**Digital Nurture 4.0 – Week 4**

Spring REST using Spring Boot 3

HANDSON EXERCISES

Filename: 1. spring-rest-handson

**Handson 1:Create a Spring Web Project using Maven:**

**1. Visit Spring Initializer**

* URL: <https://start.spring.io/>

**2. Fill in Project Metadata:**

| Field | Value |
| --- | --- |
| Project | Maven |
| Language | Java |
| Spring Boot | Choose latest stable version |
| Group | com.cognizant |
| Artifact | spring-learn |
| Name | spring-learn |
| Package Name | com.cognizant.springlearn |
| Packaging | Jar |
| Java Version | 17 (or your version) |

**3. Add Dependencies:**

* Spring Boot DevTools
* Spring Web

Click **GENERATE**, and a .zip file will be downloaded.

**4. Extract ZIP and Set Up Project**

* Extract the downloaded ZIP file into your workspace directory.
* Open **Eclipse IDE**.

**5. Import into Eclipse**

1. File → Import
2. Select **Maven > Existing Maven Projects** → Click Next
3. Click Browse, select the extracted spring-learn folder
4. Click Finish

**6.Build the Project**

open Command Prompt and run this command**:**

mvn clean package

**7. Verify main() Method - Log Output**

**open**

src/main/java/com/cognizant/springlearn/SpringLearnApplication.java

Modify and include a log print inside main():

package com.cognizant.springlearn;

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("Main method executed - SpringLearnApplication Started!");

}

}

**8. Run the Application**

* Right-click SpringLearnApplication.java
* Choose Run As → Java Application

**PROGRAM:**

*SpringLearnApplication.java:*

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

}

*pom.xml:*

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.cognizant</groupId>

<artifactId>spring-learn</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>spring-learn</name>

<description>Spring Boot Web Project - Learn Demo</description>

<packaging>jar</packaging>

<properties>

<java.version>17</java.version>

<spring.boot.version>3.2.4</spring.boot.version> <!-- Or your Spring Boot version -->

</properties>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>${spring.boot.version}</version>

<relativePath/> <!-- Lookup parent from repository -->

</parent>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<version>${spring.boot.version}</version>

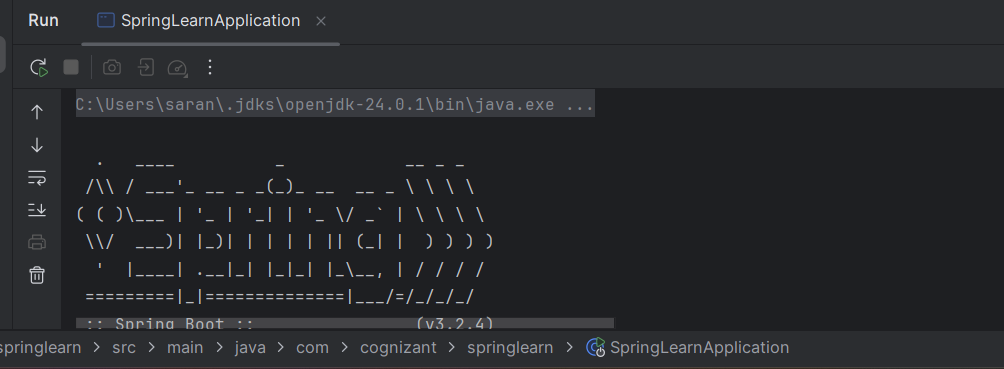
</plugin>

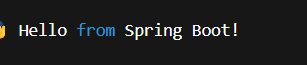
</plugins>

</build>

</project>

**OUTPUT:**

****



**Hands on 4:Spring Core – Load Country from Spring Configuration XML**

Step 1: Create a Java class named Country

\* Define two instance variables: code and name.

\* Create a **no-argument constructor** and add a debug log inside

LOGGER.debug("Inside Country Constructor.");

\*Generate **getters** and **setters** for both code and name.

* + In each getter and setter, add debug messages:

LOGGER.debug("Inside setCode()");

LOGGER.debug("Inside getCode()");

\* Override the toString() method to return a readable string of country details.

**Step 2: Create Spring Configuration XML**

➤ File: country.xml

➤ Location: src/main/resources

1. Use the <bean> tag to declare a bean with:
   * id="country"
   * class="com.cognizant.springlearn.Country"
2. Use <property> tags to set the values of code and name:

<bean id="country" class="com.cognizant.springlearn.Country">

<property name="code" value="IN"/>

<property name="name" value="India"/>

</bean>

**Step 3: Create the Main Application Class**

➤ File: SpringLearnApplication.java

➤ Package: com.cognizant.springlearn

1. Create a static method named displayCountry().
2. Inside this method, do the following:
   * Create the Spring Application Context:

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

* + Get the country bean:

Country country = context.getBean("country", Country.class);

* + Log the country details using:

LOGGER.debug("Country : {}", country.toString());

1. In the main() method, call the displayCountry() method.

**Step 4: Add Dependencies**

Add these dependencies to your pom.xml:

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.30</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>2.0.13</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-simple</artifactId>

<version>2.0.13</version>

</dependency>

</dependencies>

**Step 5: Run the Program**

1. Right-click on SpringLearnApplication.java.
2. Run the main() method.
3. Observe the output in the console.

**PROGRAM:**

*Pom.xml:*

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>springlearn</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <properties>  
 <java.version>24</java.version>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.30</version>  
 </dependency>  
 <dependency>  
 <groupId>org.slf4j</groupId>  
 <artifactId>slf4j-api</artifactId>  
 <version>2.0.13</version>  
 </dependency>  
 <dependency>  
 <groupId>org.slf4j</groupId>  
 <artifactId>slf4j-simple</artifactId>  
 <version>2.0.13</version>  
 </dependency>  
  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.13.2</version>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
   
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.11.0</version>  
 <configuration>  
 <source>24</source>  
 <target>24</target>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

*Country.java:*

package com.cognizant.springlearn;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class Country {  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(Country.class);  
  
 private String code;  
 private String name;  
  
 public Country() {  
 *LOGGER*.debug("Inside Country Constructor.");  
 }  
  
 public String getCode() {  
 *LOGGER*.debug("Inside getCode()");  
 return code;  
 }  
  
 public void setCode(String code) {  
 *LOGGER*.debug("Inside setCode()");  
 this.code = code;  
 }  
  
 public String getName() {  
 *LOGGER*.debug("Inside getName()");  
 return name;  
 }  
  
 public void setName(String name) {  
 *LOGGER*.debug("Inside setName()");  
 this.name = name;  
 }  
  
 @Override  
 public String toString() {  
 return "Country [code=" + code + ", name=" + name + "]";  
 }  
}

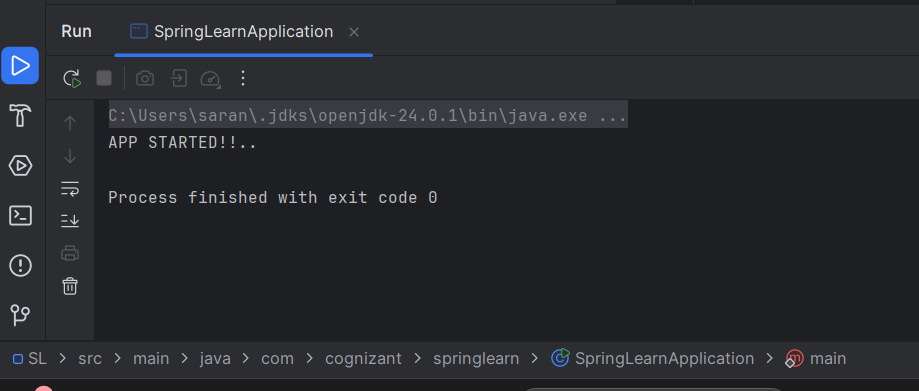
*country.xml:*

<?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  
 http://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="country" class="com.cognizant.springlearn.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
</beans>

*SpringLearnApplication:*

package com.cognizant.springlearn;  
  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class SpringLearnApplication {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(SpringLearnApplication.class);  
  
 public static void displayCountry() {  
 *LOGGER*.debug("START - displayCountry()");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = context.getBean("country", Country.class);  
  
 *LOGGER*.debug("Country : {}", country.toString());  
 *LOGGER*.debug("END - displayCountry()");  
 }  
  
 public static void main(String[] args) {  
 *displayCountry*();  
 System.*out*.println("APP STARTED!!..");  
 }  
}

**Output:**



Filename: 2. spring-rest-handson

**Exercise 1 : Hello WorldRESTful Web Service**

Step 1: Open IntelliJ & Create New Maven Project

1. Open IntelliJ IDEA
2. Click "New Project"
3. Select Maven on the left
4. Click Next

Step 2: Fill Maven Details

| Field | Value |
| --- | --- |
| GroupId | com.cognizant |
| ArtifactId | springlearn |
| Name | springlearn |

Then click Finish

**Step 3: Add Full Working pom.xml**

Right-click the project → Open pom.xml and **replace everything** with this:

**Step 4: Create Main Class**

Right-click src/main/java → New → Package → com.cognizant.springlearn

Then inside that package:

Create a Java class → SpringLearnApplication.java

**Step 5: Create Controller Class**

Right-click com.cognizant.springlearn → New → Package → controller  
Inside controller package, create: HelloController.java

**Step 6: Run the App**

Right-click SpringLearnApplication.java → **Run**

**PROGRAM:**

*Pom.xml:*

<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>springlearn</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <packaging>jar</packaging>  
  
 <properties>  
 <java.version>17</java.version> </properties>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.1.0</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
   
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>

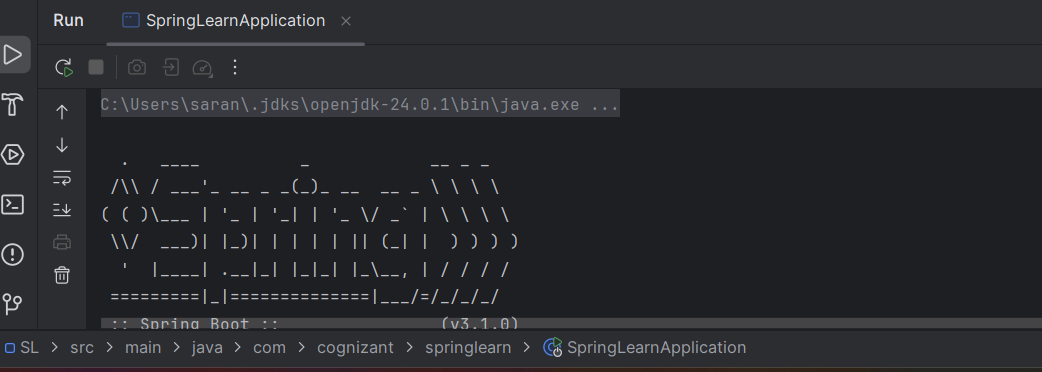
*HelloController.java:*

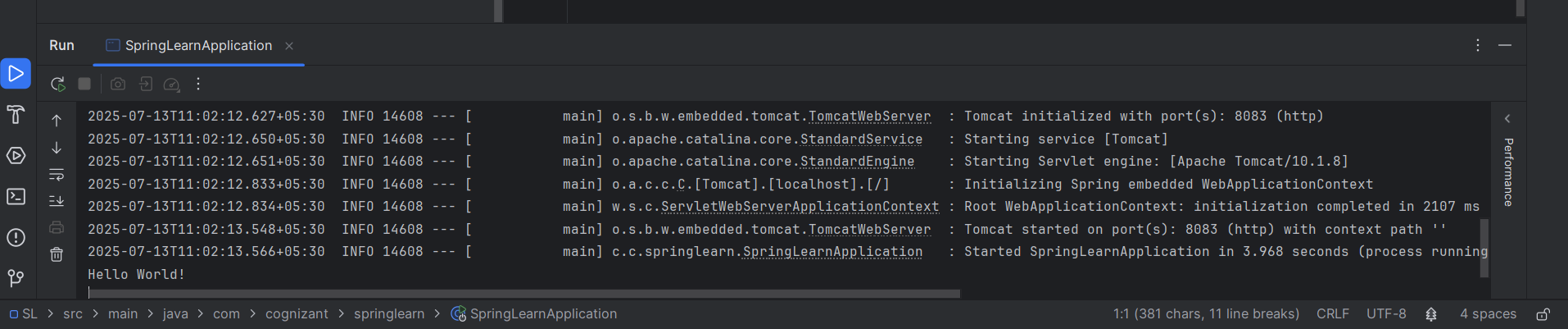
package com.cognizant.springlearn.controller;  
  
  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
  
public class HelloController {  
  
 private static 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;  
 }  
}

*SpringLearnApplication:*

package com.cognizant.springlearn;  
  
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("Hello World!");  
 }  
}

**OUTPUT:**

****

****

**Exercise 2: REST- Country Web Service:**

**Step 1: Create country.xml**

In src/main/resources → Right-click → New → File → country.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

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="country" class="com.cognizant.springlearn.Country">

<property name="code" value="IN"/>

<property name="name" value="India"/>

</bean>

</beans>

**Step 2: Create Country Java Class**

Package: com.cognizant.springlearn  
 File: Country.java

package com.cognizant.springlearn;

public class Country {

private String code;

private String name;

public Country() {

System.out.println("Inside Country Constructor");

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

**Step 3: Create CountryController Class**

Package: com.cognizant.springlearn.controller  
 File: CountryController.java

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

public class CountryController {

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

public Country getCountryIndia() {

LOGGER.info("START - getCountryIndia()");

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

Country country = context.getBean("country", Country.class);

LOGGER.info("END - getCountryIndia()");

return country;

}

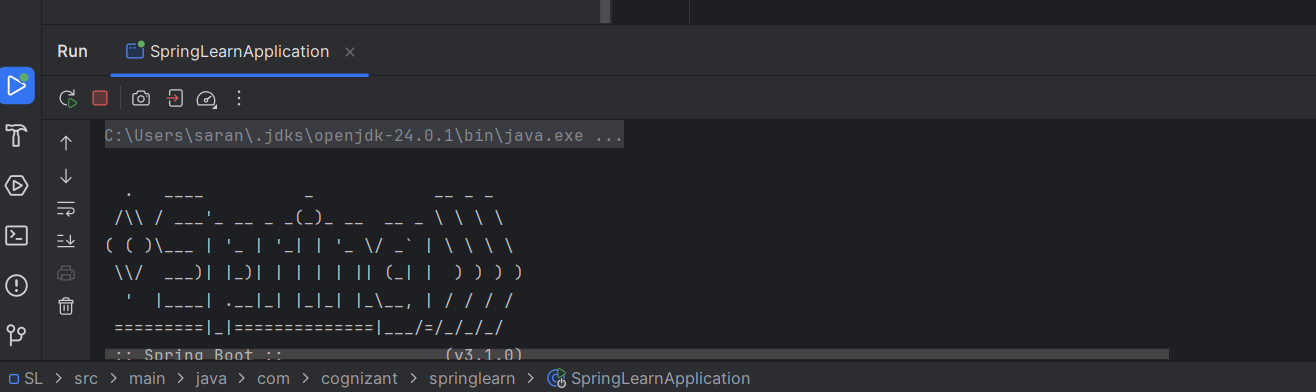
}

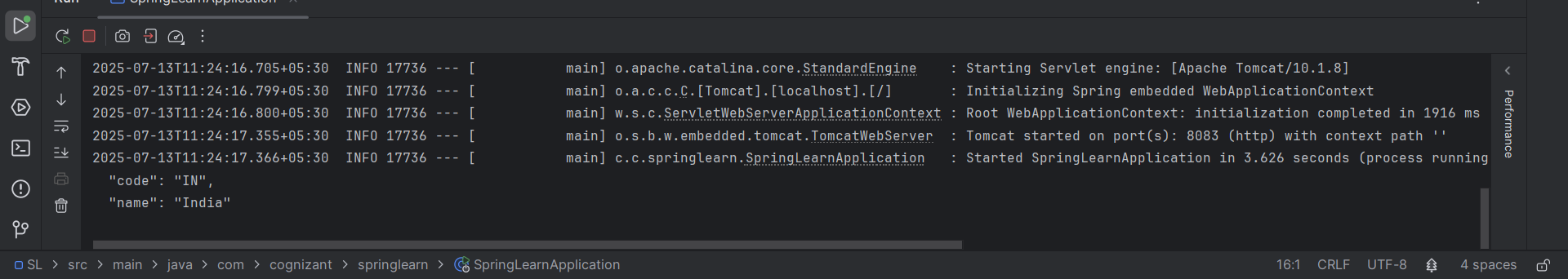
**Step 4: Run the App**

Run SpringLearnApplication.java

package com.cognizant.springlearn;  
  
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);  
 }  
}

**OUTPUT:**



****

**Exercise 3: REST- Get Country based on country code:**

1. Update country.xml to Include All 4 Countries

src/main/resources/country.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

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

<bean class="com.cognizant.springlearn.Country">

<property name="code" value="IN"/>

<property name="name" value="India"/>

</bean>

<bean class="com.cognizant.springlearn.Country">

<property name="code" value="US"/>

<property name="name" value="United States"/>

</bean>

<bean class="com.cognizant.springlearn.Country">

<property name="code" value="DE"/>

<property name="name" value="Germany"/>

</bean>

<bean class="com.cognizant.springlearn.Country">

<property name="code" value="JP"/>

<property name="name" value="Japan"/>

</bean>

</list>

</constructor-arg>

</bean>

</beans>

**2.Country.java**

package com.cognizant.springlearn;

public class Country {

private String code;

private String name;

public Country() {

System.out.println("Inside Country Constructor");

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

**3. CountryService.java**

com.cognizant.springlearn.service

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.Country;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

public Country getCountry(String code) {

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

List<Country> countryList = context.getBean("countryList", List.class);

return countryList.stream()

.filter(c -> c.getCode().equalsIgnoreCase(code))

.findFirst()

.orElse(null);

}

}

**4. CountryController.java** com.cognizant.springlearn.controller

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.Country;

import com.cognizant.springlearn.service.CountryService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;public class CountryController {

@Autowired

private CountryService countryService;

public Country getCountry(@PathVariable String code) {

return countryService.getCountry(code);

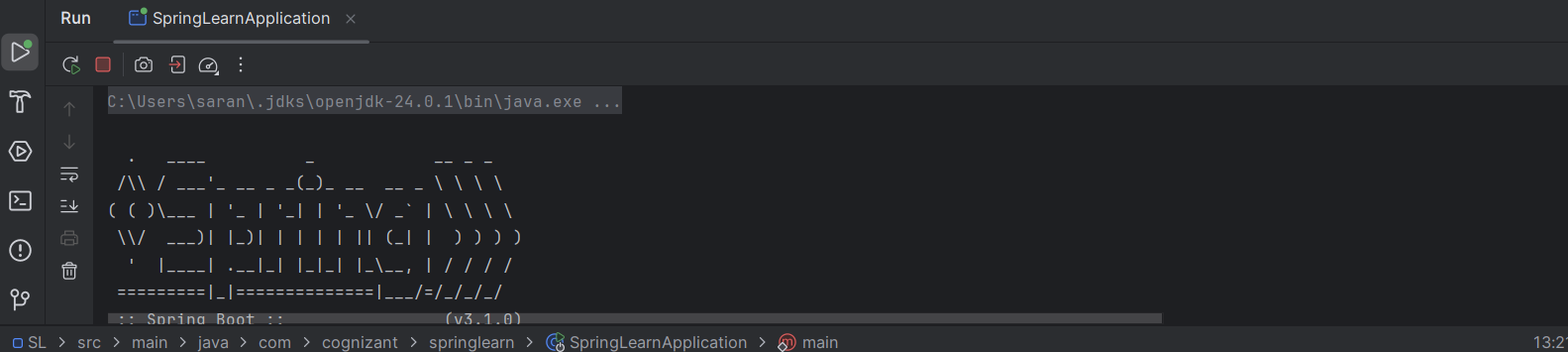
}

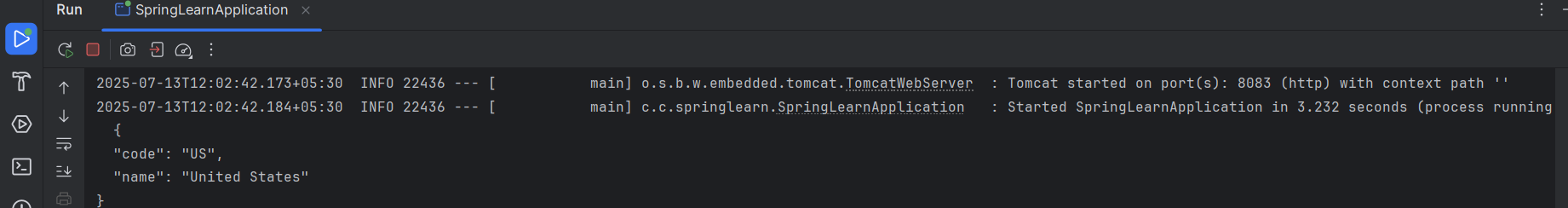
}

**5. Run the App**

Right-click → SpringLearnApplication.java → **Run**

**Output:**



****

Filename: 5. JWT -Handson:

**Exercise 1:Create authentication service that returns JWT**

**Step 1: Add JWT & Spring Security Dependencies to pom.xml**

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt</artifactId>

<version>0.9.1</version>

</dependency>

**Step 2: Create SecurityConfig.java**

package com.cognizant.jwt.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.web.SecurityFilterChain;

@Configuration

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http.csrf().disable()

.authorizeHttpRequests(auth -> auth

.requestMatchers("/authenticate").permitAll()

.anyRequest().authenticated()

)

.httpBasic(); // Enables -u user:pwd style

return http.build();

}

}

**Step 3: Create User in application.properties**

server.port=8090

spring.security.user.name=user

spring.security.user.password=pwd

**Step 4: Create AuthenticationController.java**

package com.cognizant.jwt.controller;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import org.springframework.web.bind.annotation.\*;

import javax.servlet.http.HttpServletRequest;

import java.util.Date;

import java.util.Base64;

@RestController

public class AuthenticationController {

private final String SECRET\_KEY = "mySecretKey";

public String generateToken(HttpServletRequest request) {

// Get "Authorization" header from request

String authHeader = request.getHeader("Authorization");

if (authHeader == null || !authHeader.startsWith("Basic ")) {

throw new RuntimeException("Missing or invalid Authorization header.");

}

// Decode username:password

String base64Credentials = authHeader.substring("Basic ".length());

byte[] credDecoded = Base64.getDecoder().decode(base64Credentials);

String credentials = new String(credDecoded);

String[] values = credentials.split(":", 2);

String username = values[0];

String jwt = Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date())

.setExpiration(new Date(System.currentTimeMillis() + 60 \* 60 \* 1000)) // 1 hour

.signWith(SignatureAlgorithm.HS256, SECRET\_KEY)

.compact();

return "{\"token\":\"" + jwt + "\"}";

}

}

**Step 5: Test with curl**

curl -s -u user:pwd <http://localhost:8090/authenticate>

**Output:**

