Introduction to Spring Boot

Agenda

1. Introduction to Spring Boot
2. Hello World - Example

Objectives

At the end of this module, you will be able to:

- Introduction to Spring Boot

Introduction to Spring Boot

What is Spring Boot?

- Spring Boot is a module of the Spring Framework.
- It is used to create stand-alone, production-grade Spring Based Applications with minimum efforts.
- It is developed on top of the core Spring Framework.
- Spring Boot follows a layered architecture in which each layer communicates with the layer directly below or above it.

Spring Boot vs Spring

- **Spring**: Spring framework is the most popular application development framework of Java. The main feature of the Spring Framework is dependency Injection or Inversion of Control (IoC). With the help of Spring Framework, we can develop a loosely coupled application. It is better to use if application type or characteristics are purely defined.

- **Spring Boot**: Spring Boot is a module of Spring Framework. It allows us to build a stand-alone application with minimal or zero configurations. It is better to use if we want to develop a simple Spring-based application or RESTful services.

Spring Boot vs Spring MVC

- **Spring Boot**: Spring Boot makes it easy to quickly bootstrap and start developing a Spring-based application. It avoids a lot of boilerplate code. It hides a lot of complexity behind the scene so that the developer can quickly get started and develop Spring-based applications easily.

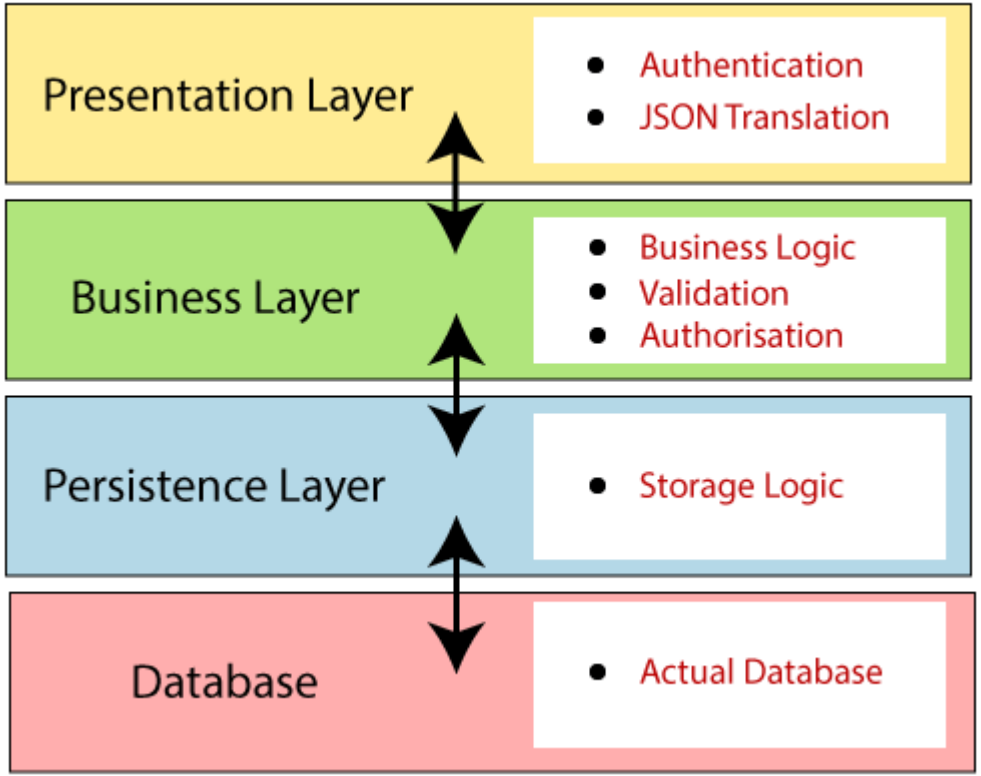
- **Spring MVC**: Spring MVC is a Web MVC Framework for building web applications. It contains a lot of configuration files for various capabilities. It is an HTTP oriented web application development framework.

Spring Boot Layers

- Before understanding the **Spring Boot Architecture**, we must know the different layers and classes present in it. There are four layers in Spring Boot are as follows:

  - Presentation Layer
  - Business Layer
  - Persistence Layer
  - Database Layer

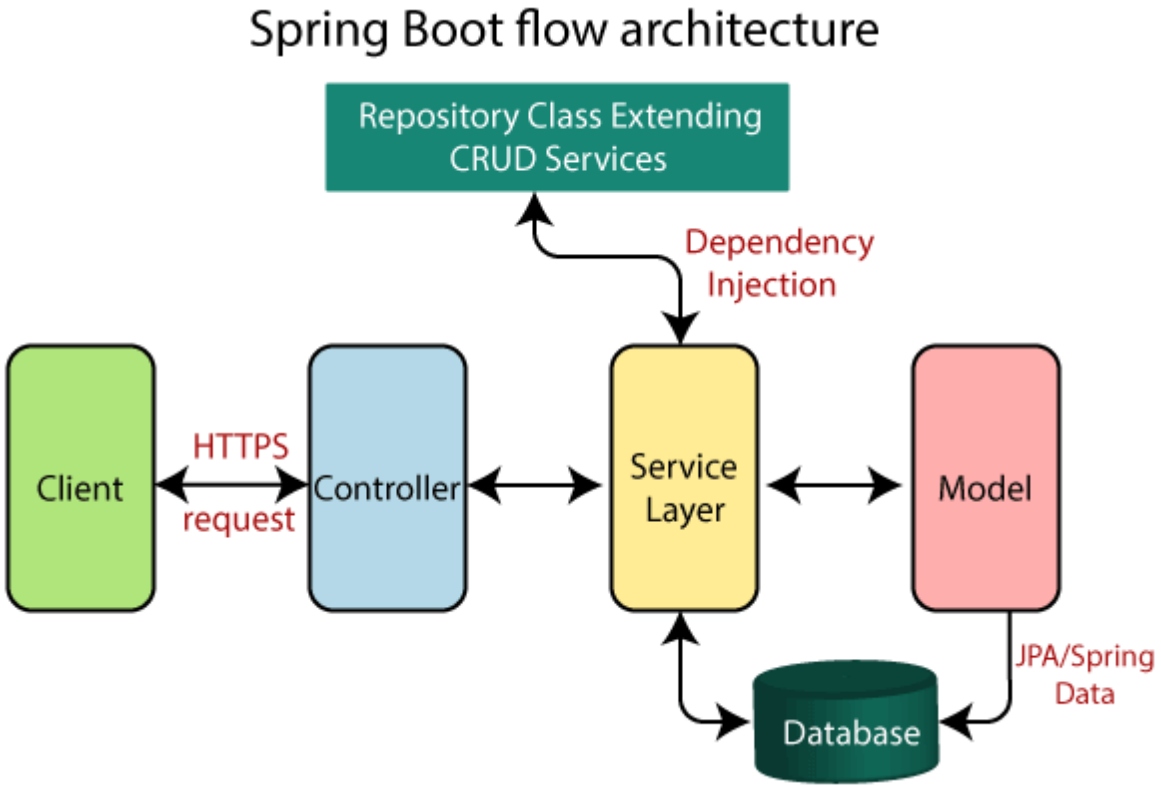
Spring Boot Layers



Spring Boot Layers

- **Presentation Layer**: The presentation layer handles the HTTP requests, translates the JSON parameter to object, and authenticates the request and transfer it to the business layer. In short, it consists of views i.e., frontend part.
- **Business Layer**: The business layer handles all the business logic. It consists of service classes and uses services provided by data access layers. It also performs authorization and validation.
- **Persistence Layer**: The persistence layer contains all the storage logic and translates business objects from and to database rows.
- **Database Layer**: In the database layer, **CRUD** (create, retrieve, update, delete) operations are performed.

Spring Boot Flow Architecture



Spring Boot Flow Architecture

- Now we have validator classes, view classes, and utility classes.
- Spring Boot uses all the modules of Spring-like Spring MVC, Spring Data, etc. The architecture of Spring Boot is the same as the architecture of Spring MVC, except one thing: there is no need for DAO and DAOImpl classes in Spring boot.
- Creates a data access layer and performs CRUD operation.
- The client makes the HTTP requests.
- The request goes to the controller, and the controller maps that request and handles it. After that, it calls the service logic if required.
- In the service layer, all the business logic performs. It performs the logic on the data that is mapped to JPA with model classes.
- A JSP page is returned to the user if no error occurred.

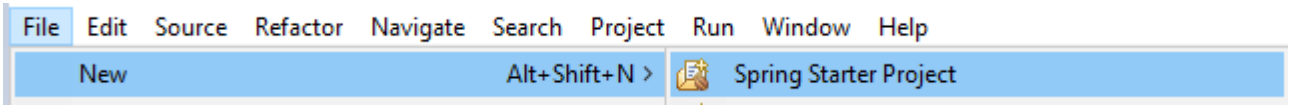
What is Spring Initializr?

- Spring Initializr is a web-based tool provided by the Pivotal Web Service.
- With the help of Spring Initializr, we can easily generate the structure of the Spring Boot Project.
- It offers extensible API for creating JVM-based projects.

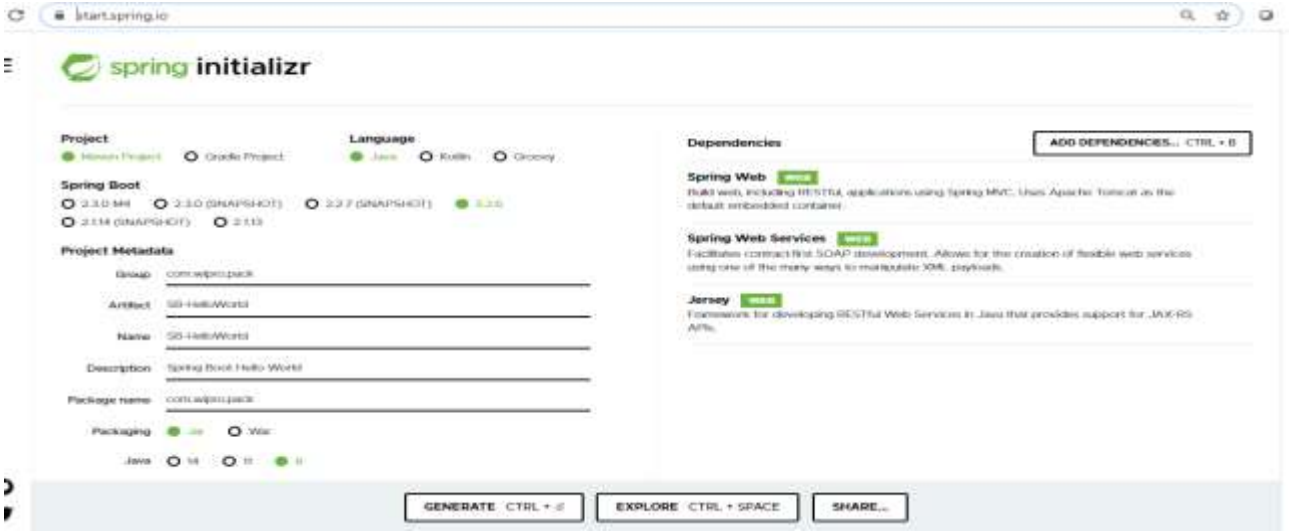
Hello World

Spring Boot Project Creation.

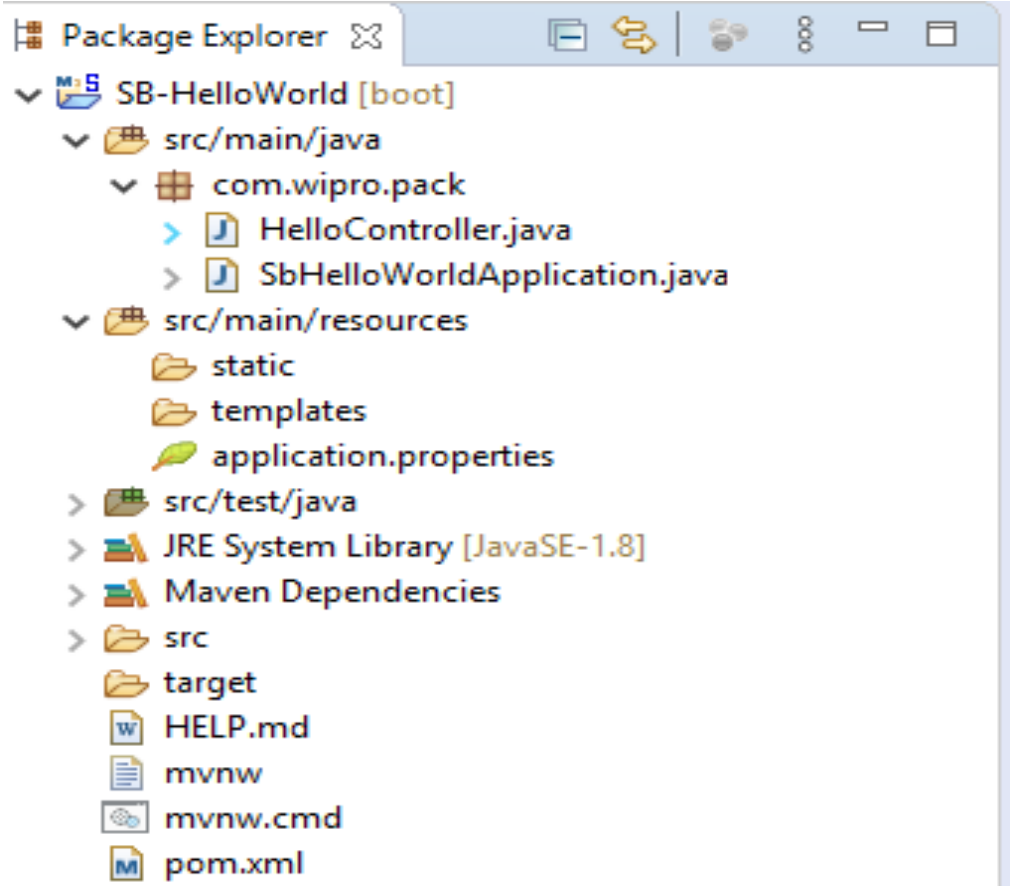
- Using STS :



- Using Spring Initializr : start.spring.io



Spring Boot Project Structure.



Spring Boot – Hello World.

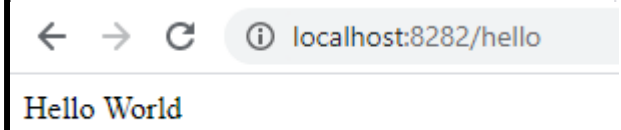
```java
package com.wipro.pack;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class SbHelloWorldApplication {

		public static void main(String[] args) {
		SpringApplication.run(SbHelloWorldApplication.class, args);
		}
}
```

```java
package com.wipro.pack;
import org.springframework.web.bind.annotation.RequestMapping;
import import org.springframework.stereotype.Controller;
@Controller
public class HelloController {

		@RequestMapping("/hello")
		public String sayHello() {
				return "Hello World";
		}
}
```



Summary

In this session, you have learned about:

- Introduction to Spring Boot

Thank you