**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:  
   
**Method:** GET  
 **URL:** /hello  
 **Controller:** com.cognizant.spring-learn.controller.HelloController  
 **Method Signature:** public String sayHello()  
 **Method Implementation:** return hard coded string "Hello World!!"  
 **Sample Request**: <http://localhost:8083/hello> **Sample Response:** Hello World!!   
   
**IMPORTANT NOTE**: Don't forget to include start and end log in the sayHello() method.  
   
Try the URL <http://localhost:8083/hello> in both chrome browser and postman.  
   
SME to explain the following aspects:

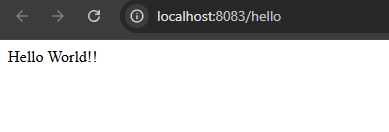
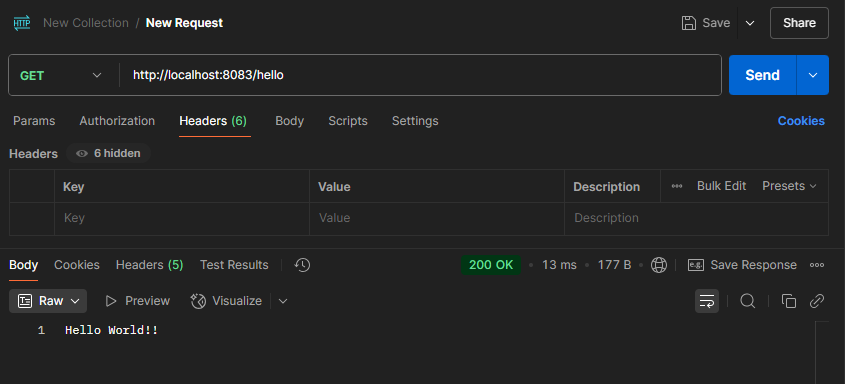
* In network tab of developer tools show the HTTP header details received
* In postman click on "Headers" tab to view the HTTP header details received

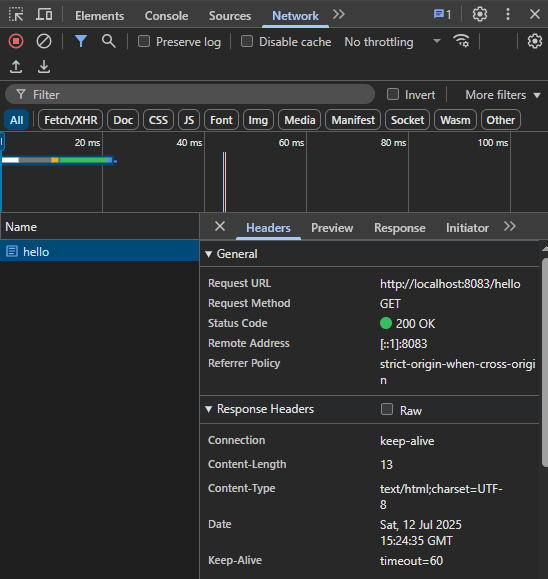
**Solution**

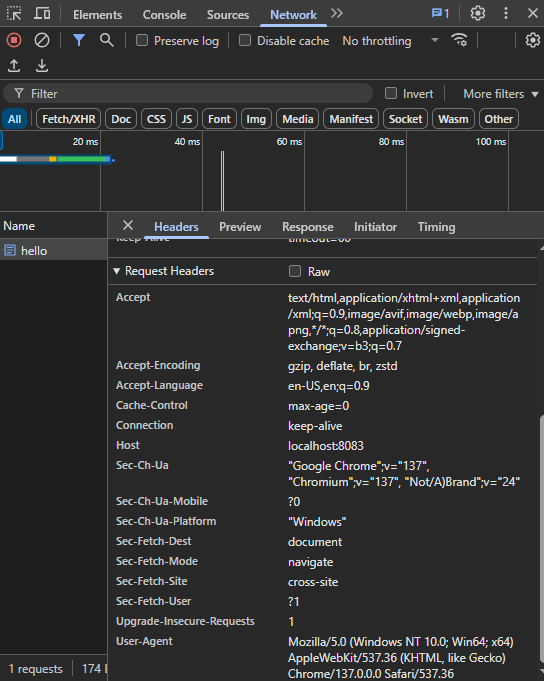
**HelloController.java**

**package com.cognizant.springapplearn.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 message = "Hello World!!";**  
 **LOGGER.info("END sayHello()");**  
 **return message;**  
 **}**  
**}**

**Output**







**Conclusion**

* **Status Code:** 200 (API is working correctly).
* **Response Body:** Hello World!!
* **Content-Type:** text/html;charset=UTF-8
* **Other headers:** Browser-specific values like user-agent, accept, host, etc.

**REST - Country Web Service**   
   
Write a REST service that returns India country details in the earlier created spring learn application.  
   
**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  
 **Sample Request**: <http://localhost:8083/country> **Sample Response**:

{

"code": "IN",

"name": "India"

}

SME to explain the following aspects:

* What happens in the controller method?
* How the bean is converted into JSON reponse?
* In network tab of developer tools show the HTTP header details received
* In postman click on "Headers" tab to view the HTTP header details received

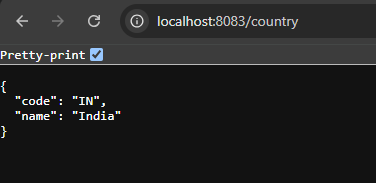
**Solution**

**Country.java**

package com.cognizant.springapplearn.model;  
  
public class Country {  
 private String code;  
 private String name;  
 public Country() {  
 }  
 public Country(String code, String name) {  
 this.code = code;  
 this.name = 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;  
 }  
}

**CountryController.java**

package com.cognizant.springapplearn.controller;  
import com.cognizant.springapplearn.model.Country;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.core.io.ClassPathResource;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
@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("country");  
 LOGGER.info("END - getCountryIndia()");  
 return country;  
 }  
}



* What happens in the controller method?

When we type <http://localhost:8083/country> in the browser or Postman, it hits the controller method getCountryIndia().

Inside this method:

* We tell Spring to load a bean named "in" from the XML file (which contains India’s data like code and name).
* This bean is a normal Java object (Country class).
* The method then returns this object.

Spring automatically understands that this is a REST method and sends back the object as a response.

* How the bean is converted into JSON reponse?

Spring Boot uses a tool called **Jackson** behind the scenes.

Here’s what happens:

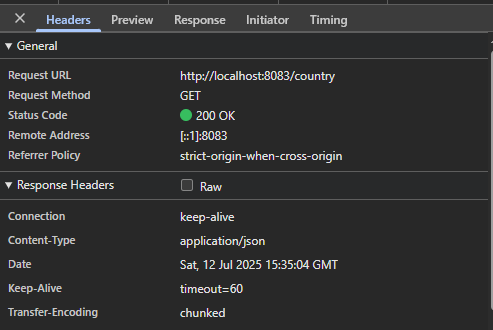
* We return a Java object (Country).
* Spring sees that this method is part of a REST API (@RestController or @ResponseBody).
* It uses Jackson to **automatically convert** the Java object into a JSON format.
* So instead of seeing a raw object, we get a response like:

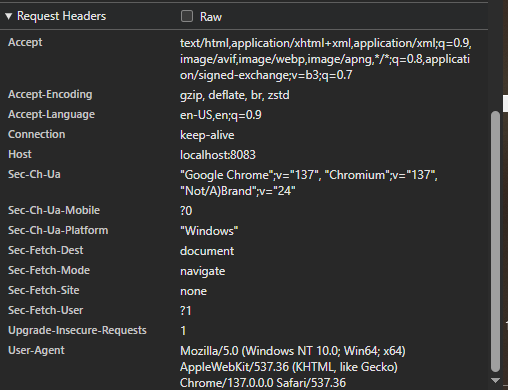
json

CopyEdit

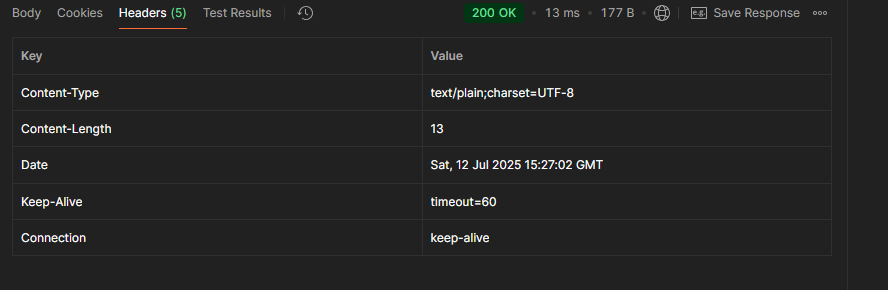
{  
 "code": "IN",  
 "name": "India"  
}

* In network tab of developer tools show the HTTP header details received





* In postman click on "Headers" tab to view the HTTP header details received



**REST - Get all countries**   
   
Write a REST service that returns all the countries.  
   
**Controller**: com.cognizant.spring-learn.controller.CountryController  
 **Method Annotation**: @GetMapping("/countries")  
 **Method Name**: getAllCountries()  
 **Method Implementation**: Load country list from country.xml and return  
   
**Sample Request**: <http://localhost:8083/countries> **Sample Response**:

[

{ "code": "IN", "name": "India"},

{ "code": "US", "name": "United States"},

{ "code": "JP", "name": "Japan"},

{ "code": "DE", "name": "Germany"}

]

**Solution**

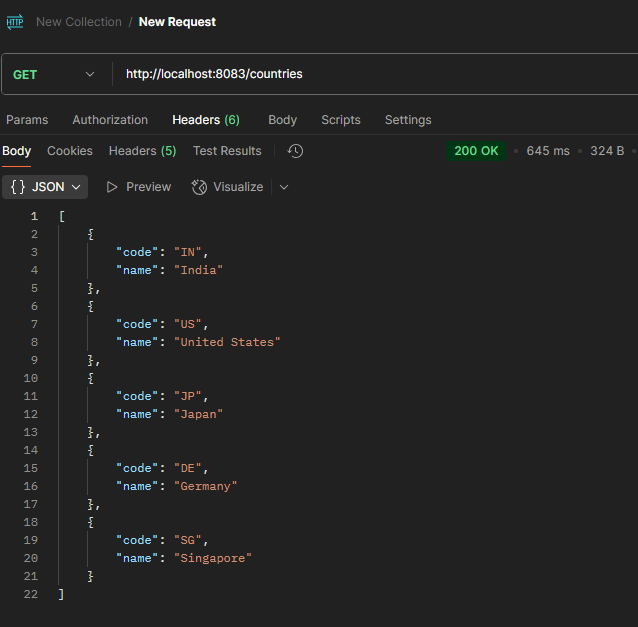
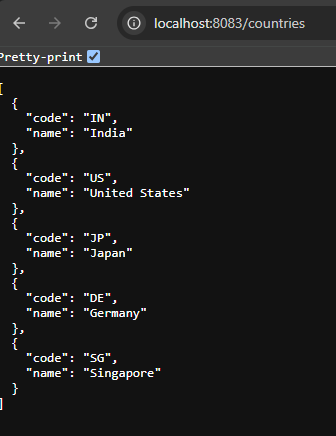
**country.xml**

<?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.springapplearn.model.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
 <bean id="in" class="com.cognizant.springapplearn.model.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
  
 <bean id="us" class="com.cognizant.springapplearn.model.Country">  
 <property name="code" value="US"/>  
 <property name="name" value="United States"/>  
 </bean>  
  
 <bean id="jp" class="com.cognizant.springapplearn.model.Country">  
 <property name="code" value="JP"/>  
 <property name="name" value="Japan"/>  
 </bean>  
  
 <bean id="de" class="com.cognizant.springapplearn.model.Country">  
 <property name="code" value="DE"/>  
 <property name="name" value="Germany"/>  
 </bean>  
 <bean id="sg" class="com.cognizant.springapplearn.model.Country">  
 <property name="code" value="SG"/>  
 <property name="name" value="Singapore"/>  
 </bean>  
 <bean id="countryList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <ref bean="in"/>  
 <ref bean="us"/>  
 <ref bean="jp"/>  
 <ref bean="de"/>  
 <ref bean="sg"/>  
 </list>  
 </constructor-arg>  
 </bean>  
  
</beans>

**CountryController.java**

package com.cognizant.springapplearn.controller;  
  
import com.cognizant.springapplearn.model.Country;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.context.ApplicationContext;  
import org.springframework.core.io.ClassPathResource;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
import java.util.List;  
  
@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("country");  
 LOGGER.info("END - getCountryIndia()");  
 return country;  
 }  
 **@GetMapping("/countries")**  
 **public List<Country> getAllCountries() {**  
 **LOGGER.info("START - getAllCountries()");**  
  
 **ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");**  
 **List<Country> countryList = context.getBean("countryList", List.class);**  
  
 **LOGGER.info("END - getAllCountries()");**  
 **return countryList;**  
 **}**  
  
}

**Ouput**



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

* Get the country code using @PathVariable
* Get country list from country.xml
* Iterate through the country list
* Make a case insensitive matching of country code and return the country.
* Lambda expression can also be used instead of iterating the country list

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

{

"code": "IN",

"name": "India"

}

**Solution**

**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.springapplearn.model.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
 <bean id="in" class="com.cognizant.springapplearn.model.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
  
 <bean id="us" class="com.cognizant.springapplearn.model.Country">  
 <property name="code" value="US"/>  
 <property name="name" value="United States"/>  
 </bean>  
  
 <bean id="jp" class="com.cognizant.springapplearn.model.Country">  
 <property name="code" value="JP"/>  
 <property name="name" value="Japan"/>  
 </bean>  
  
 <bean id="de" class="com.cognizant.springapplearn.model.Country">  
 <property name="code" value="DE"/>  
 <property name="name" value="Germany"/>  
 </bean>  
 <bean id="sg" class="com.cognizant.springapplearn.model.Country">  
 <property name="code" value="SG"/>  
 <property name="name" value="Singapore"/>  
 </bean>  
 <bean id="countryList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <ref bean="in"/>  
 <ref bean="us"/>  
 <ref bean="jp"/>  
 <ref bean="de"/>  
 <ref bean="sg"/>  
 </list>  
 </constructor-arg>  
 </bean>  
  
</beans>

**CountryController.java**

package com.cognizant.springapplearn.controller;  
  
import com.cognizant.springapplearn.model.Country;  
import com.cognizant.springapplearn.service.CountryService;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.ApplicationContext;  
import org.springframework.core.io.ClassPathResource;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import java.util.List;  
  
@RestController  
public class CountryController {  
 private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);  
 @Autowired  
 private CountryService countryService;  
 @RequestMapping("/country")  
 public Country getCountryIndia() {  
 LOGGER.info("START - getCountryIndia()");  
 ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = (Country) context.getBean("country");  
 LOGGER.info("END - getCountryIndia()");  
 return country;  
 }  
 @GetMapping("/countries")  
 public List<Country> getAllCountries() {  
 LOGGER.info("START - getAllCountries()");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countryList = context.getBean("countryList", List.class);  
  
 LOGGER.info("END - getAllCountries()");  
 return countryList;  
 }  
  **@GetMapping("/countries/{code}")**  
 **public Country getCountry(@PathVariable String code) {**  
 **return countryService.getCountry(code);**  
 **}**  
  
}

**CountryService.java**

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

**Output**

