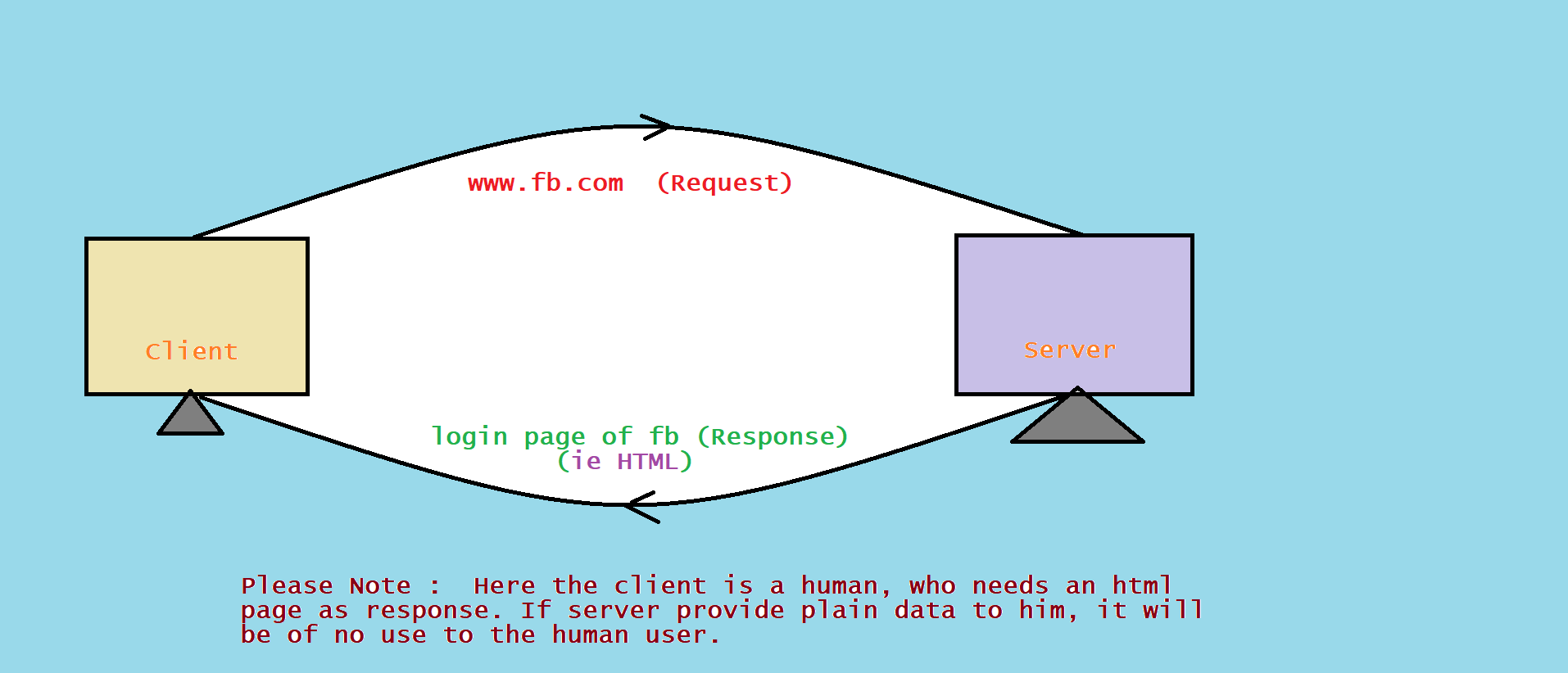
**How to Start with REST API using Spring Boot**

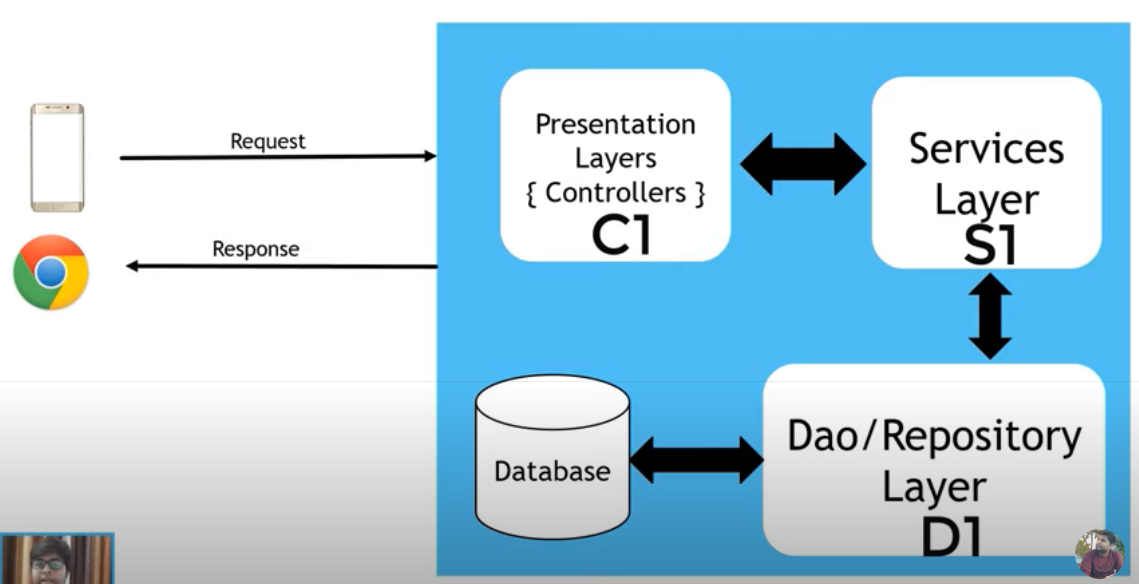
A **RESTful API is** an application program interface (**API**) that uses HTTP requests to GET, PUT, POST and DELETE data. An **API** for a website **is** code that allows two software programs to communicate with each other.

Normally what we do is - We make a web app in which a user/client hit a particular url to the server and gets its result in the form of html pages.

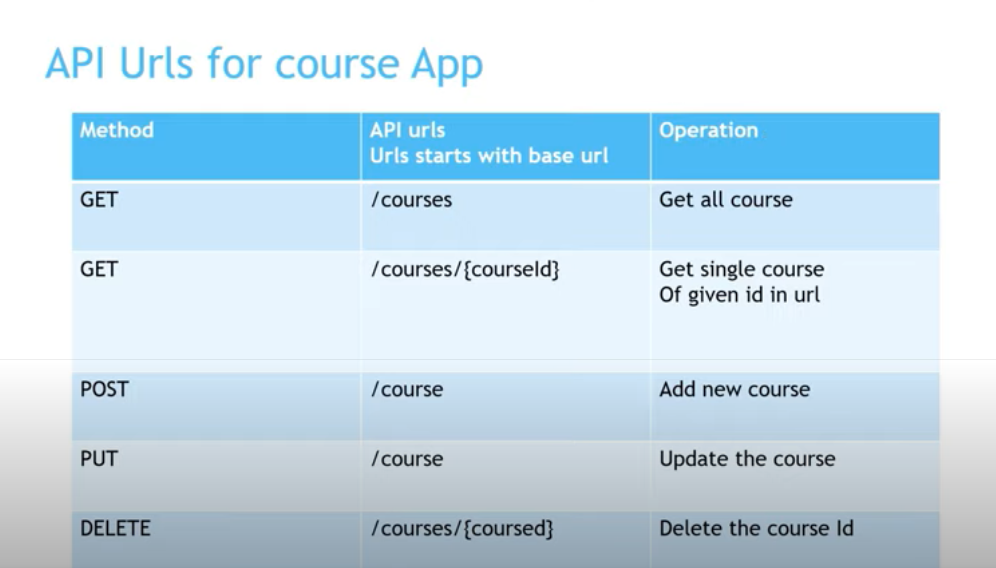


But suppose, what if user is a machine and want to hit any server to get data only, not an html page, because the machine/program/application will use data in its own program and machine will display the data in its own style.

**Flow of Control**



We will create the following APIs



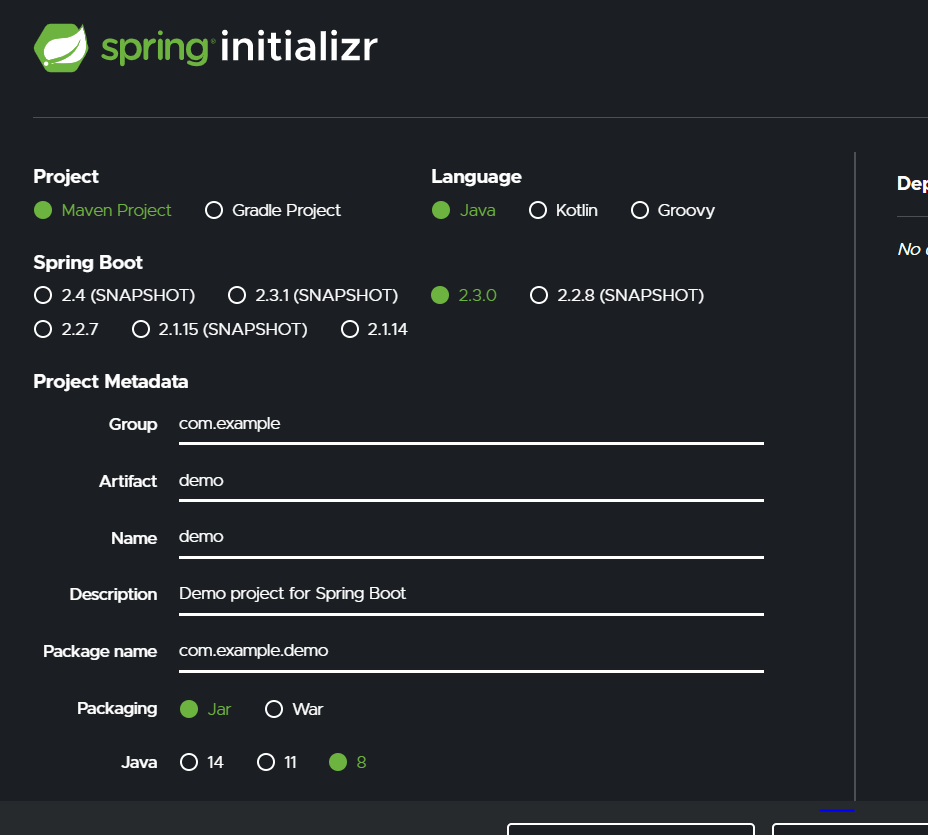
**Project Software Setup**

1. STS to create Spring or Spring Boot Starter Project
2. POSTMAN to hit our API and check the responses

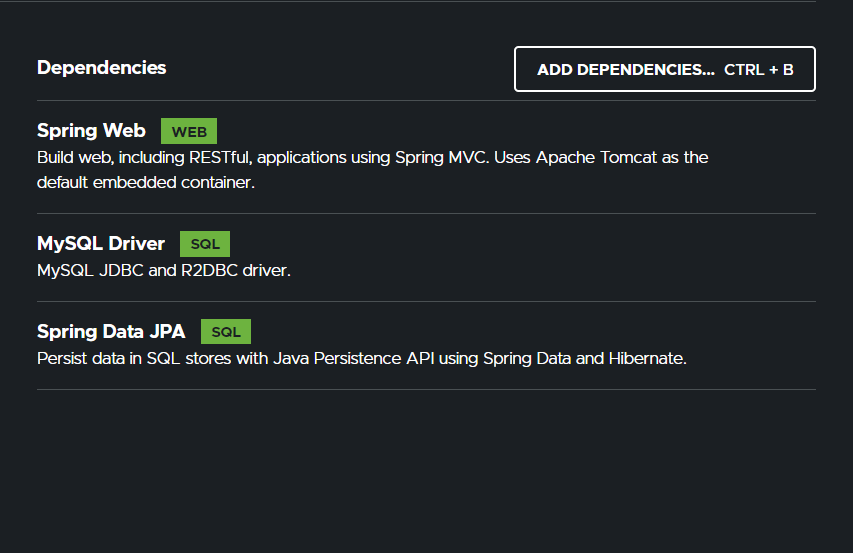
**Creating the Project**

1. For eclipse or netbeans user

Use Spring Initializr



**Adding Dependency**

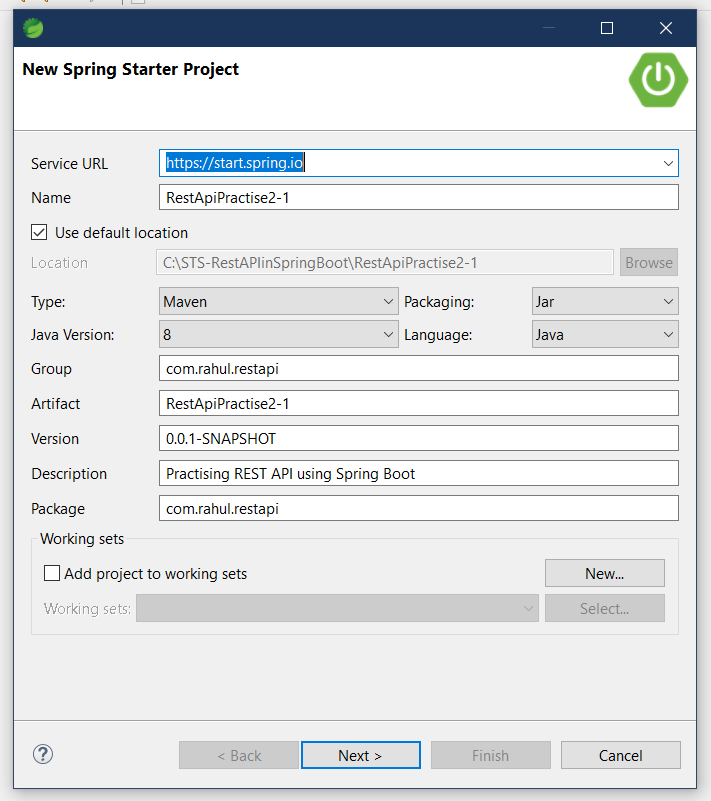


Now Generate or Download this project using button given there.

Finally import In the Eclipse.

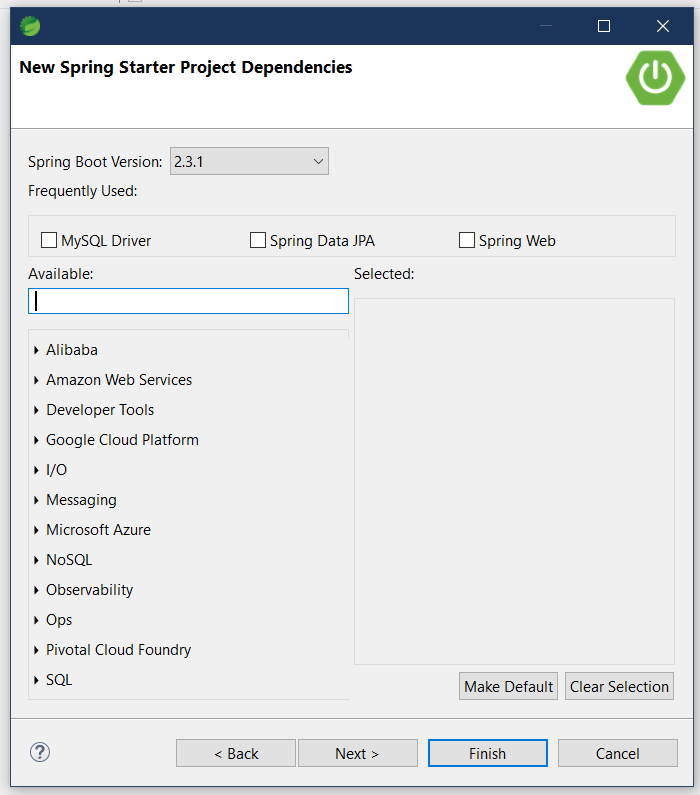
1. **For STS**

**Create Spring Starter Project**



**Add dependency**

1. **Spring Web (for creating RESTful Services)**
2. **MySQL Driver (for jdbc)**
3. **Spring Data JPA (……)**



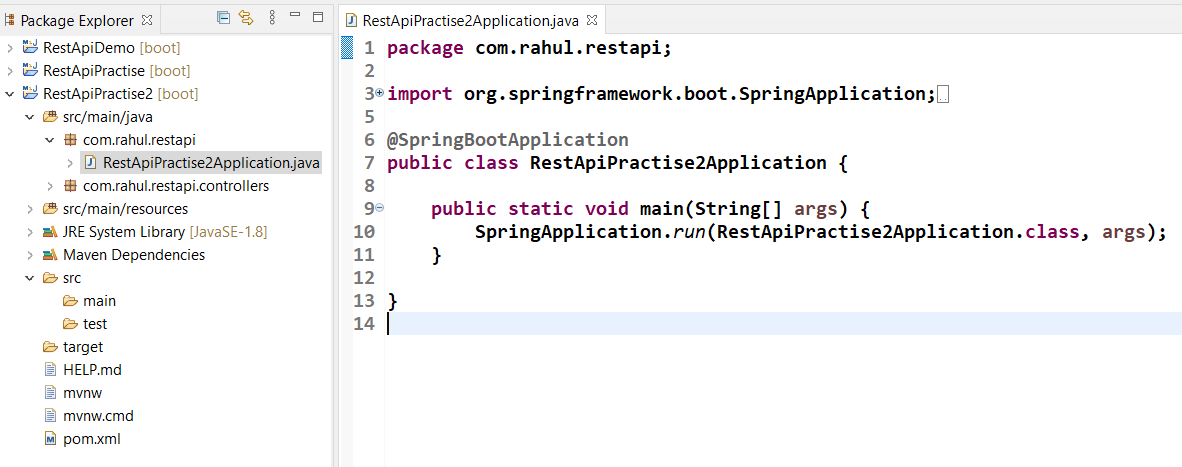
**See the POM.xml**



**Please Note : We are not going to use database, so we don’t need Spring Data JPA dependency in our project for now. So we need to remove it, otherwise it will cause error.**

**Cut its dependency lines from pom.xml and paste it somewhere, it will be required later on.**

We can also remove MySQL driver dependency if we are not going to work with database.



* **The class “RestApiPractise2Application” is the starting point of the execution of the Application.**
* **It has the main() method.**
* **Any other class must be created under this package “com.rahul.restapi”, it is due to the working idealogy of the SpringBoot Application that is it starts scanning from this package and subpackages.**
* **If any class is created outside the main package, it will not be of any use or it will not be under control of spring boot.**
* **For eg. the new package “com.rahul.restapi.controllers” is subpackage of parent package “com.rahul.restapi” .**

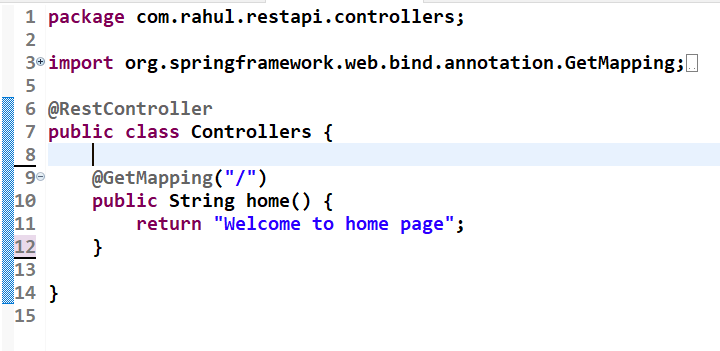
**Creating our first Controller.**

1. **Create a package (com.rahul.restapi.controllers)**
2. **Create a class (say Controller)**
3. **Annotate this class with @RestController**

* **It means this class will be our front-end controller.**
* **That is, any request coming to our application will come to this class.**
* **And then a suitable method will be invoked based on the request and mapping of urls.**

1. **Create a method that will be invoked , if a request is made like**

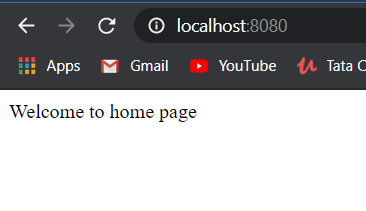
**“ localhost:8080 “ and this method should return a String “Welcome to the HomePage”.**



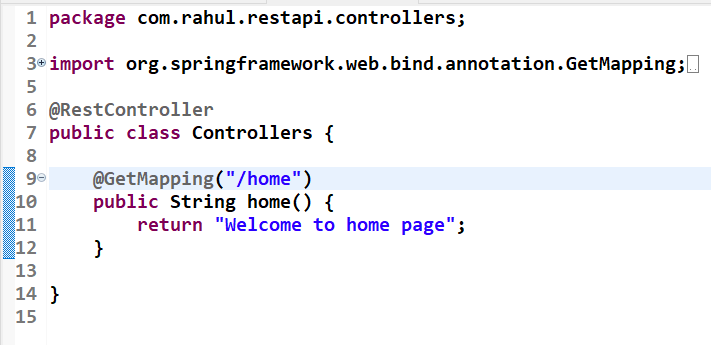
1. **Testing**

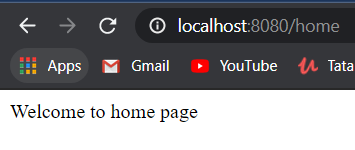
**Run the project as Spring Boot App.**

**Open any browser, hit localhost:8080**

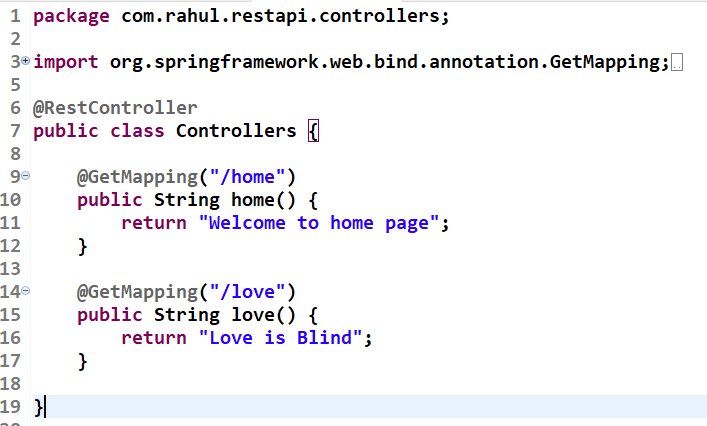


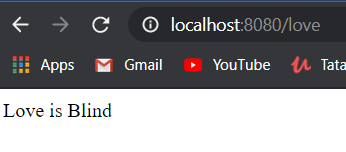
1. **So project is working fine.**
2. **So it’s a GET request. So when we hit the localhost:8080, the controller invoked the home() method.**
3. **How did he know that he had to hit home() method …?**
4. **He knows because of mapping, the @GetMapping(“/”).**
5. **The mapping says that any get request should hit this method.**
6. **“/” means no specific method. Any one.**
7. **We can change it like making it “/home”.**
8. **Do the change and check it.**





1. **We can create n number of methods to handle n request.**
2. **Lets make an another one. It will return a quote saying “Love is Blind”. When a user hits localhost:8080/love.**





**So based on our url, suitable method should be invoked and it is sured by mapping of the request to the corresponding methods.**