**1. Hello World RESTful Web Service**

**Steps:**

1. Open STS4 and import the Spring Boot project (spring-learn).
2. Create a new package: com.cognizant.spring\_learn.controller.
3. Inside the package, create a class named HelloController.

package com.cognizant.springlearn.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class HelloController {

@GetMapping("/hello")

public String sayHello() {

return "Hello World!!";

}

}

1. Annotate the class with @RestController.
2. Create a method sayHello() and annotate it with @GetMapping("/hello").
3. Return the hardcoded string "Hello World!!" from the method.
4. Run the application from SpringLearnApplication.java.

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) {

SpringApplication.run(SpringlearnApplication.class, args);

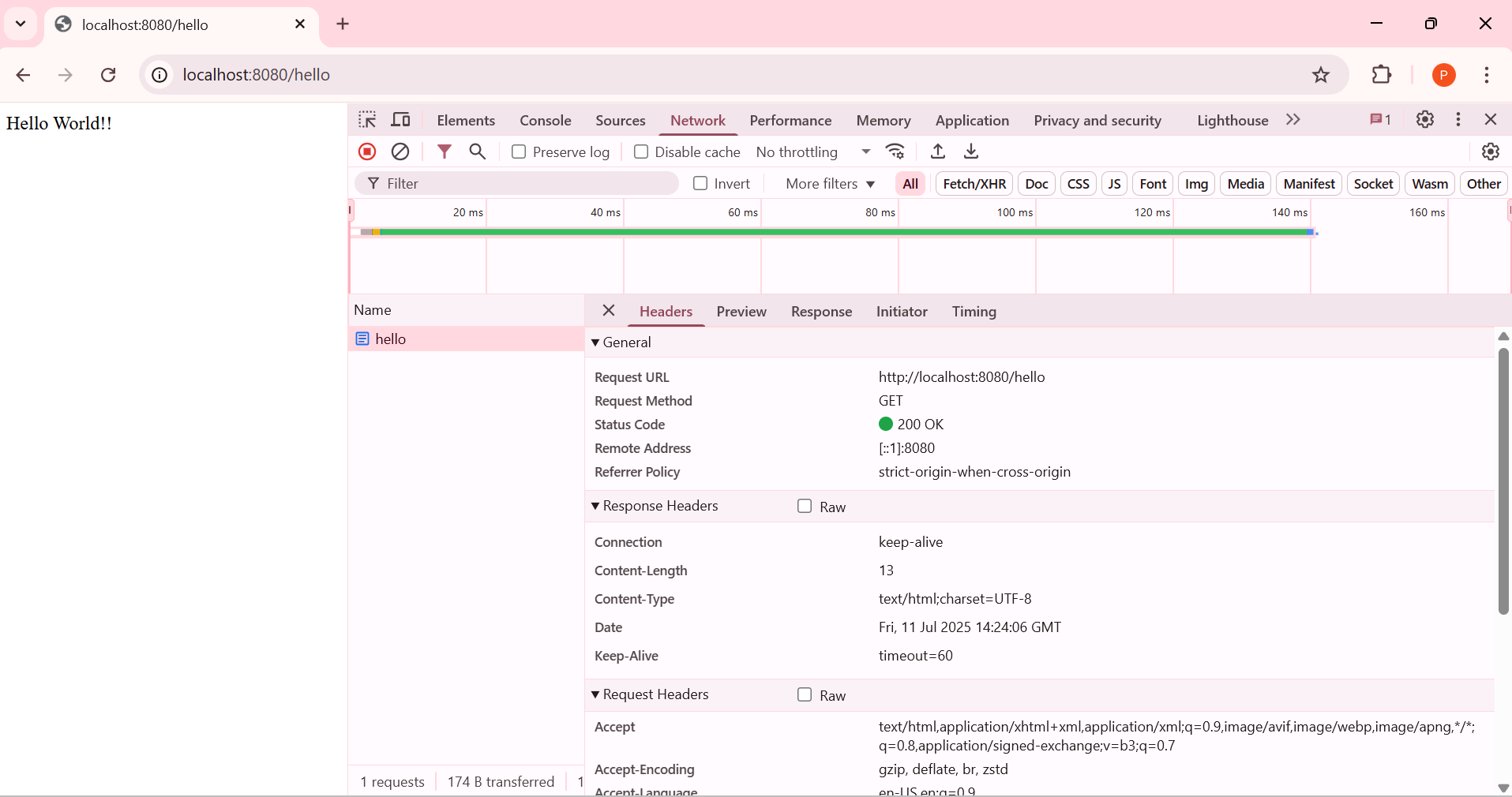
}

}

1. Open browser and access: <http://localhost:8080/hello>

**Output:**

****



**2. REST - Country Web Service**

**Steps:**

1. Create a Country Java class with code, name, getters, setters, and toString().

package com.cognizant.spring\_learn;

public class Country {

private String code;

private String name;

public Country() {}

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 + "]";

}

}

1. Create a Spring XML config file country.xml in src/main/resources and define a country bean for India.

<?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>

1. Create a package com.cognizant.spring\_learn.controller.
2. Inside it, create a class CountryController and annotate it with @RestController.

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.Country;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class CountryController {

@RequestMapping("/country")

public Country getCountryIndia() {

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

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

return country;

}

}

1. Create a method getCountryIndia() with @RequestMapping("/country") that loads the bean from country.xml and returns it.
2. Run the application from SpringLearnApplication.java.

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

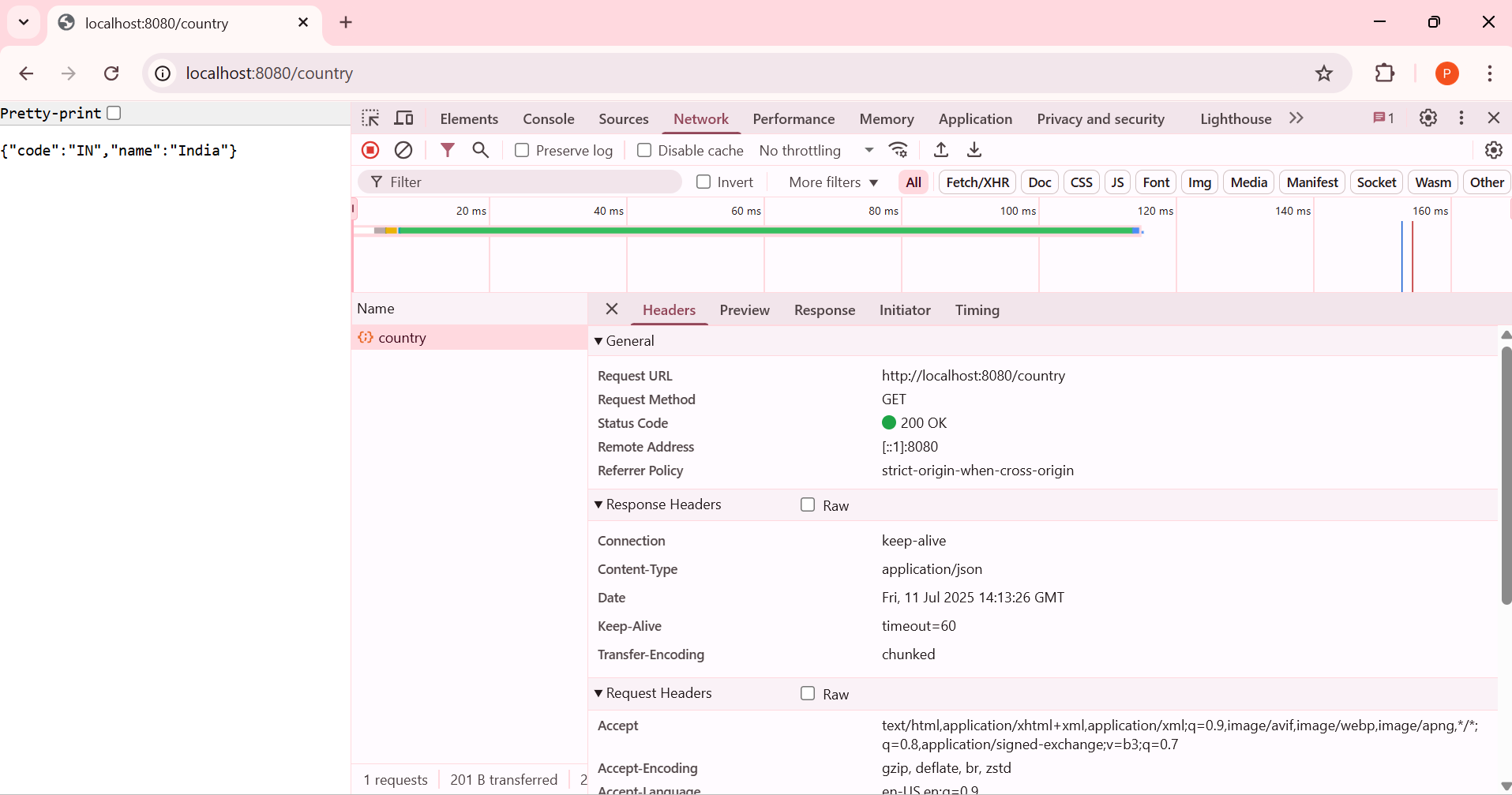
SpringApplication.run(SpringLearnApplication.class, args);

}

}

1. Test the API in browser at http://localhost:8080/country.

**Output:**

****

What happens in the controller method?

* @RequestMapping("/country") maps the URL /country to the method.
* ApplicationContext loads country.xml from resources.
* It retrieves the country bean defined in XML.
* The bean is returned from the method.

How the bean is converted into JSON?

* Spring Boot uses Jackson (automatically included via spring-boot-starter-web) to convert Java objects to JSON.
* Since Country is a simple POJO, Spring auto-converts it to JSON in the response body.

**3. REST - Get country based on country code**

**Steps:**

1. Create a Country Java class with code, name, getters, setters, and toString().

package com.cognizant.spring\_learn;

public class Country {

private String code;

private String name;

public Country() {}

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 + "]";

}

}

1. Create a Spring XML config country.xml with multiple country beans and a countryList bean.

<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.Country">

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

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

<constructor-arg>

<list>

<ref bean="in"/>

<ref bean="us"/>

<ref bean="de"/>

<ref bean="jp"/>

</list>

</constructor-arg>

</bean>

</beans>

1. Create a service class CountryService in com.cognizant.spring\_learn.service.

package com.cognizant.spring\_learn.service;

import com.cognizant.spring\_learn.Country;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.util.List;

public class CountryService {

public Country getCountry(String code) {

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

List<Country> countryList = context.getBean("countryList", List.class);

return countryList.stream()

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

.findFirst()

.orElse(null);

}

}

1. In the service, load countryList from country.xml and use a case-insensitive match to return the country.
2. Create a REST controller CountryController in com.cognizant.spring\_learn.controller.

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.Country;

import com.cognizant.spring\_learn.service.CountryService;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class CountryController {

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

public Country getCountry(@PathVariable String code) {

CountryService countryService = new CountryService();

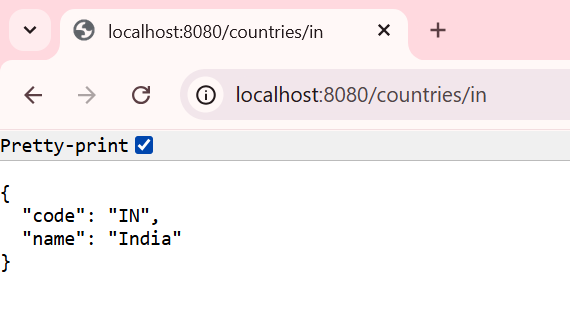
return countryService.getCountry(code);

}

}

1. Create a method getCountry(@PathVariable String code) and annotate it with @GetMapping("/countries/{code}").
2. In this method, call countryService.getCountry(code) and return the result.
3. Run the application and test the API at http://localhost:8080/countries/in.

**Output:**

****