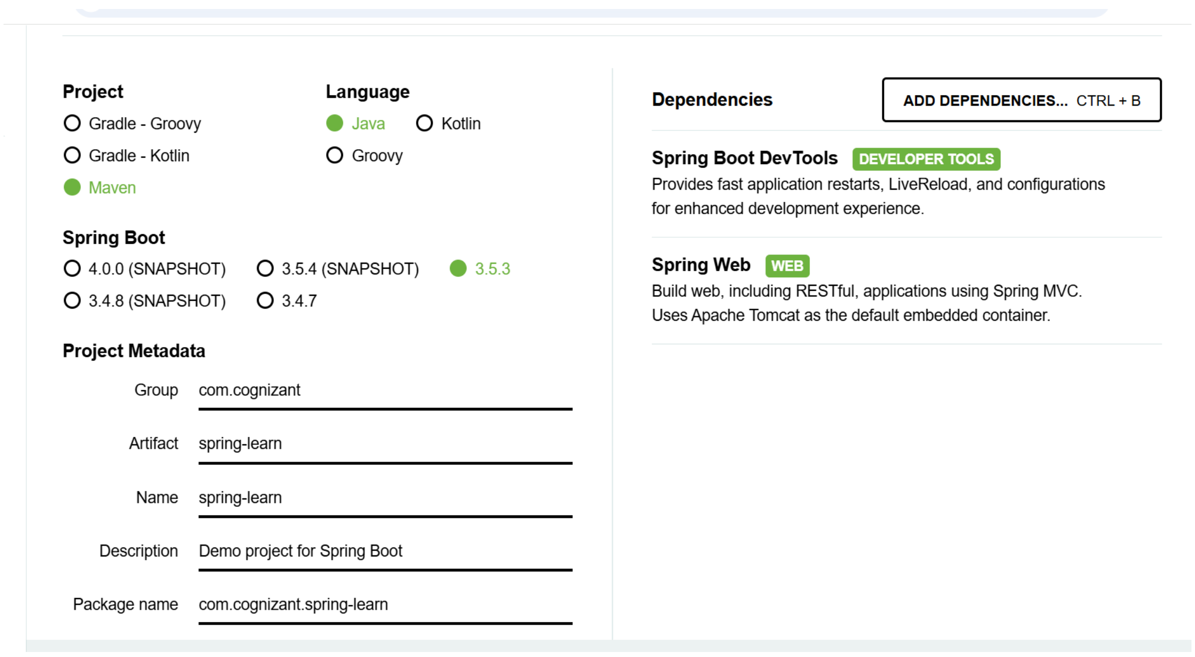
Name:- Vidya C S TRACK: JAVA FSE

I. spring-rest-handson-1

1) Create a Spring Web Project using Maven

Step 1:-Creating and setting up the project with all required dependencies



Project
- Gradle - Groovy
- Gradle - Kotlin
- [✓ Maven]

Language
- [✓ Java]
- Kotlin
- Groovy

Spring Boot
- 4.0.0 (SNAPSHOT)
- 3.5.4 (SNAPSHOT)
- [✓ 3.5.3]
- 3.4.8 (SNAPSHOT)
- 3.4.7

Project Metadata

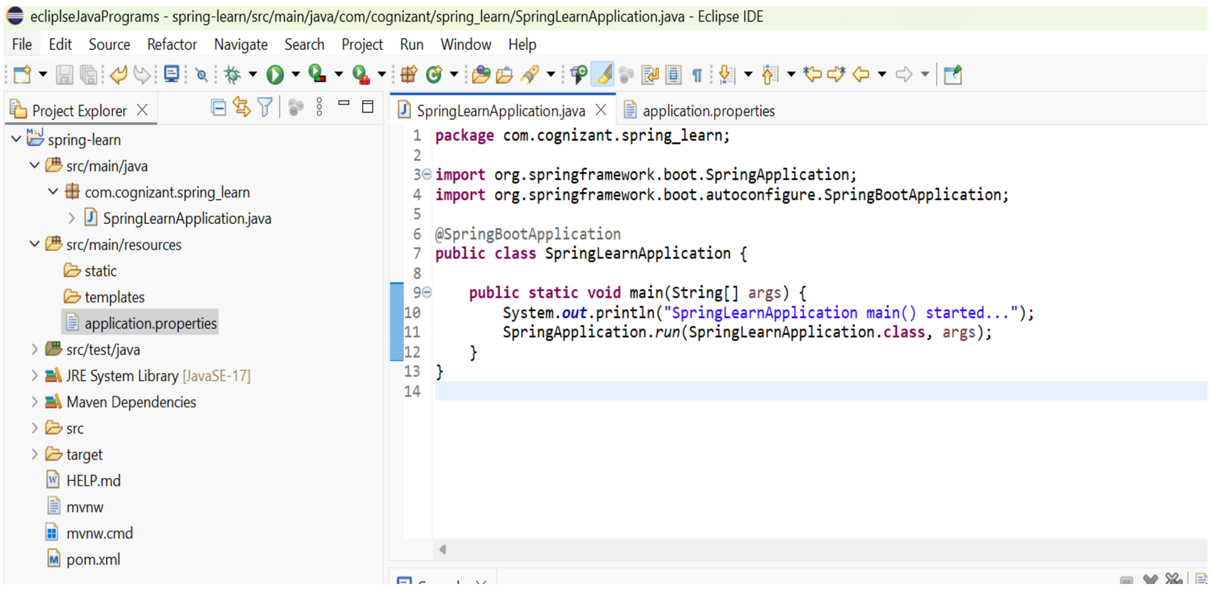
Group	com.cognizant
Artifact	spring-learn
Name	spring-learn
Description	Demo project for Spring Boot
Package name	com.cognizant.spring-learn

Dependencies ADD DEPENDENCIES... CTRL + B

Spring Boot DevTools DEVELOPER TOOLS
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Step 2 :- Filling the up the main function code



eclipseJavaPrograms - spring-learn/src/main/java/com/cognizant/spring_learn/SpringLearnApplication.java - Eclipse IDE

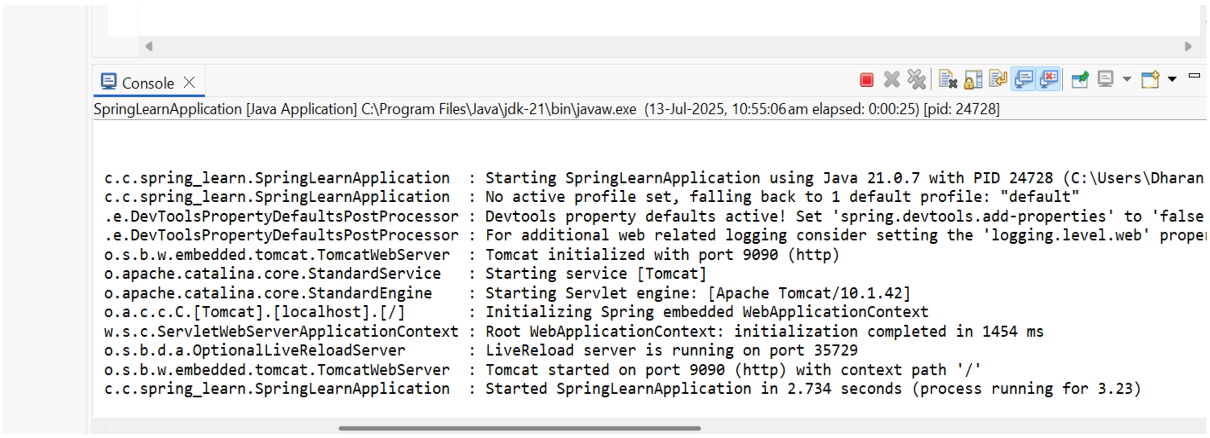
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
SpringLearnApplication.java
application.properties

```
spring-learn
  src/main/java
    com.cognizant.spring_learn
      SpringLearnApplication.java
  src/main/resources
    static
    templates
    application.properties
  src/test/java
  JRE System Library [JavaSE-17]
  Maven Dependencies
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

```java
package com.cognizant.spring_learn;

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

@SpringBootApplication
public class SpringLearnApplication {

    public static void main(String[] args) {
        System.out.println("SpringLearnApplication main() started...");
        SpringApplication.run(SpringLearnApplication.class, args);
    }
}
```

Step 3:- Run the Application

Output :-



Console

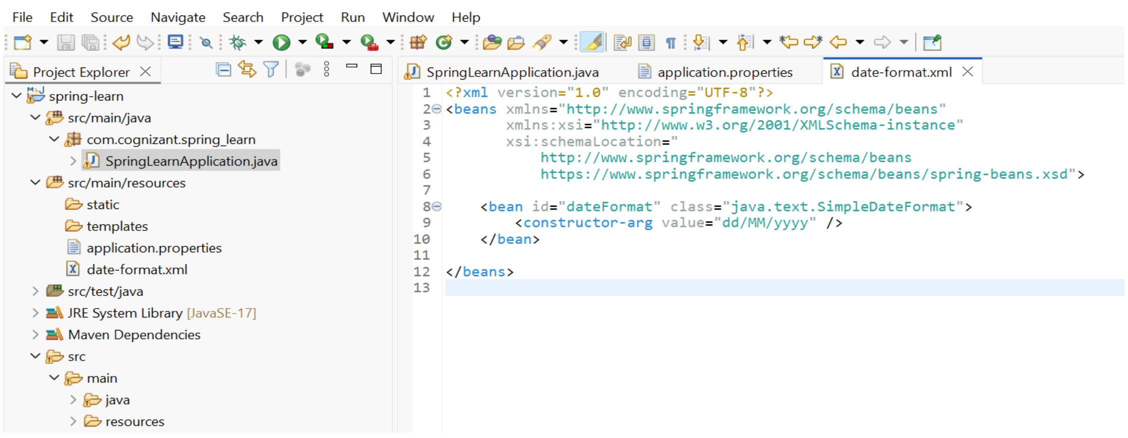
SpringLearnApplication [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (13-Jul-2025, 10:55:06am elapsed: 0:00:25) [pid: 24728]

```
c.c.spring_learn.SpringLearnApplication   : Starting SpringLearnApplication using Java 21.0.7 with PID 24728 (C:\Users\Dharan
c.c.spring_learn.SpringLearnApplication   : No active profile set, falling back to 1 default profile: "default"
.e.DevToolsPropertyDefaultsPostProcessor  : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false
.e.DevToolsPropertyDefaultsPostProcessor  : For additional web related logging consider setting the 'logging.level.web' prope
o.s.b.w.embedded.tomcat.TomcatWebServer   : Tomcat initialized with port 9090 (http)
o.apache.catalina.core.StandardService    : Starting service [Tomcat]
o.apache.catalina.core.StandardEngine     : Starting Servlet engine: [Apache Tomcat/10.1.42]
o.a.c.c.C.[Tomcat].[localhost].[/]        : Initializing Spring embedded WebApplicationContext
w.s.c.ServletWebServerApplicationContext  : Root WebApplicationContext: initialization completed in 1454 ms
o.s.b.d.a.OptionalLiveReloadServer        : LiveReload server is running on port 35729
o.s.b.w.embedded.tomcat.TomcatWebServer   : Tomcat started on port 9090 (http) with context path '/'
c.c.spring_learn.SpringLearnApplication   : Started SpringLearnApplication in 2.734 seconds (process running for 3.23)
```

Hands on 2:

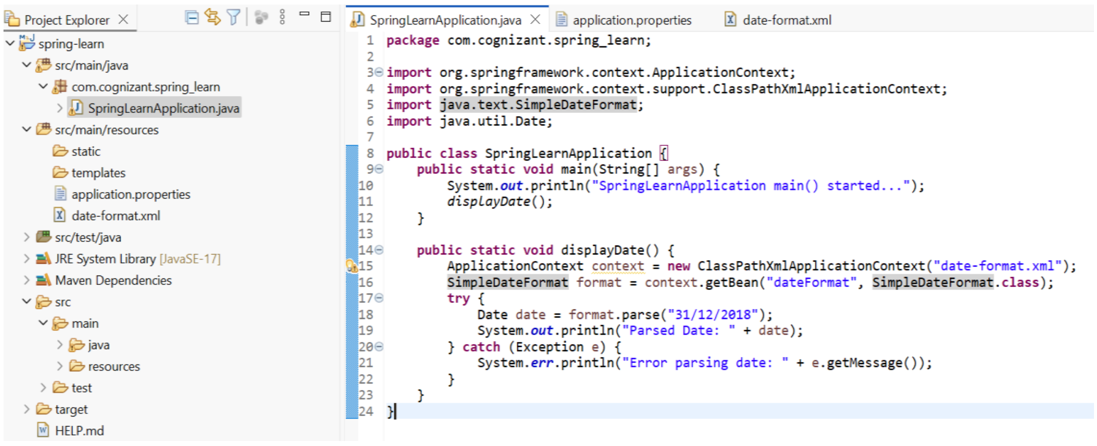
Spring Core – Load SimpleDateFormat from Spring Configuration XML

1. **Create the XML Configuration File**



```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
           http://www.springframework.org/schema/beans
           https://www.springframework.org/schema/beans/spring-beans.xsd">

    <bean id="dateFormat" class="java.text.SimpleDateFormat">
        <constructor-arg value="dd/MM/yyyy" />
    </bean>

</beans>
```

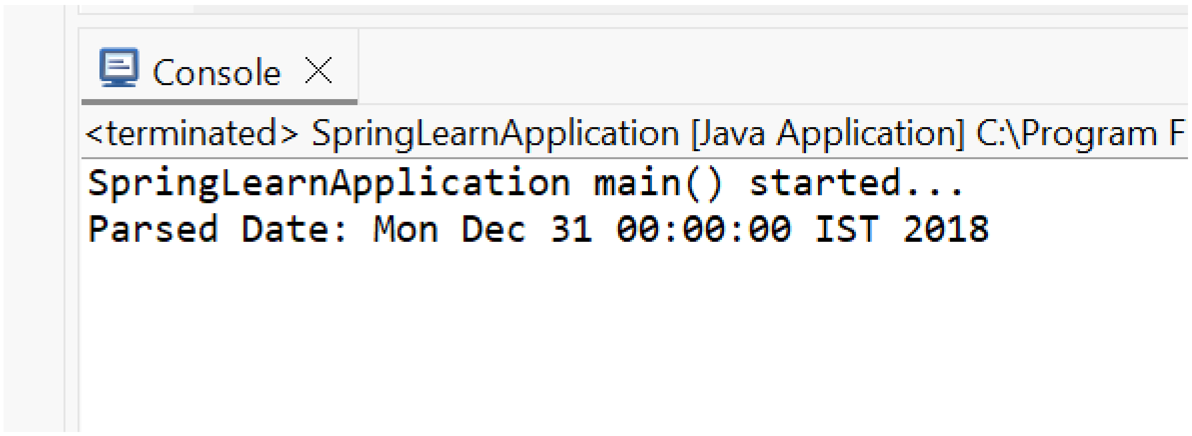
2.Modify SpringLearnApplication.java



```java
package com.cognizant.spring_learn;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import java.text.SimpleDateFormat;
import java.util.Date;

public class SpringLearnApplication {
    public static void main(String[] args) {
        System.out.println("SpringLearnApplication main() started...");
        displayDate();
    }

    public static void displayDate() {
        ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");
        SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);
        try {
            Date date = format.parse("31/12/2018");
            System.out.println("Parsed Date: " + date);
        } catch (Exception e) {
            System.err.println("Error parsing date: " + e.getMessage());
        }
    }
}
```

Step 3:- Run the Application

Output :-

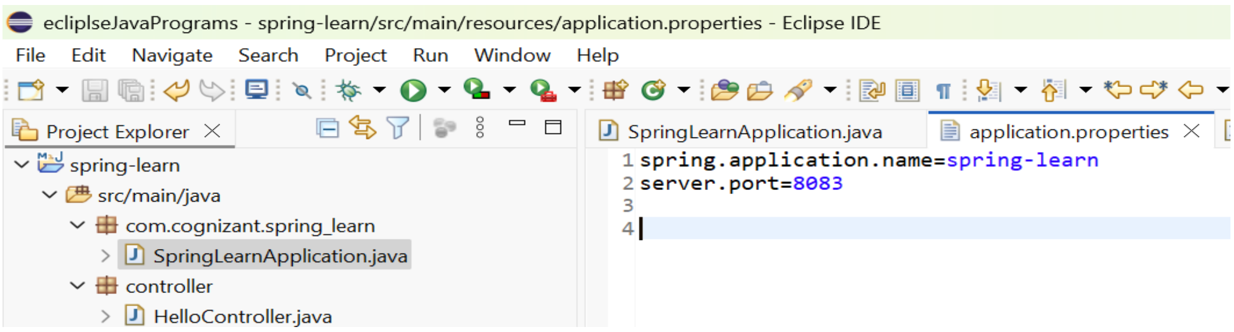


```
<terminated> SpringLearnApplication [Java Application] C:\Program F
SpringLearnApplication main() started...
Parsed Date: Mon Dec 31 00:00:00 IST 2018
```

II. spring-rest-handson-2

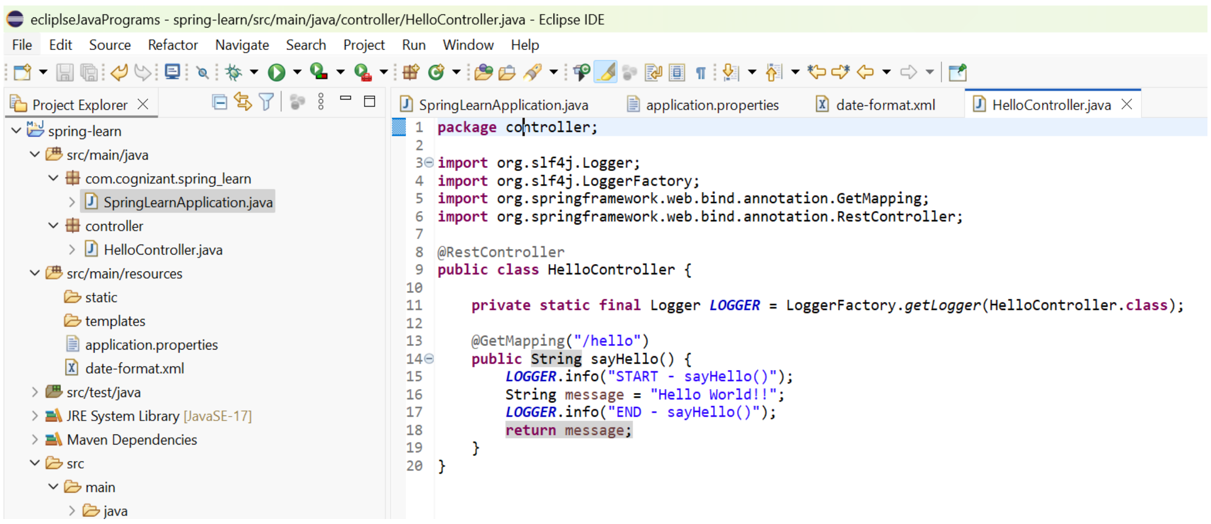
1) Hello World RESTful Web Service

Step 1:- Set Server Port in application.properties


eclipseJavaPrograms - spring-learn/src/main/resources/application.properties - Eclipse IDE
File
Edit
Navigate
Search
Project
Run
Window
Help
Project Explorer
spring-learn
src/main/java
com.cognizant.spring_learn
SpringLearnApplication.java
controller
HelloController.java
SpringLearnApplication.java
application.properties

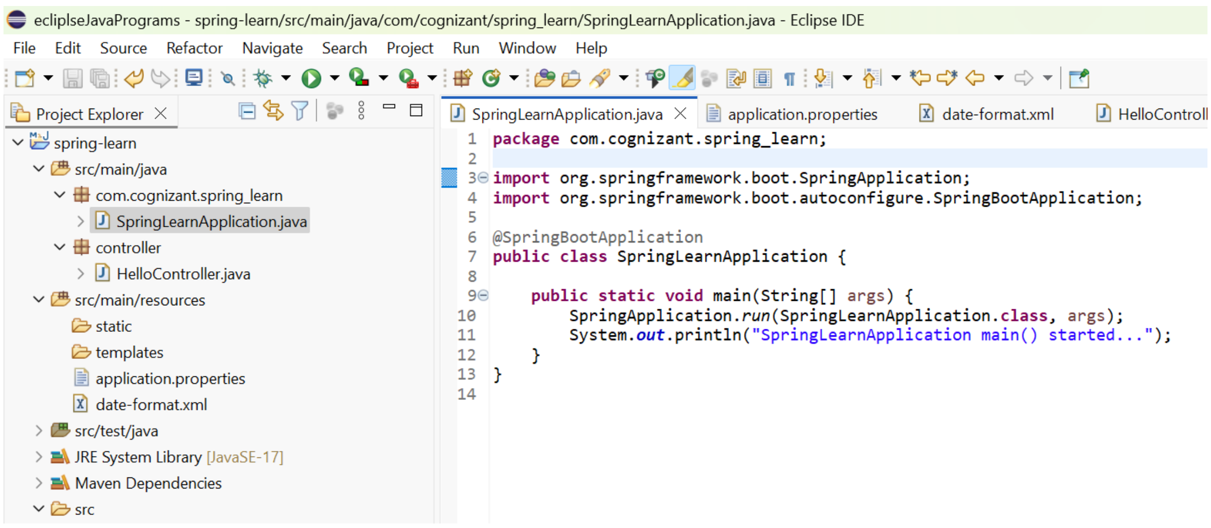
```
spring.application.name=spring-learn
server.port=8083

```

Step 2:- Create the Controller Class


eclipseJavaPrograms - spring-learn/src/main/java/controller/HelloController.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
spring-learn
src/main/java
com.cognizant.spring_learn
SpringLearnApplication.java
controller
HelloController.java
src/main/resources
static
templates
application.properties
date-format.xml
src/test/java
JRE System Library [JavaSE-17]
Maven Dependencies
src
main
java
SpringLearnApplication.java
application.properties
date-format.xml
HelloController.java

```java
package controller;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HelloController {

    private static final Logger LOGGER = LoggerFactory.getLogger(HelloController.class);

    @GetMapping("/hello")
    public String sayHello() {
        LOGGER.info("START - sayHello()");
        String message = "Hello World!!";
        LOGGER.info("END - sayHello()");
        return message;
    }
}
```

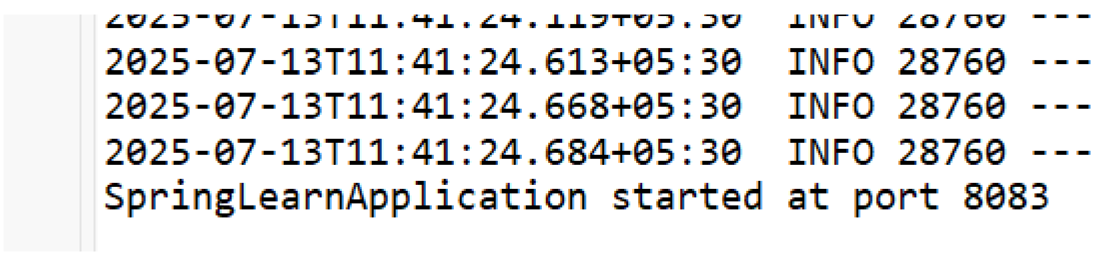
Step 3:- Verify Main Class Has Component Scan


eclipseJavaPrograms - spring-learn/src/main/java/com/cognizant/spring_learn/SpringLearnApplication.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
spring-learn
src/main/java
com.cognizant.spring_learn
SpringLearnApplication.java
controller
HelloController.java
src/main/resources
static
templates
application.properties
date-format.xml
src/test/java
JRE System Library [JavaSE-17]
Maven Dependencies
src
SpringLearnApplication.java
application.properties
date-format.xml
HelloControll

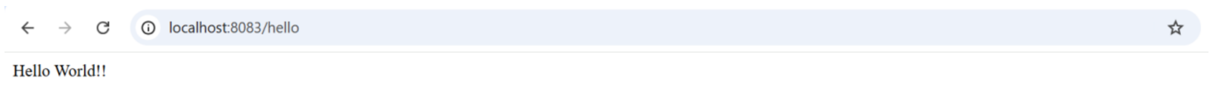
```java
package com.cognizant.spring_learn;

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

@SpringBootApplication
public class SpringLearnApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringLearnApplication.class, args);
        System.out.println("SpringLearnApplication main() started...");
    }
}
```

Step 4:- Run the Application

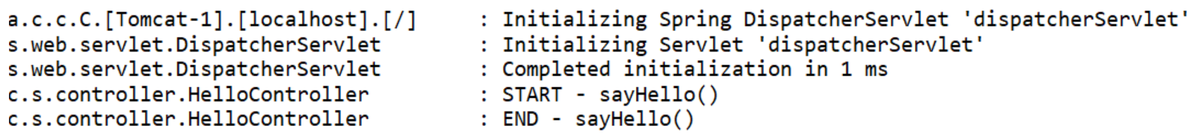
Output:-



```
2025-07-13T11:41:24.119+05:30  INFO 28760 ---
2025-07-13T11:41:24.613+05:30  INFO 28760 ---
2025-07-13T11:41:24.668+05:30  INFO 28760 ---
2025-07-13T11:41:24.684+05:30  INFO 28760 ---
SpringLearnApplication started at port 8083
```



```
Hello World!!
```



```
a.c.c.C.[Tomcat-1].[localhost].[/]      : Initializing Spring DispatcherServlet 'dispatcherServlet'
s.web.servlet.DispatcherServlet         : Initializing Servlet 'dispatcherServlet'
s.web.servlet.DispatcherServlet         : Completed initialization in 1 ms
c.s.controller.HelloController          : START - sayHello()
c.s.controller.HelloController          : END - sayHello()
```