

## 1.spring-rest-handson

### Create a Spring Web Project using Maven

#### 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 Web Project</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>

<build>

    <plugins>

        <plugin>

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

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

        </plugin>

    </plugins>

</build>
```

```

        </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)

    {

        SpringApplication.run(SpringLearnApplication.class, args);

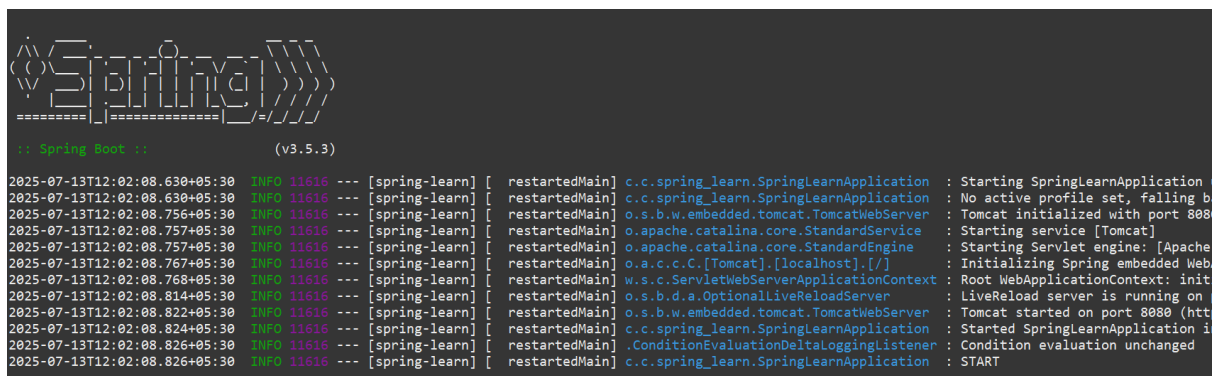
        LOGGER.info("START");

    }

}

```

## Output :



```

[ Spring Boot ] (v3.5.3)
2025-07-13T12:02:08.630+05:30 INFO 11616 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : Starting SpringLearnApplication
2025-07-13T12:02:08.630+05:30 INFO 11616 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : No active profile set, falling b
2025-07-13T12:02:08.756+05:30 INFO 11616 --- [spring-learn] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 808
2025-07-13T12:02:08.757+05:30 INFO 11616 --- [spring-learn] [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-07-13T12:02:08.757+05:30 INFO 11616 --- [spring-learn] [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache
2025-07-13T12:02:08.767+05:30 INFO 11616 --- [spring-learn] [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded Web
2025-07-13T12:02:08.768+05:30 INFO 11616 --- [spring-learn] [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: init
2025-07-13T12:02:08.814+05:30 INFO 11616 --- [spring-learn] [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on
2025-07-13T12:02:08.822+05:30 INFO 11616 --- [spring-learn] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (htt
2025-07-13T12:02:08.824+05:30 INFO 11616 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : Started SpringLearnApplication i
2025-07-13T12:02:08.826+05:30 INFO 11616 --- [spring-learn] [ restartedMain] .ConditionEvaluationDeltaLoggingListener : Condition evaluation unchanged
2025-07-13T12:02:08.826+05:30 INFO 11616 --- [spring-learn] [ restartedMain] c.c.spring_learn.SpringLearnApplication : START

```

## Spring Core – Load Country from Spring Configuration XML

### pom.xml

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

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

  <modelVersion>4.0.0</modelVersion>

  <parent>

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

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

    <version>3.5.3</version>

    <relativePath/>

  </parent>

  <groupId>com.cognizant</groupId>

  <artifactId>spring-learn</artifactId>

  <version>0.0.1-SNAPSHOT</version>

  <name>spring-learn</name>

  <description>Spring Web Project</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>
    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

## country.xml

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

<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd>

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

## application.properties

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

## Country.java

```
package com.cognizant.spring_learn;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class Country
{
    private static final Logger LOGGER = LoggerFactory.getLogger(Country.class);

    private String code;

    private String name;

    public Country() {
        LOGGER.debug("Inside Country Constructor.");
    }

    public String getCode() {
        LOGGER.debug("Inside getCode()");
    }
}
```

```

        return code;
    }

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

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

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

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

```

### SpringLearnApplication.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);
}

```





- name – matches the setter method `setCode(String code)`
- value – literal value to assign

---

### ◆ **ApplicationContext vs ClassPathXmlApplicationContext**

Feature	ApplicationContext	ClassPathXmlApplicationContext
Interface	Yes	No (Concrete class)
Purpose	Main Spring container interface	Loads XML config from classpath
Usage	Used to get Spring beans	One way to implement ApplicationContext

#### **Example:**

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

This reads the XML configuration and instantiates the beans defined.

---

### ◆ **What Happens in `context.getBean("country")`**

1. Spring loads `country.xml`.
2. It parses `<bean>` definitions.
3. Instantiates `Country` class (calls constructor).
4. Injects property values using setters (`setCode`, `setName`).
5. Registers the bean with ID `country`.
6. When `context.getBean("country", Country.class)` is called:
  - Spring returns a reference to the managed `Country` object.

---

### **Summary**

Component	Purpose
<code>Country.java</code>	POJO with log messages in constructor and methods
<code>country.xml</code>	Configures the bean and its property values
<code>SpringLearnApplication.java</code>	Loads context, fetches and prints bean
<code>context.getBean()</code>	Retrieves a fully initialized bean instance