## Spring REST Assignment

## As per document : 2. spring-rest-handson

## Student Name: Divyansh Tiwari

## SuperSet ID: 6394658

## Week 4 – Spring REST

## Hands On: Hello World RESTful Web Service

### Objective

Create a simple REST controller that returns a hardcoded "Hello World!!" response.

### Steps Followed

- Created HelloController.java in com.cognizant.spring\_learn.controller

- Annotated class with @RestController

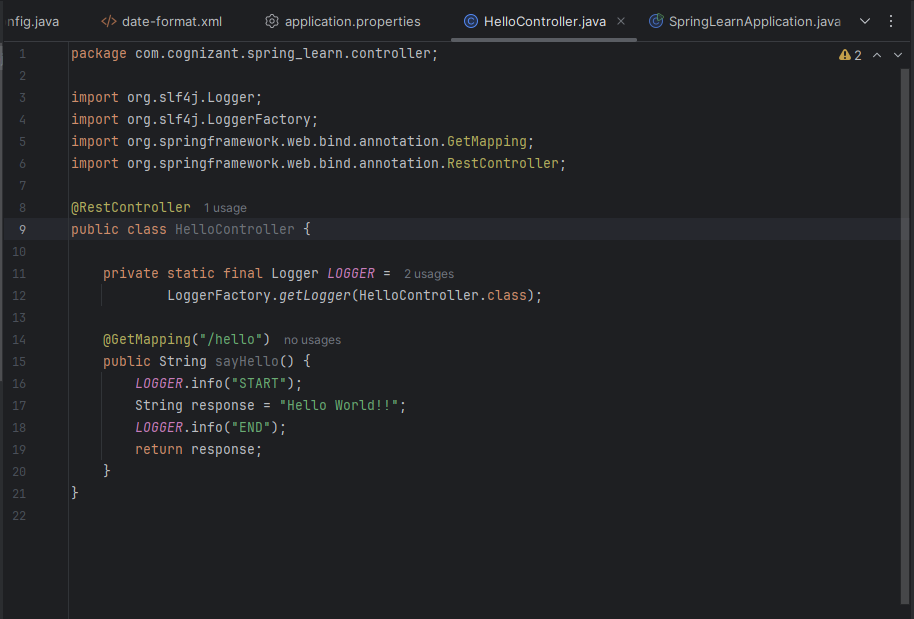
- Added method sayHello() annotated with @GetMapping("/hello")

- Returned string "Hello World!!"

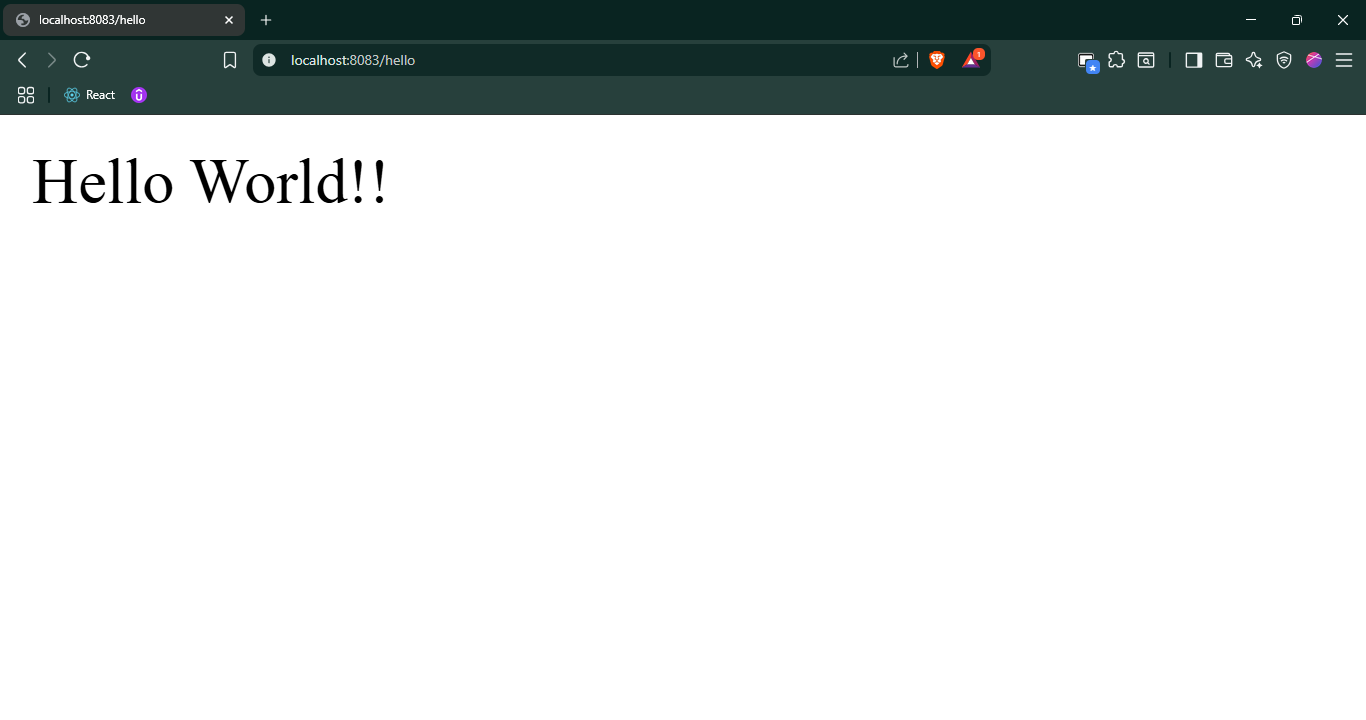
- Started the application and accessed endpoint using browser and Postman

### Screenshots

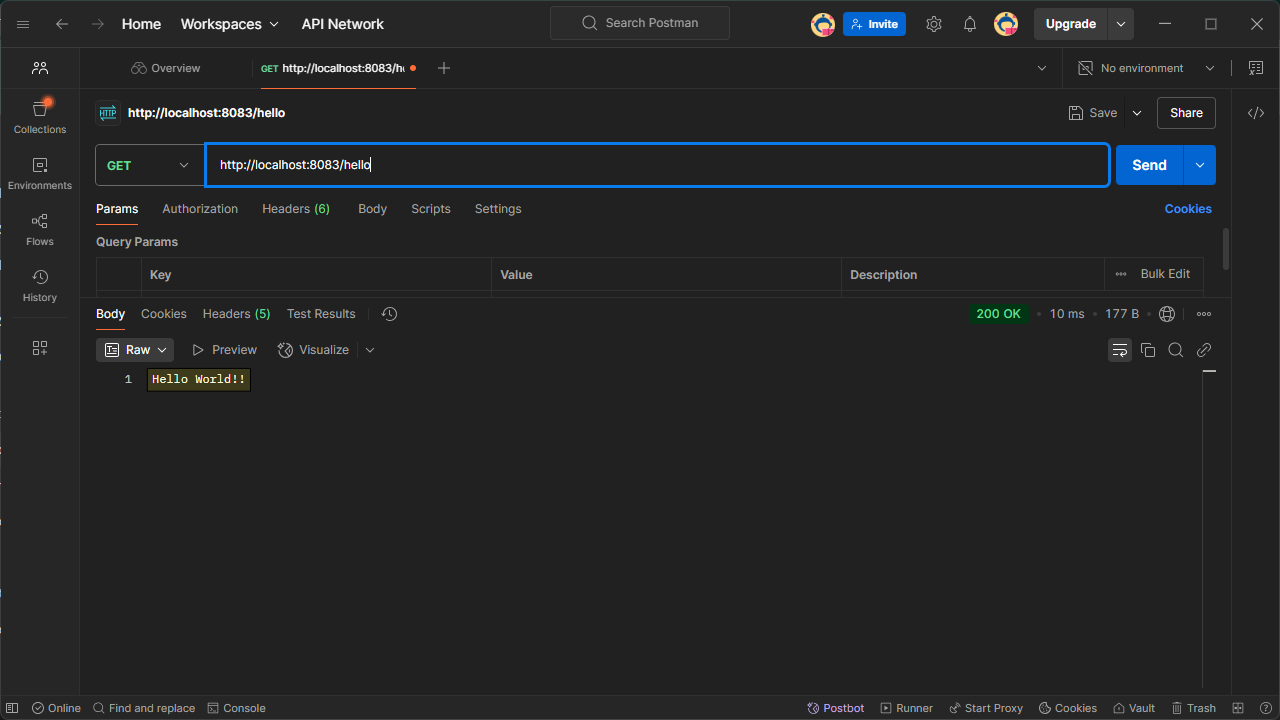
[Screenshot: HelloController.java in IntelliJ]



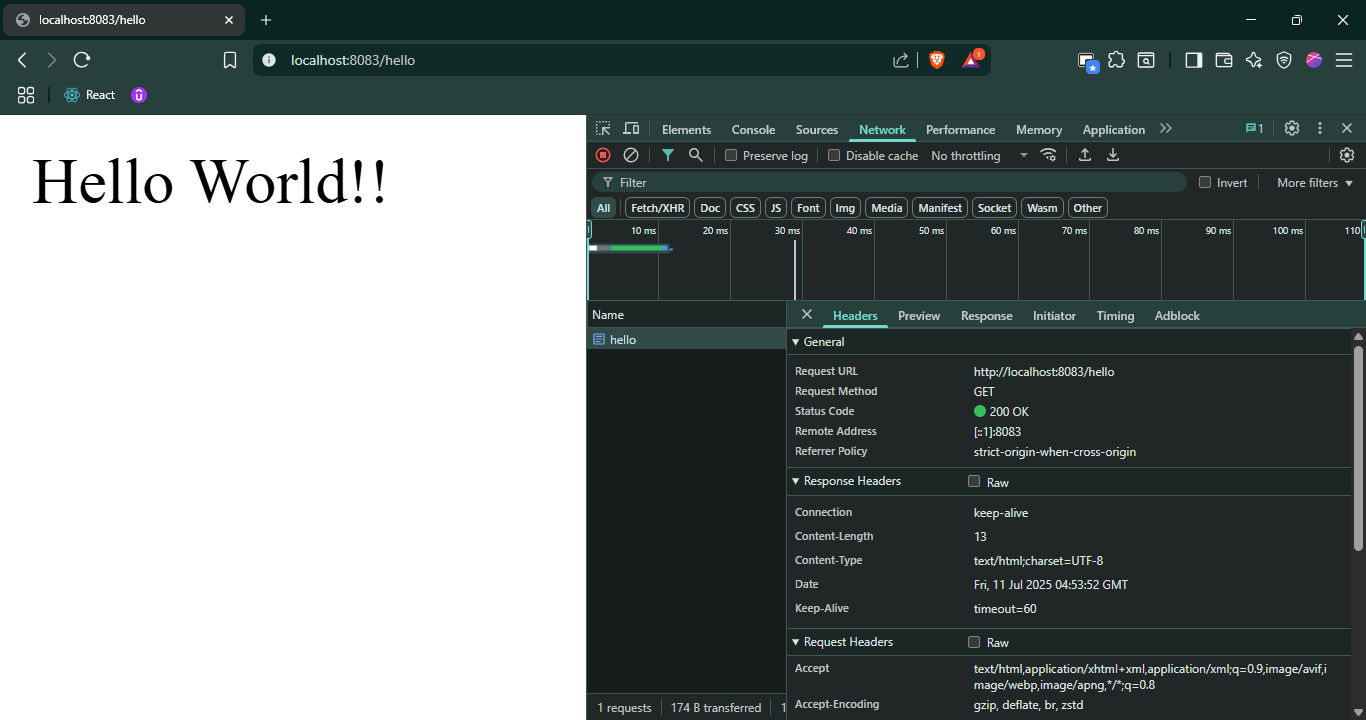
[Screenshot: Browser showing 'Hello World!!' response]

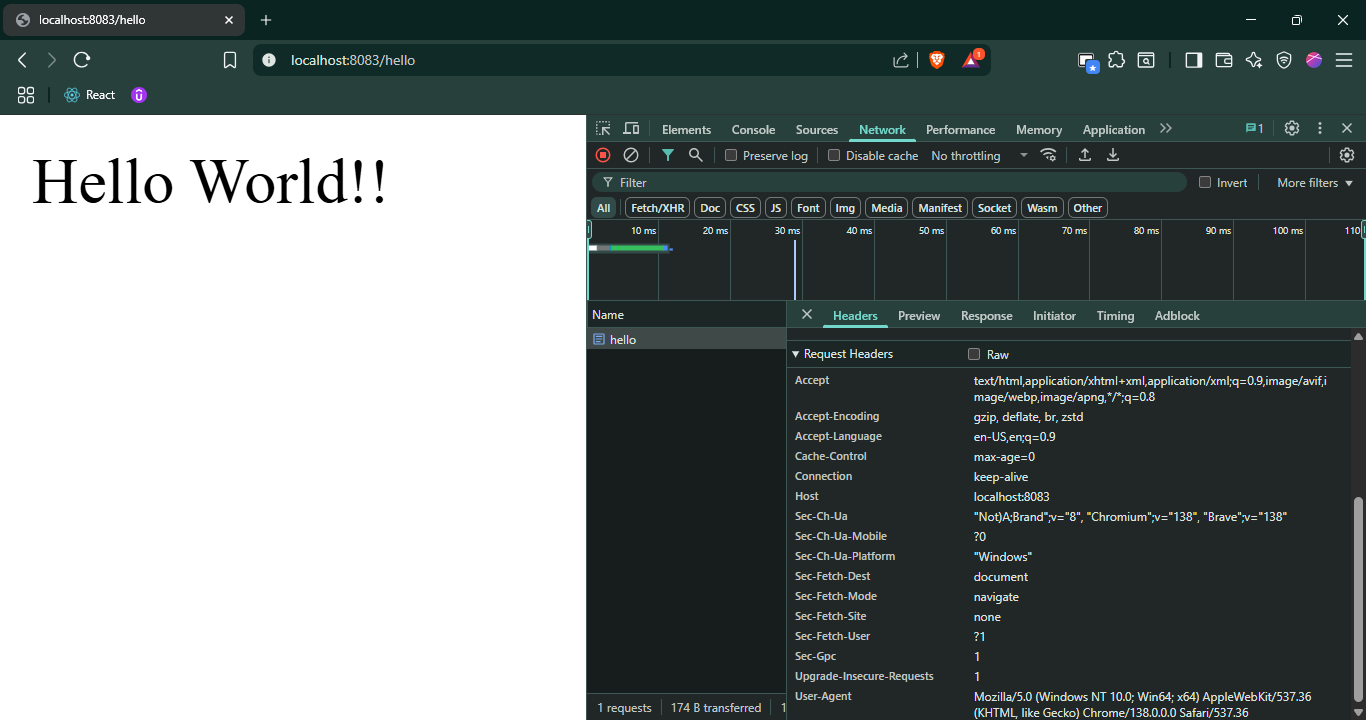


[Screenshot: Postman GET request to /hello]



[Screenshot: DevTools > Network tab showing request and response headers]





## REST – Country Web Service

### Objective

Expose a /country endpoint that returns a bean defined in XML (country.xml).

### Steps Followed

- Created CountryController.java with @RequestMapping("/country")

- Loaded country bean from country.xml using ClassPathXmlApplicationContext

- Returned a Country object with code = "IN", name = "India"

- Accessed endpoint using browser and Postman

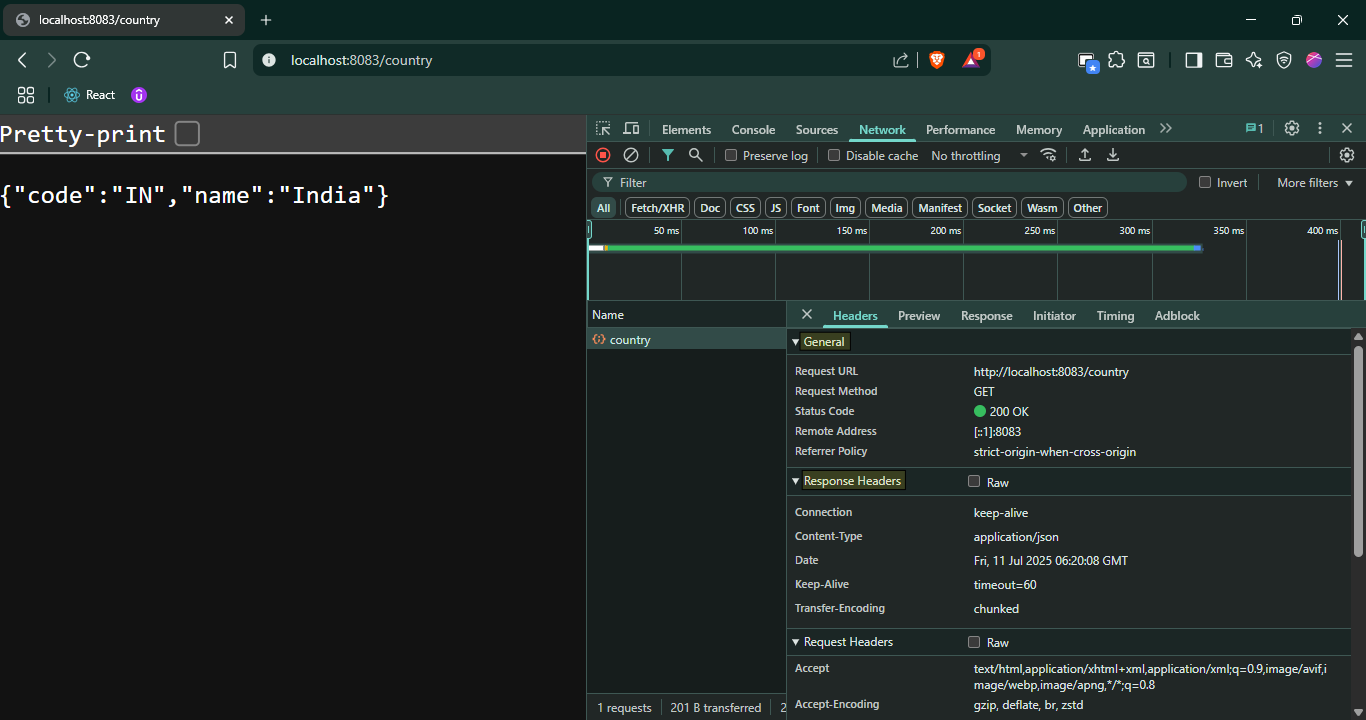
### Screenshots

[Screenshot: country.xml with <bean id="country" ... />]

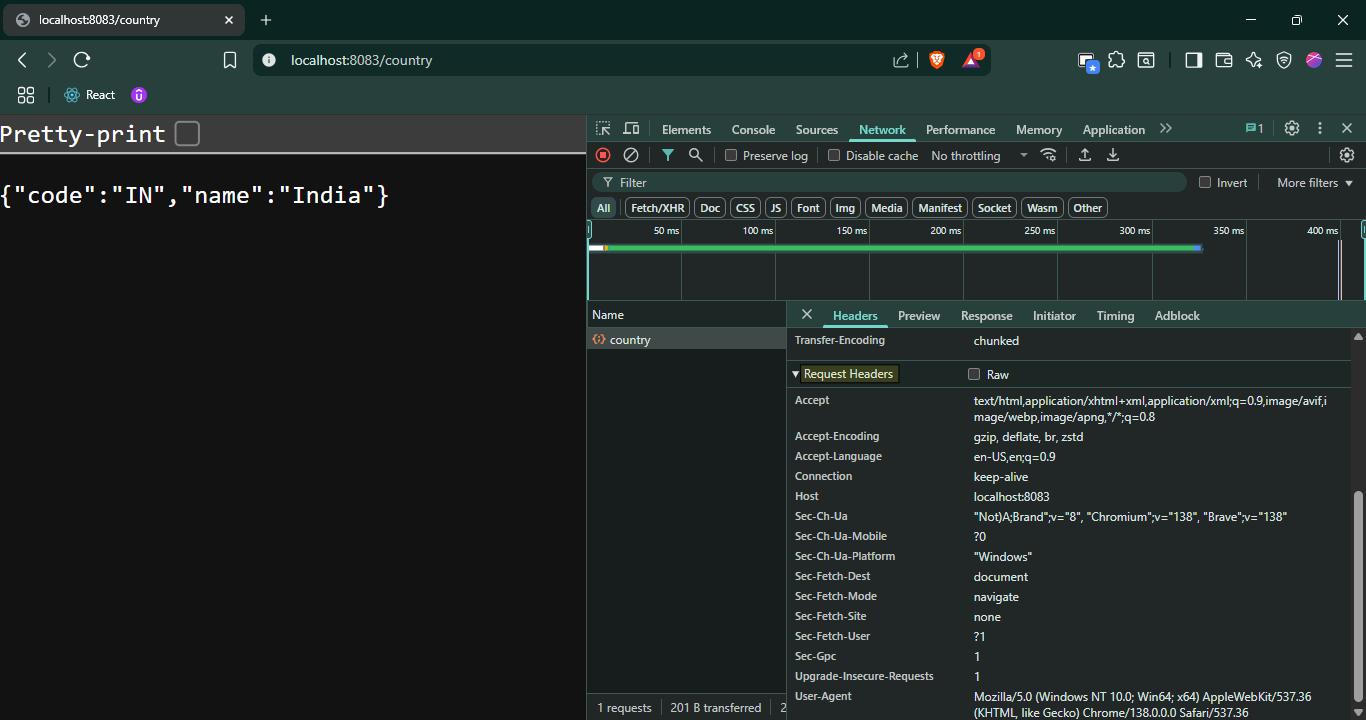
[Screenshot: Controller method for /country endpoint]

[Screenshot: Browser output for /country]

General and Response Header

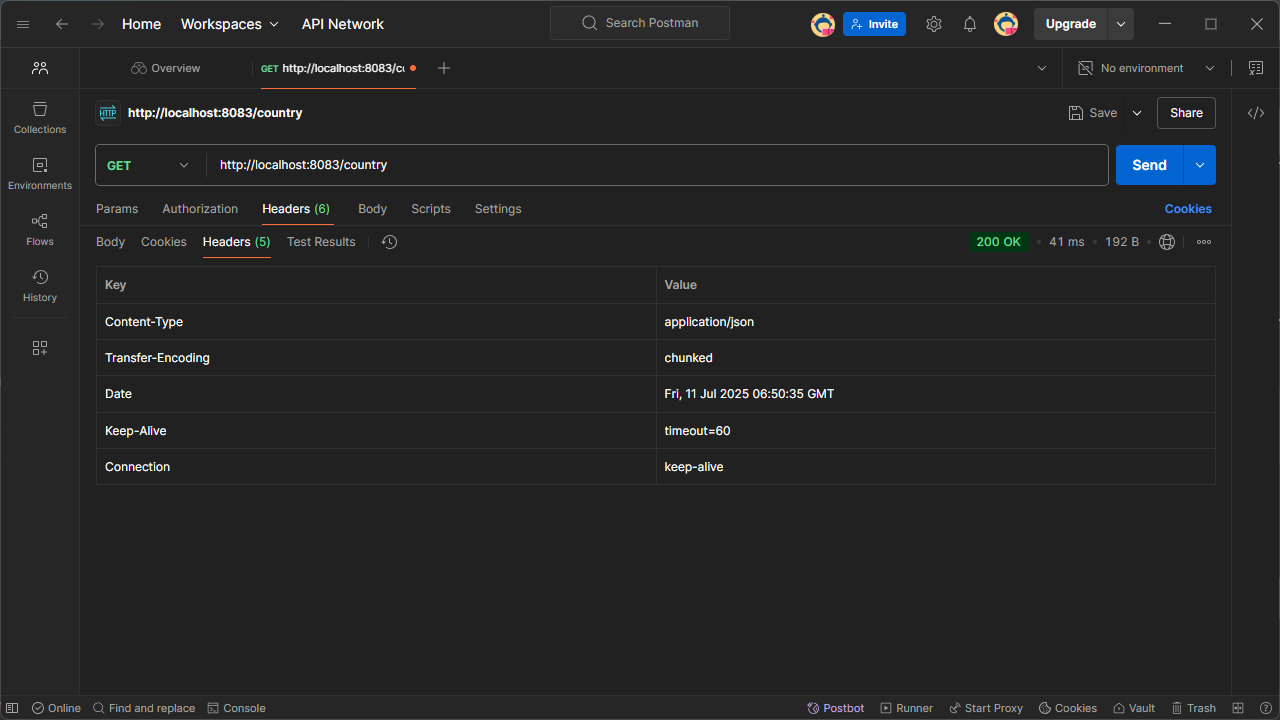


Response Headers

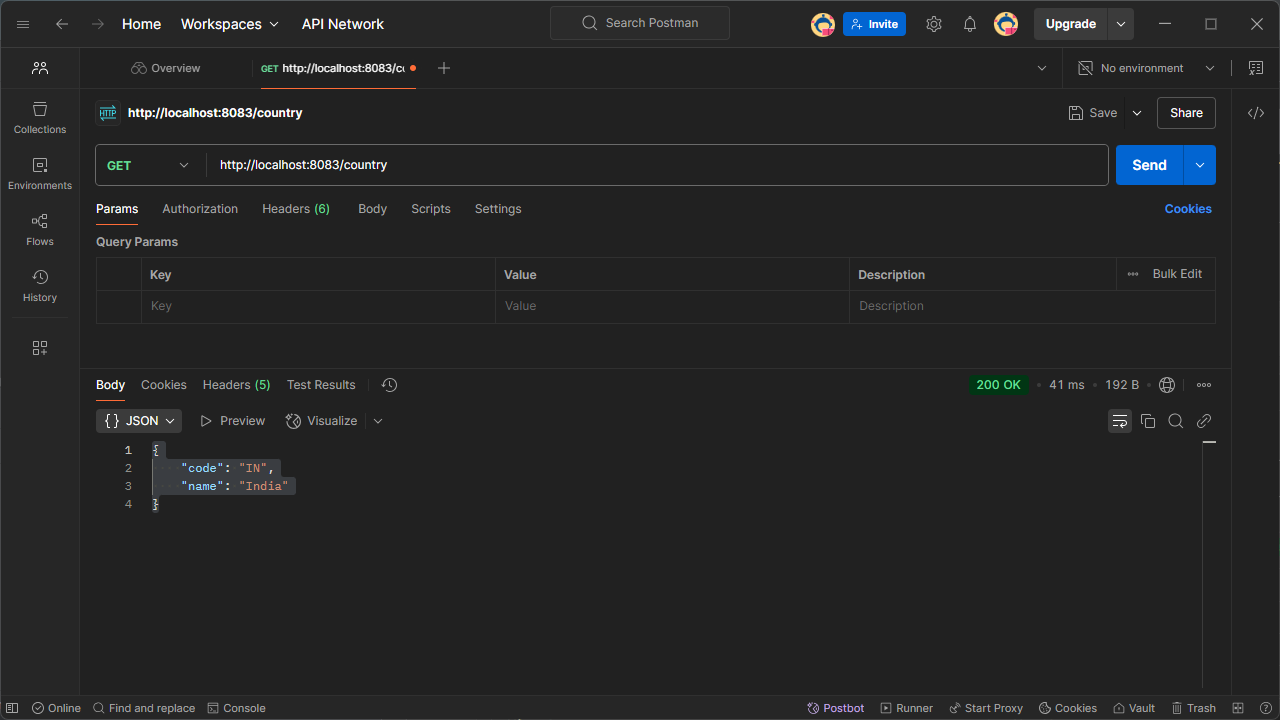


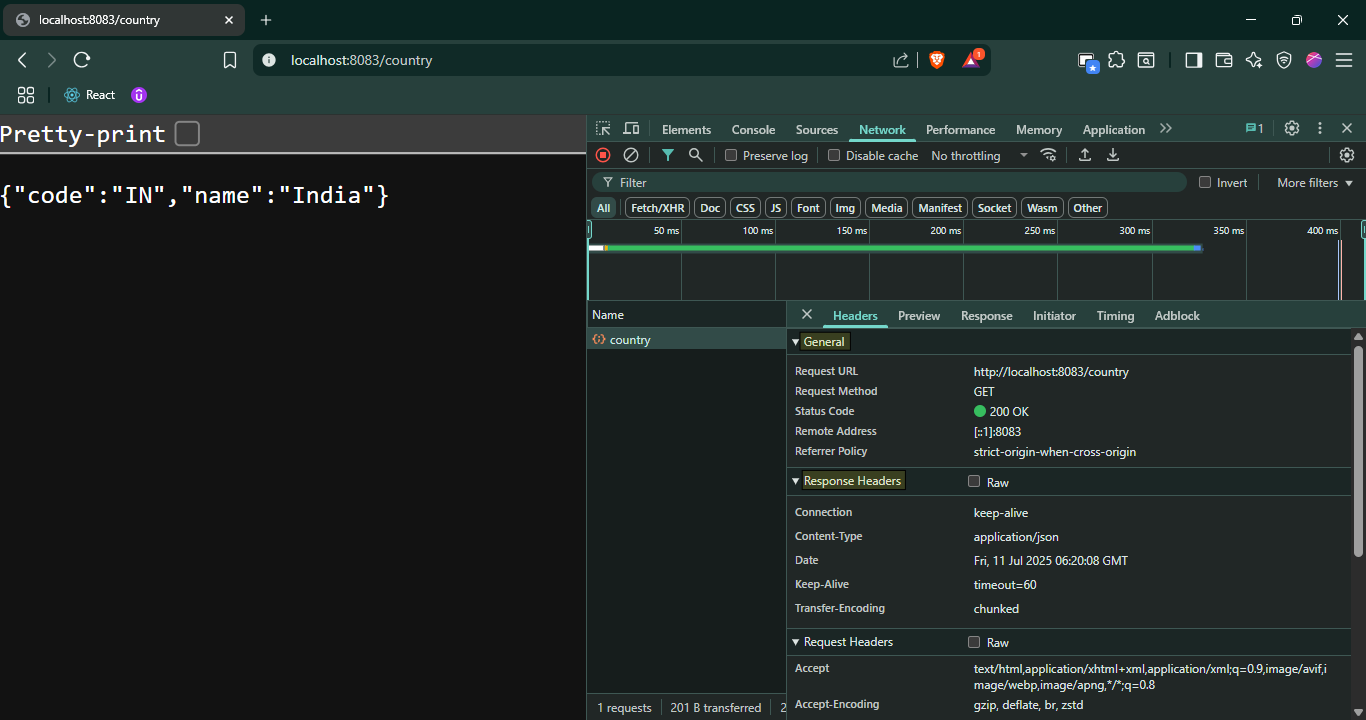
[Screenshot: Postman response for /country]

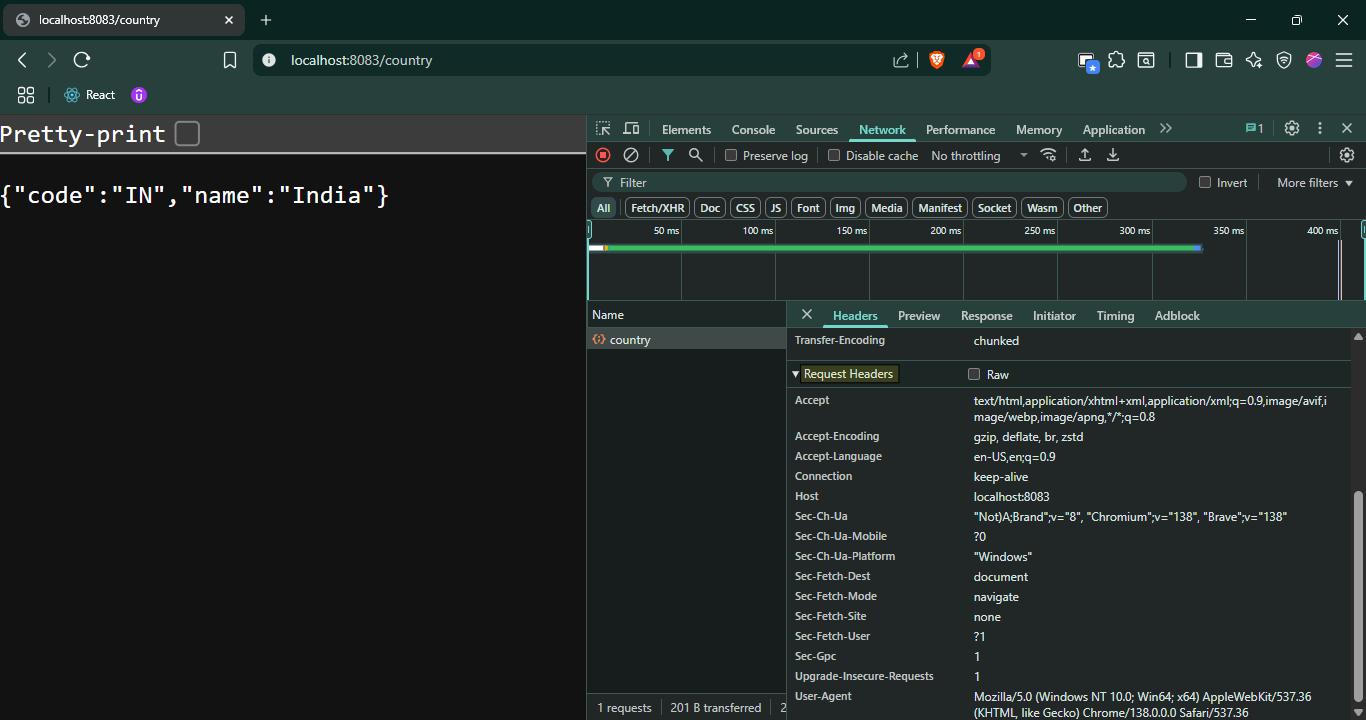
Postman Headers



Postman JSON Output



[Screenshot: DevTools > Network tab and Postman Headers tab]



## REST – Get all countries

### Objective

Return a list of countries defined in XML via a REST endpoint.

### Steps Followed

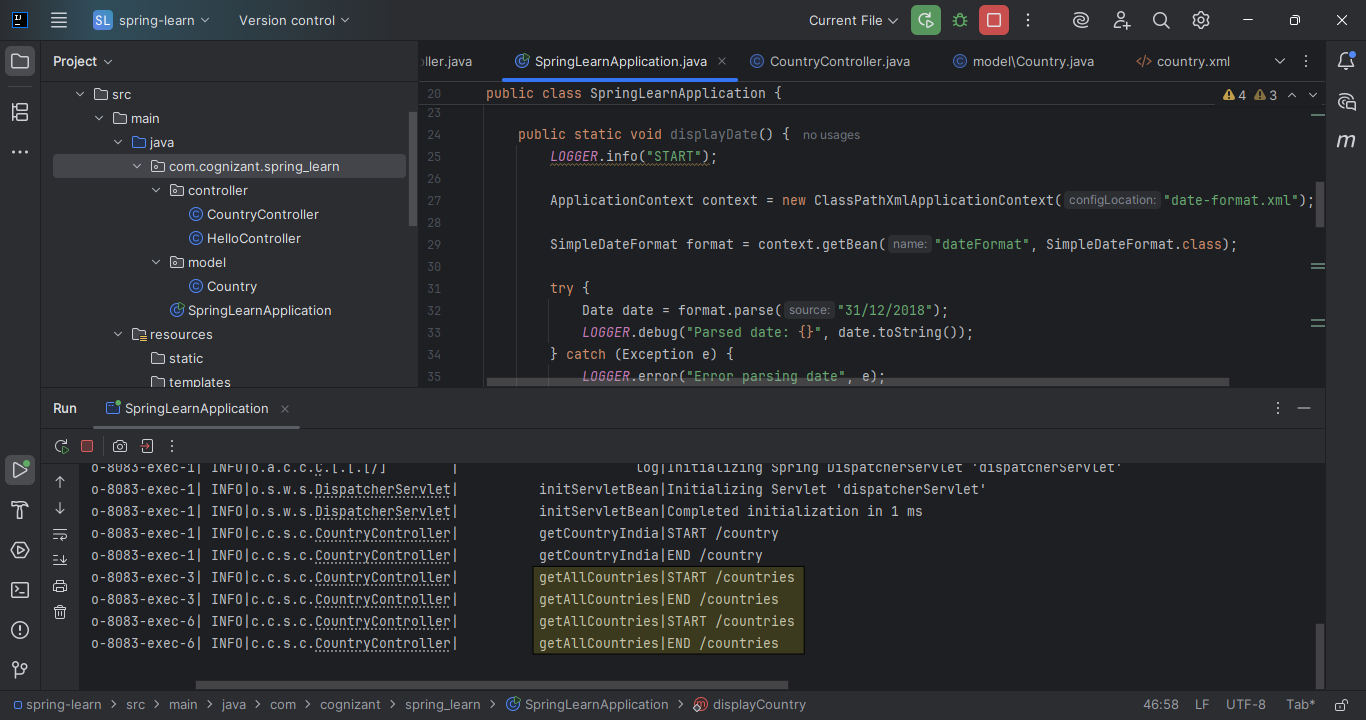
- Updated country.xml with multiple country beans

- Defined countryList bean containing the country references

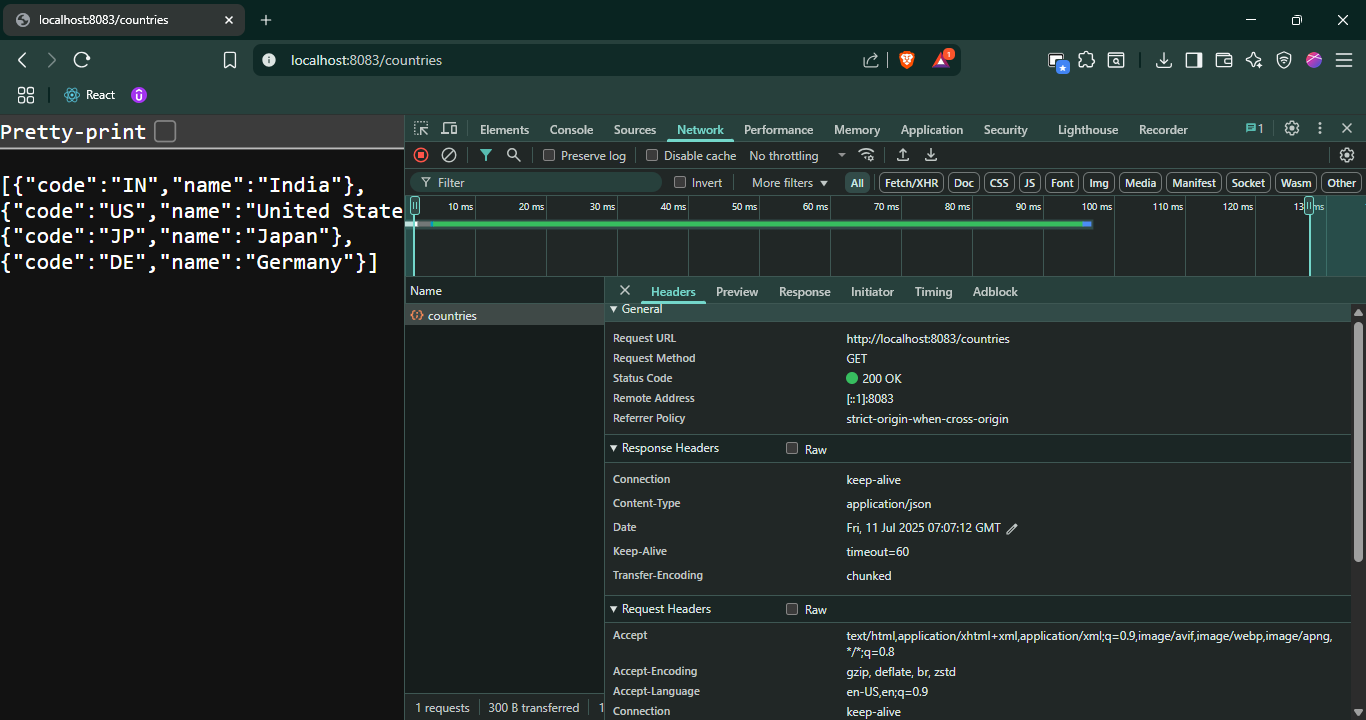
- Created /countries endpoint to return List<Country>

- Verified output via browser and Postman

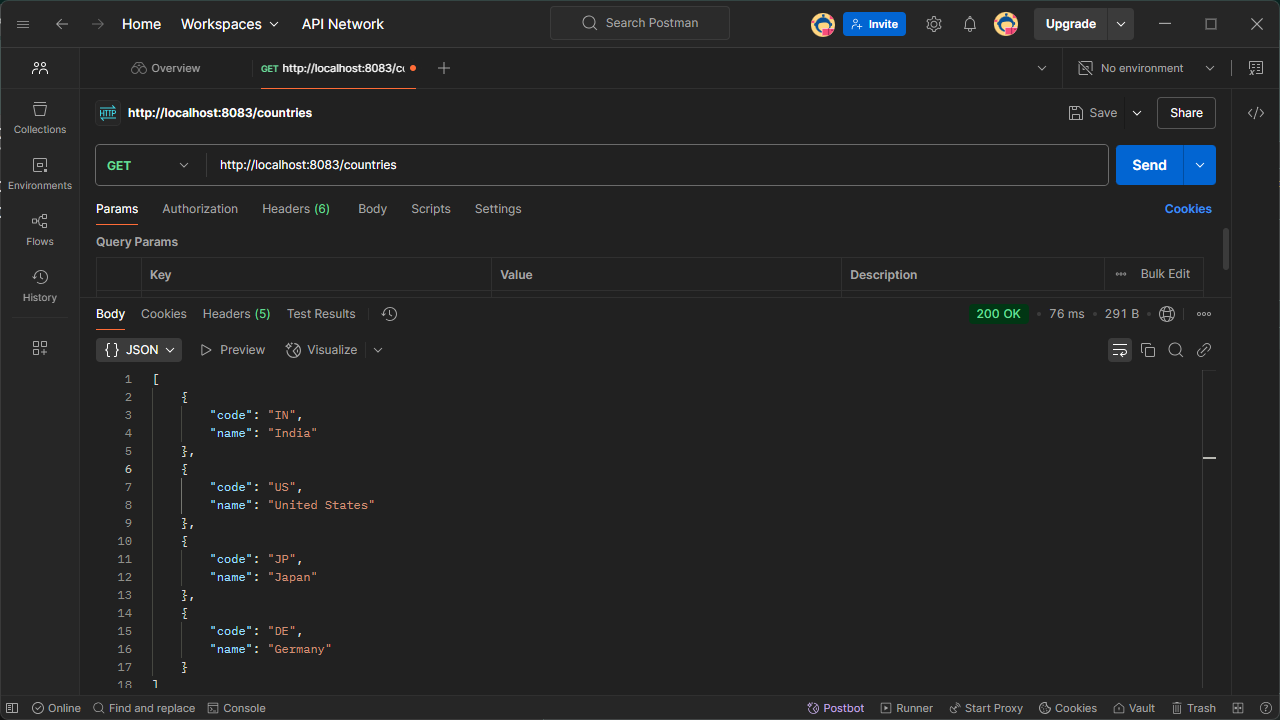
### Screenshots

[Screenshot: Application Started Console Log]

[Screenshot: country.xml with list of countries]



[Screenshot: Postman JSON response for /countries]



## REST – Get country based on country code

### Objective

Use @PathVariable to fetch a specific country from the list by code.

### Steps Followed

- Created /countries/{code} endpoint in CountryController

- Implemented CountryService with getCountry(code) method

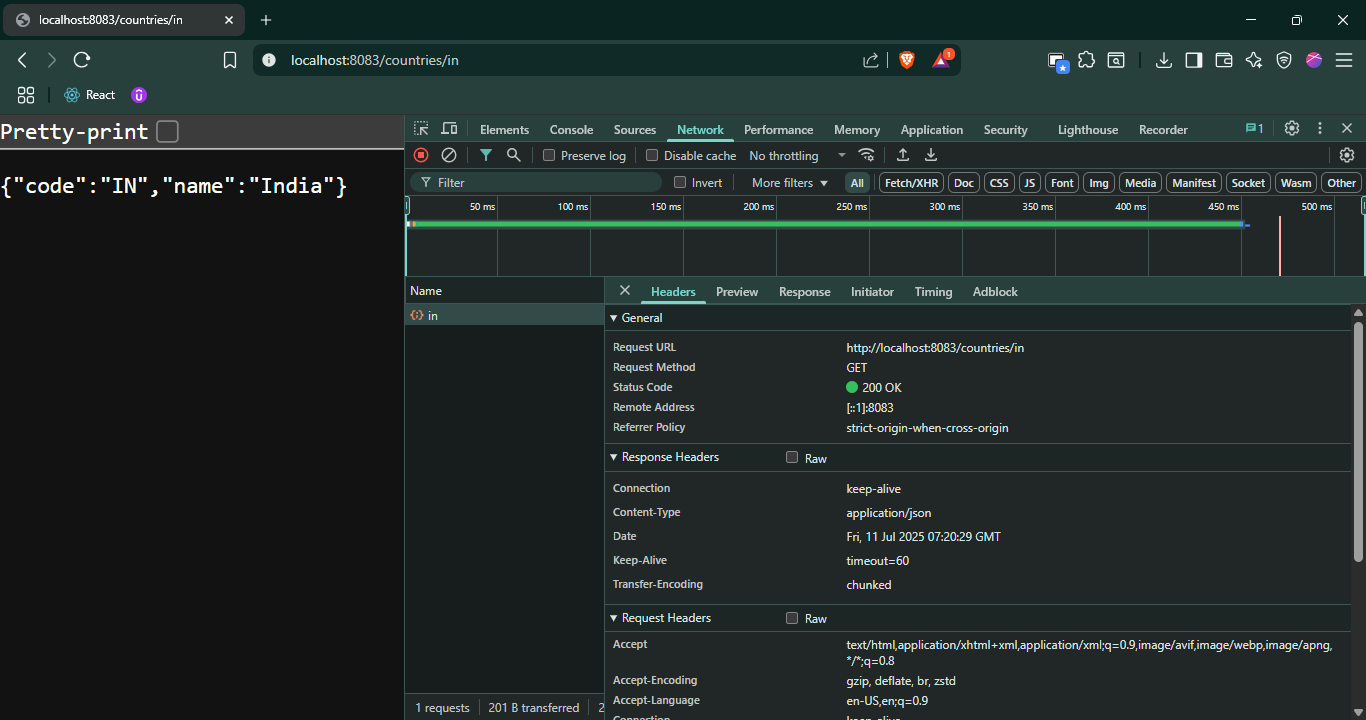
- Performed case-insensitive match on code

- Returned matching Country or threw CountryNotFoundException

- Tested with multiple codes via Postman

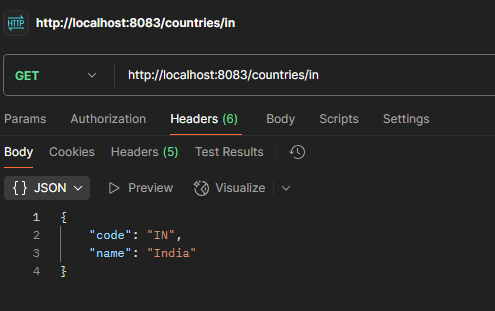
### Screenshots

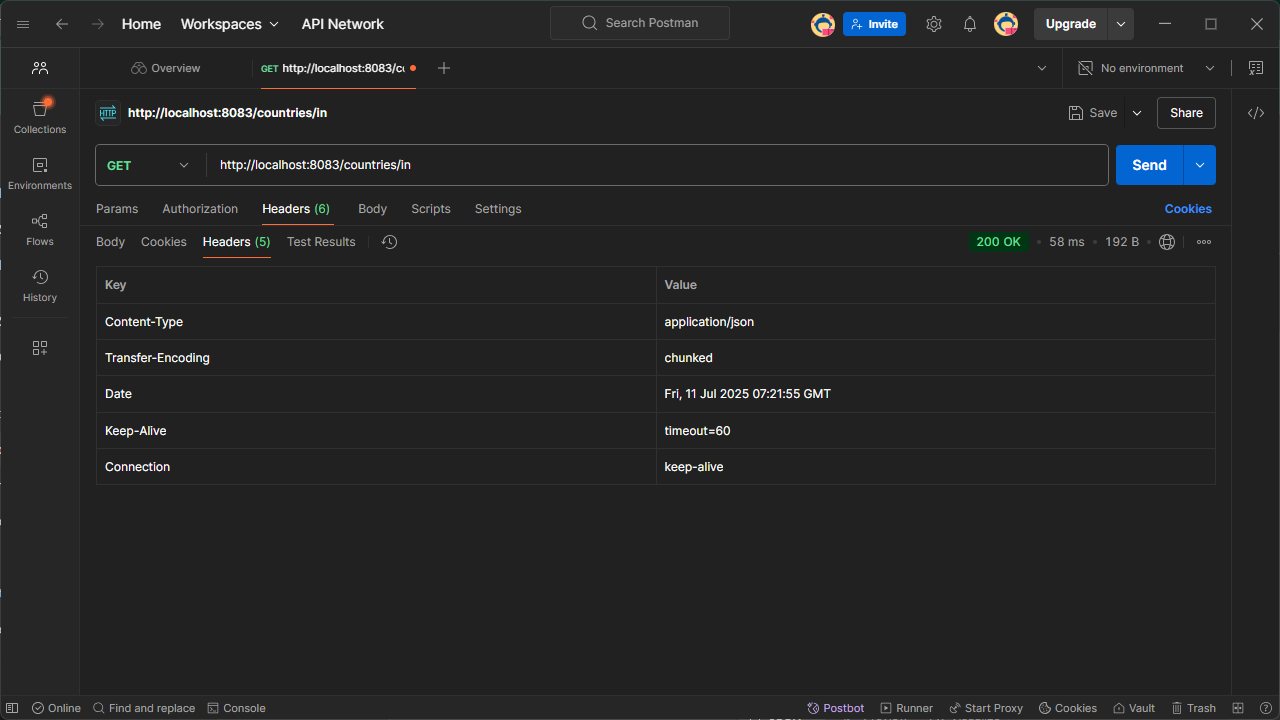
[Screenshot: Controller method with @PathVariable]



[Screenshot: Service method for country lookup]

[Screenshot: Postman success response for /countries/IN]





## REST – Get country exceptional scenario

### Objective

Gracefully handle the case where the country code is not present.

### Steps Followed

- Created CountryNotFoundException class

- Threw exception in service when code not found

- Handled exception in controller using @ExceptionHandler

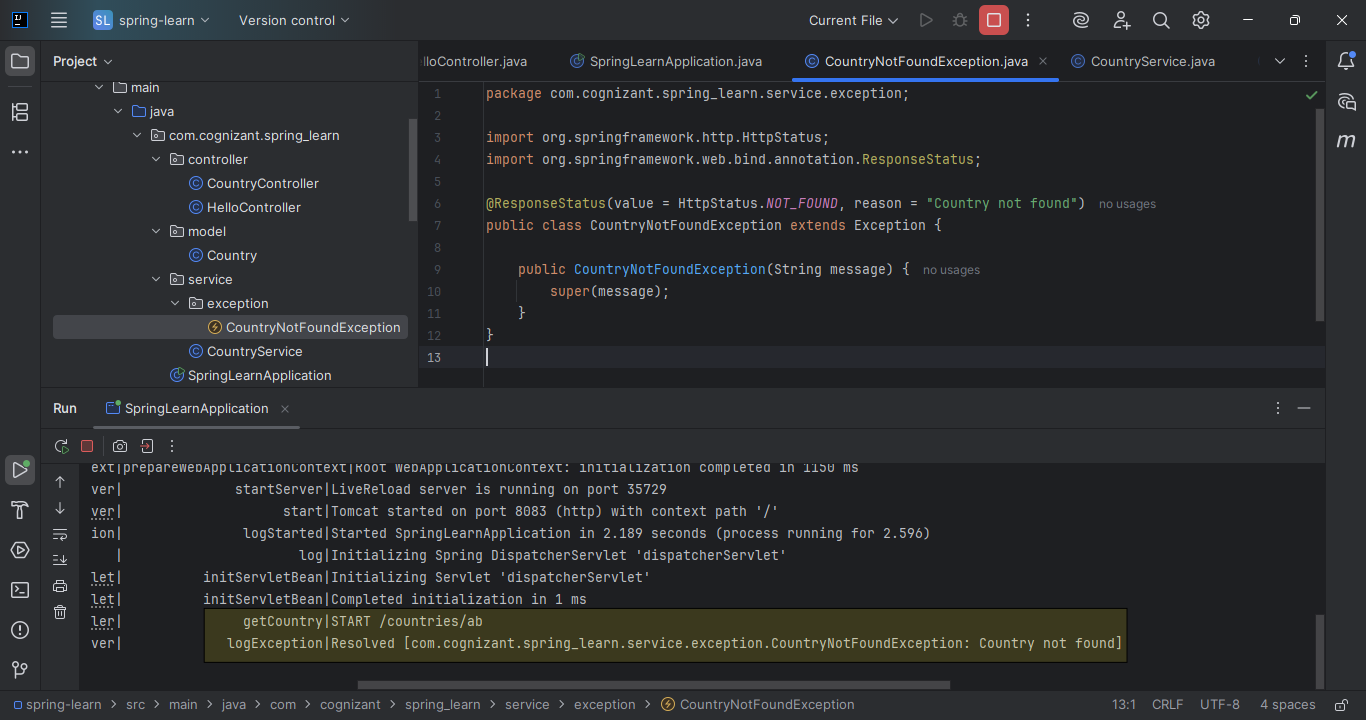
- Returned JSON error response with HTTP 400

- Verified using Postman and curl

- Implemented and tested using MockMVC with JUnit

### Screenshots

[Screenshot: CountryNotFoundException.java and console log]



[Screenshot: Exception handler method in controller]

[Screenshot: Postman error response for /countries/ab]

