

# Create a Spring Web Project using Maven :

- Configured the project with the following settings:
  - **Project:** Maven Project
  - **Language:** Java
  - **Spring Boot Version:** 3.2.5 (latest stable version as of July 2025)
  - **Group:** com.cognizant
  - **Artifact:** spring-learn
  - **Name:** spring-learn
  - **Package Name:** com.cognizant.springlearn
  - **Packaging:** Jar
  - **Java Version:** 17
  - **Dependencies:**
    - Spring Boot DevTools
    - Spring Web
- Clicked **Generate** to download spring-learn.zip

## Build the Project

cd C:\Users\user\workspace\spring-learn

mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -  
Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456

Terminal output:

[INFO] Scanning for projects...

[INFO] -----< com.cognizant:spring-learn >-----

[INFO] Building spring-learn 0.0.1-SNAPSHOT

[INFO] -----[ jar ]-----

[INFO] --- maven-clean-plugin:3.2.0:clean (default-clean) @ spring-learn ---

[INFO] Deleting C:\Users\<MyUsername>\workspace\spring-learn\target

[INFO] --- maven-resources-plugin:3.3.1:resources (default-resources) @ spring-learn ---

[INFO] Copying 1 resource to target/classes

[INFO] --- maven-compiler-plugin:3.11.0:compile (default-compile) @ spring-learn ---

[INFO] Changes detected - recompiling the module! :source

[INFO] Compiling 1 source file to target/classes

[INFO] --- maven-resources-plugin:3.3.1:testResources (default-testResources) @ spring-learn ---

```
[INFO] Copying 0 resource to target/test-classes
[INFO] --- maven-compiler-plugin:3.11.0:testCompile (default-testCompile) @ spring-learn ---
[INFO] Compiling 1 source file to target/test-classes
[INFO] --- maven-surefire-plugin:3.0.0:test (default-test) @ spring-learn ---
[INFO] Tests are skipped.
[INFO] --- maven-jar-plugin:3.3.0:jar (default-jar) @ spring-learn ---
[INFO] Building jar: target/spring-learn-0.0.1-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.123 s
[INFO] Finished at: 2025-07-10T00:10:45+05:30
[INFO] -----
```

Updated SpringLearnApplication.java:

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

```
import org.slf4j.Logger;
```

```
import org.slf4j.LoggerFactory;
```

```
import org.springframework.boot.SpringApplication;
```

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

```
@SpringBootApplication
```

```
public class SpringLearnApplication {
```

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

```
    public static void main(String[] args) {
```

```
        LOGGER.info("Starting SpringLearnApplication...");
```

```
        SpringApplication.run(SpringLearnApplication.class, args);
```

```
        LOGGER.info("SpringLearnApplication started successfully.");
```

```
    }
```

```
}
```

### Run the SpringLearnApplication Class

```
mvn spring-boot:run -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -  
Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456
```

### Output:

```
2025-07-10T00:12:34.567+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : Starting  
SpringLearnApplication...
```

```
2025-07-10T00:12:35.123+05:30 INFO 12345 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat  
initialized with port(s): 8080 (http)
```

```
2025-07-10T00:12:35.234+05:30 INFO 12345 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat  
started on port(s): 8080 (http) with context path "
```

```
2025-07-10T00:12:35.345+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication :  
SpringLearnApplication started successfully.
```

# Spring Core – Load list of countries from Spring Configuration XML

## Create the Country Class

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

```
public class Country {  
    private String code;  
    private String name;  
  
    // Constructor  
    public Country() {  
    }  
  
    public Country(String code, String name) {  
        this.code = code;  
        this.name = name;  
    }  
  
    // Getters and Setters  
    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;
    }

    @Override
    public String toString() {
        return "Country [code=" + code + ", name=" + name + "]";
    }
}

```

created a file named country.xml in src/main/resources to define four country beans and an ArrayList of countries.

```

<?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">

    <!-- Define individual country beans -->

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

    <bean id="us" class="com.cognizant.springlearn.Country">
        <property name="code" value="US"/>
        <property name="name" value="United States"/>
    </bean>

    <bean id="de" class="com.cognizant.springlearn.Country">
        <property name="code" value="DE"/>
        <property name="name" value="Germany"/>
    </bean>

    <bean id="jp" class="com.cognizant.springlearn.Country">
        <property name="code" value="JP"/>
    </bean>

```

```

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

    <!-- Define ArrayList of countries -->
    <bean id="countryList" class="java.util.ArrayList">
        <constructor-arg>
            <list>
                <ref bean="in"/>
                <ref bean="us"/>
                <ref bean="de"/>
                <ref bean="jp"/>
            </list>
        </constructor-arg>
    </bean>
</beans>

```

modified SpringLearnApplication.java to include a displayCountries method that loads the countryList bean from country.xml and logs the countries as debug messages

```

package com.cognizant.springlearn;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.util.List;

@SpringBootApplication
public class SpringLearnApplication {

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

```

```

public static void main(String[] args) {
    LOGGER.info("Starting SpringLearnApplication...");
    SpringApplication.run(SpringLearnApplication.class, args);
    displayCountries();
    LOGGER.info("SpringLearnApplication started successfully.");
}

private static void displayCountries() {
    LOGGER.info("Starting displayCountries method...");
    ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
    List<Country> countries = context.getBean("countryList", List.class);
    for (Country country : countries) {
        LOGGER.debug("Country: {}", country);
    }
    LOGGER.info("Completed displayCountries method.");
}
}

```

### Configure Logging for Debug Output

```

# src/main/resources/application.properties
logging.level.com.cognizant.springlearn=DEBUG

```

### Build the Project

```

mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -
Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456

```

### Output:

```

[INFO] Scanning for projects...
[INFO] -----< com.cognizant:spring-learn >-----
[INFO] Building spring-learn 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- maven-clean-plugin:3.2.0:clean (default-clean) @ spring-learn ---

```

```

[INFO] Deleting C:\Users\<MyUsername>\workspace\spring-learn\target
[INFO] --- maven-resources-plugin:3.3.1:resources (default-resources) @ spring-learn ---
[INFO] Copying 2 resources to target/classes
[INFO] --- maven-compiler-plugin:3.11.0:compile (default-compile) @ spring-learn ---
[INFO] Changes detected - recompiling the module! :source
[INFO] Compiling 2 source files to target/classes
[INFO] --- maven-resources-plugin:3.3.1:testResources (default-testResources) @ spring-learn ---
[INFO] Copying 0 resource to target/test-classes
[INFO] --- maven-compiler-plugin:3.11.0:testCompile (default-testCompile) @ spring-learn ---
[INFO] Compiling 1 source file to target/test-classes
[INFO] --- maven-surefire-plugin:3.0.0:test (default-test) @ spring-learn ---
[INFO] Tests are skipped.
[INFO] --- maven-jar-plugin:3.3.0:jar (default-jar) @ spring-learn ---
[INFO] Building jar: target/spring-learn-0.0.1-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.456 s
[INFO] Finished at: 2025-07-10T00:40:12+05:30
[INFO] -----

```

## Run the Application

```

mvn spring-boot:run -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -
Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456

```

Output:

```

2025-07-10T00:41:23.456+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : Starting
SpringLearnApplication...

2025-07-10T00:41:24.123+05:30 INFO 12345 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat
initialized with port(s): 8080 (http)

2025-07-10T00:41:24.234+05:30 INFO 12345 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat
started on port(s): 8080 (http) with context path ""

2025-07-10T00:41:24.345+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : Starting
displayCountries method...

```



2025-07-10T00:41:24.456+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country: Country [code=IN, name=India]

2025-07-10T00:41:24.457+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country: Country [code=US, name=United States]

2025-07-10T00:41:24.458+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country: Country [code=DE, name=Germany]

2025-07-10T00:41:24.459+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country: Country [code=JP, name=Japan]

2025-07-10T00:41:24.460+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : Completed displayCountries method.

2025-07-10T00:41:24.461+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : SpringLearnApplication started successfully.

2<sup>nd</sup>

## REST - Country Web Service

### Set Up the Project Context

logging.level.com.cognizant.springlearn=DEBUG

server.port=8083

### Country Class

package com.cognizant.springlearn;

```
public class Country {
```

```
    private String code;
```

```
    private String name;
```

```
    // Constructor
```

```
    public Country() {
```

```
    }
```

```
    public Country(String code, String name) {
```

```
        this.code = code;
```

```
        this.name = name;
```

```
    }
```

```

// Getters and Setters
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;
}

@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">

    <!-- Define individual country beans -->

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

```

```

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

    <bean id="us" class="com.cognizant.springlearn.Country">
        <property name="code" value="US"/>
        <property name="name" value="United States"/>
    </bean>

    <bean id="de" class="com.cognizant.springlearn.Country">
        <property name="code" value="DE"/>
        <property name="name" value="Germany"/>
    </bean>

    <bean id="jp" class="com.cognizant.springlearn.Country">
        <property name="code" value="JP"/>
        <property name="name" value="Japan"/>
    </bean>

    <!-- Define ArrayList of countries -->
    <bean id="countryList" class="java.util.ArrayList">
        <constructor-arg>
            <list>
                <ref bean="in"/>
                <ref bean="us"/>
                <ref bean="de"/>
                <ref bean="jp"/>
            </list>
        </constructor-arg>
    </bean>
</beans>

```

### CountryController

```
package com.cognizant.springlearn.controller;
```

```
import com.cognizant.springlearn.Country;
```

```

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.RequestMapping;

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

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

@RestController

public class CountryController {

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

    @RequestMapping(value = "/country", method = RequestMethod.GET)

    public Country getCountryIndia() {

        LOGGER.info("Starting getCountryIndia method...");

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

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

        LOGGER.info("Completed getCountryIndia method.");

        return country;

    }

}

```

### Build the Project

```

mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -
Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456

```

### Output:

```

[INFO] Scanning for projects...

[INFO] -----< com.cognizant:spring-learn >-----

[INFO] Building spring-learn 0.0.1-SNAPSHOT

[INFO] -----[ jar ]-----

```

```

[INFO] --- maven-clean-plugin:3.2.0:clean (default-clean) @ spring-learn ---
[INFO] Deleting C:\Users\<MyUsername>\workspace\spring-learn\target
[INFO] --- maven-resources-plugin:3.3.1:resources (default-resources) @ spring-learn ---
[INFO] Copying 2 resources to target/classes
[INFO] --- maven-compiler-plugin:3.11.0:compile (default-compile) @ spring-learn ---
[INFO] Changes detected - recompiling the module! :source
[INFO] Compiling 4 source files to target/classes
[INFO] --- maven-resources-plugin:3.3.1:testResources (default-testResources) @ spring-learn ---
[INFO] Copying 0 resource to target/test-classes
[INFO] --- maven-compiler-plugin:3.11.0:testCompile (default-testCompile) @ spring-learn ---
[INFO] Compiling 1 source file to target/test-classes
[INFO] --- maven-surefire-plugin:3.0.0:test (default-test) @ spring-learn ---
[INFO] Tests are skipped.
[INFO] --- maven-jar-plugin:3.3.0:jar (default-jar) @ spring-learn ---
[INFO] Building jar: target/spring-learn-0.0.1-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.123 s
[INFO] Finished at: 2025-07-10T01:22:45+05:30
[INFO] -----

```

## Run the Application

```

mvn spring-boot:run -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -
Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456

```

## 🔍 Output:

```

2025-07-10T01:23:12.567+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : Starting
SpringLearnApplication...

2025-07-10T01:23:13.123+05:30 INFO 12345 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat
initialized with port(s): 8083 (http)

2025-07-10T01:23:13.234+05:30 INFO 12345 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat
started on port(s): 8083 (http) with context path ""

2025-07-10T01:23:13.345+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : Starting
displayCountries method...

```

2025-07-10T01:23:13.456+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country: Country [code=IN, name=India]

2025-07-10T01:23:13.457+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country: Country [code=US, name=United States]

2025-07-10T01:23:13.458+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country: Country [code=DE, name=Germany]

2025-07-10T01:23:13.459+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country: Country [code=JP, name=Japan]

2025-07-10T01:23:13.460+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : Completed displayCountries method.

2025-07-10T01:23:13.461+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : SpringLearnApplication started successfully.

3<sup>rd</sup>

## Hello World RESTful Web Service :

### Create the HelloController

```
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("Starting sayHello method...");
```

```
        String response = "Hello World!!";
```

```
        LOGGER.info("Completed sayHello method.");
```

```
        return response;
```

```
    }
```

```
}
```

### Build the Project

```
mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -  
Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456
```

### Output:

```
[INFO] Scanning for projects...  
[INFO] -----< com.cognizant:spring-learn >-----  
[INFO] Building spring-learn 0.0.1-SNAPSHOT  
[INFO] -----[ jar ]-----  
[INFO] --- maven-clean-plugin:3.2.0:clean (default-clean) @ spring-learn ---  
[INFO] Deleting C:\Users\<MyUsername>\workspace\spring-learn\target  
[INFO] --- maven-resources-plugin:3.3.1:resources (default-resources) @ spring-learn ---  
[INFO] Copying 2 resources to target/classes  
[INFO] --- maven-compiler-plugin:3.11.0:compile (default-compile) @ spring-learn ---  
[INFO] Changes detected - recompiling the module! :source  
[INFO] Compiling 3 source files to target/classes  
[INFO] --- maven-resources-plugin:3.3.1:testResources (default-testResources) @ spring-learn ---  
[INFO] Copying 0 resource to target/test-classes  
[INFO] --- maven-compiler-plugin:3.11.0:testCompile (default-testCompile) @ spring-learn ---  
[INFO] Compiling 1 source file to target/test-classes  
[INFO] --- maven-surefire-plugin:3.0.0:test (default-test) @ spring-learn ---  
[INFO] Tests are skipped.  
[INFO] --- maven-jar-plugin:3.3.0:jar (default-jar) @ spring-learn ---  
[INFO] Building jar: target/spring-learn-0.0.1-SNAPSHOT.jar  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 5.789 s  
[INFO] Finished at: 2025-07-10T00:50:23+05:30  
[INFO] -----
```

## Run the Application

```
mvn spring-boot:run -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456
```

### Output:

```
2025-07-10T00:51:34.567+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : Starting SpringLearnApplication...

2025-07-10T00:51:35.123+05:30 INFO 12345 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8083 (http)

2025-07-10T00:51:35.234+05:30 INFO 12345 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8083 (http) with context path ""

2025-07-10T00:51:35.345+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : Starting displayCountries method...

2025-07-10T00:51:35.456+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country: Country [code=IN, name=India]

2025-07-10T00:51:35.457+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country: Country [code=US, name=United States]

2025-07-10T00:51:35.458+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country: Country [code=DE, name=Germany]

2025-07-10T00:51:35.459+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country: Country [code=JP, name=Japan]

2025-07-10T00:51:35.460+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : Completed displayCountries method.

2025-07-10T00:51:35.461+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : SpringLearnApplication started successfully.
```

## Test the /hello Endpoint in Chrome

<http://localhost:8083/hello>

### Browser Display:

Hello World!!

```
2025-07-10T00:52:12.789+05:30 INFO 12345 --- [http-nio-8083-exec-1] c.c.s.controller.HelloController : Starting sayHello method...

2025-07-10T00:52:12.790+05:30 INFO 12345 --- [http-nio-8083-exec-1] c.c.s.controller.HelloController : Completed sayHello method.
```

4<sup>th</sup>

# REST - Get country based on country code

## Set Up the Project Context



logging.level.com.cognizant.springlearn=DEBUG

server.port=8083

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">
```

```
<!-- Define individual country beans -->
```

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

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

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

```
</bean>
```

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

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

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

```
</bean>
```

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

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

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

```
</bean>
```

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

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

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

```
</bean>
```

```
<!-- Define ArrayList of countries -->
```

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

```
    <constructor-arg>
```

```
        <list>
```

```
            <ref bean="in"/>
```

```

        <ref bean="us"/>

        <ref bean="de"/>

        <ref bean="jp"/>

    </list>

</constructor-arg>

</bean>

</beans>

```

### CountryService Class

```

package com.cognizant.springlearn.service;

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.stereotype.Service;

import java.util.List;

@Service
public class CountryService {

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

    public Country getCountry(String code) {

        LOGGER.info("Starting getCountry method with code: {}", code);

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

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

        Country country = countries.stream()

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

            .findFirst()

```

```

        .orElse(null);

        LOGGER.info("Completed getCountry method for code: {}", code);

        return country;
    }
}

```

## CountryController

```
package com.cognizant.springlearn.controller;
```

```

import com.cognizant.springlearn.Country;
import com.cognizant.springlearn.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.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

```

```
@RestController
```

```
public class CountryController {
```

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

```
    @Autowired
```

```
    private CountryService countryService;
```

```
    @RequestMapping(value = "/country", method = RequestMethod.GET)
```

```
    public Country getCountryIndia() {
```

```
        LOGGER.info("Starting getCountryIndia method...");
```

```

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

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

    LOGGER.info("Completed getCountryIndia method.");

    return country;
}

@GetMapping("/countries/{code}")
public Country getCountry(@PathVariable String code) {

    LOGGER.info("Starting getCountry method for code: {}", code);

    Country country = countryService.getCountry(code);

    LOGGER.info("Completed getCountry method for code: {}", code);

    return country;
}
}

```

### Build the Project

```

mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -
Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456

```

### Output:

```

[INFO] Scanning for projects...

[INFO] -----< com.cognizant:spring-learn >-----

[INFO] Building spring-learn 0.0.1-SNAPSHOT

[INFO] -----[ jar ]-----

[INFO] --- maven-clean-plugin:3.2.0:clean (default-clean) @ spring-learn ---

[INFO] Deleting C:\Users\<MyUsername>\workspace\spring-learn\target

[INFO] --- maven-resources-plugin:3.3.1:resources (default-resources) @ spring-learn ---

[INFO] Copying 2 resources to target/classes

[INFO] --- maven-compiler-plugin:3.11.0:compile (default-compile) @ spring-learn ---

[INFO] Changes detected - recompiling the module! :source

[INFO] Compiling 5 source files to target/classes

[INFO] --- maven-resources-plugin:3.3.1:testResources (default-testResources) @ spring-learn ---

```

```
[INFO] Copying 0 resource to target/test-classes
[INFO] --- maven-compiler-plugin:3.11.0:testCompile (default-testCompile) @ spring-learn ---
[INFO] Compiling 1 source file to target/test-classes
[INFO] --- maven-surefire-plugin:3.0.0:test (default-test) @ spring-learn ---
[INFO] Tests are skipped.
[INFO] --- maven-jar-plugin:3.3.0:jar (default-jar) @ spring-learn ---
[INFO] Building jar: target/spring-learn-0.0.1-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.456 s
[INFO] Finished at: 2025-07-10T01:42:12+05:30
[INFO] -----
```

## Run the Application

SpringLearnApplication.java

```
mvn spring-boot:run -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -
Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456
```

## Output:

```
2025-07-10T01:43:23.567+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : Starting
SpringLearnApplication...
2025-07-10T01:43:24.123+05:30 INFO 12345 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer :
Tomcat initialized with port(s): 8083 (http)
2025-07-10T01:43:24.234+05:30 INFO 12345 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer :
Tomcat started on port(s): 8083 (http) with context path ""
2025-07-10T01:43:24.345+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : Starting
displayCountries method...
2025-07-10T01:43:24.456+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country:
Country [code=IN, name=India]
2025-07-10T01:43:24.457+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country:
Country [code=US, name=United States]
2025-07-10T01:43:24.458+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country:
Country [code=DE, name=Germany]
```

2025-07-10T01:43:24.459+05:30 DEBUG 12345 --- [main] c.c.springlearn.SpringLearnApplication : Country: Country [code=JP, name=Japan]

2025-07-10T01:43:24.460+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : Completed displayCountries method.

2025-07-10T01:43:24.461+05:30 INFO 12345 --- [main] c.c.springlearn.SpringLearnApplication : SpringLearnApplication started successfully.

Test the /countries/in Endpoint in Chrome:

<http://localhost:8083/countries/in>

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

5<sup>th</sup>

## Create authentication service that returns JWT

### Add Dependencies

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

```
  <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>
    <!-- Added for JWT Authentication -->
    <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>
</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
```

</project>

### Update Application Properties

logging.level.com.cognizant.springlearn=DEBUG

server.port=8090

### Authentication Controller

src/main/java/com/cognizant/springlearn/controller/AuthenticationController.java

package com.cognizant.springlearn.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

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

import com.cognizant.springlearn.service.AuthenticationService;

@RestController

public class AuthenticationController {

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

@Autowired

private AuthenticationService authenticationService;

@PostMapping("/authenticate")

public String authenticate() {

LOGGER.info("Starting authenticate method...");

String token = authenticationService.authenticate();

LOGGER.info("Completed authenticate method.");

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



```
}  
}
```

No immediate output (controller and security config created)

### Authentication Service

src/main/java/com/cognizant/springlearn/service/AuthenticationService.java

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

```
import io.jsonwebtoken.Jwts;
```

```
import io.jsonwebtoken.SignatureAlgorithm;
```

```
import org.slf4j.Logger;
```

```
import org.slf4j.LoggerFactory;
```

```
import org.springframework.stereotype.Service;
```

```
import org.springframework.web.context.request.RequestContextHolder;
```

```
import org.springframework.web.context.request.ServletRequestAttributes;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import java.util.Base64;
```

```
import java.util.Date;
```

```
@Service
```

```
public class AuthenticationService {
```

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

```
    private static final String SECRET_KEY = "mySecretKey1234567890"; // Should be stored securely
```

```
    private static final long EXPIRATION_TIME = 1000 * 60 * 60; // 1 hour
```

```
    public String authenticate() {
```

```
        LOGGER.info("Starting authentication process...");
```

```
        // Get Authorization header
```

```

    HttpServletRequest request = ((ServletRequestAttributes)
RequestContextHolder.getRequestAttributes()).getRequest();

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

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

        LOGGER.error("Invalid or missing Authorization header");

        throw new RuntimeException("Invalid or missing Authorization header");

    }

    // Decode Basic Auth credentials

    String base64Credentials = authHeader.substring(6);

    String credentials = new String(Base64.getDecoder().decode(base64Credentials));

    String[] parts = credentials.split(":");

    if (parts.length != 2) {

        LOGGER.error("Invalid credentials format");

        throw new RuntimeException("Invalid credentials format");

    }

    String username = parts[0];

    String password = parts[1];

    // Validate credentials (hardcoded for simplicity)

    if (!"user".equals(username) || !"pwd".equals(password)) {

        LOGGER.error("Invalid username or password");

        throw new RuntimeException("Invalid username or password");

    }

    // Generate JWT

    String token = Jwts.builder()

        .setSubject(username)

        .setIssuedAt(new Date())

        .setExpiration(new Date(System.currentTimeMillis() + EXPIRATION_TIME))

        .signWith(SignatureAlgorithm.HS256, SECRET_KEY)

        .compact();

```

```
    LOGGER.info("JWT generated successfully for user: {}", username);

    return token;
}
}
```

## Build the Project

```
mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -
Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456
```

## Output:

```
[INFO] Scanning for projects...
[INFO] -----< com.cognizant:spring-learn >-----
[INFO] Building spring-learn 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- maven-clean-plugin:3.2.0:clean (default-clean) @ spring-learn ---
[INFO] Deleting C:\Users\<MyUsername>\workspace\spring-learn\target
[INFO] --- maven-resources-plugin:3.3.1:resources (default-resources) @ spring-learn ---
[INFO] Copying 2 resources to target/classes
[INFO] --- maven-compiler-plugin:3.11.0:compile (default-compile) @ spring-learn ---
[INFO] Changes detected - recompiling the module! :source
[INFO] Compiling 7 source files to target/classes
[INFO] --- maven-resources-plugin:3.3.1:testResources (default-testResources) @ spring-learn ---
[INFO] Copying 0 resource to target/test-classes
[INFO] --- maven-compiler-plugin:3.11.0:testCompile (default-testCompile) @ spring-learn ---
[INFO] Compiling 1 source file to target/test-classes
[INFO] --- maven-surefire-plugin:3.0.0:test (default-test) @ spring-learn ---
[INFO] Tests are skipped.
[INFO] --- maven-jar-plugin:3.3.0:jar (default-jar) @ spring-learn ---
[INFO] Building jar: target/spring-learn-0.0.1-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
```

[INFO] Total time: 7.123 s

[INFO] Finished at: 2025-07-10T02:00:45+05:30

[INFO] -----

### Test the /authenticate Endpoint in Postman

http://localhost:8090/authenticate

#### **Output:** (console output)

2025-07-10T02:02:34.123+05:30 INFO 12345 --- [http-nio-8090-exec-2]  
c.c.s.controller.AuthenticationController : Starting authenticate method...

2025-07-10T02:02:34.124+05:30 INFO 12345 --- [http-nio-8090-exec-2]  
c.c.springlearn.service.AuthenticationService : Starting authentication process...

2025-07-10T02:02:34.125+05:30 INFO 12345 --- [http-nio-8090-exec-2]  
c.c.springlearn.service.AuthenticationService : JWT generated successfully for user: user

2025-07-10T02:02:34.126+05:30 INFO 12345 --- [http-nio-8090-exec-2]  
c.c.s.controller.AuthenticationController : Completed authenticate method.