Unit – 04: IoT APIs and its Integration

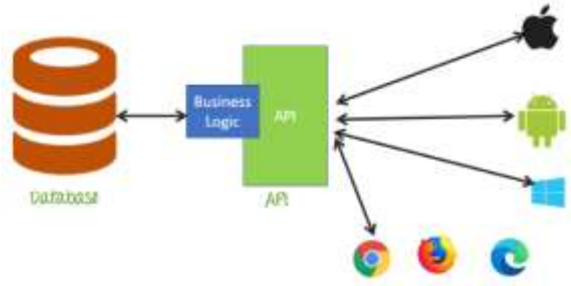
- Explain APIs and its use
- Explanation of given IoT APIs along with its applications
- REST
- SOAP
- JSON
- MQTT, Broker, subscriber, publisher
- Programming API using Python

Questions to be discussed:

1. What do you mean by IoT API? What are the application of API?
2. Explain different types of IoT API.
3. Discuss MQTT Broker subscriber/publisher model.
4. How to create an API using python?
5. Write short notes on:
    a. REST
    b. SOAP
    c. JSON

What is an API?

- API stands for Application Programming Interface.
- An API allows two or more IoT devices interact with each other.
- APIs are the points of interaction between an IoT device and the internet within the network.
- It is a set of rules that are shared by a particular service.
- An API acts as a layer between your application and external service.
- You do not need to know the internal structure and features of the service, you just send and receive.



Applications of API:

- Desktop applications.
- Web applications.
- APIs make mobile applications possible.
- APIs connect devices to the internet.
- APIs define the networks or the information passed between applications, systems, and devices.
- Connect everyday things like automobiles, doorbells, dishwashers, and wearable devices.

API Types in IoT:

1. SOAP
2. REST
3. JSON

SOAP

- SOAP stands for Simple Object Access Protocol.
- They make building a communication bridge between the servers and the clients.
- It is much complex but also well defined in terms of security.
- Banking transactions or any other applications that need high security use this API.
- It works over HTTP, HTTPS, SMTP, XMPP.
- Designed with large enterprise applications.

REST:

- REST stands for Representational State Transfer.
- It is a software architecture that defines a pattern for client and server communications over a network.
- REST provides a set of constraints for software architecture to promote performance, scalability, simplicity, and reliability in the system.
- It is one of the most popular web API used.
- The client-server communication is simple here and is based on URL or HTTP protocols.

Differentiating between SOAP API and REST API:

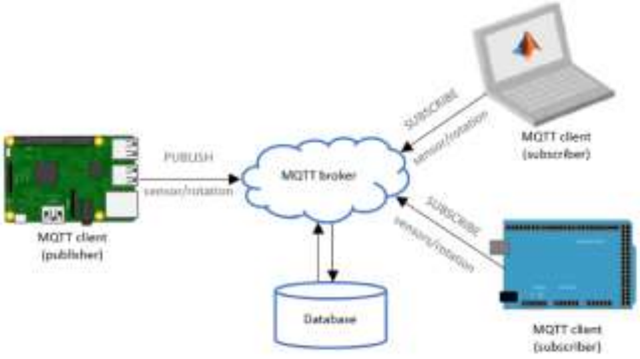
SOAP API	REST API
SOAP stands for Simple Object Access Protocol.	REST stands for Representational State Transfer.
Transports data in standard XML format.	Generally transports data in JSON.
Because it is XML based and relies on SOAP, it works with WSDL	It works with GET, POST, PUT, DELETE
It works over HTTP, HTTPS, SMTP, XMPP	It works over HTTP and HTTPS
Highly structured.	Less structured.
Designed with large enterprise applications.	Designed with mobile devices in mind.

JSON:

- JSON stands for JavaScript Object Notation.
- It has one of the best request efficiency API.
- It uses JSON formatting which is much reliable and simple to use.
- It is human-readable as well as machine-readable and is designed in such a way that they call methods.

MQTT, Broker, subscriber, publisher:

- MQTT stands for Message Queuing Telemetry Transport.
- It is a publish-subscribe architecture that is developed to connect devices over wireless networks.
- It is a simple and lightweight protocol that runs over TCP/IP sockets, WebSockets, and SSL.
- MQTT has two components:
  1. MQTT broker.
  2. MQTT client.



MQTT broker:

- An MQTT broker is a central point of communication.
- The broker is responsible for dispatching all messages between the clients.

MQTT client:

- An MQTT client is any device (computer or a mobile phone) that connects to the broker.
- A client that sends messages is a publisher.
- A client that receives messages is a subscriber.
- To receive a message, the client must subscribe to the topic of that message.
- You can publish and subscribe to MQTT messages using MQTT Publish and MQTT Subscribe blocks. These blocks support MQTT only over TCP/IP sockets.

How to create an API using Python?

- There are different ways to create an API in Python the most used being FastAPI and Flask.

How to create an API in Python with Flask:

- First of all, create an API in Python using Flask we must install the flask and flask-restful packages.
- Once we have the libraries installed, we have to to create our server, as we did with FastAPI.
- We can do this with the following command:

```
from flask import Flask
app = Flask()
```

- In addition, we will have to indicate that the server to launch together with the port where it should be launched. To do this, at the end of our app we must include the following code:

```
if __name__ == '__main__':
app.run(debug=True, port=8000)
```

- Once we have our server created and we have indicated on which port it should run, we can start creating our APIs.
- To create an API in Python with Flask, we have to indicate: the endpoint, the method and the function that should be executed on that endpoint.
- Let's see an example with an API that simply returns the text "Hello world!".

```
from flask import Flask, jsonify, request,send_file
app = Flask()
@app.route('/my-first-api', method = ['GET'])
def hello():
return "Hello world!"
```

Note:

- Flask is a framework that allows you to create entire web applications.
- FastAPI performs the validation of the data types that we perform, as well as the entire documentation process, saving us a lot of time compared to whether We do it in Flask.