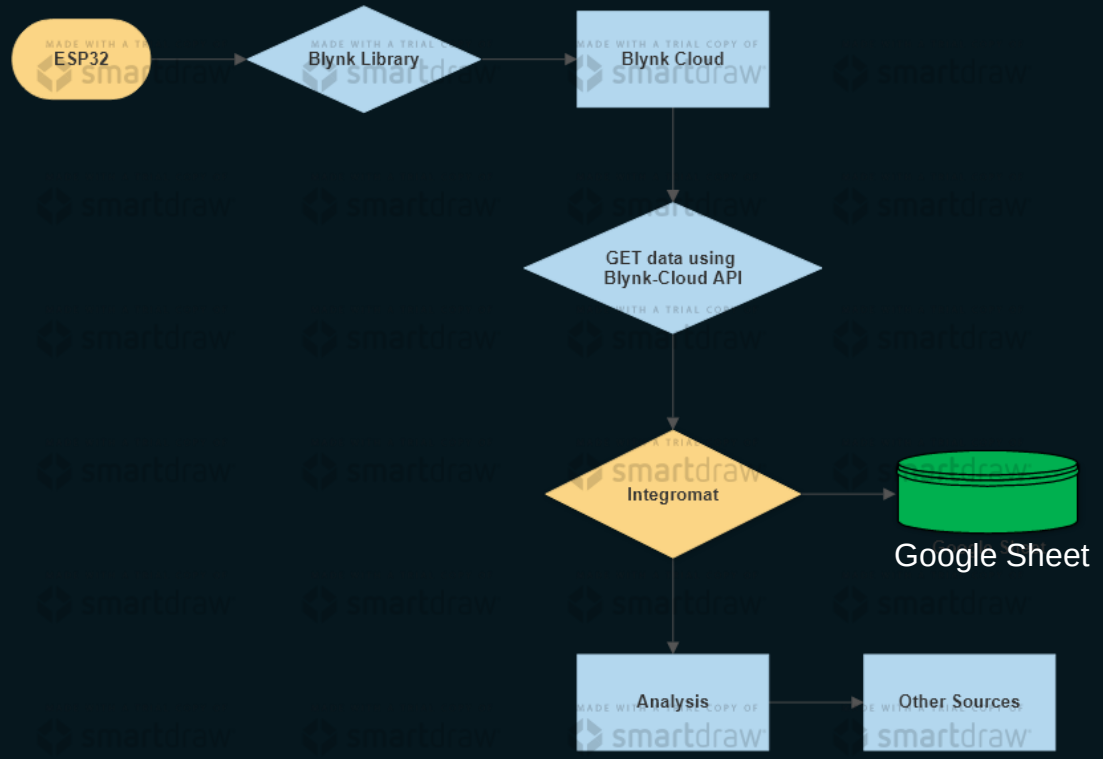
Turns out, we can even Create a Graph with that Data.

And Embed that Chart anywhere in our Website.

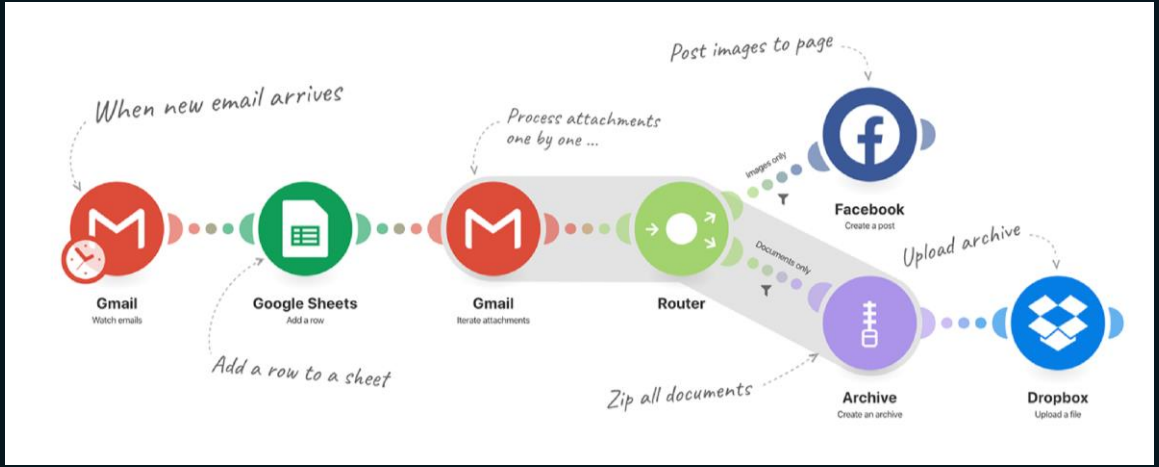
# Access all Code (Data Management)

shorturl.at/cgoI7

# Blynk IoT Data Flow

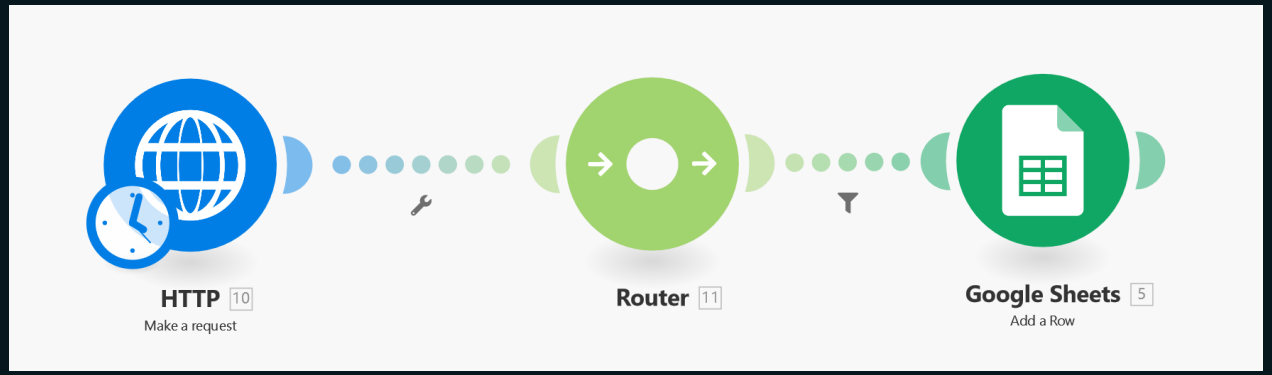


# What is integromat ?




A powerful integration platform that allows you to visualize, design and automate your work in minutes.

# Integromat Configuration (Blynk IoT)




| | A | B | C |
|---|---|---|---|
| 1 | time | value | Status |
| 2 | 2021-10-23T07:3 | 48.381 | 200 |
| 3 | 2021-10-23T07:4 | 46.063 | 200 |
| 4 | 2021-10-23T07:4 | 0 | 200 |
| 5 | 2021-10-23T07:4 | 47.084 | 200 |
| 6 | 2021-10-23T07:5 | 48.116 | 200 |
| 7 | 2021-10-23T07:5 | 48.656 | 200 |

# Integromat Configuration (Blynk IoT)

1. Module Name – HTTP

- URL - https://blr1.blynk.cloud/external/api/update?token={token}&{pin}={value}
- Method – GET
- Body Type – RAW
- Content Type – JSON (application/json)
- Parse Response – ✔

# Integromat Configuration (Blynk IoT)

2. Module Name – Router

- Filters Blynk Data before entering Google sheet Module.

# Integromat Configuration (Blynk IoT)

3. Module Name – Google Sheet (Add a Row)

- Connection – Google Account (Which has the Spreadsheet file created)
- Mode – Select spreadsheet and sheet
- Spreadsheet - <name of the spreadsheet>
- Sheet - <sheet no.> By default, Sheet1.
- Table Contains Headers – Yes
- Values –

  time (A) -> now (Date and Time section)

  value (B) -> Data[]  (May not see this option unless Run the Scenario at least once )

  status (C) -> Status code

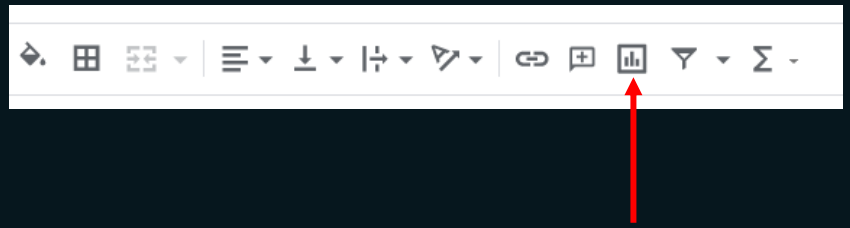
# Spreadsheet Configuration (Blynk IoT)

- Include the headers, i.e. time, value and status.
- Run the scenario to check if new row has been added on the respective headers.
- Select the Insert Chart icon:

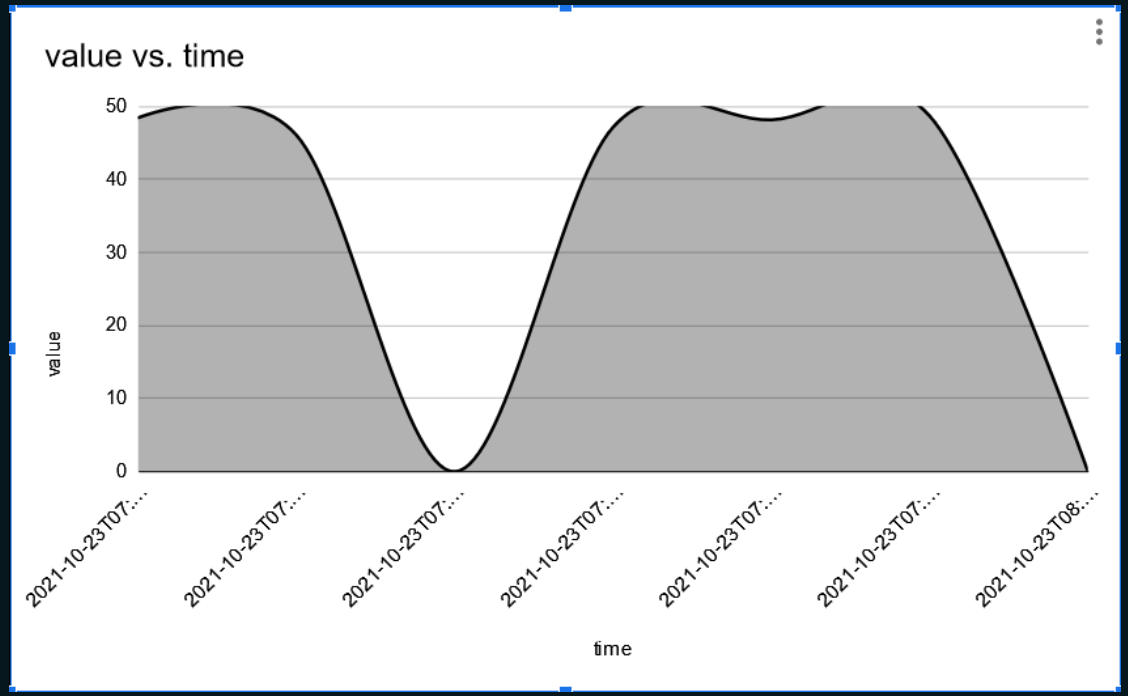
  Chart Type – Line/Area/Smooth Line

  Data Range > Select Data Range > In the Chart, press SHIFT and select ‘A’ and ‘B’. All the rows of A and B will get selected.
- X Axis -> Time, Series -> Value
- Customize the Chart further.

| | A | B | C |
|---|---|---|---|
| 1 | time | value | Status |
| 2 | 2021-10-23T07:3 | 48.381 | 200 |
| 3 | 2021-10-23T07:4 | 46.063 | 200 |
| 4 | 2021-10-23T07:4 | 0 | 200 |
| 5 | 2021-10-23T07:4 | 47.084 | 200 |
| 6 | 2021-10-23T07:5 | 48.116 | 200 |
| 7 | 2021-10-23T07:5 | 48.656 | 200 |



# Data Chart (Blynk IoT)



# Data Chart (Blynk IoT) - Organized

| | A | B | C | D |
|---|---|---|---|---|
| 1 | Date | time | value | Status |
| 2 | 23.10.2021 | 14:45:23 | 49.425 | 200 |
| 3 | 23.10.2021 | 14:46:42 | 49.173 | 200 |
| 4 | 23.10.2021 | 14:48:07 | 49.125 | 200 |
| 5 | 23.10.2021 | 14:48:43 | 50.084 | 200 |
| 6 | 23.10.2021 | 14:52:23 | 49.545 | 200 |

Date and time

VARIABLES

timestamp

now

FUNCTIONS

addSeconds addMinutes addHours addDays

addMonths addYears setSecond setMinute

setHour setDay setDate setMonth

setYear formatDate parseDate

Values

Date (A)

formatDate( now ; DD.MM.YYYY )

time (B)

formatDate( now ; HH:mm:ss )

value (C)

10. data[1]

Status (D)

10. Status code

# Blynk IoT API Documentation

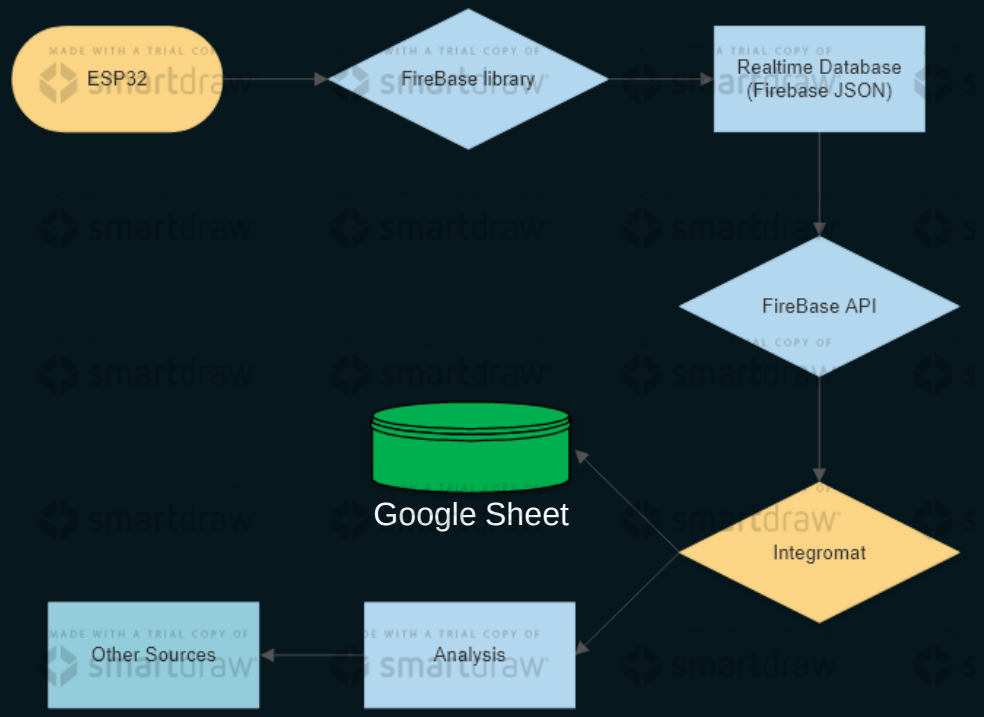
https://docs.blynk.io/en/blynk.cloud/https-api-overview

# Firebase Realtime Database (JSON)

Firebase is a platform to develop Mobile and Web Applications. It provides tools for tracking analytics, reporting and fixing app crashes, creating marketing and product experiment.

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in realtime.

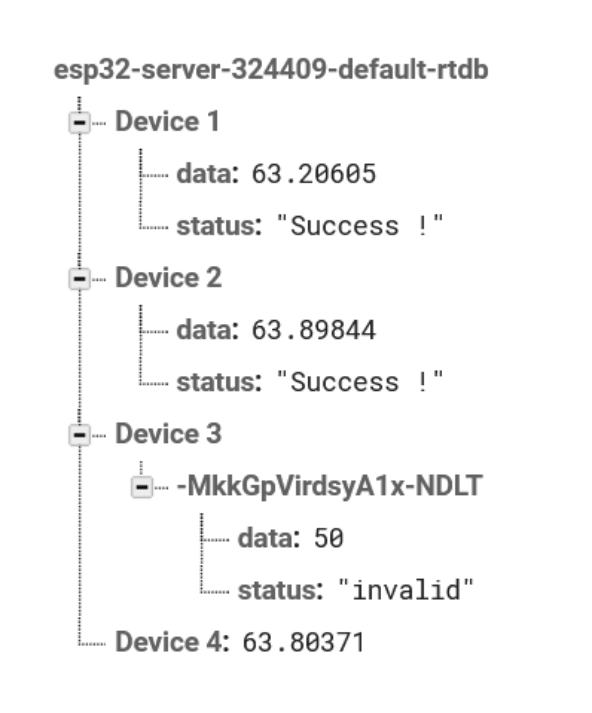
# FireBase Data Flow



# Google FireBase to store IoT Data

**Set Up a Realtime Database in Firebase, with appropriate objects**

- Realtime Database > Create Database > Location (US) > Start in ‘Locked mode’ and Enable (wait for a min. for the setup)
- To create objects, Add child (click + ) beside ‘null’ > Enter ‘name’ and ‘value’ > Click ‘Add’
- To have a hierarchy, enter ‘name’ and add child (click + beside ‘value’. (As given in the image)
- In the ESP32 code, fill the blank spaces with appropriate details.
- For Project ID, go to Settings > Projects Settings > General > Get the Project ID.
- For Secret Key, go to Project Settings > Service Accounts > Database Secrets > Copy the Secret Key.



# FireBase API Documentation

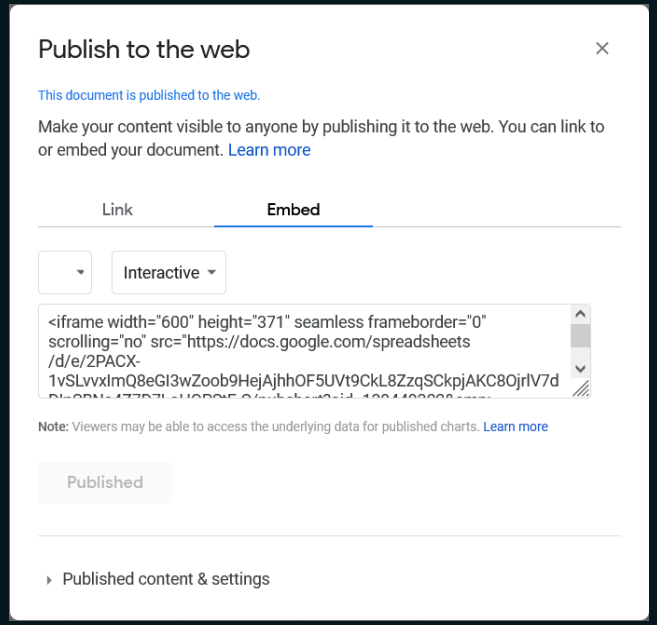
https://firebase.google.com/docs/reference/rest/database

# Embedding Chart to any Website

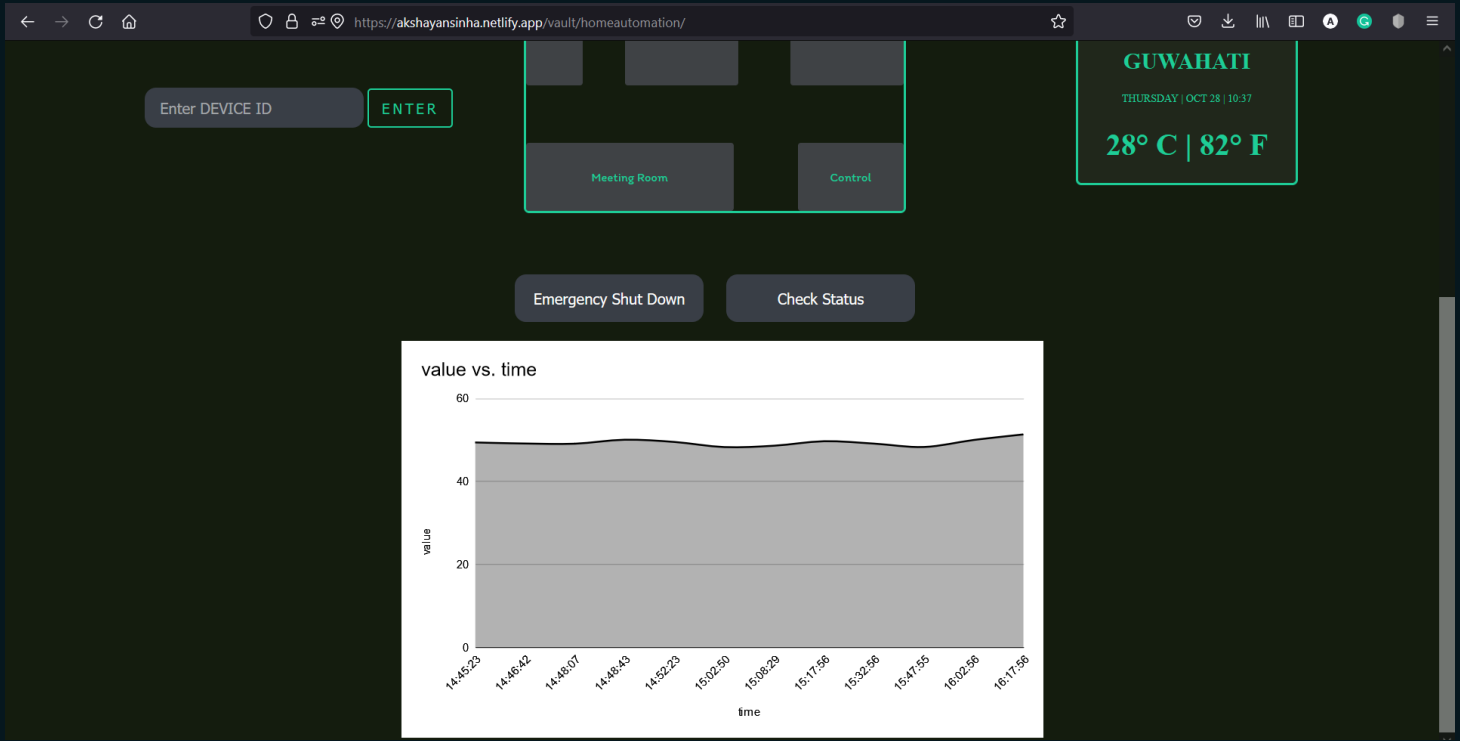
**Use the iframe tag to view the chart in your own Website.**

**How to do that?**

- Click on the Chart > click ‘three dots’ on top right corner.
- Select ‘Publish chart’ > select ‘Embed’ > Click Publish
- The iframe tag for the chart will be generated. Which can be used anywhere in the html file of a website.



# Embedding Chart to any Website



# Your own REST – API service

**To build a REST API,**

- NodeJS Application (Server side program)
- Using Express framework.
- MongoDB (database) to store  (optional)

**Pre-installations required:**

- **Visual Studio Code (IDE)**
- **Node JS**

# Using Online Template for Overview

https://codesandbox.io/

Create Sandbox > Explore Templates > Search and select ‘node-express-server-rest-api’

# ESP32 REST Server with DB

https://esp32-rest-server.herokuapp.com/

# APIs with Python

**Like an ESP32, we can Write a Program on Python as well, to send or receive HTTP request to/from any Web Service.**

- Install requests library, using ‘pip install requests’
- Sample Code –

```
import requests

url = "http://<base_url>"
response = requests.request("GET", url + "/")
print(response)
```

# APIs with Python

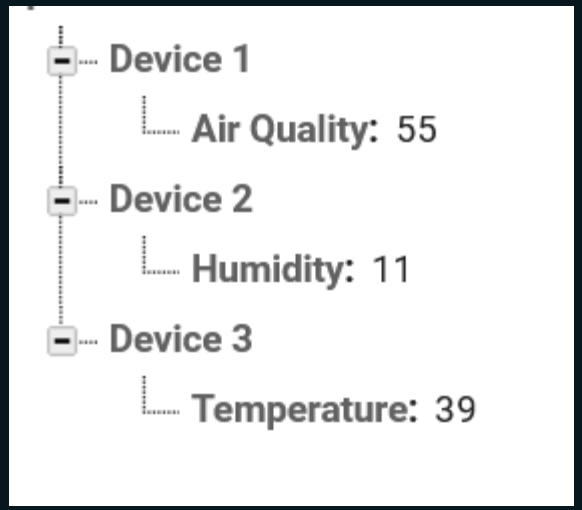
- Install json library, using ‘pip install json’
- Sample Code –

```
import requests
import json

url = "https://esp32-server-324409-default-rtdb.firebaseio.com/Device 4.json"

querystring = {"auth":"4gmoOLRzpGxxxxxxxxxxuzq8Ugz0aABOT6pf"}

response = requests.request("GET", url, params=querystring)


line = json.loads(response.text)
#data = str(line['data'])
print(line)
```

# APIs with Python (GitHub Link)

shorturl.at/nvF01

# TASK - 3




Use the FireBase API documentation, to get Data from the Realtime Database and Write a Python program to Print Data in a structured format, as shown below -

Air Quality -> XX

Humidity -> XX %

Temperature -> XX ℃

As a reference, Follow the Structure of FireBase as in the Image.

Insert any int/float value to the Database, do NOT use 'XX' in the submission.

# TASK - 4

**Write an ESP32 Program on Arduino IDE, to control an IoT Car**

- Use the WebServer code from Github as a reference.
- Create functions like forward, backward, left and right.
- Use proper endpoint on the 4 functions.
- Each endpoint must have a heading of it’s Movement.
- Go to File > Preferences > Show Verbose output - (Compilation ✔)
- Compile and Verify the Code (take screenshot of screen).

# TASK - 5

**Write a Python code to Send Data to the ThingSpeak or Blynk Cloud**

- Use the FireBase and City Weather API code as a reference.
- Use User input, to get data from the User.
- Use that data to Send to Blynk Cloud/ThingSpeak.
- (if any) Another Member, use another Python program, to GET that data, use Json Parsing Method to access that.
- Use any real-life scenario to Implicate the Program.

# Submission Link

https://forms.gle/zooomBdxnR1YoVsp8

# Thank You

Reach me in the below Socials:

LinkedIn - linkedin.com/in/akshayansinha/

Hackster - hackster.io/akshayansinha

Instagram - instagram.com/akshayansinha/

Github - github.com/hippyaki

Mail - akshayan.sinha@gmail.com