

Cognizant Digital Nurture 4.0 – Java FSE

Week 4 – Hands-On Report

Name: Shaik Sulthan - 6431169

Track: Deep Skilling – Java Full Stack Engineer

Batch: 2025

Create RESTful Web Service to handle POST request of Country

A new RESTful Web Service method to handle POST request of Country. Follow steps below to incorporate the same:

- Create new method in CountryController based on the following details:
 - Annotation - `@PostMapping()`
 - Method Signature - `public void addCountry()`
- Within this method include "Start" logger.
- Start the web application
- Open Git Bash
- Execute the following curl command, to invoke the web service:
 - `-i` to display the headers
 - `-X` to define the HTTP method type
 - `-s` silent mode, so that performance details are not displayed

Main file :

```
package com.cognizant.spring_learn;

import com.cognizant.spring_learn.model.Country;
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 {
    public static void main(String[] args) {
        SpringApplication.run(SpringLearnApplication.class, args);

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

        Country country = (Country) context.getBean("country");

        System.out.println(country);
    }
}
```

```
}  
}
```

Country class:

```
package com.cognizant.spring_learn.model;  
  
public class Country {  
    private String code;  
    private String name;  
  
    public String getCode() {  
        return code;  
    }  
    public void setCode(String code) {  
        this.code = code;  
    }  
  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    @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  
        https://www.springframework.org/schema/beans/spring-beans.xsd">  
  
    <bean id="country" class="com.cognizant.spring_learn.model.Country">  
        <property name="code" value="IN"/>  
        <property name="name" value="India"/>  
    </bean>  
</beans>
```

OUTPUT:

```
.support.ClassPathXmlApplicationContext@2c8d66b2  
17:07:13.363 [main] DEBUG org.springframework.beans.factory.xml.XmlBeanDefinitionReader -- Loaded 1 bean definitions from class path  
resource [country.xml]  
17:07:13.463 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory -- Creating shared instance of singleton  
bean 'country'  
17:07:13.463 [main] DEBUG com.cognizant.springlearn.Country -- Inside Country Constructor.  
17:07:13.658 [main] DEBUG com.cognizant.springlearn.Country -- Inside setCode()  
17:07:13.658 [main] DEBUG com.cognizant.springlearn.Country -- Inside setName()  
17:07:13.695 [main] DEBUG com.cognizant.springlearn.SpringLearnApplication -- Country : Country{code='IN', name='India'}
```