IOT COMMUNICATION MODELS

INTRODUCTION

IoT devices are found everywhere and will enable circulatory intelligence in the future. For operational perception, it is important and useful to understand how various IoT devices communicate with each other. Communication models used in IoT have great value. The IoTs allow people and things to be connected any time, any space, with anything and anyone, using any network and any service.

TYPES OF IOT COMMUNICATION MODEL

There are several different types of models available in an IoT system that used to communicate between the system and server.

- Request-Response model
- Publish-Subscribe model
- Push-Pull model
- Exclusive Pair model

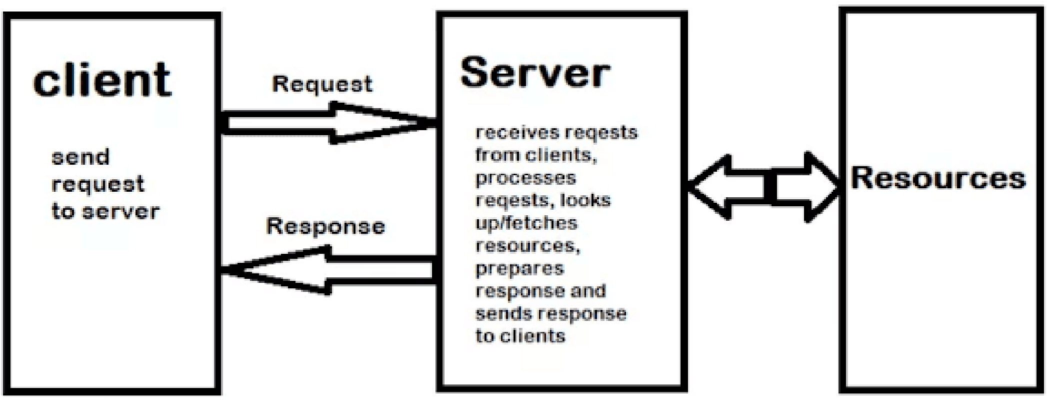
REQUEST-RESPONSE COMMUNICATION MODEL

This model is a communication model in which a client sends the request for data to the server and the server responds according to the request. When a server receives a request it fetches the data, retrieves the resources and prepares the response, and then sends the data back to the client. In simple terms, we can say that in the request-response model server send the response of equivalent on the request of the client. In this model, HTTP works as a request-response protocol between a client and server.

This model follows a client-server architecture.

- The client, when required, requests the information from the server. This request is usually in the encoded format.
- This model is stateless since the data between the requests is not retained and each request is independently handled.
- The server Categories the request, and fetches the data from the database and its resource representation. This data is converted to response and is transferred in an encoded format to the client. The client, in turn, receives the response.

Example: When we search a query on a browser then the browser submits an HTTP request to the server and then the server returns a response to the browser (client).



Request-Response Communication Model

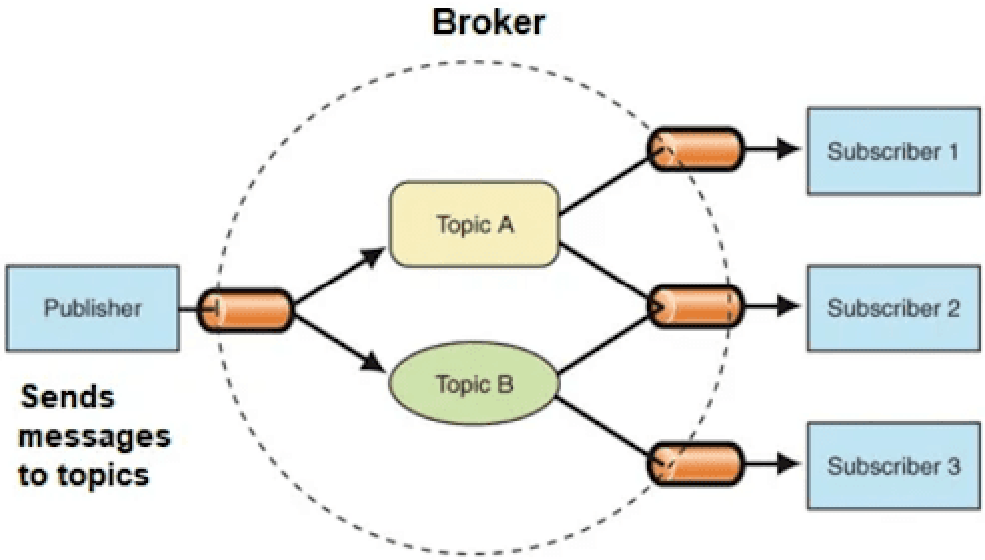
PUBLISH-SUBSCRIBE COMMUNICATION MODEL

In this communication model, we have a broker between publisher and consumer. Here publishers are the source of data but they are not aware of consumers. They send the data managed by the brokers and when a consumer subscribes to a topic that managed by the broker and when the broker receives data from the publisher it sends the data to all the subscribed consumers.

This model comprises three entities: Publishers, Brokers, and Consumers.

- Publishers are the source of data. It sends the data to the topic which are managed by the broker. They are not aware of consumers.
- Consumers subscribe to the topics which are managed by the broker.
- Hence, Brokers responsibility is to accept data from publishers and send it to the appropriate consumers.
- The broker only has the information regarding the consumer to which a particular topic belongs to which the publisher is unaware of.

Example: On the website many times we subscribed to their newsletters using our email address. These email addresses managed by some third-party services and when a new article published on the website it directly sends to the broker and then the broker send these new data or post to all the subscribers.



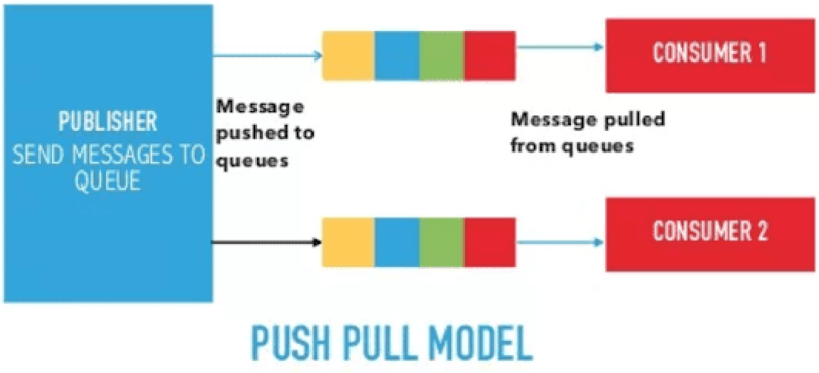
.

PUSH-PULL COMMUNICATION MODEL

It is a communication model in which the data push by the producers in a queue and the consumers pull the data from the queues. Here also producers are not aware of the consumers.

The push-pull model constitutes data publishers, data consumers, and data queues.

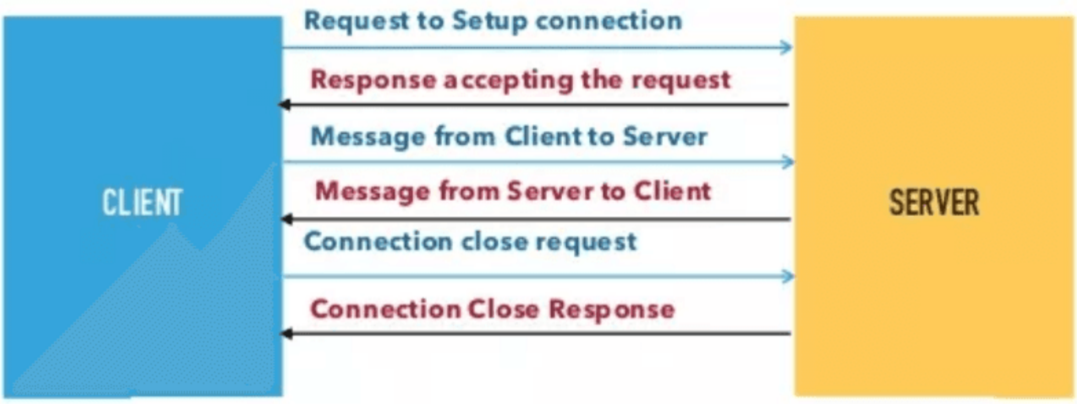
- Publishers and Consumers are not aware of each other.
- Publishers publish the message/data and push it into the queue. The consumers, present on the other side, pull the data out of the queue. Thus, the queue acts as the buffer for the message when the difference occurs in the rate of push or pull of data on the side of a publisher and consumer.
- Queues help in decoupling the messaging between the producer and consumer. Queues also act as a buffer which helps in situations where there is a mismatch between the rate at which the producers push the data and consumers pull the data.



Example: When we visit a website we saw a number of posts that published in a queue and according to our requirements, we click on a post and start reading it.

EXCLUSIVE PAIR COMMUNICATION MODEL

It is a bidirectional fully duplex communication model that uses a persistent connection between the client and server. Here first set up a connection between the client and the server and remains open until the client sends a close connection request to the server. The Server has the record of all the connections which has been opened. This is a state-full connection model and the server is aware of all open connections. Web Socket based communication API is fully based on this model.


CLIENT
Request to Setup connection
Response accepting the request
Message from Client to Server
Message from Server to Client
Connection close request
Connection Close Response
SERVER

EXCLUSIVE PAIR COMMUNICATION MODEL