Spring Boot Web MVC

Introduction to Spring MVC:

MVC Architecture:

- It is an Enterprise Application Architectural design pattern.

- MVC demands layered application development.
  Eg: Constructing a building, managing a Restaurant, etc

- **M** stands for **Model Layer** which contains **Business logic** and **Persistence logics**.
  Eg: **Accountant**

- **V** stands for **View Layer** which contains only **Presentation Logic.**
  Eg: Beautician or Make up man.

- **C** stands for **Controller Layer** which contains **Integration Logic** / **Connectivity Logic**.
  Eg: **Team Lead**

What is Spring MVC?

- It is designed based on the **MVC architectural design pattern**.

Advantages of Spring MVC:

1. Support to develop MVC based web applications.
2. Supports Form backup support
3. Supports to plug with other MVC frameworks like **Struts, JSF, etc.**
4. Supports different view types like JSP, Velocity, XML, PDF, Tiles, etc.

Front Controller:

- It is a **Java EE design pattern.**
- It is responsible for the following activities:
  1. It intercepts every request coming from client and then dispatches or forwards requests to an appropriate **controller**.
  2. It routes all the requests to corresponding controller.
  3. Load and map all urls to handle the requests.
- **DispatcherServlet** is act as front controller in spring mvc

Spring Boot Web MVC:

- Spring Boot has provided one starter for web application.

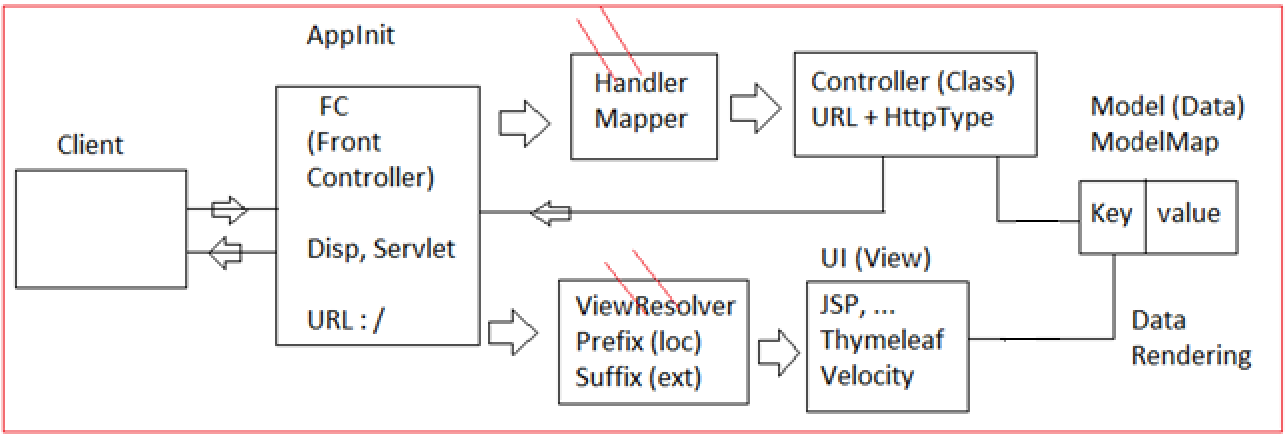
- It is similar to Spring WEB MVC execution process but reduces work done by programmer for,
  a. Maven project creation and setup.
  b. Pom.xml dependencies and plugins.
  c. Writing common code (AppInit, AppConfig).
  d. Handle Runtime Environment and creating JAR/WARs.

- Such process is taken care by spring boot and called as "**AutoConfiguration**".

- Even coming to Handler Mapping is configured by **Front Controller.**

- **ViewResolver** needs not to be configured. But Programmer has to provide (Prefix and Suffix) using properties/yml file.

- Front Controller (**DispatcherServlet**) is configured by spring boot and mapped to URL = "/"



- **Front Controller, ViewResolver, and HandlerMapper** are taken care by **Spring Boot** where Controller and UI files should be defined by Programmer.

- Reading data from **Model (I)** or **ModelMap(C)** at runtime and send to UI is known as **Data Rendering** and it is implemented using **Expression Language Programming**.

- Programmer should provide inputs like **port number** and **view resolver** details using Properties or yml file.
  Example:

  application.properties
  ```
  server.port=8181
  spring.mvc.view.prefix=/WEB-INF/views/
  spring.mvc.view.suffix=.jsp
  ```

- Default port no mapped to ‘**8080’** by using key ‘**server.port’** and it can be changeable.

- Spring boot has provided 3 embedded servers. [No download and No install]
  Eg: Tomcat (default server), Jetty and Undertow.
  Tomcat provided by Apache.
  Jetty Provided by Eclipse.
  Undertow provided by JBoss.

- In general, Tomcat contains 2 engines **Servlet Engine (Catalina)** and **JSP Engine (JASPER).**
  In Spring boot, tomcat comes with only **Servlet Engine**.
  That’s why it is also called as light weight engine that works for “**DispatcherServlet**”, nothing else.

- To work with JSP files in Spring Boot WEB apps, we need to add dependencies in pom.xml.
  Eg:

```
<dependency>
        <groupId>org.apache.tomcat.embed</groupId>
        <artifactId>tomcat-embed-jasper</artifactId>
        <scope>provided</scope> //Version taken care by Spring boot (and Tomcat)
</dependency>
```

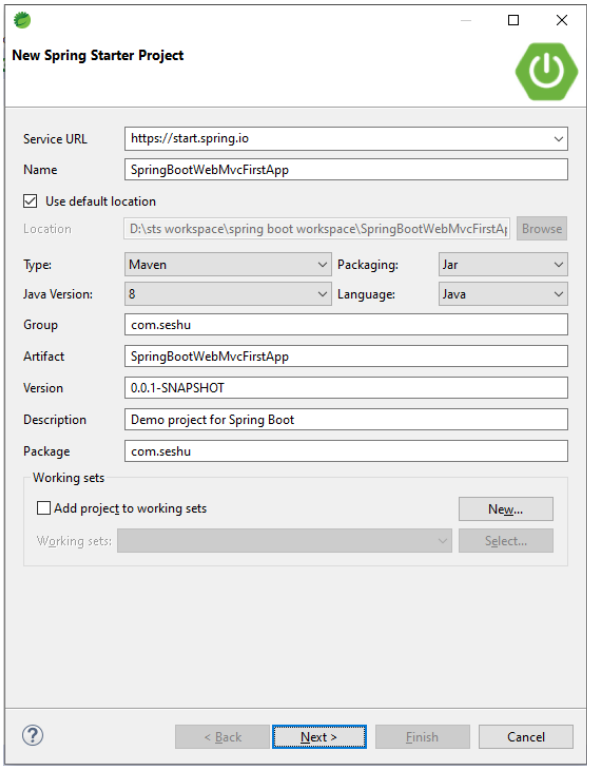
- To avoid/remove tomcat server (default server) from Boot application, we need to add <Exclusion> for Tomcat under web dependency. Given as,
  Eg:

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
        <exclusions>
           <exclusion>
               <groupId>org.springframework.boot</groupId>
               <artifactId>spring-boot-starter-tomcat</artifactId>
           </exclusion>
        </exclusions>
</dependency>
```

- Default Static Resource Handler is added to folder **static** and **template** which are provided under **src/main/resources** folder.

First Spring Boot Web MVC Application

Step 1: Create Spring Starter and add dependencies (**Spring Web, Spring Data JPA, H2 Database**)



New Spring Starter Project

Service URL: https://start.spring.io

Name: SpringBootWebMvcFirstApp

☑ Use default location

Location: D:\sts workspace\spring boot workspace\SpringBootWebMvcFirstAp

Type: Maven Packaging: Jar

Java Version: 8 Language: Java

Group: com.seshu

Artifact: SpringBootWebMvcFirstApp

Version: 0.0.1-SNAPSHOT

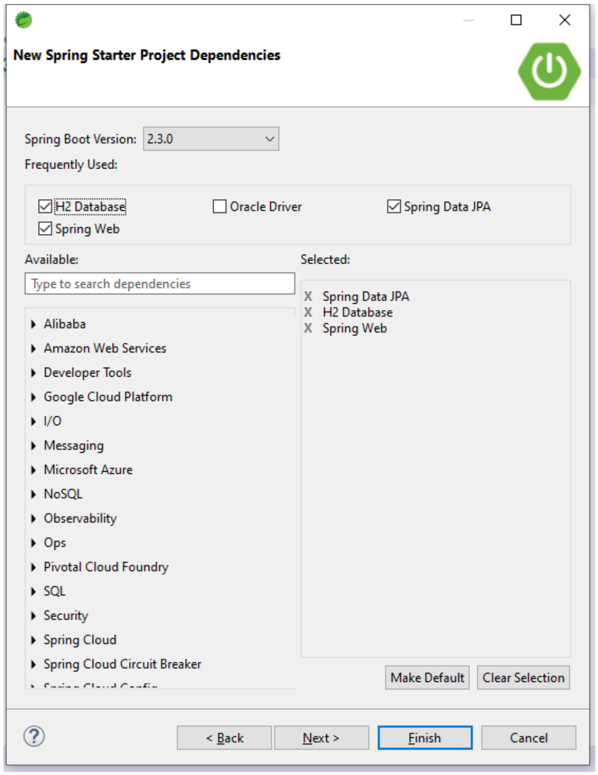
Description: Demo project for Spring Boot

Package: com.seshu

Working sets

☐ Add project to working sets

Working sets:



New Spring Starter Project Dependencies

Spring Boot Version: 2.3.0

Frequently Used:

☑ H2 Database ☐ Oracle Driver ☑ Spring Data JPA

☑ Spring Web

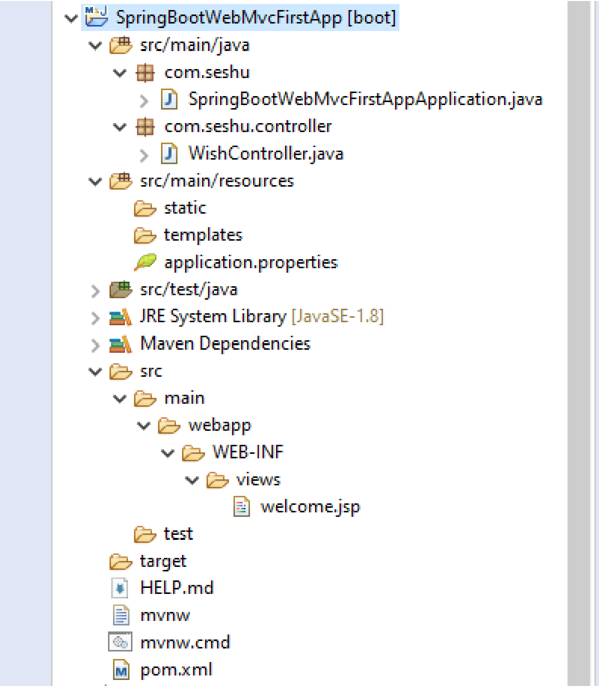
Available:

Type to search dependencies

- Alibaba
- Amazon Web Services
- Developer Tools
- Google Cloud Platform
- I/O
- Messaging
- Microsoft Azure
- NoSQL
- Observability
- Ops
- Pivotal Cloud Foundry
- SQL
- Security
- Spring Cloud
- Spring Cloud Circuit Breaker
- Spring Cloud Config

Selected:

X Spring Data JPA
X H2 Database
X Spring Web



```
SpringBootWebMvcFirstApp [boot]
  src/main/java
    com.seshu
      SpringBootWebMvcFirstAppApplication.java
    com.seshu.controller
      WishController.java
  src/main/resources
    static
    templates
    application.properties
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
    main
      webapp
        WEB-INF
          views
            welcome.jsp
    test
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

Step 2: Open pom.xml file and add below dependency to work with JSP pages only.

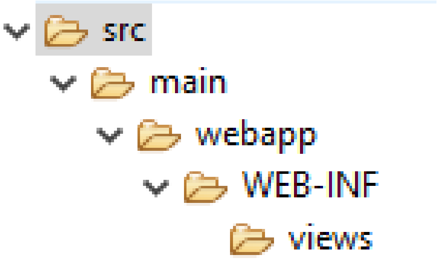
```xml
<dependency>
	<groupId>org.apache.tomcat.embed</groupId>
	<artifactId>tomcat-embed-jasper</artifactId>
	<scope>provided</scope>
</dependency>
```

Step 3: Create fallowing folder structure in src/main/

a. webapp folder under main

b. WEB-INF under webapp

c. views under WEB-INF



```
src
  main
    webapp
      WEB-INF
        views
```

Step 4: - Define WishController class under ‘src/main/java’

```
package com.seshu.controller;

import org.springframework.stereotype.Component;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;

@Component
@RequestMapping("wish")
public class WishController {
      @RequestMapping("welcome")
      public String welcomePage(Model m) {
            m.addAttribute("msg", "Welcome to Spring Boot");
            return "welcome";
      }
}
```

Step 5:- Define welcome.jsp Page under ‘**views’** folder

welcome.jsp

```
${msg}
```

Step 6: - Define **application.properties** given below

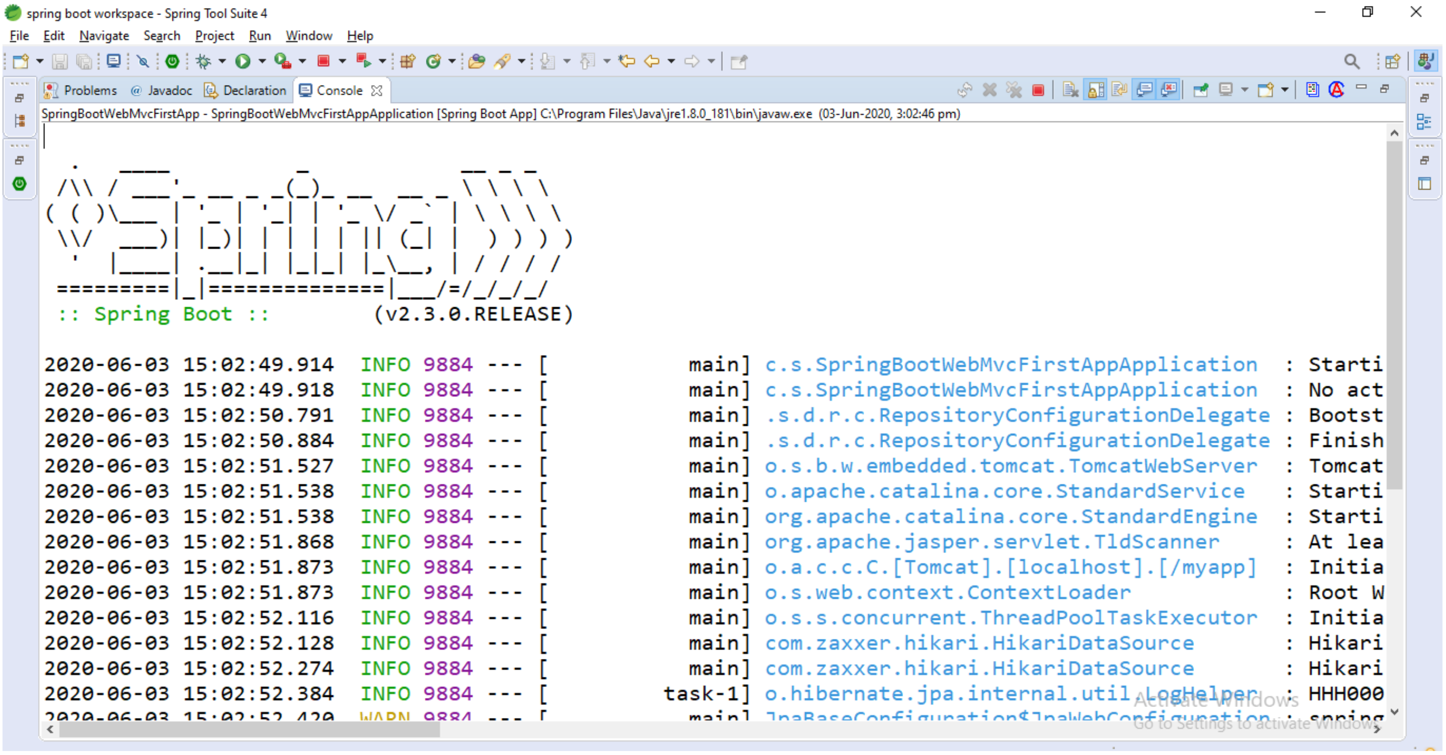
```
server.port=8181
server.servlet.context-path=/myapp
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp
```

Step 7: Run Starter class

```
package com.seshu;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

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


spring boot workspace - Spring Tool Suite 4
File Edit Navigate Search Project Run Window Help
Problems
Javadoc
Declaration
Console

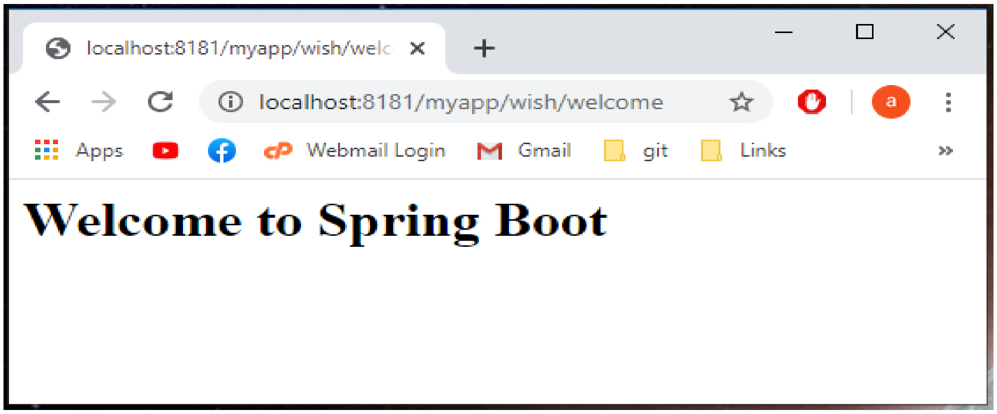
SpringBootWebMvcFirstApp - SpringBootWebMvcFirstAppApplication [Spring Boot App] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (03-Jun-2020, 3:02:46 pm)

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.3.0.RELEASE)

2020-06-03 15:02:49.914  INFO 9884 --- [           main] c.s.SpringBootWebMvcFirstAppApplication  : Starti
2020-06-03 15:02:49.918  INFO 9884 --- [           main] c.s.SpringBootWebMvcFirstAppApplication  : No act
2020-06-03 15:02:50.791  INFO 9884 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootst
2020-06-03 15:02:50.884  INFO 9884 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finish
2020-06-03 15:02:51.527  INFO 9884 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat
2020-06-03 15:02:51.538  INFO 9884 --- [           main] o.apache.catalina.core.StandardService   : Starti
2020-06-03 15:02:51.538  INFO 9884 --- [           main] org.apache.catalina.core.StandardEngine  : Starti
2020-06-03 15:02:51.868  INFO 9884 --- [           main] org.apache.jasper.servlet.TldScanner     : At lea
2020-06-03 15:02:51.873  INFO 9884 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/myapp]  : Initia
2020-06-03 15:02:51.873  INFO 9884 --- [           main] o.s.web.context.ContextLoader            : Root W
2020-06-03 15:02:52.116  INFO 9884 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initia
2020-06-03 15:02:52.128  INFO 9884 --- [           main] com.zaxxer.hikari.HikariDataSource       : Hikari
2020-06-03 15:02:52.274  INFO 9884 --- [           main] com.zaxxer.hikari.HikariDataSource       : Hikari
2020-06-03 15:02:52.384  INFO 9884 --- [         task-1] o.hibernate.jpa.internal.util.LogHelper  : HHH000
2020-06-03 15:02:52.420  WARN 9884 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring
```

Step 8: Open the fallowing url in browser.

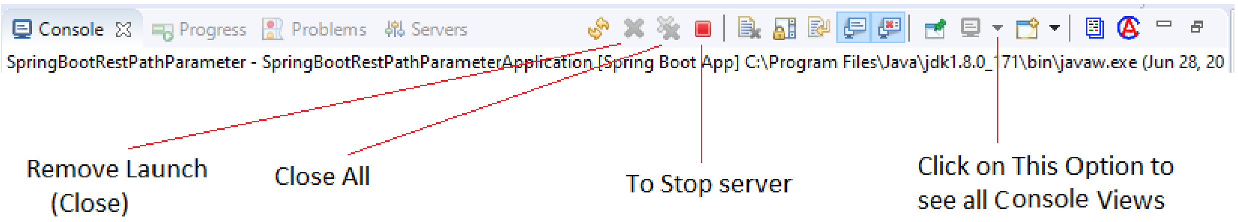
http://localhost:8181/myapp/wish/welcome


localhost:8181/myapp/wish/welc
localhost:8181/myapp/wish/welcome
Apps
Webmail Login
Gmail
git
Links

Welcome to Spring Boot

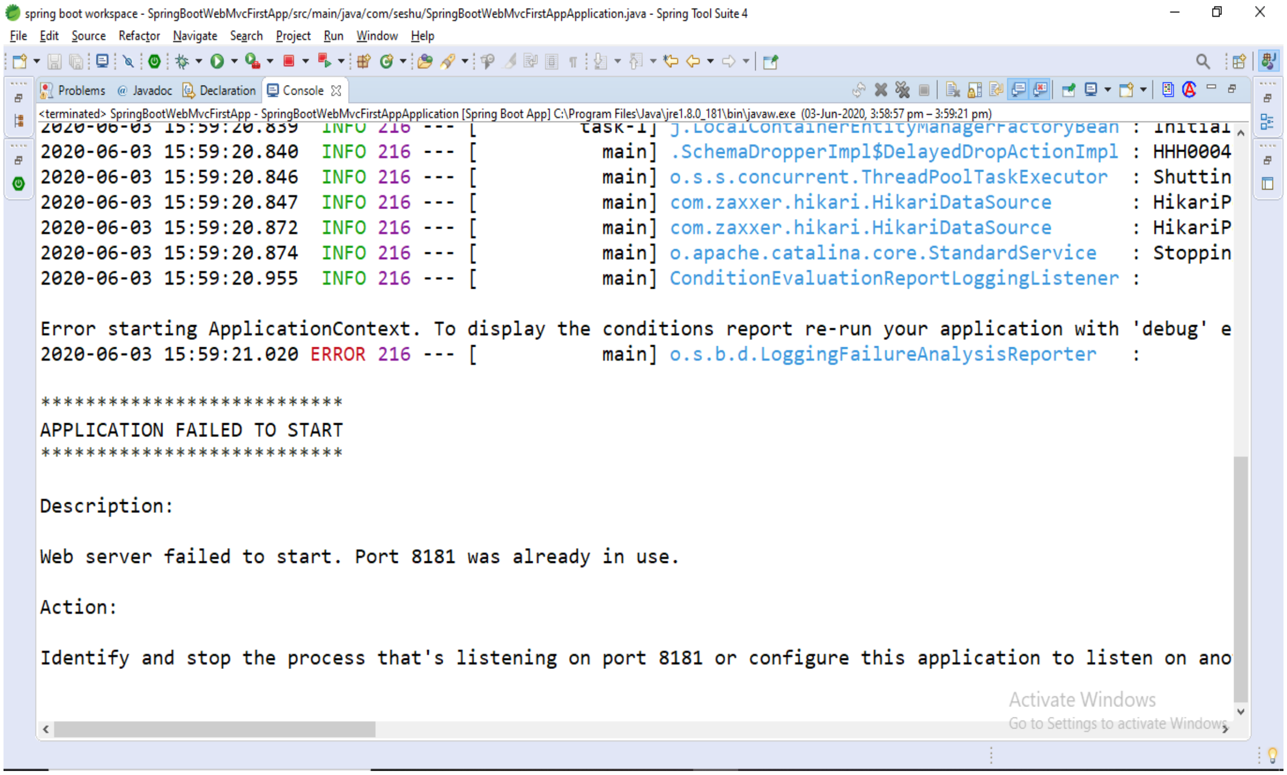
Note:

- We can run Starter class only one time if type is web application with one port number.
  If we want to run again, then must stop last process which is already running. Otherwise we get error.
  ->Go to Console option.
  ->Look at Console Symbol.
  ->Click on DropDown (In case of multiple).
  ->Click on Red Color Box Symbol (Stop Symbol).



```
SpringBootRestPathParameter - SpringBootRestPathParameterApplication [Spring Boot App] C:\Program Files\Java\jdk1.8.0_171\bin\javaw.exe (Jun 28, 20
```

- If we want to run again, without stopping last process we will get error.


spring boot workspace - SpringBootWebMvcFirstApp/src/main/java/com/seshu/SpringBootWebMvcFirstAppApplication.java - Spring Tool Suite 4
File
Edit
Source
Refactor
Navigate
Search
Project
Run
Window
Help
Problems
@ Javadoc
Declaration
Console
Activate Windows
Go to Settings to activate Windows.

```
<terminated> SpringBootWebMvcFirstApp - SpringBootWebMvcFirstAppApplication [Spring Boot App] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (03-Jun-2020, 3:58:57 pm – 3:59:21 pm)
2020-06-03 15:59:20.839  INFO 216 --- [          task-1] j.LocalContainerEntityManagerFactoryBean : Initial
2020-06-03 15:59:20.840  INFO 216 --- [           main] .SchemaDropperImpl$DelayedDropActionImpl : HHH0004
2020-06-03 15:59:20.846  INFO 216 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Shuttin
2020-06-03 15:59:20.847  INFO 216 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariP
2020-06-03 15:59:20.872  INFO 216 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariP
2020-06-03 15:59:20.874  INFO 216 --- [           main] o.apache.catalina.core.StandardService   : Stoppin
2020-06-03 15:59:20.955  INFO 216 --- [           main] ConditionEvaluationReportLoggingListener :

Error starting ApplicationContext. To display the conditions report re-run your application with 'debug' e
2020-06-03 15:59:21.020 ERROR 216 --- [           main] o.s.b.d.LoggingFailureAnalysisReporter   :

***************************
APPLICATION FAILED TO START
***************************

Description:

Web server failed to start. Port 8181 was already in use.

Action:

Identify and stop the process that's listening on port 8181 or configure this application to listen on ano
```