

Ex 6: Create authentication service that returns JWT

As part of first step of JWT process, the user credentials needs to be sent to authentication service request that generates and returns the JWT.

Ideally when the below curl command is executed that calls the new authentication service, the token should be responded. Kindly note that the credentials are passed using -u option.

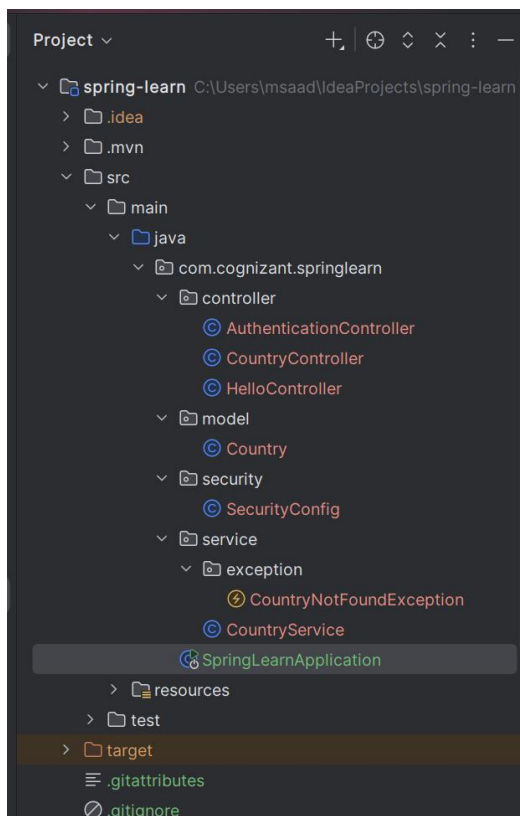
Request

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

Response

```
{"token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiaWF0IjoxNTcwMzc5NDc0LCJleHAiOjE1NzAzODA2NzR9.t3LRv1CV-hwKfoqZYlaVQqEUIBloWcWn0ft3tgv0dL0"}
```

Folder Structure:



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>spring-learn</description>
    <url/>
    <licenses>
        <license/>
    </licenses>
    <developers>
        <developer/>
    </developers>
    <scm>
        <connection/>
        <developerConnection/>
        <tag/>
        <url/>
    </scm>
    <properties>
        <java.version>17</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>
</project>
```

```

<!-- Spring Security -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>

<!-- JWT Library -->
<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> <!-- or jjwt-gson -->
  <version>0.11.5</version>
  <scope>runtime</scope>
</dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>

</project>

```

SecurityConfig.java (in package security)

```

package com.cognizant.springlearn.security;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.authentication.AuthenticationManager;
import
org.springframework.security.config.annotation.authentication.configuration.AuthenticationConfiguration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;

```

```
import org.springframework.security.web.SecurityFilterChain;
import org.springframework.security.provisioning.InMemoryUserDetailsManager;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetails;
```

@Configuration

```
public class SecurityConfig {
```

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

@Bean

```
    public PasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }
```

@Bean

```
    public InMemoryUserDetailsManager userDetailsService() {
        UserDetails admin = User
            .withUsername("admin")
            .password(passwordEncoder().encode("pwd"))
            .roles("ADMIN")
            .build();

        UserDetails user = User
            .withUsername("user")
            .password(passwordEncoder().encode("pwd"))
            .roles("USER")
            .build();

        return new InMemoryUserDetailsManager(admin, user);
    }
```

@Bean

```
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http.csrf().disable()
            .httpBasic()
            .and()
            .authorizeHttpRequests()
            .requestMatchers("/authenticate").hasAnyRole("USER", "ADMIN")
            .anyRequest().authenticated();

        return http.build();
    }
```

@Bean

```
    public AuthenticationManager authenticationManager(AuthenticationConfiguration config) throws
    Exception {
        return config.getAuthenticationManager();
    }
}
```

AuthenticationController.java (in the package controller)

```
package com.cognizant.springlearn.controller;
import io.jsonwebtoken.JwtBuilder;
import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.SignatureAlgorithm;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestHeader;
import org.springframework.web.bind.annotation.RestController;
import io.jsonwebtoken.security.Keys;

import java.util.Base64;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;

import io.jsonwebtoken.security.Keys;
import java.security.Key;

@RestController
public class AuthenticationController {

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

    @GetMapping("/authenticate")
    public Map<String, String> authenticate(@RequestHeader("Authorization") String authHeader) {
        LOGGER.info("START - authenticate()");
        LOGGER.debug("Authorization Header: {}", authHeader);

        String user = getUser(authHeader);
        String token = generateJwt(user);

        Map<String, String> map = new HashMap<>();
        map.put("token", token);

        LOGGER.info("END - authenticate()");
        return map;
    }

    private String getUser(String authHeader) {
        String encodedCredentials = authHeader.replace("Basic ", "");
        byte[] decodedBytes = Base64.getDecoder().decode(encodedCredentials);
        String decodedCredentials = new String(decodedBytes); // "user:pwd"
        LOGGER.debug("Decoded credentials: {}", decodedCredentials);
        return decodedCredentials.split(":")[0]; // return "user"
    }

    private String generateJwt(String user) {
        String secret = "my-secret-key-that-is-long-enough-123456"; // At least 32 chars
```

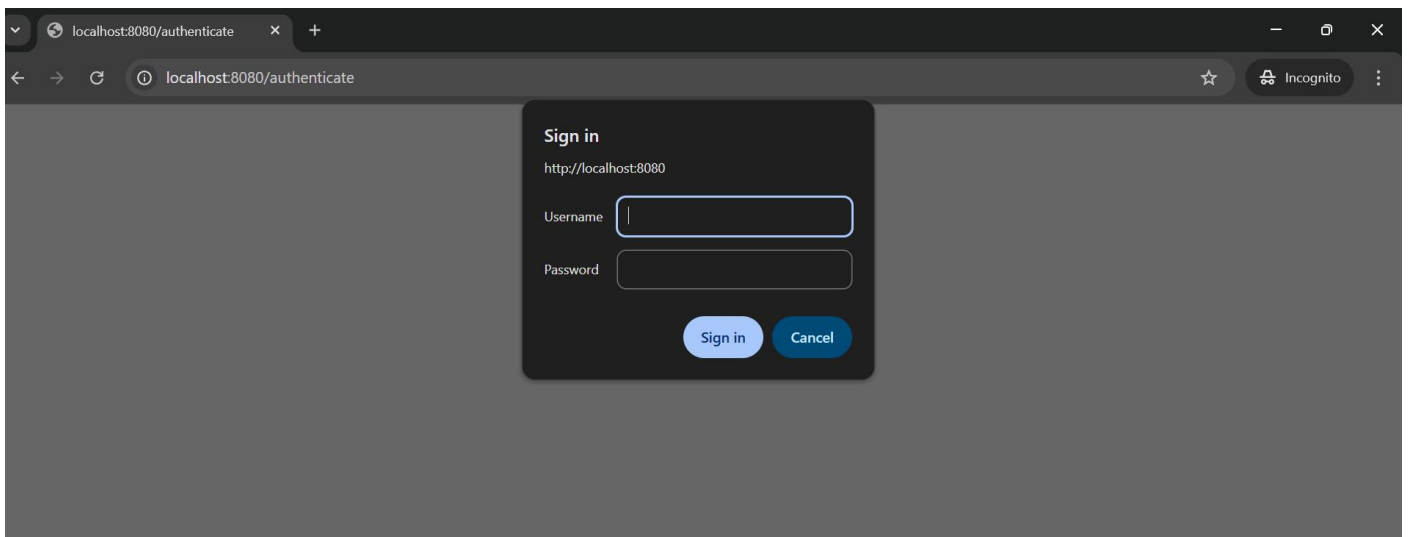
```
Key key = Keys.hmacShaKeyFor(secret.getBytes());

return Jwts.builder()
    .setSubject(user)
    .setIssuedAt(new Date())
    .setExpiration(new Date(System.currentTimeMillis() + 20 * 60 * 1000)) // 20 minutes
    .signWith(key, SignatureAlgorithm.HS256)
    .compact();
}
```

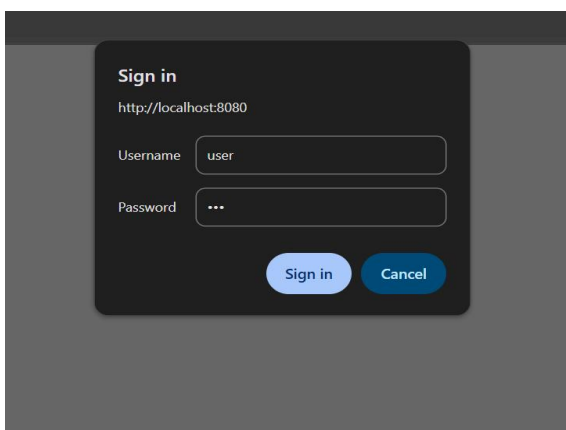
Run the `SpringLearnApplication.java` and go to browser and type,

<http://localhost:8080/authenticate>

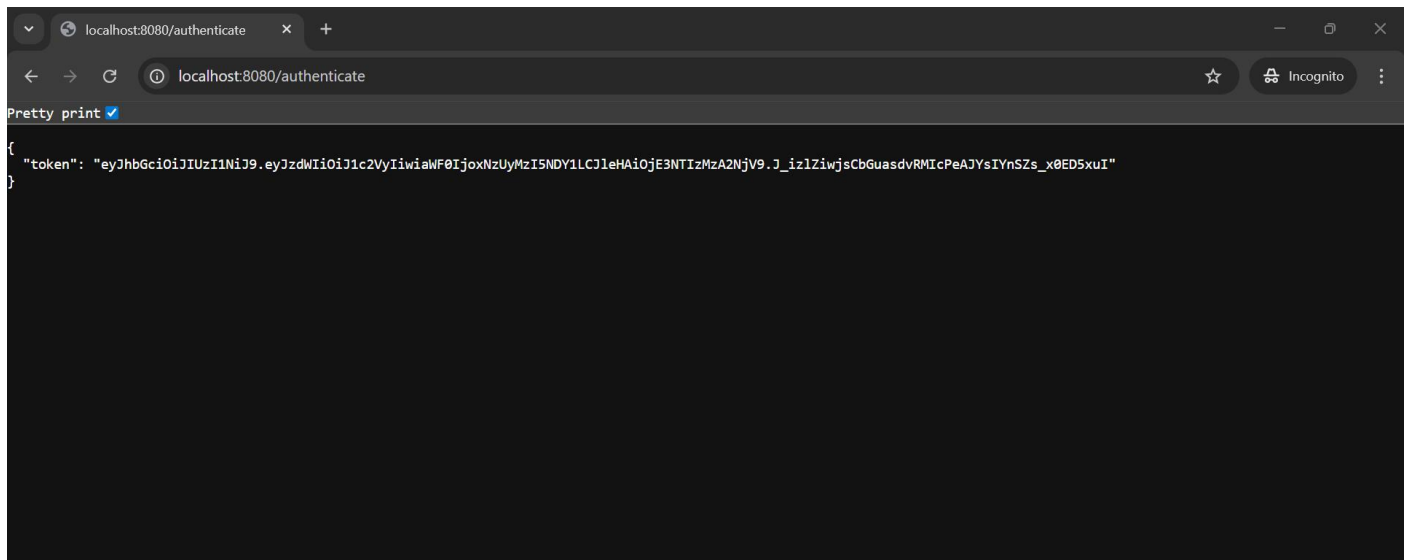
You will get a prompt to enter credentials,



I have entered “user” for username and “pwd” for password

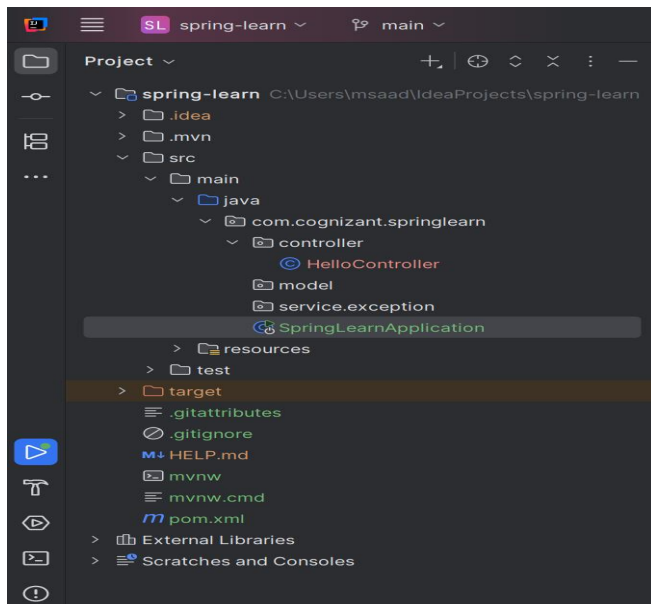


Output:



Ex 3: Hello World RESTful Web Service

Folder Structure:



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;
```

```
@RestController
public class HelloController {

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

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

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);
    }

}
```

Run the SpringLearnApplication.java

Then, open any browser and type: <http://localhost:8080/hello>

Output:



Ex 4: REST - Country Web Service

URL: /country

Controller: com.cognizant.spring-learn.controller.CountryController

Method Annotation: @RequestMapping

Method Name: getCountryIndia()

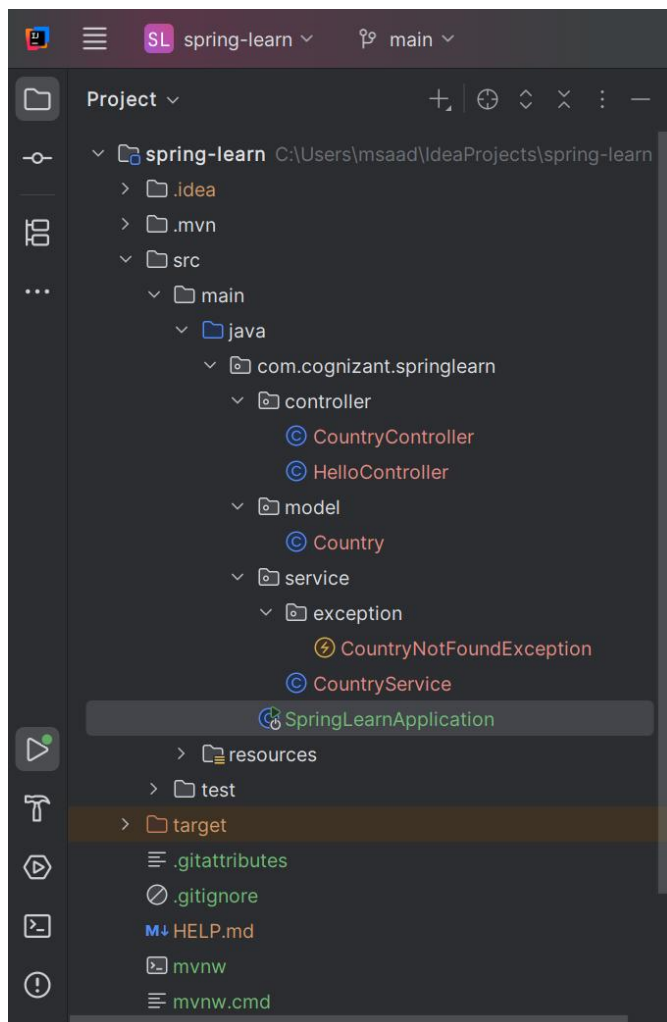
Method Implementation: Load India bean from spring xml configuration and return

Sample Request: http://localhost:8083/country

Sample Response:

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

Folder Structure:



CountryController.java (in controller package)

```
package com.cognizant.springlearn.controller;
import com.cognizant.springlearn.model.Country;
import com.cognizant.springlearn.service.CountryService;
import com.cognizant.springlearn.service.exception.CountryNotFoundException;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
public class CountryController {

    @Autowired
    private CountryService countryService;

    // GET: /country → returns India
    @GetMapping("/country")
    public Country getCountryIndia() {
        return new Country("IN", "India");
    }

    // GET: /countries → returns all countries
    @GetMapping("/countries")
    public List<Country> getAllCountries() {
        return countryService.getAllCountries();
    }

    // GET: /countries/{code} → returns a country by code
    @GetMapping("/countries/{code}")
    public Country getCountry(@PathVariable String code) throws CountryNotFoundException {
        return countryService.getCountry(code);
    }
}
```

Country.java (in model package)

```
package com.cognizant.springlearn.model;

public class Country {
    private String code;
    private String name;

    public Country() {}
    public Country(String code, String name) {
        this.code = code;
        this.name = name;
    }
}
```

```

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;
}
}

```

CountryService.java (in service package)

```

package com.cognizant.springlearn.service;
import com.cognizant.springlearn.model.Country;
import com.cognizant.springlearn.service.exception.CountryNotFoundException;
import org.springframework.stereotype.Service;

import java.util.Arrays;
import java.util.List;

@Service
public class CountryService {

    private static final List<Country> countries = Arrays.asList(
        new Country("IN", "India"),
        new Country("US", "United States"),
        new Country("JP", "Japan"),
        new Country("DE", "Germany")
    );

    public Country getCountry(String code) throws CountryNotFoundException {
        return countries.stream()
            .filter(c -> c.getCode().equalsIgnoreCase(code))
            .findFirst()
            .orElseThrow(() -> new CountryNotFoundException("Country not found"));
    }

    public List<Country> getAllCountries() {
        return countries;
    }
}

```

CountryNotFoundException.java (in service.exception package)

```
package com.cognizant.springlearn.service.exception;

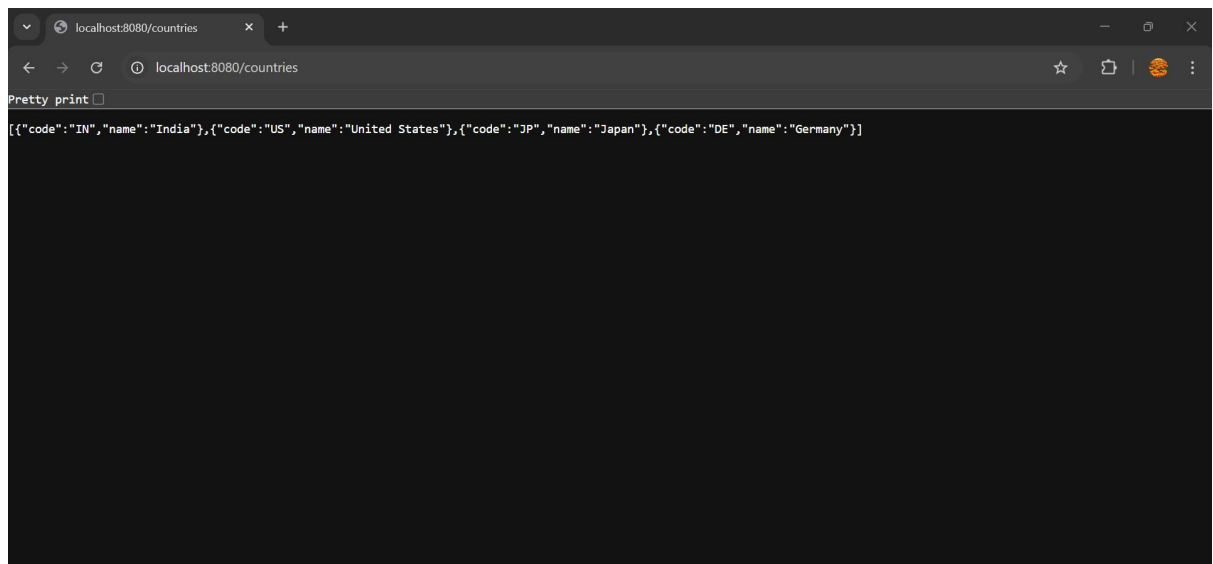
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.ResponseStatus;

@ResponseStatus(value = HttpStatus.NOT_FOUND, reason = "Country not found")
public class CountryNotFoundException extends Exception {
    public CountryNotFoundException(String message) {
        super(message);
    }
}
```

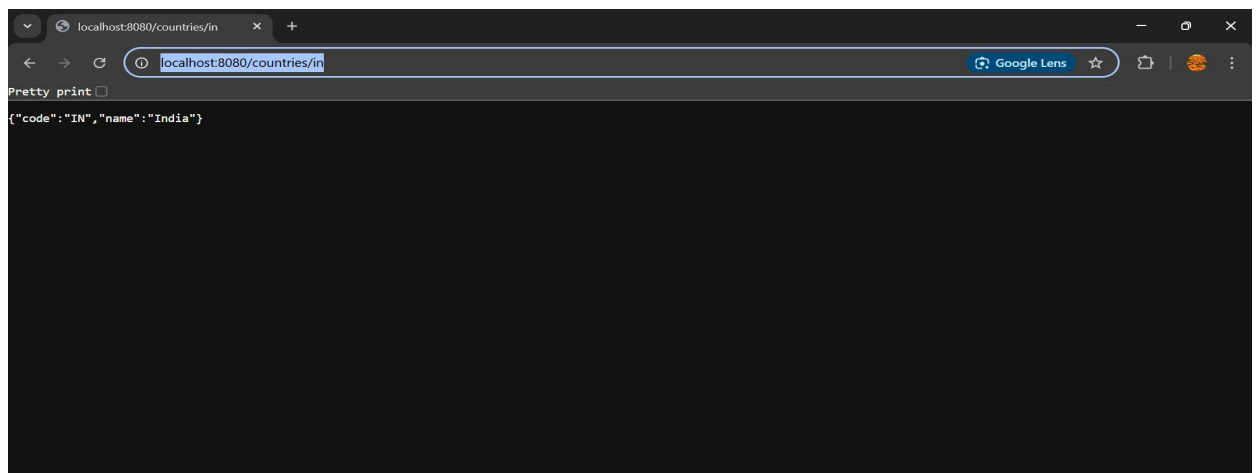
Now, run the `SPringLearnApplication.java`

Then, go to any browser and type,

- 1) <http://localhost:8080/countries> (to display all countries)



- 2) <http://localhost:8080/countries/in> (to display specific country)



Ex 5: REST - Get country based on country code

Write a REST service that returns a specific country based on country code. The country code should be case insensitive.

Controller: com.cognizant.spring-learn.controller.CountryController

Method Annotation: @GetMapping("/countries/{code}")

Method Name: getCountry(String code)

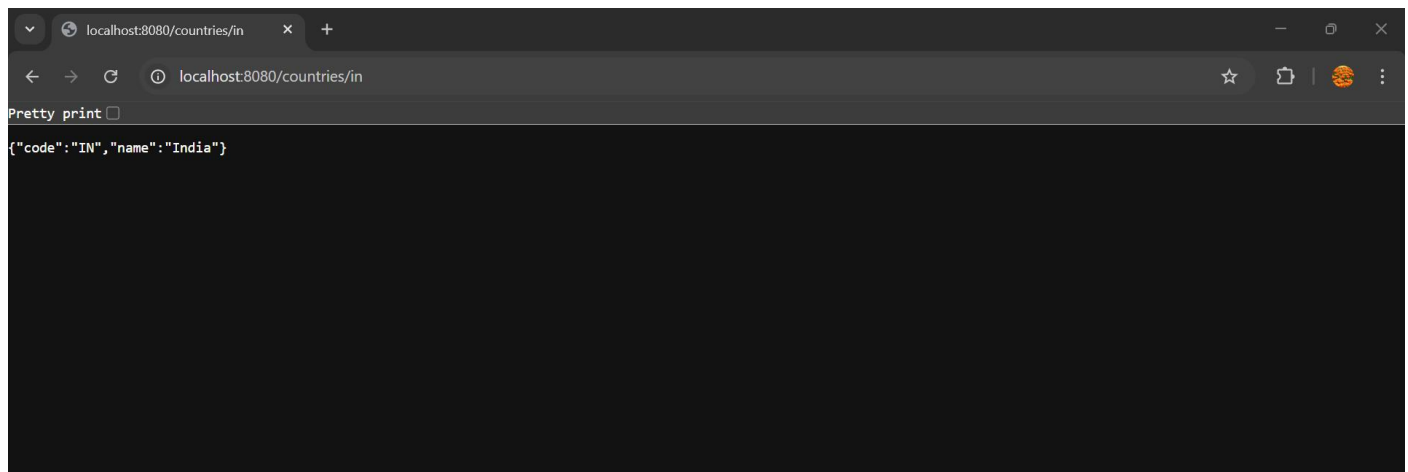
Method Implementation: Invoke countryService.getCountry(code)

Service Method: com.cognizant.spring-learn.service.CountryService.getCountry(String code)

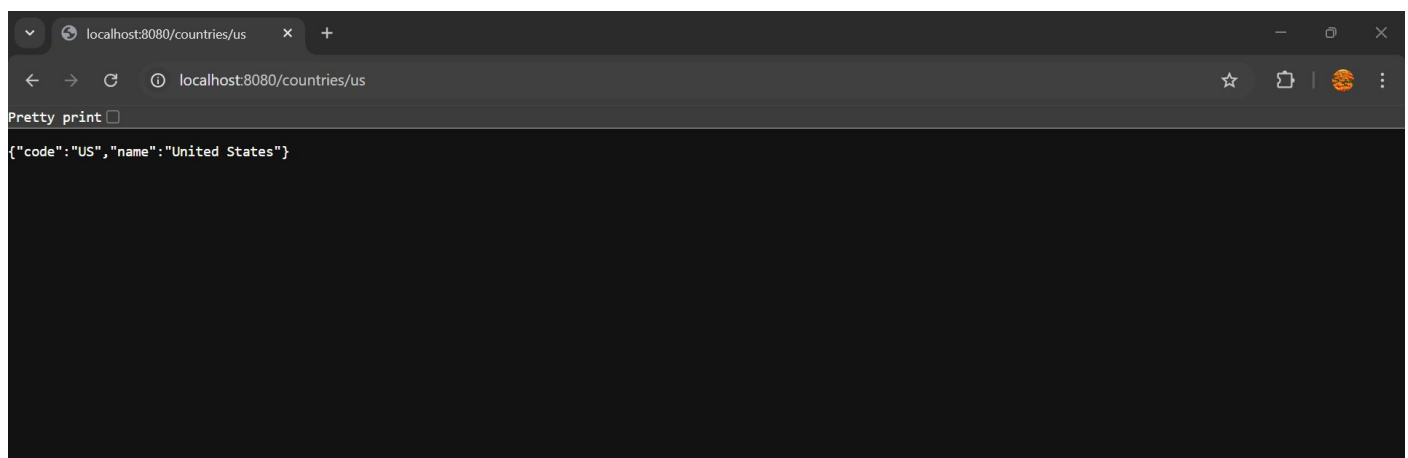
The folder structure and all the packages and classes remain the same which are used for the exercise above which is REST- Country Web Service

Output:

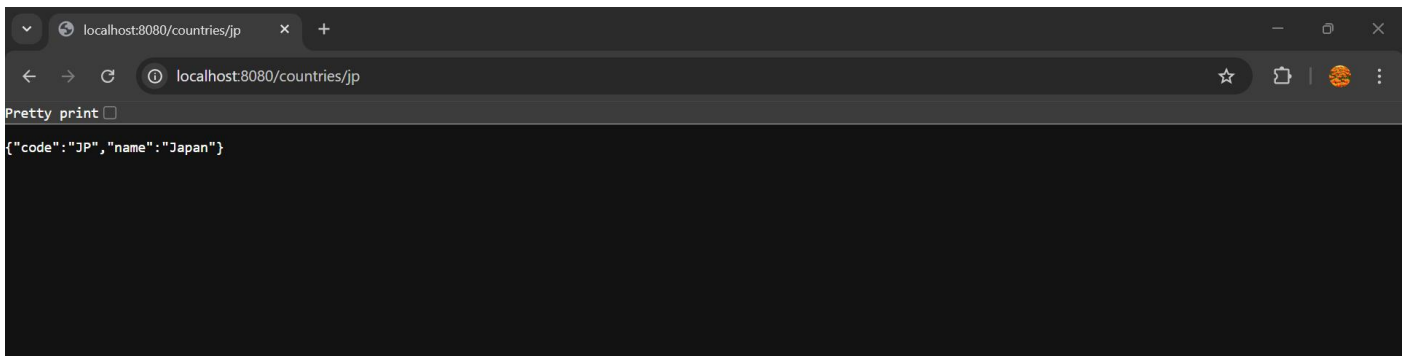
<http://localhost:8080/countries/in>



<http://localhost:8080/countries/us>

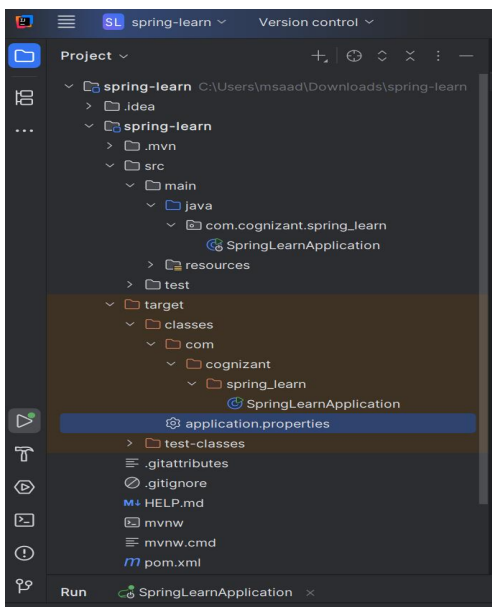


<http://localhost:8080/countries/jp>

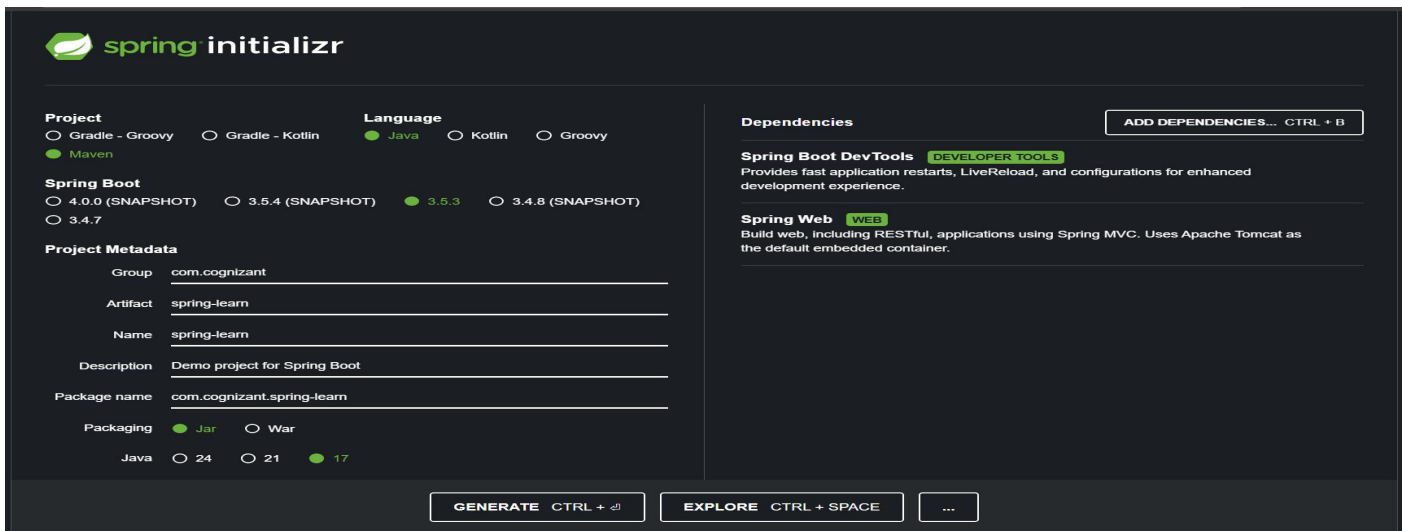


Ex 1: Create a Spring Web Project using Maven

Folder Structure:



Spring Initializr:



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>
    <url/>
    <licenses>
        <license/>
    </licenses>
    <developers>
        <developer/>
    </developers>
    <scm>
        <connection/>
        <developerConnection/>
        <tag/>
        <url/>
    </scm>
    <properties>
        <java.version>17</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>
</project>
```

```

</dependencies>

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>

</project>

```

SpringLearnApplication.java

```

package com.cognizant.spring_learn;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

```

@SpringBootApplication

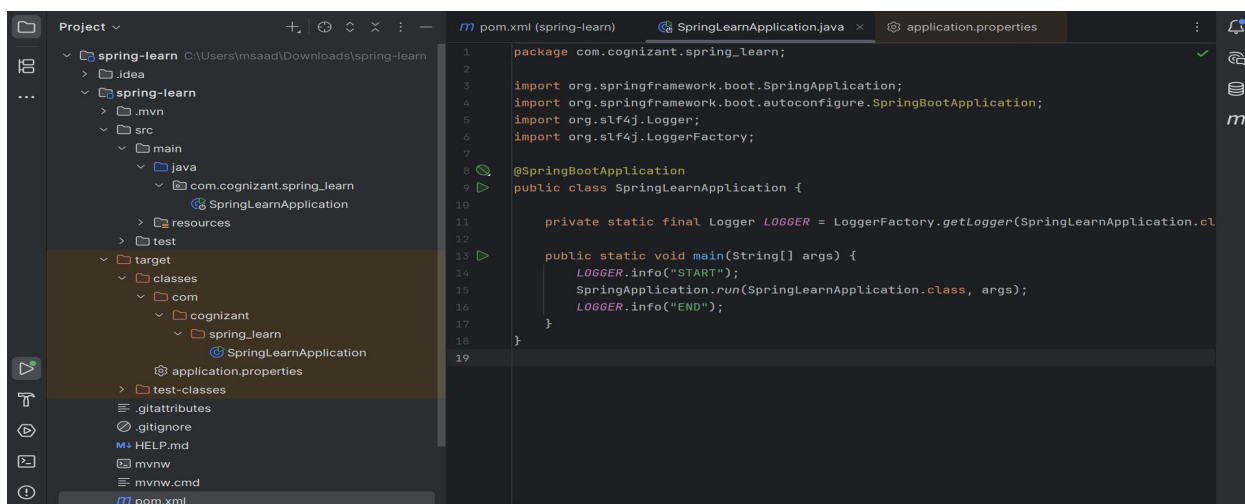
```

public class SpringLearnApplication {
    private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);
    public static void main(String[] args) {
        LOGGER.info("START");
        SpringApplication.run(SpringLearnApplication.class, args);
        LOGGER.info("END");
    }
}

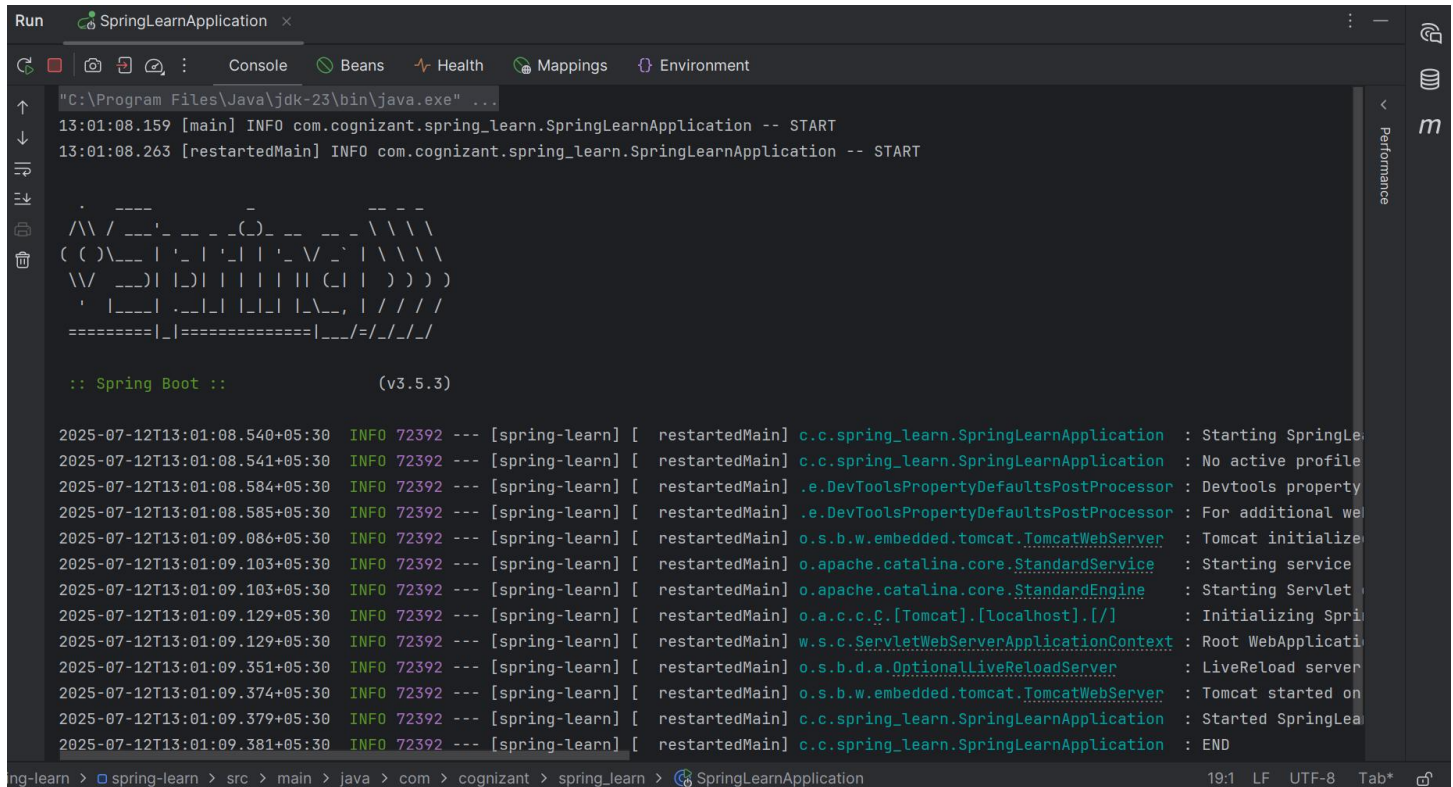
```

application.properties :

```
spring.application.name=spring-learn
```



Output:



```
"C:\Program Files\Java\jdk-23\bin\java.exe" ...
13:01:08.159 [main] INFO com.cognizant.spring_learn.SpringLearnApplication -- START
13:01:08.263 [restartedMain] INFO com.cognizant.spring_learn.SpringLearnApplication -- START

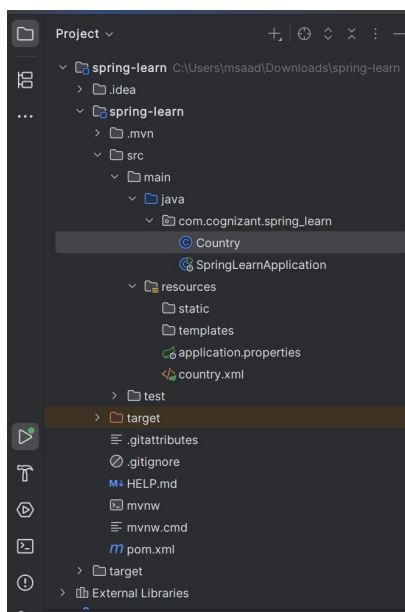
  ____
 / \  / ____'--- _(-)--- _ \ \ \ \
( ( ) \___| ' | ' | ' | ' \ \ \ \ \
 \ \ ___| | | | | | | | | ( | ) ) )
  ' |___| | | | | | | | | | / / /
=====|_|=====|___/_/_/_/_/

:: Spring Boot ::                (v3.5.3)

2025-07-12T13:01:08.540+05:30 INFO 72392 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : Starting SpringLea
2025-07-12T13:01:08.541+05:30 INFO 72392 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : No active profile
2025-07-12T13:01:08.584+05:30 INFO 72392 --- [spring-learn] [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property
2025-07-12T13:01:08.585+05:30 INFO 72392 --- [spring-learn] [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional we
2025-07-12T13:01:09.086+05:30 INFO 72392 --- [spring-learn] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialize
2025-07-12T13:01:09.103+05:30 INFO 72392 --- [spring-learn] [ restartedMain] o.apache.catalina.core.StandardService : Starting service
2025-07-12T13:01:09.103+05:30 INFO 72392 --- [spring-learn] [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet
2025-07-12T13:01:09.129+05:30 INFO 72392 --- [spring-learn] [ restartedMain] o.a.c.c.ø.[Tomcat].[localhost].[/] : Initializing Sprin
2025-07-12T13:01:09.129+05:30 INFO 72392 --- [spring-learn] [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicati
2025-07-12T13:01:09.351+05:30 INFO 72392 --- [spring-learn] [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server
2025-07-12T13:01:09.374+05:30 INFO 72392 --- [spring-learn] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on
2025-07-12T13:01:09.379+05:30 INFO 72392 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : Started SpringLea
2025-07-12T13:01:09.381+05:30 INFO 72392 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : END
```

EX 2: Spring Core – Load Country from Spring Configuration XML

Folder Structure:



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 + "];"
    }
}
```

country.xml (src/main/resources)

```
<?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>
```

application.properties (src/main/resources)

```
spring.application.name=spring-learn
logging.level.com.cognizant.spring_learn=DEBUG
```

SpringLearnApplication.java

```
package com.cognizant.spring_learn;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
```

```
@SpringBootApplication
```

```
public class SpringLearnApplication {
```

```
    private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);
    public void displayCountry() {
        LOGGER.info("START");
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
        Country country = context.getBean("country", Country.class);
        LOGGER.debug("Country : {}", country.toString());
        LOGGER.info("END");
    }
```

```
    public static void main(String[] args) {
        LOGGER.info("START");
        SpringApplication.run(SpringLearnApplication.class, args);
```

```
        SpringLearnApplication app = new SpringLearnApplication();
        app.displayCountry();
```

```
        LOGGER.info("END");
```

```
    }
}
```

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>
    <url/>
    <licenses>
        <license/>
    </licenses>
    <developers>
        <developer/>
    </developers>
    <scm>
        <connection/>
        <developerConnection/>
        <tag/>
        <url/>
    </scm>
    <properties>
        <java.version>17</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>
```

</project>

Output:

```
"C:\Program Files\Java\jdk-23\bin\java.exe" ...  
13:34:46.119 [main] INFO com.cognizant.spring_learn.SpringLearnApplication -- START  
13:34:46.223 [restartedMain] INFO com.cognizant.spring_learn.SpringLearnApplication -- START  
  
      _   _       _   _       _   _  
     / \ / ___'_ - _ -( )_ - _ - \\ \ \  
    ( )\_\\__ | '_ | '| | |_ V _`| \\ \\  
    \\ / ___) | |_| | | | | | | (_|_| )) ) )  
      ' |_____| ._|_| | |_| | |_|_\ , | // // //  
=====|_|=====|_|_/=//_/_/_/  
  
:: Spring Boot ::                (v3.5.3)
```

```

:: Spring Boot ::                (v3.5.3)

2025-07-12T13:34:46.514+05:30 INFO 46940 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : Starting SpringLe
2025-07-12T13:34:46.515+05:30 DEBUG 46940 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : Running with Spro
2025-07-12T13:34:46.516+05:30 INFO 46940 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : No active profile
2025-07-12T13:34:46.561+05:30 INFO 46940 --- [spring-learn] [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property
2025-07-12T13:34:46.561+05:30 INFO 46940 --- [spring-learn] [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional we
2025-07-12T13:34:47.074+05:30 INFO 46940 --- [spring-learn] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialize
2025-07-12T13:34:47.086+05:30 INFO 46940 --- [spring-learn] [ restartedMain] o.apache.catalina.core.StandardService : Starting service
2025-07-12T13:34:47.086+05:30 INFO 46940 --- [spring-learn] [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet
2025-07-12T13:34:47.111+05:30 INFO 46940 --- [spring-learn] [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Sprin
2025-07-12T13:34:47.111+05:30 INFO 46940 --- [spring-learn] [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicati
2025-07-12T13:34:47.372+05:30 INFO 46940 --- [spring-learn] [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server
2025-07-12T13:34:47.392+05:30 INFO 46940 --- [spring-learn] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on
2025-07-12T13:34:47.397+05:30 INFO 46940 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : Started SpringLea
2025-07-12T13:34:47.400+05:30 INFO 46940 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : START
2025-07-12T13:34:47.478+05:30 DEBUG 46940 --- [spring-learn] [ restartedMain] com.cognizant.spring_learn.Country : Inside Country Co
2025-07-12T13:34:47.478+05:30 DEBUG 46940 --- [spring-learn] [ restartedMain] com.cognizant.spring_learn.Country : Inside setCode()
2025-07-12T13:34:47.478+05:30 DEBUG 46940 --- [spring-learn] [ restartedMain] com.cognizant.spring_learn.Country : Inside setName()
2025-07-12T13:34:47.480+05:30 DEBUG 46940 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : Country : Country
2025-07-12T13:34:47.481+05:30 INFO 46940 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : END
2025-07-12T13:34:47.481+05:30 INFO 46940 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : END

```

```

n] c.c.spring_learn.SpringLearnApplication : Starting SpringLearnApplication using Java 23.0.1 with PID 46940 (C:\Users\msaad\Downloads\sp
n] c.c.spring_learn.SpringLearnApplication : Running with Spring Boot v3.5.3, Spring v6.2.8
n] c.c.spring_learn.SpringLearnApplication : No active profile set, falling back to 1 default profile: "default"
n] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable
n] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.web' property to 'DEBU
n] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
n] o.apache.catalina.core.StandardService : Starting service [Tomcat]
n] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.42]
n] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
n] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 550 ms
n] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
n] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'
n] c.c.spring_learn.SpringLearnApplication : Started SpringLearnApplication in 1.172 seconds (process running for 1.579)
n] c.c.spring_learn.SpringLearnApplication : START
n] com.cognizant.spring_learn.Country : Inside Country Constructor.
n] com.cognizant.spring_learn.Country : Inside setCode()
n] com.cognizant.spring_learn.Country : Inside setName()
n] c.c.spring_learn.SpringLearnApplication : Country : Country [code=IN, name=India]
n] c.c.spring_learn.SpringLearnApplication : END
n] c.c.spring_learn.SpringLearnApplication : END

```

