**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

**CODE:**

Application.properties

spring.application.name=spring-learn

server.port=8083

logging.level.root=INFO

logging.level.com.cognizant.spring\_learn=DEBUG

**src/main/java/com/cognizant/spring\_learn/controller/HelloController.java**

package com.cognizant.spring\_learn.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");

LOGGER.info("END");

return "Hello World!!";

}

}

**SpringLearnApplication.java**

**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) {

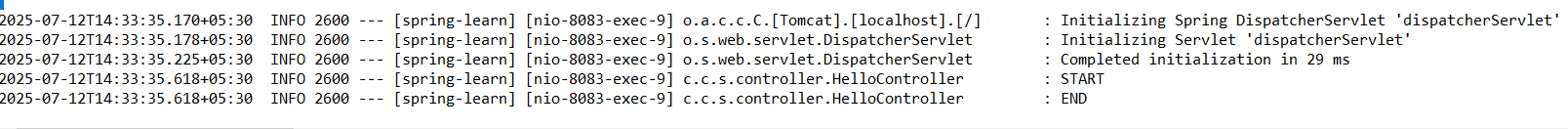
***LOGGER***.info("START");

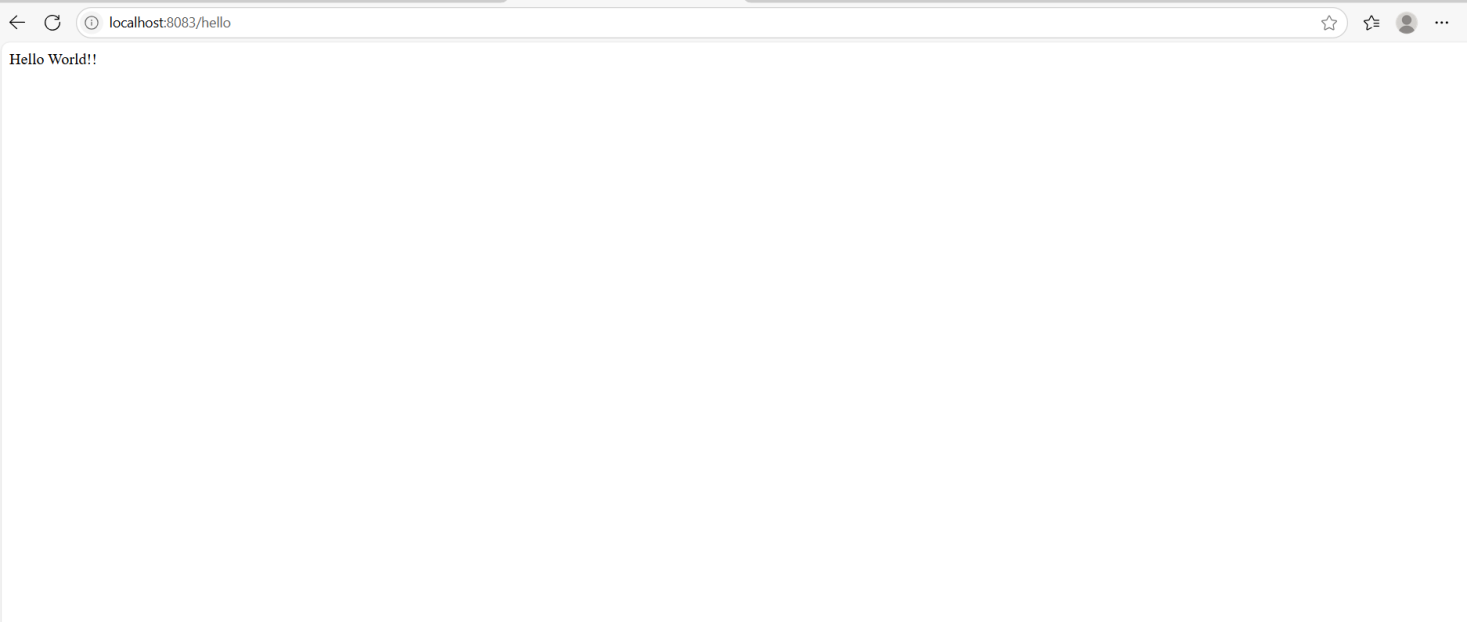
SpringApplication.*run*(SpringLearnApplication.**class**, args); ***LOGGER***.info("END");

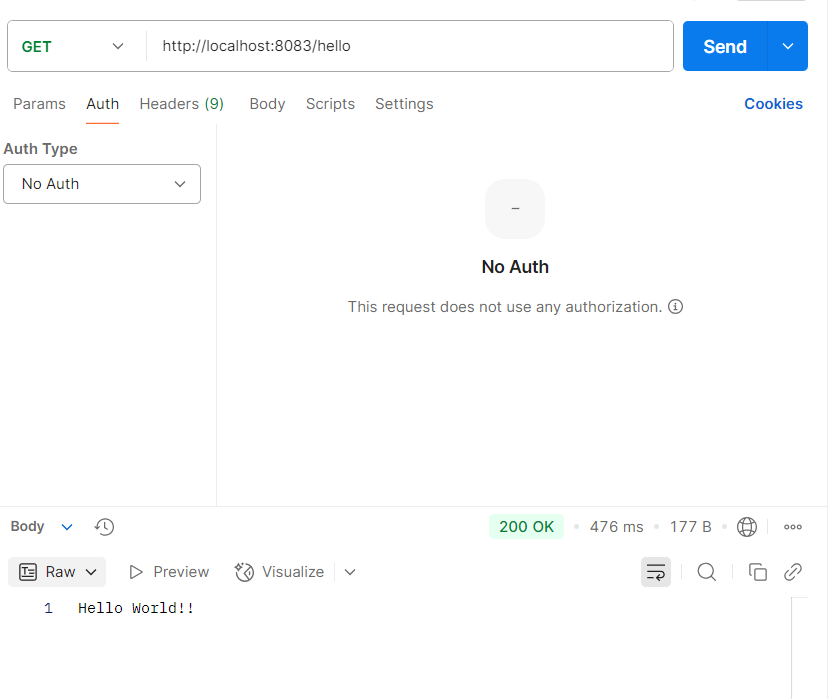
}

}

**OUTPUT:**

****

****

****

**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

**CODES:**

**src/main/java/com/cognizant/spring\_learn/controller/CountryController.java**

**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 india = (Country) context.getBean("in");

**return** india;

}

}

**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.Country"*>

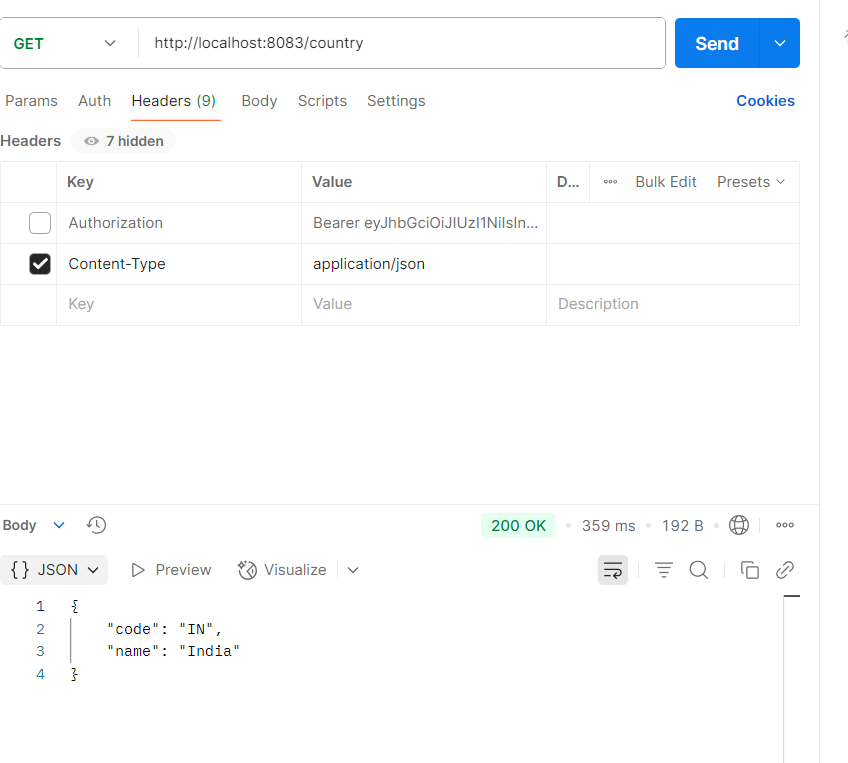
<property name=*"code"* value=*"IN"*/>

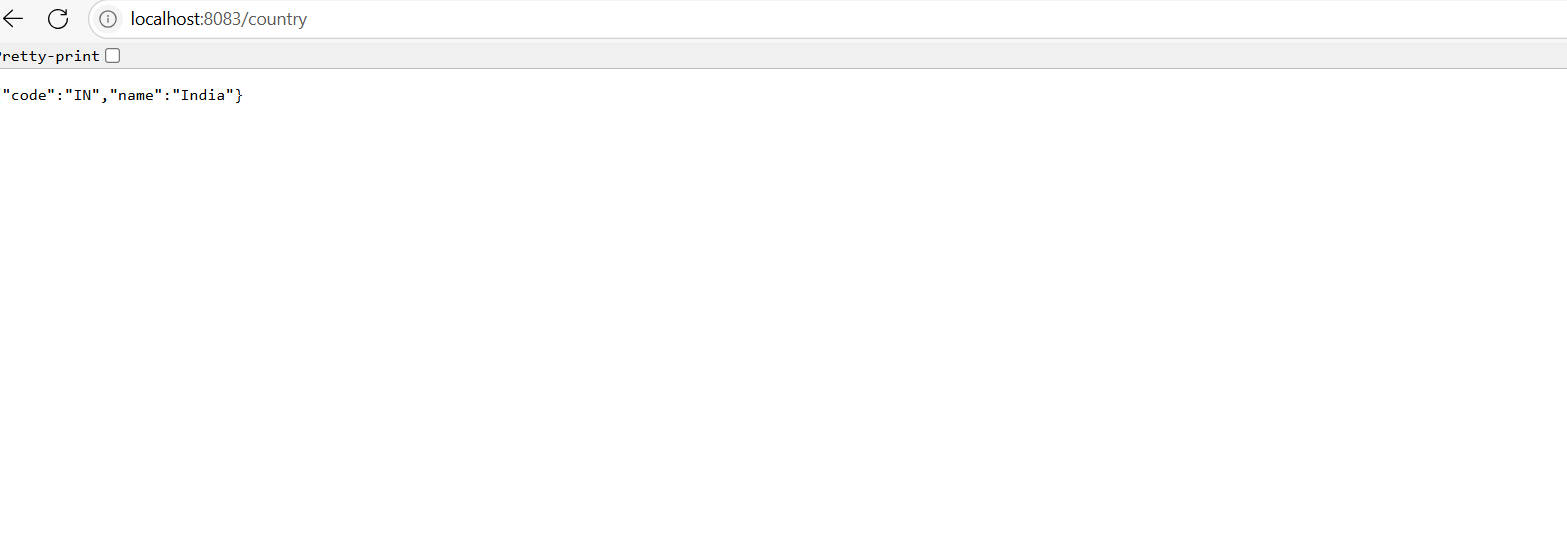
<property name=*"name"* value=*"India"*/>

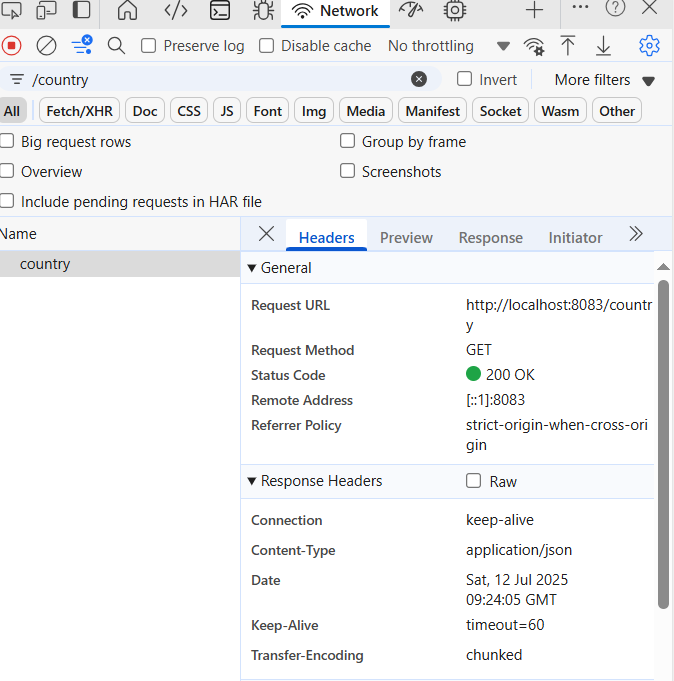
</bean>

</beans>

**OUTPUTS:**

****

****

****

**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"

}

**CODES:**

**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>

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

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

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

</bean>

</beans>

**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;

**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; }

}

**com.cognizant.spring\_learn.config.ApplicationConfig:**

package com.cognizant.spring\_learn.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.support.ClassPathXmlApplicationContext;

@Configuration

public class ApplicationConfig {

@Bean

public ClassPathXmlApplicationContext applicationContext() {

return new ClassPathXmlApplicationContext("country.xml");

}

}

**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.context.ApplicationContext;

**import** org.springframework.stereotype.Service;

**import** java.util.ArrayList;

**import** java.util.List;

@Service

**public** **class** CountryService {

@Autowired

**private** ApplicationContext context;

**public** Country getCountry(String code) {

List<Country> countries = **new** ArrayList<>();

countries.add((Country) context.getBean("in"));

countries.add((Country) context.getBean("us"));

**return** countries.stream()

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

.findFirst()

.orElse(**null**);

}

}

**src/main/java/com/cognizant/spring\_learn/controller/CountryController.java**

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

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

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

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

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

@RestController

**public** **class** CountryController {

@Autowired

**private** CountryService countryService;

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

**public** Country getCountry(@PathVariable String code) {

**return** countryService.getCountry(code);

}

}

**Application.properties**

spring.application.name=spring-learn

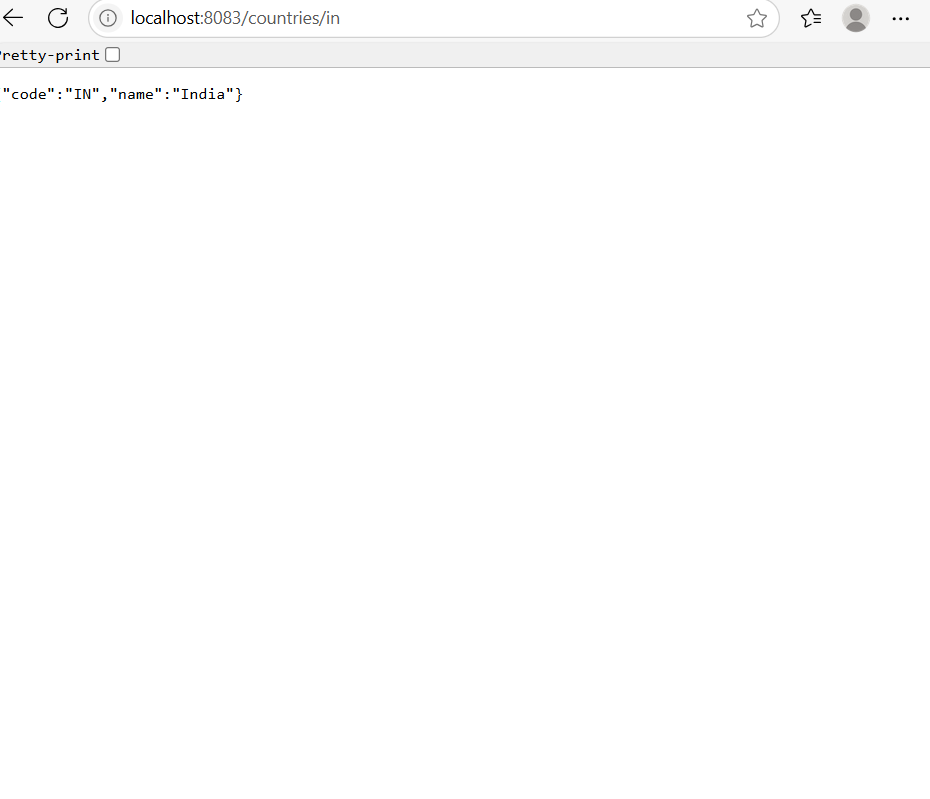
server.port=8083

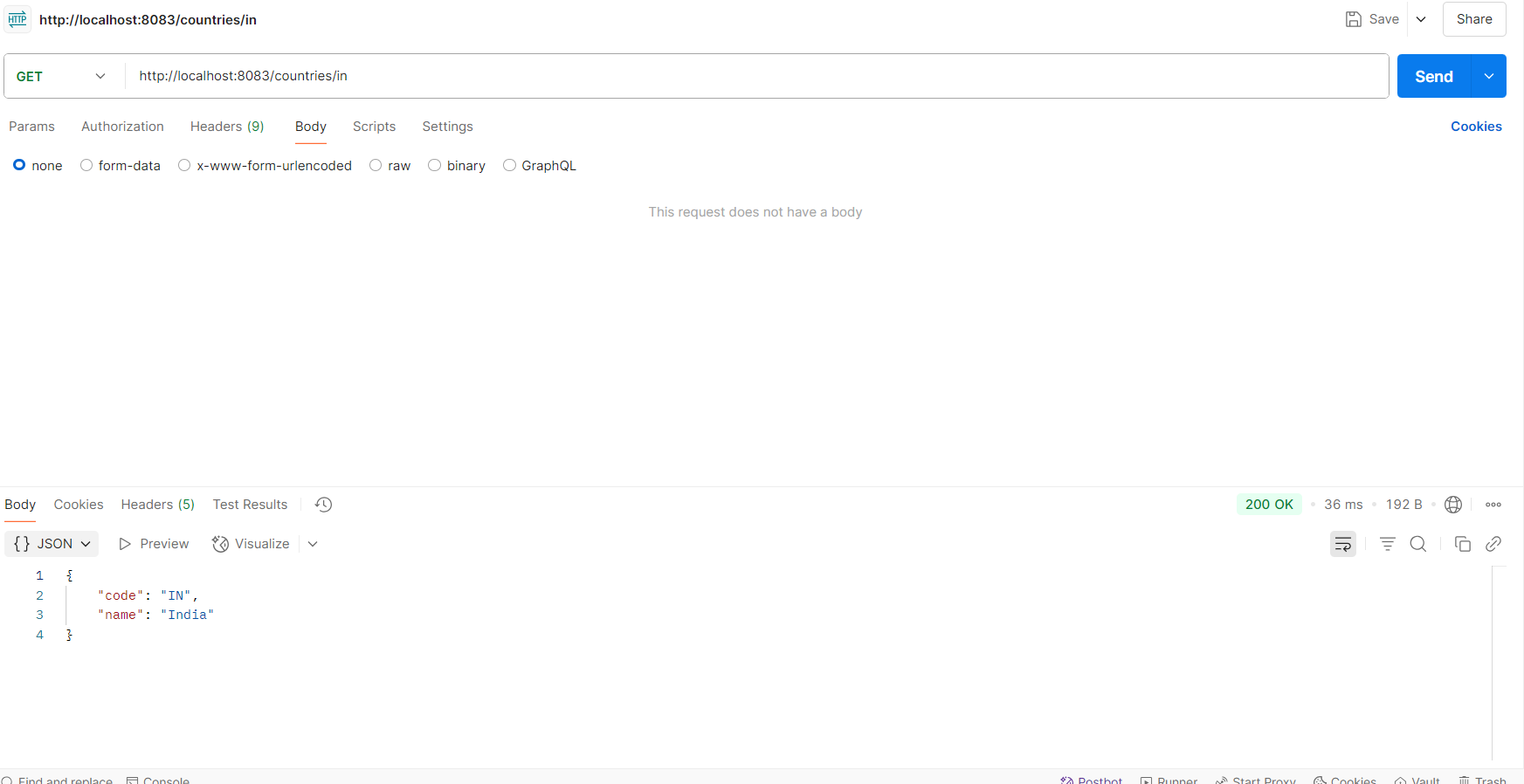
logging.level.root=INFO

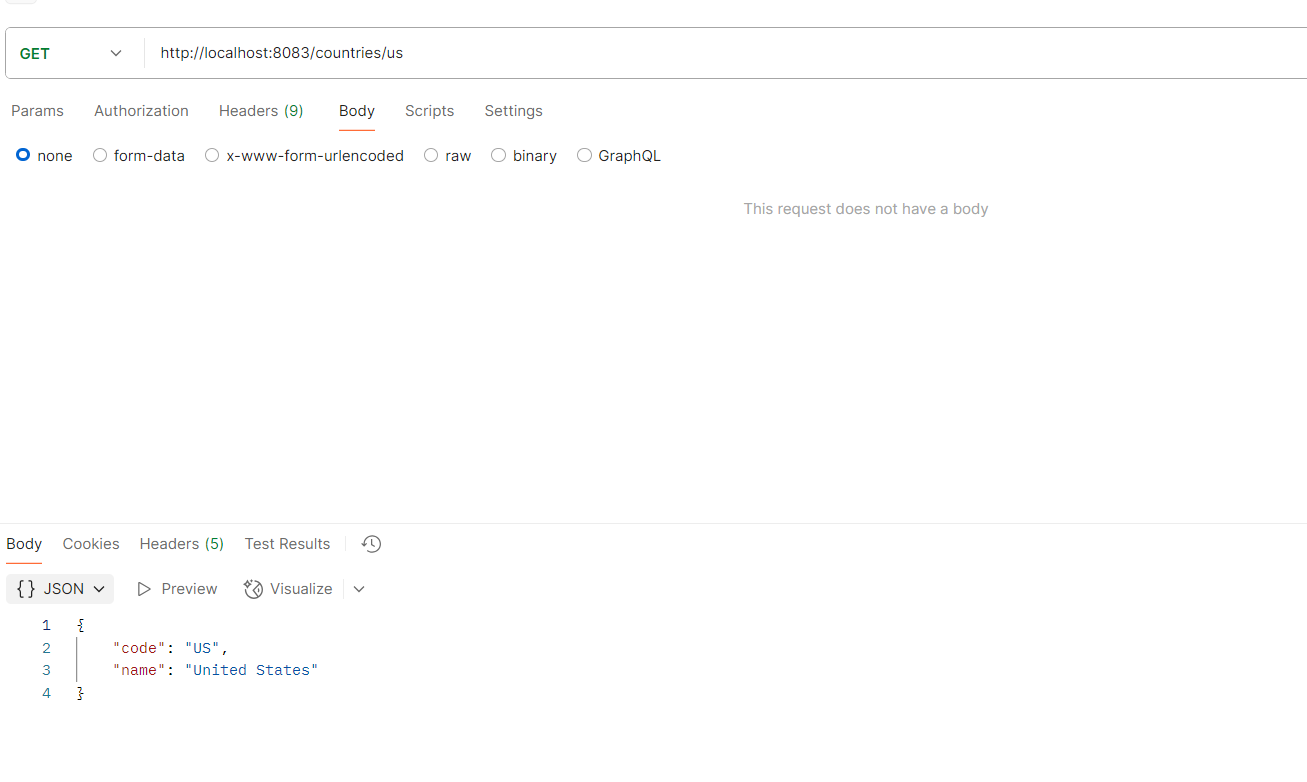
logging.level.com.cognizant.spring\_learn=DEBUG

spring.main.allow-bean-definition-overriding=true

**OUTPUTS:**

****

****

****