WEEK 4 MANDATORY HANDS ON

Spring Rest:

Exersice 1: Create a Spring Web Project using Maven

SpringLearnApplication.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);
    }
}
```

1. Create a REST Controller

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

HelloController.java

```
            package com.cognizant.spring_learn;

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

         @RestController

            public class HelloController {

          @GetMapping("/hello")
```

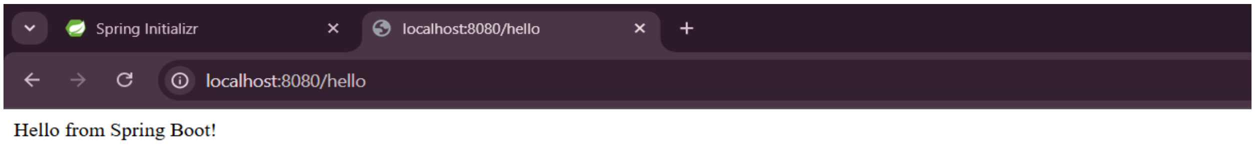
```
    public String sayHello() {
        return "Hello from Spring Boot!";
       }
   }
```

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>

    <parent>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-parent</artifactId>
      <version>3.5.3</version>
      <relativePath/>
    </parent>

    <groupId>com.cognizant</groupId>
    <artifactId>spring-learn</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>spring-learn</name>
    <description>Demo project for Spring Boot</description>

    <properties>
      <java.version>21</java.version>
    </properties>

    <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>3.5.3</version>
            </plugin>
        </plugins>
    </build>

</project>
```

OUTPUT:



Hello from Spring Boot!

Exersice 2:**Spring Core – Load Country from Spring Configuration XML**

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>
```

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>3.5.3</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>

<groupId>com.cognizant</groupId>
<artifactId>spring-learn</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>spring-learn</name>
<description>Demo project for Spring Boot</description>

<properties>
  <java.version>21</java.version>
</properties>

<dependencies>
  <!-- Spring Boot Web Starter -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <!-- Spring Boot DevTools -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
  </dependency>

  <!-- SLF4J Logging -->
  <dependency>
```

```
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-api</artifactId>
      <version>2.0.13</version>
    </dependency>

    <!-- Testing -->
    <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>
```

Java Class:

Country.java:

```
package com.cognizant.spring_learn;
```

```
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 + "]";
    }
}
```

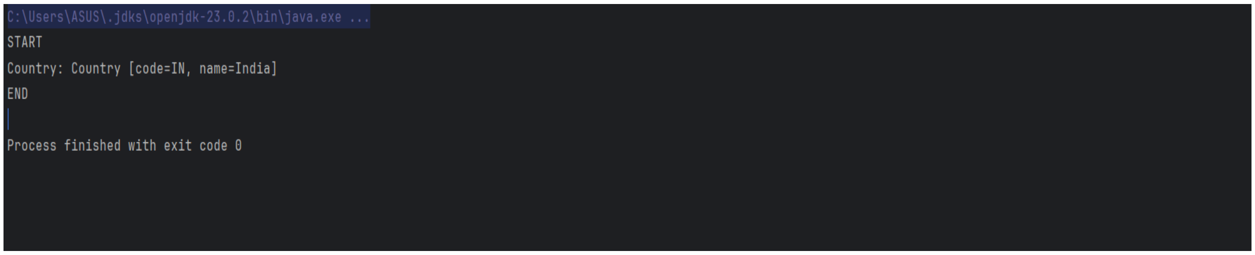
SpringLearnApplication:

```
            package com.cognizant.spring_learn;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringLearnApplication {

    public static void main(String[] args) {
        System.out.println("START");
        displayCountry();
        System.out.println("END");
    }

    public static void displayCountry() {
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
        Country country = (Country) context.getBean("country", Country.class);
        System.out.println("Country: " + country.toString());
    }
}
```

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

    <bean id="country" class="com.cognizant.spring_learn.Country">
        <property name="code" value="IN"/>
        <property name="name" value="India"/>
    </bean>

</beans>
```

OUTPUT:



```
C:\Users\ASUS\.jdks\openjdk-23.0.2\bin\java.exe ...
START
Country: Country [code=IN, name=India]
END

Process finished with exit code 0
```

Exersice 3: **Helloworld RESTFUL webservice**

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>

   <parent>
     <groupId>org.springframework.boot</groupId>
     <artifactId>spring-boot-starter-parent</artifactId>
     <version>3.5.3</version>
     <relativePath/> <!-- lookup parent from repository -->
   </parent>

   <groupId>com.cognizant</groupId>
   <artifactId>spring-learn</artifactId>
   <version>0.0.1-SNAPSHOT</version>
   <name>spring-learn</name>
   <description>Hello World RESTful Web Service Project</description>

   <properties>
     <java.version>21</java.version>
   </properties>

   <dependencies>
     <!-- Spring Boot Web Starter -->
     <dependency>
```

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

      <!-- Spring Boot DevTools (for hot reload during development) -->
      <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
        <scope>runtime</scope>
        <optional>true</optional>
      </dependency>

      <!-- Testing (optional) -->
      <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
      </dependency>
    </dependencies>

    <build>
      <plugins>
        <!-- Spring Boot Maven Plugin -->
        <plugin>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-maven-plugin</artifactId>
          <version>3.5.3</version>
        </plugin>
      </plugins>
    </build>
</project>
```

Java Class:

HelloController.java

```
        package com.cognizant.Spirng_learner.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.debug("START");
        LOGGER.debug("END");
        return "Hello World!!";
    }
}
```

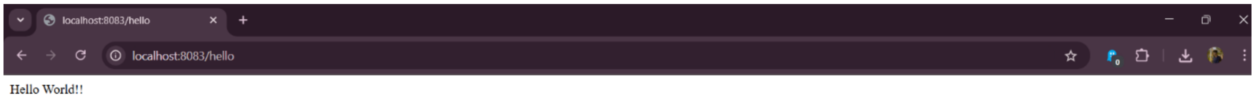
SpringlearnerApplication.java

```
        package com.cognizant.Spirng_learner;

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

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

Application.properties:

```
 spring.application.name=Springlearner
server.port=8083
```

OUTPUT:



```
Hello World!!
```

Exersice 4 :

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>

    <parent>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-parent</artifactId>
      <version>3.5.3</version>
      <relativePath/> <!-- lookup parent from repository -->
    </parent>

    <groupId>com.cognizant</groupId>
    <artifactId>spring-learn</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>spring-learn</name>
```

```
<description>Demo project for Spring Boot</description>

<properties>
  <java.version>21</java.version>
</properties>

<dependencies>
  <!-- Spring Boot Web Starter -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <!-- Spring Boot DevTools -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
  </dependency>

  <!-- SLF4J Logging -->
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>2.0.13</version>
  </dependency>

  <!-- Testing -->
  <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>
```

Java class:

Package : controller

Class :CountryController

CountryController.java

```
package com.cognizant.spring_learn.controller;

import com.cognizant.spring_learn.model.Country;
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;

@RestController
public class CountryController {

    @RequestMapping("/country")
    public Country getCountryIndia() {
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
        Country country = (Country) context.getBean("in");
        return country;
    }
}
```

Package : model

Class : Country

Country.java:

```
        package com.cognizant.spring_learn.model;

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 + "]";
```

```
   }
}
```

SpringLearnApplication.java

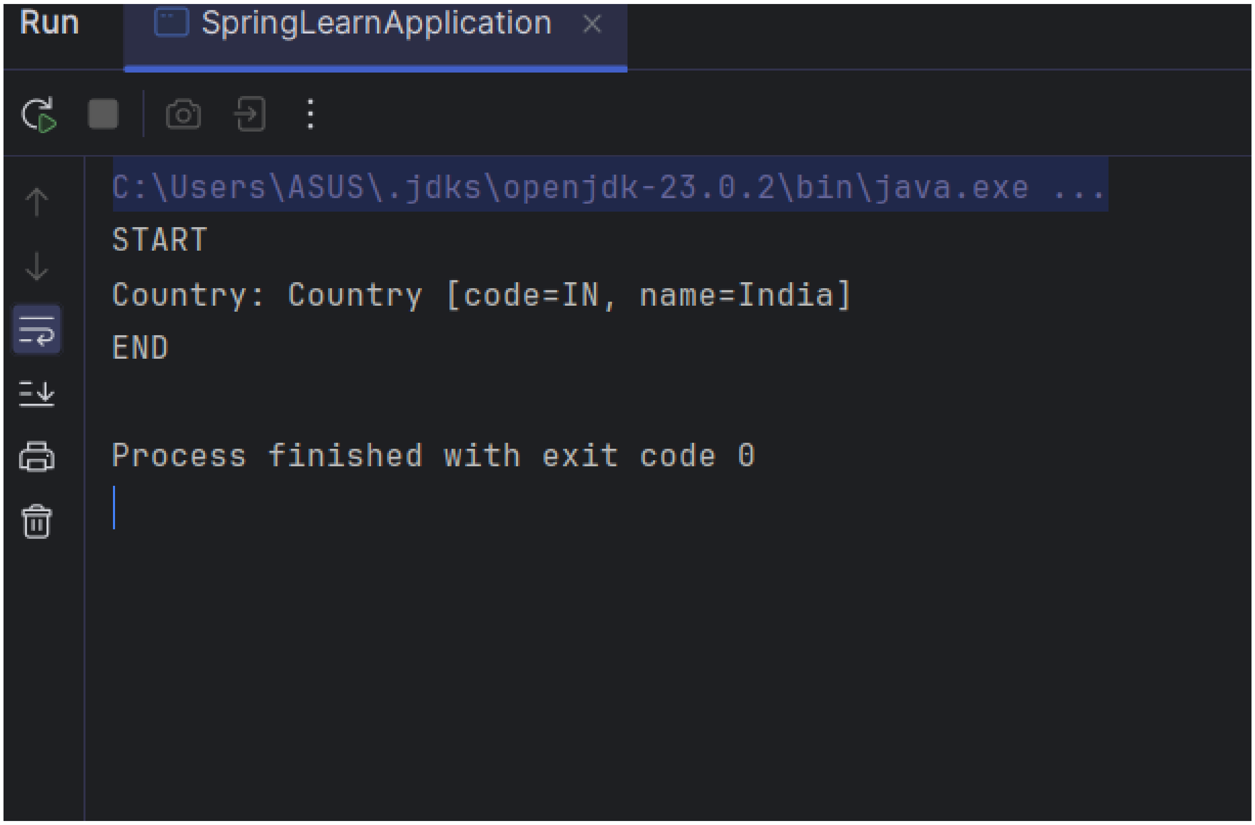
```
            package com.cognizant.spring_learn;

import com.cognizant.spring_learn.model.Country;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringLearnApplication {

   public static void main(String[] args) {
      System.out.println("START");
      displayCountry();
      System.out.println("END");
   }

   public static void displayCountry() {
      ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
      Country country = (Country) context.getBean("in");

      System.out.println("Country: " + country.toString());
   }
}
```

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="in" class="com.cognizant.spring_learn.model.Country">
       <property name="code" value="IN"/>
```

```
        <property name="name" value="India"/>
    </bean>
</beans>
```

OUTPUT:



```
C:\Users\ASUS\.jdks\openjdk-23.0.2\bin\java.exe ...
START
Country: Country [code=IN, name=India]
END

Process finished with exit code 0
```

Excersice 5: **REST - Get country based on country code**

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

    <bean id="countryList" class="java.util.ArrayList">
        <constructor-arg>
            <list>
                <bean class="com.cognizant.spring_learn.model.Country">
                    <property name="code" value="IN"/>
                    <property name="name" value="India"/>
                </bean>
                <bean class="com.cognizant.spring_learn.model.Country">
                    <property name="code" value="US"/>
                    <property name="name" value="United States"/>
                </bean>
                <bean class="com.cognizant.spring_learn.model.Country">
                    <property name="code" value="JP"/>
                    <property name="name" value="Japan"/>
                </bean>
            </list>
        </constructor-arg>
    </bean>
</beans>
```

CountryService.java:

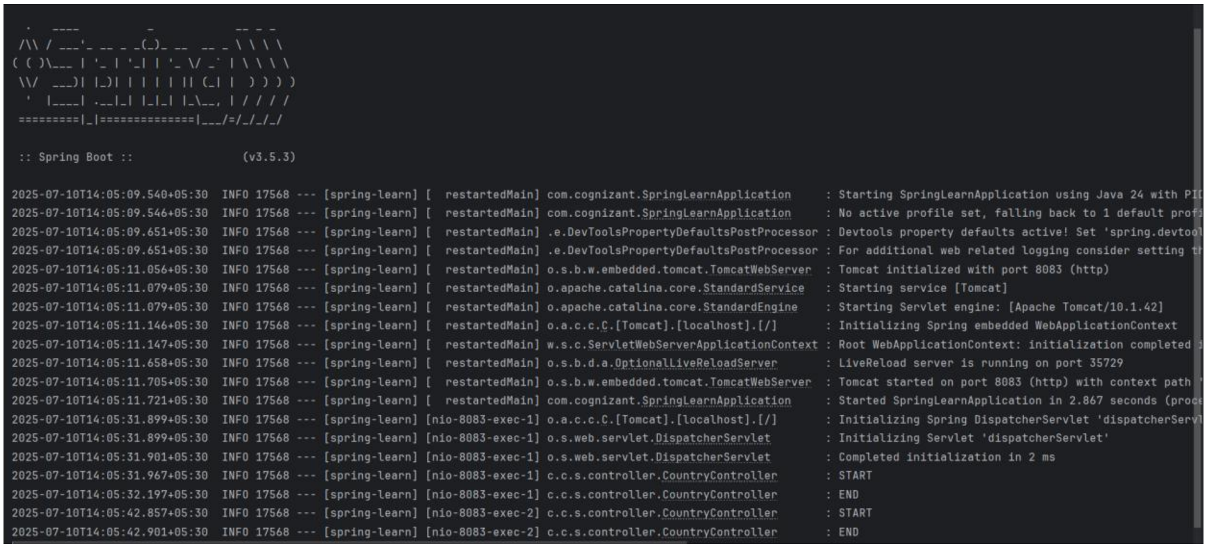
```
package com.cognizant.spring_learn.service;

import com.cognizant.spring_learn.model.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(country -> country.getCode().equalsIgnoreCase(code))
                .findFirst()
                .orElse(null);
    }
}
```

CountryController.java:

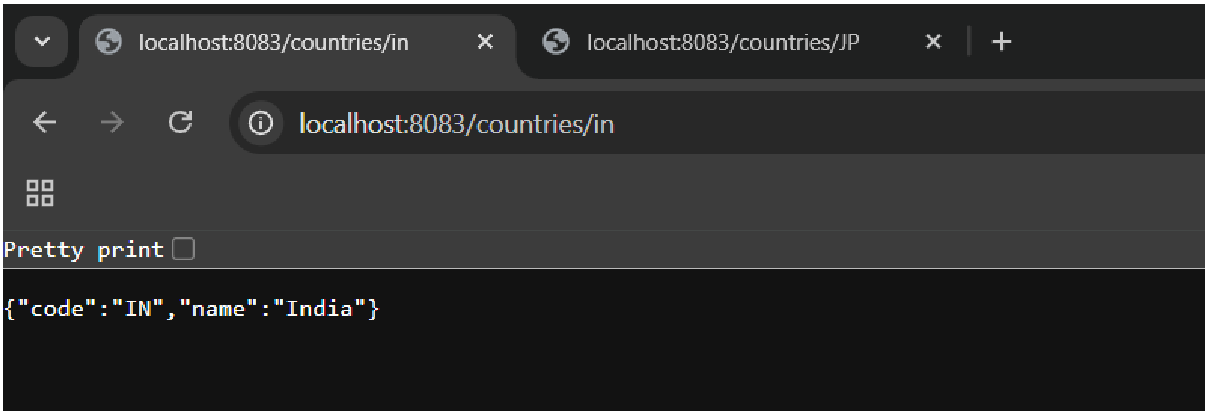
```
package com.cognizant.spring_learn.controller;

import com.cognizant.spring_learn.model.Country;
import com.cognizant.spring_learn.service.CountryService;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.web.bind.annotation.*;

@RestController
public class CountryController {

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

    @Autowired
    private CountryService countryService;

    @RequestMapping("/country")
    public Country getCountryIndia() {
        LOGGER.info("START");
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
        Country country = context.getBean("in", Country.class);
        LOGGER.info("END");
        return country;
    }

    @GetMapping("/countries/{code}")
    public Country getCountry(@PathVariable String code) {
        LOGGER.info("START");
        Country country = countryService.getCountry(code);
        LOGGER.info("END");
        return country;
    }
}.
```

Output:

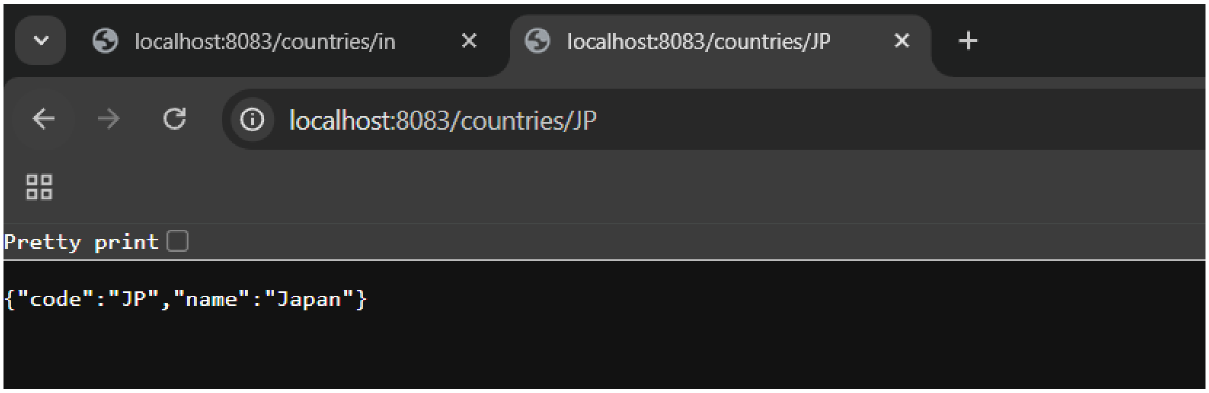


```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v3.5.3)

2025-07-10T14:05:09.540+05:30  INFO 17568 --- [spring-learn] [  restartedMain] com.cognizant.SpringLearnApplication     : Starting SpringLearnApplication using Java 24 with PI
2025-07-10T14:05:09.546+05:30  INFO 17568 --- [spring-learn] [  restartedMain] com.cognizant.SpringLearnApplication     : No active profile set, falling back to 1 default profi
2025-07-10T14:05:09.651+05:30  INFO 17568 --- [spring-learn] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtool
2025-07-10T14:05:09.651+05:30  INFO 17568 --- [spring-learn] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting th
2025-07-10T14:05:11.056+05:30  INFO 17568 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8083 (http)
2025-07-10T14:05:11.079+05:30  INFO 17568 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2025-07-10T14:05:11.079+05:30  INFO 17568 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.42]
2025-07-10T14:05:11.146+05:30  INFO 17568 --- [spring-learn] [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2025-07-10T14:05:11.147+05:30  INFO 17568 --- [spring-learn] [  restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed 
2025-07-10T14:05:11.658+05:30  INFO 17568 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
2025-07-10T14:05:11.705+05:30  INFO 17568 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8083 (http) with context path 
2025-07-10T14:05:11.721+05:30  INFO 17568 --- [spring-learn] [  restartedMain] com.cognizant.SpringLearnApplication     : Started SpringLearnApplication in 2.867 seconds (proce
2025-07-10T14:05:31.899+05:30  INFO 17568 --- [spring-learn] [nio-8083-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServl
2025-07-10T14:05:31.899+05:30  INFO 17568 --- [spring-learn] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2025-07-10T14:05:31.901+05:30  INFO 17568 --- [spring-learn] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 2 ms
2025-07-10T14:05:31.967+05:30  INFO 17568 --- [spring-learn] [nio-8083-exec-1] c.c.s.controller.CountryController       : START
2025-07-10T14:05:32.197+05:30  INFO 17568 --- [spring-learn] [nio-8083-exec-1] c.c.s.controller.CountryController       : END
2025-07-10T14:05:42.857+05:30  INFO 17568 --- [spring-learn] [nio-8083-exec-2] c.c.s.controller.CountryController       : START
2025-07-10T14:05:42.901+05:30  INFO 17568 --- [spring-learn] [nio-8083-exec-2] c.c.s.controller.CountryController       : END
```

In browser:


localhost:8083/countries/in
localhost:8083/countries/JP
localhost:8083/countries/in
Pretty print □

```
{"code":"IN","name":"India"}
```


localhost:8083/countries/in
localhost:8083/countries/JP
localhost:8083/countries/JP
Pretty print □

```
{"code":"JP","name":"Japan"}
```

Exersice 6: **Create authentication service that returns JWT**

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>

    <parent>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-parent</artifactId>
      <version>3.5.3</version>
      <relativePath/>
    </parent>

    <groupId>com.cognizant</groupId>
    <artifactId>jwt-auth</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>jwt-auth</name>
    <description>JWT Auth Project</description>
```

```xml
<properties>
  <java.version>24</java.version>
</properties>

<dependencies>
  <!-- Spring Boot Web and Security -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
  </dependency>

  <!-- JSON Web Token - JJWT -->
  <dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt-api</artifactId>
    <version>0.11.5</version>
  </dependency>
  <dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt-impl</artifactId>
    <version>0.11.5</version>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt-jackson</artifactId>
    <version>0.11.5</version>
    <scope>runtime</scope>
  </dependency>
```

```
      <!-- Servlet API -->
      <dependency>
        <groupId>jakarta.servlet</groupId>
        <artifactId>jakarta.servlet-api</artifactId>
        <version>6.0.0</version>
        <scope>provided</scope>
      </dependency>

      <!-- Development and Testing -->
      <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>
      <dependency>
        <groupId>org.springframework.security</groupId>
        <artifactId>spring-security-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>
```

- Create package named jwt_auth and config,controller
- Path src/main/java/com/cognizant/jwt_auth/controller, src/main/java/com/cognizant/jwt_auth/config.
- Then create a class named AuthenticationController.java in src/main/java/com/cognizant/jwt_auth/controller/AuthenticationController.java.

AuthenticationController.java:

```
package com.cognizant.jwt_auth.controller;

import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.SignatureAlgorithm;
import io.jsonwebtoken.security.Keys;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
import jakarta.servlet.http.HttpServletRequest;
import java.security.Key;
import java.util.Base64;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;

@RestController
public class AuthenticationController {

  private final Key secretKey =
Keys.hmacShaKeyFor("ThisIsASecretKeyThatIsAtLeast32Chars!".getBytes());
```

```
    @GetMapping("/authenticate")
    public Map<String, String> authenticate(HttpServletRequest request) {
        String authHeader = request.getHeader("Authorization");

        if (authHeader == null || !authHeader.startsWith("Basic ")) {
            throw new RuntimeException("Missing or invalid Authorization header");
        }

        String base64Credentials = authHeader.substring("Basic ".length());
        String credentials = new
String(Base64.getDecoder().decode(base64Credentials));
        String[] values = credentials.split(":", 2);

        String username = values[0];
        String password = values[1];

        if (!username.equals("user") || !password.equals("pwd")) {
            throw new RuntimeException("Invalid credentials");
        }

        String token = Jwts.builder()
                .setSubject(username)
                .setIssuedAt(new Date())
                .setExpiration(new Date(System.currentTimeMillis() + 600000)) // 10
minutes
                .signWith(secretKey, SignatureAlgorithm.HS256)
                .compact();

        Map<String, String> response = new HashMap<>();
        response.put("token", token);
        return response;
    }
}
```

SecurityConfig.java:

```
package com.cognizant.jwt_auth.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.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.provisioning.InMemoryUserDetailsManager;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
@EnableWebSecurity
public class SecurityConfig {

    @Bean
    public UserDetailsService userDetailsService() {
        UserDetails user = User
                .withUsername("user")
                .password("{noop}pwd") // {noop} means no encoding
                .roles("USER")
                .build();
        return new InMemoryUserDetailsManager(user);
    }

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http.csrf(csrf -> csrf.disable())
                .authorizeHttpRequests(auth -> auth
                        .requestMatchers("/authenticate").authenticated()
                        .anyRequest().permitAll()
                )
                .httpBasic(); // enables basic auth
        return http.build();
    }
}
```

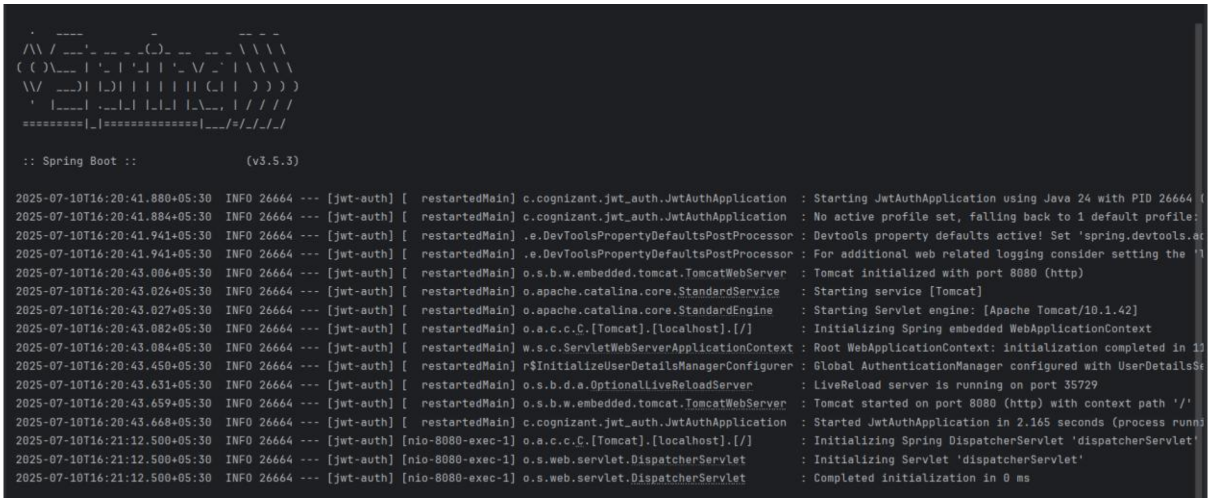
Application.properties:

```
spring.application.name=jwt-auth
server.port=8080
```

JwtAuthApplication.java:

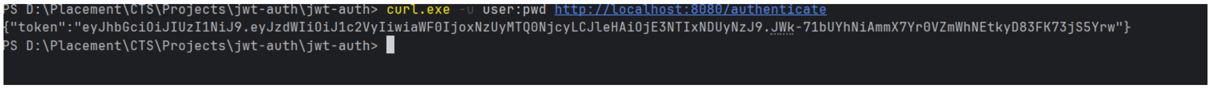
```
package com.cognizant.jwt_auth;

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

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

Output:



```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v3.5.3)

2025-07-10T16:20:41.880+05:30  INFO 26664 --- [jwt-auth] [  restartedMain] c.cognizant.jwt_auth.JwtAuthApplication  : Starting JwtAuthApplication using Java 24 with PID 26664 (
2025-07-10T16:20:41.884+05:30  INFO 26664 --- [jwt-auth] [  restartedMain] c.cognizant.jwt_auth.JwtAuthApplication  : No active profile set, falling back to 1 default profile:
2025-07-10T16:20:41.941+05:30  INFO 26664 --- [jwt-auth] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.ac
2025-07-10T16:20:41.941+05:30  INFO 26664 --- [jwt-auth] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'l
2025-07-10T16:20:43.006+05:30  INFO 26664 --- [jwt-auth] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8080 (http)
2025-07-10T16:20:43.026+05:30  INFO 26664 --- [jwt-auth] [  restartedMain] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2025-07-10T16:20:43.027+05:30  INFO 26664 --- [jwt-auth] [  restartedMain] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.42]
2025-07-10T16:20:43.082+05:30  INFO 26664 --- [jwt-auth] [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2025-07-10T16:20:43.084+05:30  INFO 26664 --- [jwt-auth] [  restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 11
2025-07-10T16:20:43.450+05:30  INFO 26664 --- [jwt-auth] [  restartedMain] r$InitializeUserDetailsManagerConfigurer : Global AuthenticationManager configured with UserDetailsSe
2025-07-10T16:20:43.631+05:30  INFO 26664 --- [jwt-auth] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
2025-07-10T16:20:43.659+05:30  INFO 26664 --- [jwt-auth] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context path '/'
2025-07-10T16:20:43.668+05:30  INFO 26664 --- [jwt-auth] [  restartedMain] c.cognizant.jwt_auth.JwtAuthApplication  : Started JwtAuthApplication in 2.165 seconds (process runni
2025-07-10T16:21:12.500+05:30  INFO 26664 --- [jwt-auth] [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-07-10T16:21:12.500+05:30  INFO 26664 --- [jwt-auth] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2025-07-10T16:21:12.500+05:30  INFO 26664 --- [jwt-auth] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 0 ms
```

The terminal token generation:



```
PS D:\Placement\CTS\Projects\jwt-auth\jwt-auth> curl.exe -u user:pwd http://localhost:8080/authenticate
{"token":"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiaWF0IjoxNzUyMTQ0NjcyLCJleHAiOjE3NTIxNDUyNzJ9.JWk-71bUYhNiAmmX7Yr0VZmWhNEtkyD83FK73jS5Yrw"}
PS D:\Placement\CTS\Projects\jwt-auth\jwt-auth> 
```