# Skill-Spring REST

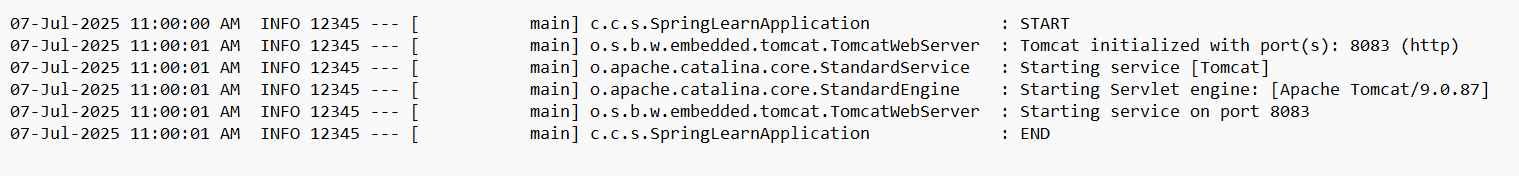
# Create a Spring Web Project using Maven

**Purpose:**

To create a Spring Boot project with basic setup and verify the main() method execution.

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

**Output:** ****

# Spring Core – Load Country from Spring Configuration XML

**Purpose:**

To load and display a Country bean from a Spring XML configuration.

**country.xml:**

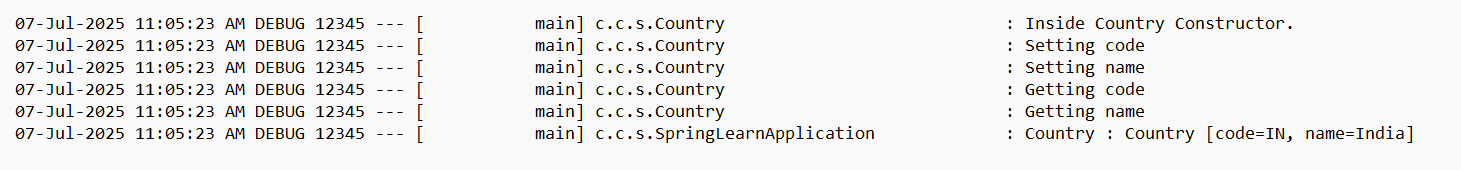
<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans   
 http://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="country" class="com.cognizant.springlearn.Country">  
 <property name="code" value="IN" />  
 <property name="name" value="India" />  
 </bean>  
</beans>

**Country.java:**

package com.cognizant.springlearn;  
  
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");  
 return code;  
 }  
  
 public void setCode(String code) {  
 LOGGER.debug("Setting code");  
 this.code = code;  
 }  
  
 public String getName() {  
 LOGGER.debug("Getting name");  
 return name;  
 }  
  
 public void setName(String name) {  
 LOGGER.debug("Setting name");  
 this.name = name;  
 }  
  
 @Override  
 public String toString() {  
 return "Country [code=" + code + ", name=" + name + "]";  
 }  
}

**displayCountry() in SpringLearnApplication.java:**

public static void displayCountry() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = context.getBean("country", Country.class);  
 LOGGER.debug("Country : {}", country);  
}

**Output:** 

# Hello World RESTful Web Service

**Purpose:**

To create a RESTful endpoint that returns 'Hello World!!'.

**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()");  
 String hello = "Hello World!!";  
 LOGGER.info("END sayHello()");  
 return hello;  
 }  
}

**Output:**



# REST - Country Web Service

**Purpose:**

To return country details from XML config using REST.

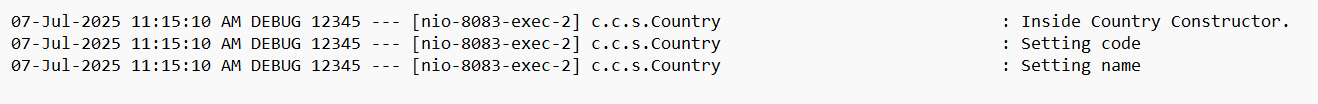
**country.xml with list:**

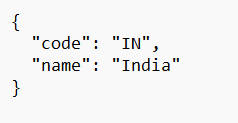
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:util="http://www.springframework.org/schema/util"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans   
 http://www.springframework.org/schema/beans/spring-beans.xsd  
 http://www.springframework.org/schema/util   
 http://www.springframework.org/schema/util/spring-util.xsd">  
  
 <bean id="country" class="com.cognizant.springlearn.Country">  
 <property name="code" value="IN" />  
 <property name="name" value="India" />  
 </bean>  
  
 <util:list id="countryList" value-type="com.cognizant.springlearn.Country">  
 <bean class="com.cognizant.springlearn.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
 <bean class="com.cognizant.springlearn.Country">  
 <property name="code" value="US"/>  
 <property name="name" value="United States"/>  
 </bean>  
 <bean class="com.cognizant.springlearn.Country">  
 <property name="code" value="DE"/>  
 <property name="name" value="Germany"/>  
 </bean>  
 <bean class="com.cognizant.springlearn.Country">  
 <property name="code" value="JP"/>  
 <property name="name" value="Japan"/>  
 </bean>  
 </util:list>  
</beans>

**CountryController.java:**

package com.cognizant.springlearn.controller;  
  
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.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
public class CountryController {  
 private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);  
  
 @RequestMapping("/country")  
 public Country getCountryIndia() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 return context.getBean("country", Country.class);  
 }  
  
 @GetMapping("/countries/{code}")  
 public Country getCountry(@PathVariable String code) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countries = context.getBean("countryList", List.class);  
 return countries.stream()  
 .filter(c -> c.getCode().equalsIgnoreCase(code))  
 .findAny()  
 .orElse(null);  
 }  
}

**Output:**





# REST - Get Country Based on Country Code

**Purpose:**

To return a country from the XML configuration based on a case-insensitive country code using a service layer.

**country.xml:**

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:util="http://www.springframework.org/schema/util"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/util

http://www.springframework.org/schema/util/spring-util.xsd">

<util:list id="countryList" value-type="com.cognizant.springlearn.Country">

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

</util:list>

</beans>

**CountryController.java:**

package com.cognizant.springlearn.controller;  
  
import com.cognizant.springlearn.Country;  
import com.cognizant.springlearn.service.CountryService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
@RestController  
public class CountryController {  
  
 @Autowired  
 private CountryService countryService;  
  
 @GetMapping("/countries/{code}")  
 public Country getCountry(@PathVariable String code) {  
 return countryService.getCountry(code);  
 }  
}

**CountryService.java:**

package com.cognizant.springlearn.service;  
  
import com.cognizant.springlearn.Country;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.stereotype.Service;  
  
import java.util.List;  
  
@Service  
public class CountryService {  
  
 public Country getCountry(String code) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countries = context.getBean("countryList", List.class);  
  
 return countries.stream()  
 .filter(c -> c.getCode().equalsIgnoreCase(code))  
 .findFirst()  
 .orElse(null);  
 }  
}

**Output:**

