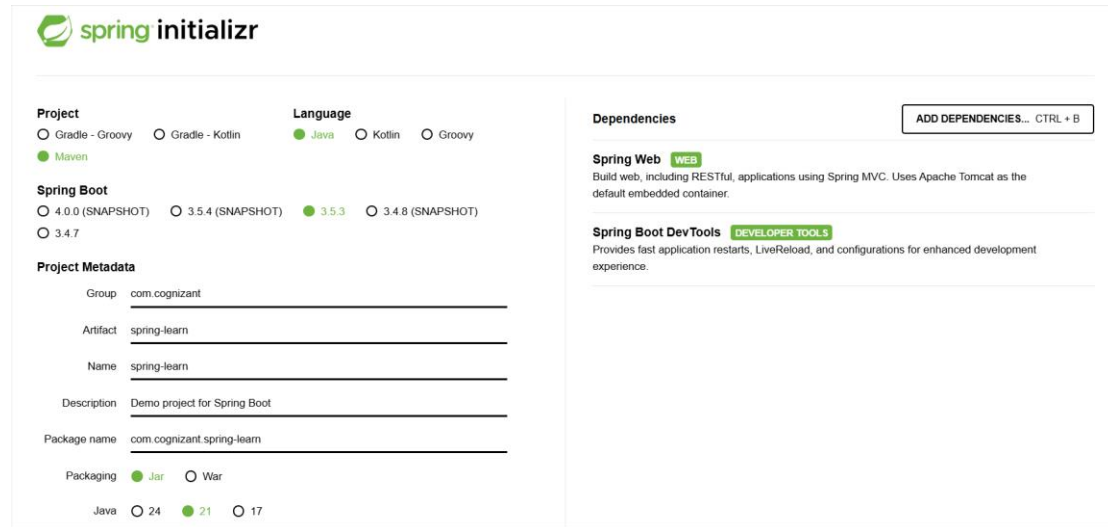


## Hands on 1

## Create a Spring Web Project using Maven



The image shows the Spring Initializr web interface. It is configured for a Spring Web project using Maven. The language is set to Java, and the Spring Boot version is 3.5.3. The project metadata includes a group ID of com.cognizant, an artifact ID of spring-learn, and a package name of com.cognizant.spring-learn. The packaging is set to Jar, and the Java version is 21.

**Project**  
☐ Gradle - Groovy ☐ Gradle - Kotlin ☒ **Java** ☐ Kotlin ☐ Groovy

**Spring Boot**  
☐ 4.0.0 (SNAPSHOT) ☐ 3.5.4 (SNAPSHOT) ☒ **3.5.3** ☐ 3.4.8 (SNAPSHOT)  
☐ 3.4.7

**Project Metadata**

Group:

Artifact:

Name:

Description:

Package name:

Packaging: ☒ **Jar** ☐ War

Java: ☐ 24 ☒ **21** ☐ 17

**Dependencies** ADD DEPENDENCIES... CTRL + B

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

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

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>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>
    </plugin>
  </plugins>
</build>
```

```
</project>
```

```
spring.application.name=spring-learn
server.port=9092
```

```
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) {  
    System.out.println("Application started...");  
    SpringApplication.run(SpringLearnApplication.class, args);  
    System.out.println("Application ended.");  
}
```

```

C:\Program Files\Java\jdk-21\bin\java.exe ...
Application started...
Application started...

  ____
 /\ /'_.'___ /_.'___ /\ /_.'___
 C )_.'___ /_.'___ /_.'___ /\ /_.'___
 /\ /_.'___ /_.'___ /_.'___ /\ /_.'___
  '___ /_.'___ /_.'___ /_.'___ /\ /_.'___
 =====|_|=====/_.'___/_.'___

:: Spring Boot ::                (v3.5.3)

2025-07-14T17:22:15:22.063+05:30 INFO 21688 --- [spring-learn] [ restartedMain] c.c.springlearn.SpringLearnApplication : Starting SpringLearnApplication using Java 21.0.2 wit
2025-07-14T17:22:15:22.067+05:30 INFO 21688 --- [spring-learn] [ restartedMain] c.c.springlearn.SpringLearnApplication : No active profile set, falling back to 1 default prof
2025-07-14T17:22:15:22.156+05:30 INFO 21688 --- [spring-learn] [ restartedMain] o.devToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtool
2025-07-14T17:22:15:22.157+05:30 INFO 21688 --- [spring-learn] [ restartedMain] o.devToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting t
2025-07-14T17:22:15:23.604+05:30 INFO 21688 --- [spring-learn] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 9092 (http)
2025-07-14T17:22:15:23.632+05:30 INFO 21688 --- [spring-learn] [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-07-14T17:22:15:23.633+05:30 INFO 21688 --- [spring-learn] [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.42]
2025-07-14T17:22:15:23.716+05:30 INFO 21688 --- [spring-learn] [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-07-14T17:22:15:23.718+05:30 INFO 21688 --- [spring-learn] [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: Initialization completed
2025-07-14T17:22:15:24.187+05:30 INFO 21688 --- [spring-learn] [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2025-07-14T17:22:15:24.226+05:30 INFO 21688 --- [spring-learn] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 9092 (http) with context path
2025-07-14T17:22:15:24.237+05:30 INFO 21688 --- [spring-learn] [ restartedMain] c.c.springlearn.SpringLearnApplication : Started SpringLearnApplication in 2.854 seconds (proc
Application ended.

```

## Spring Core – Load Country from Spring Configuration XML

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

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
```

```
  <groupId>com.cognizant</groupId>
  <artifactId>spring-learn</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>spring-learn</name>
  <description>Spring Core - Load Country Bean from XML</description>
```

```
  <properties>
    <java.version>21</java.version>
    <spring.boot.version>3.5.3</spring.boot.version>
  </properties>
```

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

```
    <!-- Spring Context (needed for XML config) -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
    </dependency>
```

```
    <!-- DevTools -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
      <scope>runtime</scope>
    </dependency>
```

```
    <!-- Logging -->
    <dependency>
      <groupId>org.slf4j</groupId>
```

```

    <artifactId>slf4j-api</artifactId>
  </dependency>
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-simple</artifactId>
    <scope>runtime</scope>
  </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>
      <version>${spring.boot.version}</version>
    </plugin>
  </plugins>
</build>
</project>

```

### application.properties

```

spring.application.name=spring-learn
logging.level.org.springframework=INFO
server.port=8083
logging.level.com.cognizant.spring_learn=DEBUG
logging.pattern.console=%d{yyMMdd} | %d{HH:mm:ss.SSS} | %-
20.20thread | %5p | %-25.25logger{25} | %25M | %m%n

```

### 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>
```

### 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("Getting Code: {}", code);
        return code;
    }
```

```
public void setCode(String code) {  
    LOGGER.debug("Setting Code: {}", code);  
    this.code = code;  
}
```

```
public String getName() {  
    LOGGER.debug("Getting Name: {}", name);  
    return name;  
}
```

```
public void setName(String name) {  
    LOGGER.debug("Setting Name: {}", name);  
    this.name = name;  
}
```

```
@Override  
public String toString() {  
    return "Country{" +  
        "code=" + code + "\" +  
        ", name=" + name + "\" +  
        '}'";  
}
```

### SpringLearnApplication.java

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

```
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;
```

```
@SpringBootApplication  
public class SpringLearnApplication {
```

```
    private static final Logger LOGGER =
```

```
LoggerFactory.getLogger(SpringLearnApplication.class);
```

```
public static void main(String[] args) {
    SpringApplication.run(SpringLearnApplication.class, args);
    displayCountry();
}
```

```
public static 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");
```

## OUTPUT:

```

C:\Program Files\Java\jdk-21\bin\java.exe" ...

  ____
 / \  / ___ \  / ___ \  / ___ \  / ___ \  / ___ \  / ___ \  / ___ \  / ___ \
( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )
 \ /  \___/  \___/  \___/  \___/  \___/  \___/  \___/  \___/  \___/  \___/
  ____

:: Spring Boot ::                (v3.5.3)

250716|16:43:46.286|restartedMain|INFO|s.SpringLearnApplication|logStarting|Starting SpringLearnApplication using Java 21.0.2 with PID 19236 (C:\Users\B PREM SAI S
250716|16:43:46.289|restartedMain|DEBUG|s.SpringLearnApplication|logStarting|Running with Spring Boot v3.5.3, Spring v6.2.8
250716|16:43:46.210|restartedMain|INFO|s.SpringLearnApplication|logStartupProfileInfo|No active profile set, falling back to 1 default profile: "default"
250716|16:43:46.291|restartedMain|INFO|ertyDefaultsPostProcessor|log|Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to d
250716|16:43:46.292|restartedMain|INFO|ertyDefaultsPostProcessor|log|For additional web related logging consider setting the 'logging.level.web' property to
250716|16:43:47.586|restartedMain|INFO|s.b.w.e.t.TomcatWebServer|initialize|Tomcat initialized with port 8083 (http)
250716|16:43:47.686|restartedMain|INFO|o.a.c.c.StandardService|logStarting service [Tomcat]
250716|16:43:47.687|restartedMain|INFO|o.a.c.c.StandardEngine|logStarting Servlet engine: [Apache Tomcat/10.1.42]
250716|16:43:47.664|restartedMain|INFO|o.a.c.c.C.[.][/] |log|Initializing Spring embedded WebApplicationContext
250716|16:43:47.665|restartedMain|INFO|bServerApplicationContext|prepareWebApplicationContext|Root WebApplicationContext: initialization completed in 1372 ms
250716|16:43:48.142|restartedMain|INFO|OptionalLiveReloadServer|startServer|LiveReload server is running on port 35729
250716|16:43:48.198|restartedMain|INFO|s.b.w.e.t.TomcatWebServer|start|Tomcat started on port 8083 (http) with context path '/'
250716|16:43:48.207|restartedMain|INFO|s.SpringLearnApplication|logStarted|STARTED SpringLearnApplication in 2.683 seconds (process running for 3.395)
250716|16:43:48.210|restartedMain|INFO|s.SpringLearnApplication|displayCountry|START
250716|16:43:48.416|restartedMain|DEBUG|c.c.spring_learn.Country |<init>|Inside Country Constructor.
250716|16:43:48.418|restartedMain|DEBUG|c.c.spring_learn.Country |setCode|Setting Code: IN
250716|16:43:48.419|restartedMain|DEBUG|c.c.spring_learn.Country |setName|Setting Name: India
250716|16:43:48.421|restartedMain|DEBUG|s.SpringLearnApplication|displayCountry|Country : Country{code='IN', name='India'}
250716|16:43:48.421|restartedMain|INFO|s.SpringLearnApplication|displayCountry|END

```



## Hello World RESTful Web Service

### 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>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>
    </plugin>
  </plugins>
</build>
```

```
</project>
```

### application.properties

```
spring.application.name=spring-learn
server.port=8083
logging.level.org.springframework=INFO
logging.level.com.cognizant.springlearn=DEBUG
```

```
logging.pattern.console=%d{yyMMdd} | %d{HH:mm:ss.SSS} | %-  
20.20thread | %5p | %-25.25logger{25} | %25M | %m%n
```

### HelloController.java

```
package com.cognizant.spring_learn.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");  
        String message = "Hello World!!";  
        LOGGER.debug("Message: {}", message);  
        LOGGER.info("END");  
        return message;  
    }  
}
```

### SpringLearnApplication.java

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

```
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

## OUTPUT

## REST - Country Web Service

**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>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>
    </plugin>
  </plugins>
</build>
```

```
</project>
```

### application.properties

```
spring.application.name=spring-learn
server.port=8083
logging.level.org.springframework=INFO
logging.level.com.cognizant.spring_learn=DEBUG
logging.pattern.console=%d{yyMMdd} | %d{HH:mm:ss.SSS} | %-
20.20thread | %5p | %-25.25logger{25} | %25M | %m%n
```

### HelloController.java

```
package com.cognizant.spring_learn.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");  
    String message = "Hello World!!";  
    LOGGER.debug("Message: {}", message);  
    LOGGER.info("END");  
    return message;  
}  
}
```

### 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.*;  
  
import java.util.List;  
  
@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.debug("Country: {}", country);  
    }  
}
```

```
    LOGGER.info("END");  
    return country;  
}
```

```
@GetMapping("/countries")  
public List<Country> getAllCountries() {  
    LOGGER.info("START");  
    ApplicationContext context = new  
ClassPathXmlApplicationContext("country.xml");  
    List<Country> list = (List<Country>) context.getBean("countryList");  
    LOGGER.debug("Countries: {}", list);  
    LOGGER.info("END");  
    return list;  
}
```

```
@GetMapping("/countries/{code}")  
public Country getCountry(@PathVariable String code) throws  
Exception {  
    LOGGER.info("START");  
    Country country = countryService.getCountry(code);  
    LOGGER.debug("Country: {}", country);  
    LOGGER.info("END");  
    return 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("getCode() called");  
    return code;  
}
```

```
public void setCode(String code) {  
    LOGGER.debug("setCode() called");  
    this.code = code;  
}
```

```
public String getName() {  
    LOGGER.debug("getName() called");  
    return name;  
}
```

```
public void setName(String name) {  
    LOGGER.debug("setName() called");  
    this.name = name;  
}
```

```
@Override  
public String toString() {  
    return "Country{" + "code=" + code + "\" + ", name=" + name + "\" +  
    '}'  
};  
}
```

### CountryService.java

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

```
import com.cognizant.spring_learn.model.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) throws Exception {
        LOGGER.info("START");
        ApplicationContext context = new
        ClassPathXmlApplicationContext("country.xml");
        List<Country> countries = (List<Country>)
        context.getBean("countryList");
```

```
        return countries.stream()
            .filter(c -> c.getCode().equalsIgnoreCase(code))
            .findFirst()
            .orElseThrow(() -> new Exception("Country not found"));
    }
}
```

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

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

```
<property name="code" value="US"/>
<property name="name" value="United States"/>
</bean>
```

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

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

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

### SpringLearnApplication.java

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

```
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("Application START");
    SpringApplication.run(SpringLearnApplication.class, args);
    LOGGER.info("Application END");
}
}

```

## OUTPUT:

```

[C:\Program Files\Java\jdk-21\bin\java.exe" ...]
17:41:52.773 [main] INFO com.cognizant.spring_learn.SpringLearnApplication -- Application START
17:41:53.199 [restartedMain] INFO com.cognizant.spring_learn.SpringLearnApplication -- Application START

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

:: Spring Boot ::                (v3.5.3)

250716|17:41:54.157|restartedMain|INFO|.s.SpringLearnApplication|logStarting|Starting SpringLearnApplication using Java 21.0.2 with PID 24036 (C:\Users\
250716|17:41:54.161|restartedMain|DEBUG|.s.SpringLearnApplication|logStarting|Running with Spring Boot v3.5.3, Spring v6.2.8
250716|17:41:54.163|restartedMain|INFO|.s.SpringLearnApplication|logStartupProfileInfo|No active profile set, falling back to 1 default profile: "default"
250716|17:41:55.175|restartedMain|INFO|ertyDefaultsPostProcessor|logToDevtools property defaults active! Set 'spring.devtools.add-properties' to '
250716|17:41:55.175|restartedMain|INFO|ertyDefaultsPostProcessor|logToFor additional web related logging consider setting the 'logging.level.web'
250716|17:41:57.035|restartedMain|INFO|s.b.w.e.t.TomcatWebServer|initialize|Tomcat initialized with port 8083 (http)
250716|17:41:57.056|restartedMain|INFO|o.a.c.c.StandardService|log|Starting service [Tomcat]
250716|17:41:57.056|restartedMain|INFO|o.a.c.c.core.StandardEngine|log|Starting Servlet engine: [Apache Tomcat/10.1.42]
250716|17:41:57.118|restartedMain|INFO|o.a.c.c.C.[.]/|log|Initializing Spring embedded WebApplicationContext
250716|17:41:57.118|restartedMain|INFO|bServerApplicationContext|prepareWebApplicationContext|Root WebApplicationContext: initialization completed in 1928 ms
250716|17:41:57.682|restartedMain|INFO|.OptionalLiveReloadServer|startServer|LiveReload server is running on port 35729
250716|17:41:57.742|restartedMain|INFO|s.b.w.e.t.TomcatWebServer|start|Tomcat started on port 8083 (http) with context path '/'
250716|17:41:57.758|restartedMain|INFO|.s.SpringLearnApplication|logStarted|Started SpringLearnApplication in 4.551 seconds (process running for 5.633)
250716|17:41:57.766|restartedMain|INFO|.s.SpringLearnApplication|main|Application END

```

← → ↻ ⓘ localhost:8083/countries

Pretty-print ☐

```
[{"code": "IN", "name": "India"}, {"code": "US", "name": "United States"}, {"code": "JP", "name": "Japan"}, {"code": "DE", "name": "Germany"}]
```

← → ↻ ⓘ localhost:8083/country

Pretty-print ☐

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

## REST - Get country based on country code

### CountryController.java

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

```
import com.cognizant.spring_learn.model.Country;
import com.cognizant.spring_learn.service.CountryService;
import com.cognizant.spring_learn.service.exception.CountryNotFoundException;
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.*;
```

```
import java.util.List;
```

```
@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 - getCountryIndia");
        ApplicationContext context = new
        ClassPathXmlApplicationContext("country.xml");
        Country country = context.getBean("in", Country.class);
        LOGGER.debug("Country: {}", country);
        LOGGER.info("END - getCountryIndia");
        return country;
    }
```

```
    @GetMapping("/countries")
    public List<Country> getAllCountries() {
        LOGGER.info("START - getAllCountries");
        ApplicationContext context = new
        ClassPathXmlApplicationContext("country.xml");
```

```
@SuppressWarnings("unchecked")
List<Country> countryList = (List<Country>) context.getBean("countryList");
```

```
LOGGER.debug("Country List: {}", countryList);
LOGGER.info("END - getAllCountries");
return countryList;
}
```

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

### CountryNotFoundException.java

```
package com.cognizant.spring_learn.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);
    }
}
```

### 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("getCode called");  
        return code;  
    }  
  
    public void setCode(String code) {  
        LOGGER.debug("setCode called");  
        this.code = code;  
    }  
  
    public String getName() {  
        LOGGER.debug("getName called");  
        return name;  
    }  
  
    public void setName(String name) {  
        LOGGER.debug("setName called");  
        this.name = name;  
    }  
  
    @Override  
    public String toString() {  
        return "Country [code=" + code + ", name=" + name + "];"  
    }  
}
```

CountryService.java

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

import com.cognizant.spring_learn.model.Country;
import com.cognizant.spring_learn.service.exception.CountryNotFoundException;
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) throws CountryNotFoundException {
        LOGGER.info("START - getCountry from service");
        ApplicationContext context = new
            ClassPathXmlApplicationContext("country.xml");

        @SuppressWarnings("unchecked")
        List<Country> countryList = (List<Country>) context.getBean("countryList");

        return countryList.stream()
            .filter(c -> c.getCode().equalsIgnoreCase(code))
            .findFirst()
            .orElseThrow(() -> new CountryNotFoundException("Country not
found"));
    }
}
```



```

[C:\Program Files\Java\jdk-21\bin\java.exe" ...
18:48:18.363 [main] INFO com.cognizant.spring_learn.SpringLearnApplication -- Application START
18:48:18.778 [restartedMain] INFO com.cognizant.spring_learn.SpringLearnApplication -- Application START

  ____  _
 / ___|| | | |
| |___| | | |
 \___|| | | |
      | | | |
      |_| |_|

:: Spring Boot ::                (v3.5.3)

250716|18:48:20.084|restartedMain| INFO|s.SpringLearnApplication| logStarting|Starting SpringLearnApplication using Java 21.0.2 with PID 23232 (C:\Users\B
250716|18:48:20.088|restartedMain| [DEBUG]|s.SpringLearnApplication| logStarting|Running with Spring Boot v3.5.3, Spring v6.2.8
250716|18:48:20.089|restartedMain| INFO|s.SpringLearnApplication| logStartupProfileInfo|No active profile set, falling back to 1 default profile: "default"
250716|18:48:20.190|restartedMain| INFO|ertyDefaultsPostProcessor| logTo|Devtools property defaults active! Set 'spring.devtools.add-properties' to '
250716|18:48:20.191|restartedMain| INFO|ertyDefaultsPostProcessor| logTo|For additional web related logging consider setting the 'logging.level.web'
250716|18:48:22.372|restartedMain| INFO|s.b.w.e.t.TomcatWebServer| initialize|Tomcat initialized with port 8083 (http)
250716|18:48:22.402|restartedMain| INFO|o.a.c.c.StandardService| log|Starting service [Tomcat]
250716|18:48:22.403|restartedMain| INFO|o.a.c.c.core.StandardEngine| log|Starting Servlet engine: [Apache Tomcat/10.1.42]
250716|18:48:22.487|restartedMain| INFO|o.a.c.c.C.[.][/] | log|Initializing Spring embedded WebApplicationContext
250716|18:48:22.489|restartedMain| INFO|bServerApplicationContext|prepareWebApplicationContext|Root WebApplicationContext: initialization completed in 2297 ms
250716|18:48:23.215|restartedMain| INFO|OptionalLiveReloadServer| startServer|LiveReload server is running on port 35729
250716|18:48:23.284|restartedMain| INFO|s.b.w.e.t.TomcatWebServer| start|Tomcat started on port 8083 (http) with context path '/'
250716|18:48:23.301|restartedMain| INFO|s.SpringLearnApplication| logStarted|Started SpringLearnApplication in 4.514 seconds (process running for 5.446)
250716|18:48:23.306|restartedMain| INFO|s.SpringLearnApplication| main|Application END

```

```

localhost:8083/countries/in

Pretty-print ☐

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

```

## 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/> <!-- 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>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>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
  </dependency>
  <dependency>
    <groupId>javax.xml.bind</groupId>
```

```
<artifactId>jaxb-api</artifactId>
<version>2.3.1</version>
</dependency>
<dependency>
  <groupId>org.glassfish.jaxb</groupId>
  <artifactId>jaxb-runtime</artifactId>
  <version>2.3.1</version>
</dependency>

<dependency>
  <groupId>io.jsonwebtoken</groupId>
  <artifactId>jjwt</artifactId>
  <version>0.9.0</version>
</dependency>

</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.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("Application START");  
    SpringApplication.run(SpringLearnApplication.class, args);  
    LOGGER.info("Application END");  
}
```

### AuthenticationController.java

```
package com.cognizant.spring_learn.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 java.util.Base64;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.Map;  
  
@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 of 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 of authenticate");  
return map;  
}
```

```
private String getUser(String authHeader) {  
    LOGGER.info("Start of getUser");  
    String encodedCredentials = authHeader.substring("Basic ".length());  
    byte[] decodedBytes =  
Base64.getDecoder().decode(encodedCredentials);  
    String decodedCredentials = new String(decodedBytes);  
    String user = decodedCredentials.split(":")[0];  
    LOGGER.debug("Extracted user: {}", user);  
    LOGGER.info("End of getUser");  
    return user;  
}
```

```
private String generateJwt(String user) {  
    LOGGER.info("Start of generateJwt");  
    JwtBuilder builder = Jwts.builder();  
    builder.setSubject(user);  
    builder.setIssuedAt(new Date());  
    builder.setExpiration(new Date((new Date()).getTime() + 1200000));  
    builder.signWith(SignatureAlgorithm.HS256, "secretkey");  
    String token = builder.compact();  
    LOGGER.debug("Generated Token: {}", token);  
    LOGGER.info("End of generateJwt");  
    return token;  
}
```

[illegible]