# Spring REST using Spring Boot 3

# spring-rest-handson

**Hello World RESTful Web Service**   
Write a REST service in the spring learn application created earlier, that returns the text "Hello World!!" using Spring Web Framework. Refer details below:

**Implementation:**

1. **Create the Controller Class**

**Controller:** com.cognizant.spring-learn.controller.HelloController

**Method:** GET  
**URL:** /hello

**Method Signature:** public String sayHello()  
**Method Implementation:** return hard coded string "Hello World!!"

****HelloController.java****

**package com.cognizant.spring\_learn.controller;**

**import org.slf4j.Logger;**

**import org.slf4j.LoggerFactory;**

**import org.springframework.web.bind.annotation.\*;**

**@RestController**

**public class HelloController {**

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

**@GetMapping("/hello")**

**public String sayHello() {**

**LOGGER.info("START");**

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

**LOGGER.info("END");**

**return response;**

**}**

**}**

1. **SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import com.cognizant.spring\_learn.Country;

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) {

LOGGER.debug("START - main()");

SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.debug("END - main()");

}

}

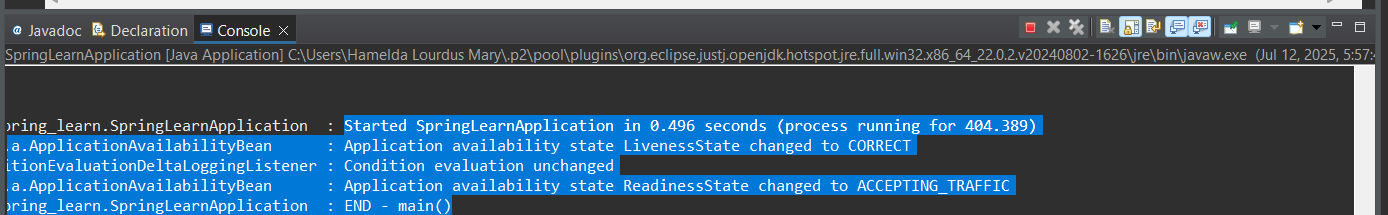
**3.application.properties**

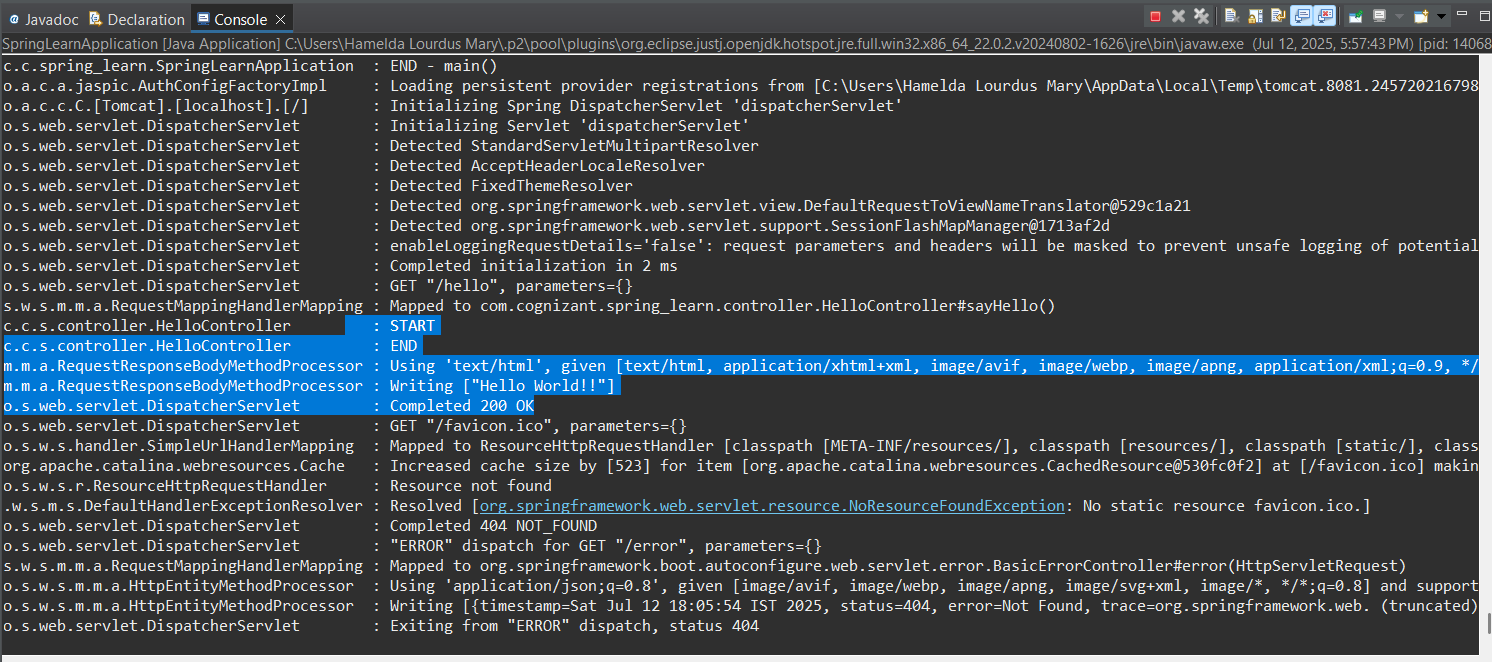
spring.application.name=spring-learn

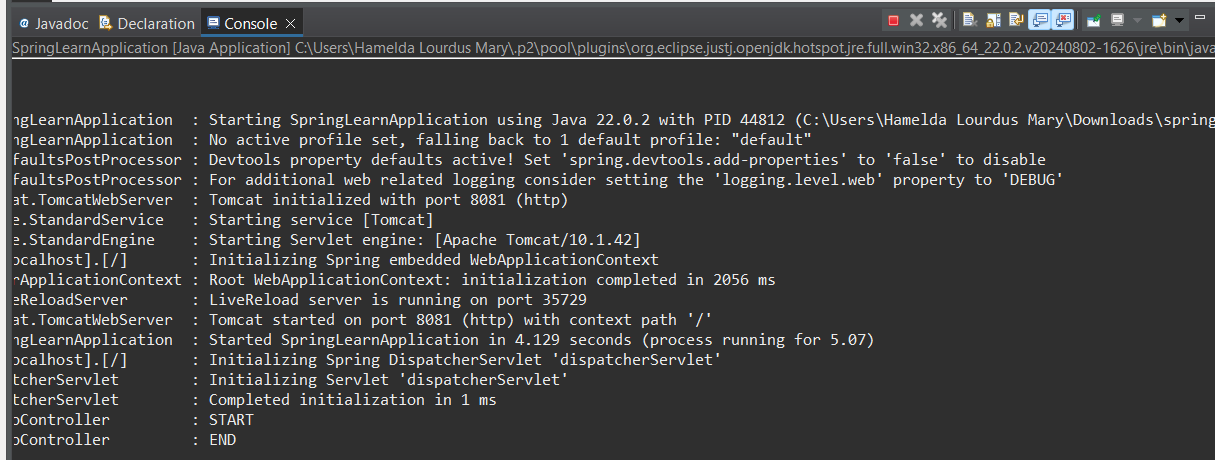
server.port=8081

logging.level.root=DEBUG

**Output:**



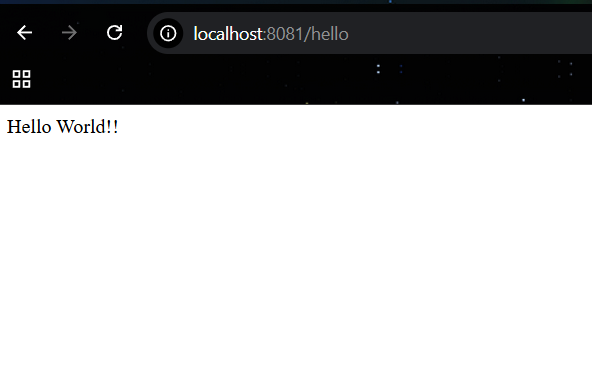




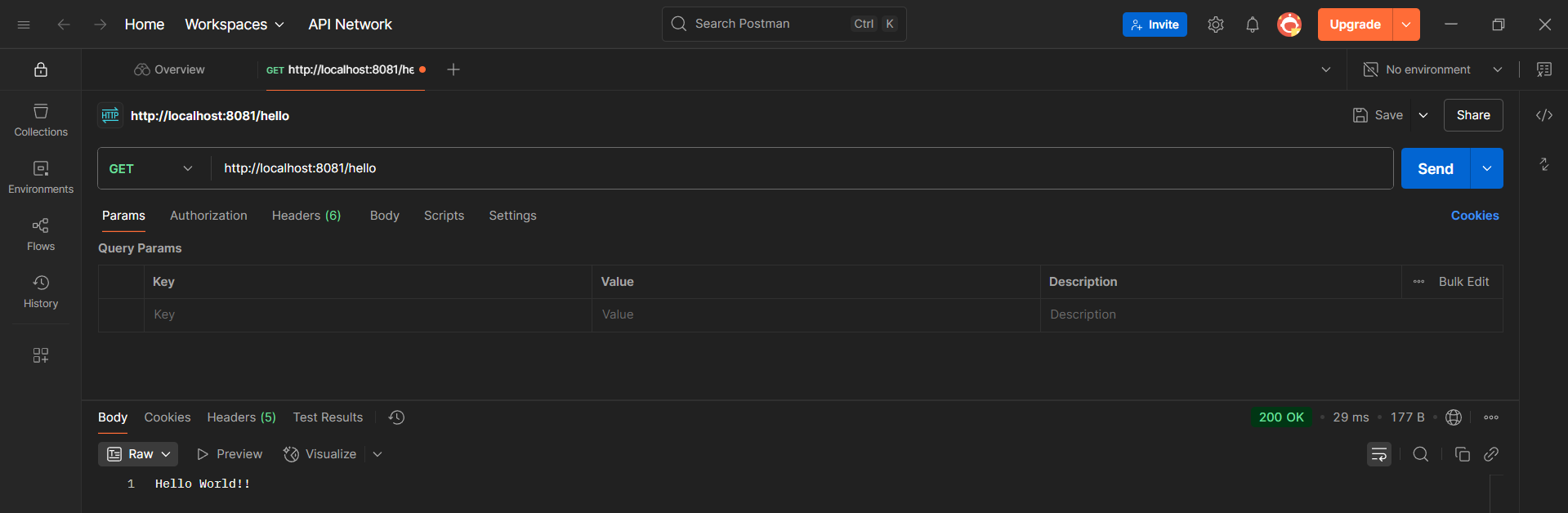
**Sample Request**: http://localhost:8083/hello  
**Sample Response:** Hello World!!

Try the URL http://localhost:8083/hello in both chrome browser and postman.

**Chrome browser:**



**Postman:**



**REST - Country Web Service**   
  
Write a REST service that returns India country details in the earlier created spring learn application.

**Implementation:**

**URL**: /country  
**Controller**: com.cognizant.spring-learn.controller.CountryController  
**Method Annotation**: @RequestMapping  
**Method Name**: getCountryIndia()

**Method Implementation**: Load India bean from spring xml configuration and return

1. **CountryController.java**

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.model.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.web.bind.annotation.\*;

@RestController

public class CountryController {

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

@RequestMapping("/country")

public Country getCountryIndia() {

LOGGER.info("START getCountryIndia()");

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

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

context.close();

LOGGER.info("END getCountryIndia()");

return country;

}

}

1. **Country.java**

src/main/java/com/cognizant/spring\_learn/model/Country.java

package com.cognizant.spring\_learn.model;

public class Country {

private String code;

private String name;

// Constructors

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;

}

}

1. **country.xml**

src/main/resources/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="in" class="com.cognizant.spring\_learn.model.Country">

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

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

</bean>

</beans>

1. **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) {

LOGGER.debug("START - main()");

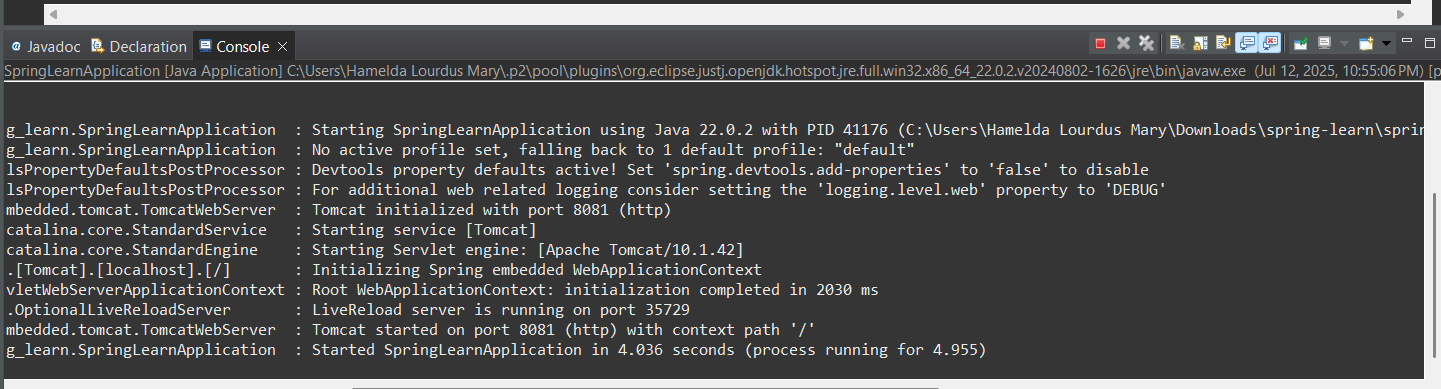
SpringApplication.run(SpringLearnApplication.class, args);

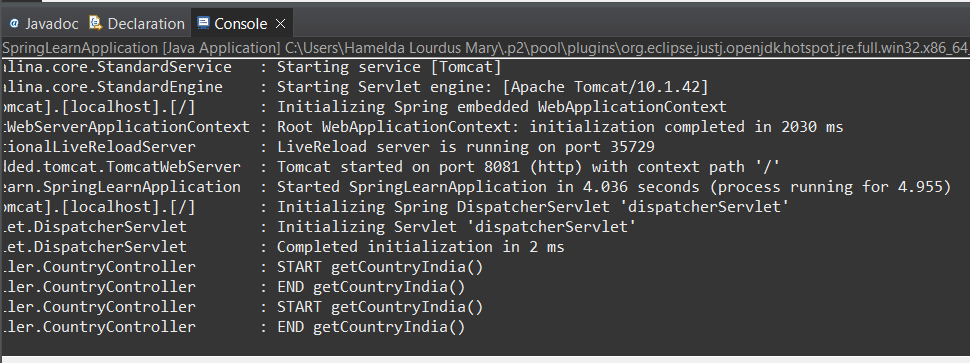
LOGGER.debug("END - main()");

}

}

**Output:**





**Sample Request**: http://localhost:8083/country  
**Sample Response**:

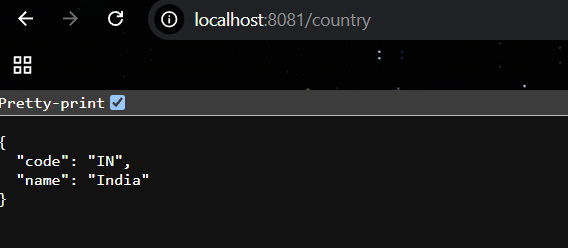
{

  "code": "IN",

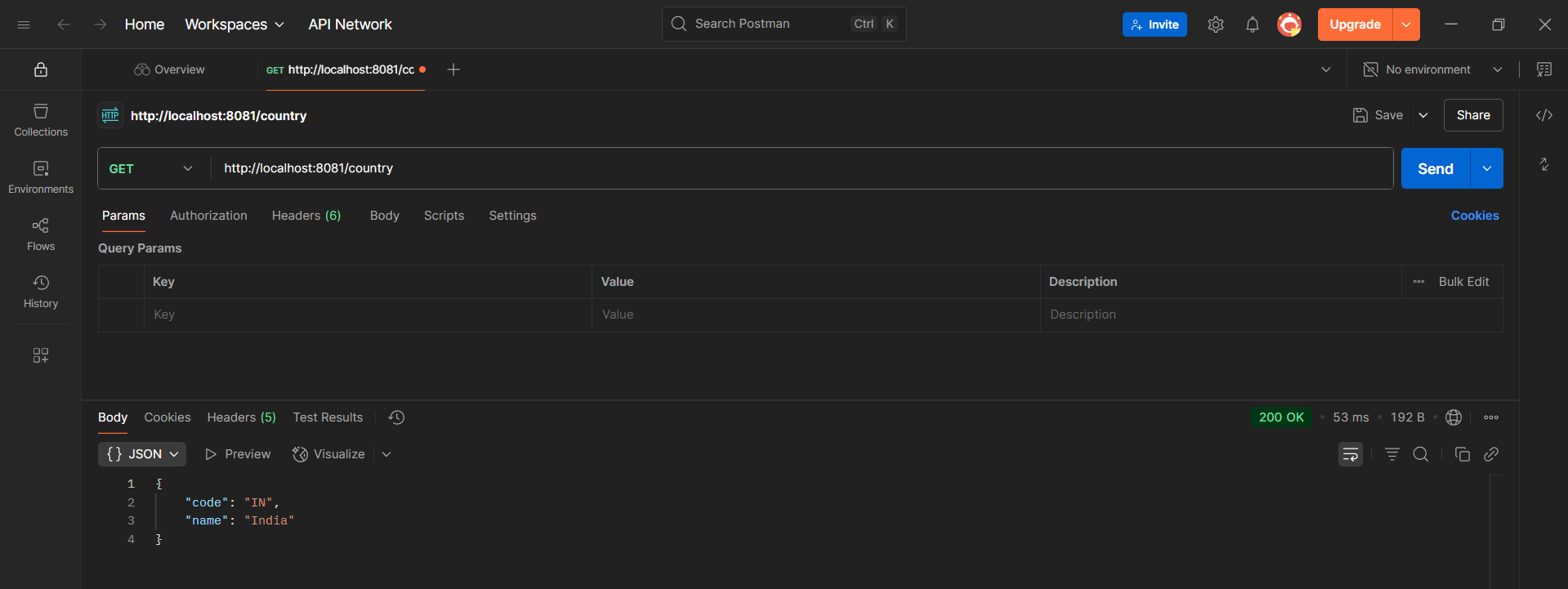
  "name": "India"

}

**Chrome Browser:**



**Postman:**



**REST - Get country based on country code**   
  
Write a REST service that returns a specific country based on country code. The country code should be case insensitive.

**Controller**: com.cognizant.spring-learn.controller.CountryController  
**Method Annotation:** @GetMapping("/countries/{code}")  
**Method Name**: getCountry(String code)  
**Method Implemetation**: Invoke countryService.getCountry(code)

1. **Country.java**

src/main/java/com/cognizant/spring\_learn/model/Country.java

package com.cognizant.spring\_learn.model;

public class Country {

private String code;

private String name;

// Constructors

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;

}

}

1. **country.xml**

src/main/resources/country.xml

<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="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

<bean class="com.cognizant.spring\_learn.model.Country">

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

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

</bean>

<bean class="com.cognizant.spring\_learn.model.Country">

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

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

</bean>

<bean class="com.cognizant.spring\_learn.model.Country">

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

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

</bean>

</list>

</constructor-arg>

</bean>

</beans>

**3.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.annotation.ImportResource;

import org.springframework.context.support.ClassPathXmlApplicationContext;

@ImportResource("classpath:country.xml")

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

LOGGER.debug("START - main()");

SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.debug("END - main()");

}

}

**4.Service Method:**com.cognizant.spring-learn.service.CountryService.getCountry(String code)

**CountryService.java**

src/main/java/com/cognizant/spring\_learn/service/CountryService.java

package com.cognizant.spring\_learn.service;

import com.cognizant.spring\_learn.model.Country;

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

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

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

@Autowired

@Qualifier("countryList")

private List<Country> countryList;

public Country getCountry(String code) {

return countryList.stream()

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

.findFirst()

.orElse(null); // Or throw an exception

}

}

1. **CountryController**

**Controller**: com.cognizant.spring-learn.controller.CountryController  
**Method Annotation:** @GetMapping("/countries/{code}")  
**Method Name**: getCountry(String code)  
**Method Implemetation**: Invoke countryService.getCountry(code)

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.web.bind.annotation.\*;

@RestController

public class CountryController {

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

@Autowired

private CountryService countryService;

@GetMapping("/countries/{code}")

public Country getCountry(@PathVariable String code) {

LOGGER.info("START getCountry()");

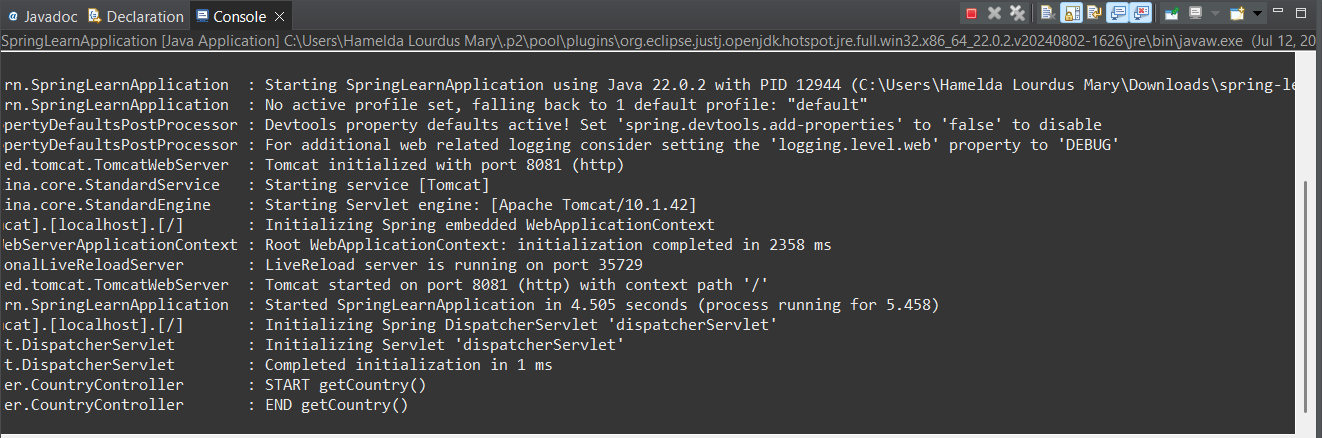
Country country = countryService.getCountry(code);

LOGGER.info("END getCountry()");

return country;

}

}

**Output:**  


**Sample Request**: <http://localhost:8083/country/in>

**Sample Response**:

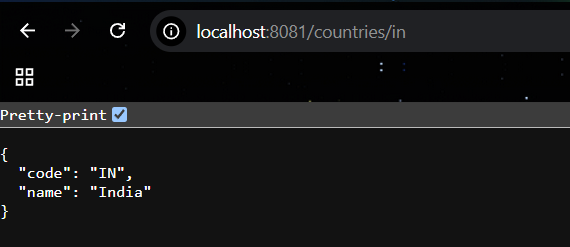
{

  "code": "IN",

  "name": "India"

}

**Chrome Browser:**



**Postman:**

