

Sync & Async Communication

=====

Sync Communication==After sending the request thread will wait for the Response before going ahead

Async Communication == After sending the request thread will not for the Response and it will go ahead with other execution

Sync ==> RestTemplate ==> class

Async ==> WebClient ==> interface

=====

Swagger UI -- Documentation , JAX - B == XML, JSON(Jackson api), HATEOS,
JUNIT5 and MOCKITO with Spring Boot Rest api, Spring Boot Mail integration, Spring Batch, MultiDb config datajpa,
association, Actuators, loggers, Remaining Concepts of Spring Security, Spring boot Apache kafka.

=====

Response from the server to client we will have Response body and Status code

HTTP Status code

=====

Success - OK --> 200-299 (2xx)

Bad request - Client error --> 400-499(4xx)

Server Error --> 500- 599 (5xx)

Information --> 100 - 199 (1xx)

Redirection --> 300 - 399(3xx)

HTTP request

Request URL (request type + url)

Request header (meta data)

Request Body (payload)

HTTP Response

Response line (Status code and Status message)

Response header (meta data)

Response Body (Payload)

application.properties vs application.yml

=====

In Spring Boot app we use application.properties or .yml file to configure our application properties

example ==> DataSource, PORT, kafka , Redis...etc

==> .properties file we will have data represented in key-value format

==> .properties file is there only for Java-spring boot application

==> YML is universal format(java, python, .Net, ansible, k8s....)

==>YML file will represent data in hierarchical format

==>Indent spacing is very important in yml file

```
server.port=8484 //application.properties

server.servlet.context-path=/TicketBookingApp //.properties file
```

```
#DataJPA
```

```
##DataSoruce Configuration
spring.datasource.url=jdbc:mysql:///telusko_db
spring.datasource.username=root
spring.datasource.password=root123
```

```
-----.yaml file -----
```

```
server:
  port:8484 //yaml file
```

```
server:
  servlet:
    context-path: //yaml file
```

```
spring:
  datasource:
    url:mysql:///telusko_db
    username:root
    password:root123
```

JAX - B ==> XML format data transfer || Jackson ---> JSON format

consumes --> In which format REST API method can accept input (data) from clients

produces --> In which formate REST API method can provide response to clients