### **1. Changes in country.xml**

* Defined four country beans: one each for India, United States, Germany, and Japan.
* Each bean contains a two-letter ISO code and the country name.
* A new countryList bean was created to group all these country beans into a list.
* The list is used in the service layer to retrieve country details dynamically.
* This XML file is placed under src/main/resources and loaded using ClassPathXmlApplicationContext.

**2. Service Creation – CountryService**

* A service class was created in the com.cognizant.spring\_learn.service package.
* It contains a method named getCountry(String code) which:  
  + Loads the countryList bean from the Spring XML configuration.
  + Performs a case-insensitive search to match the input country code.
  + Returns the corresponding country object.
* Used either traditional iteration or Java 8 Stream API for filtering.

### **3. Controller Setup – CountryController**

* A controller class was created under the com.cognizant.spring\_learn.controller package.
* It exposes a GET endpoint: /countries/{code}.
* The controller method extracts the country code from the path using @PathVariable.
* It calls the service method getCountry() with the extracted code.
* The controller returns the country object, which Spring automatically converts to a JSON response using Jackson

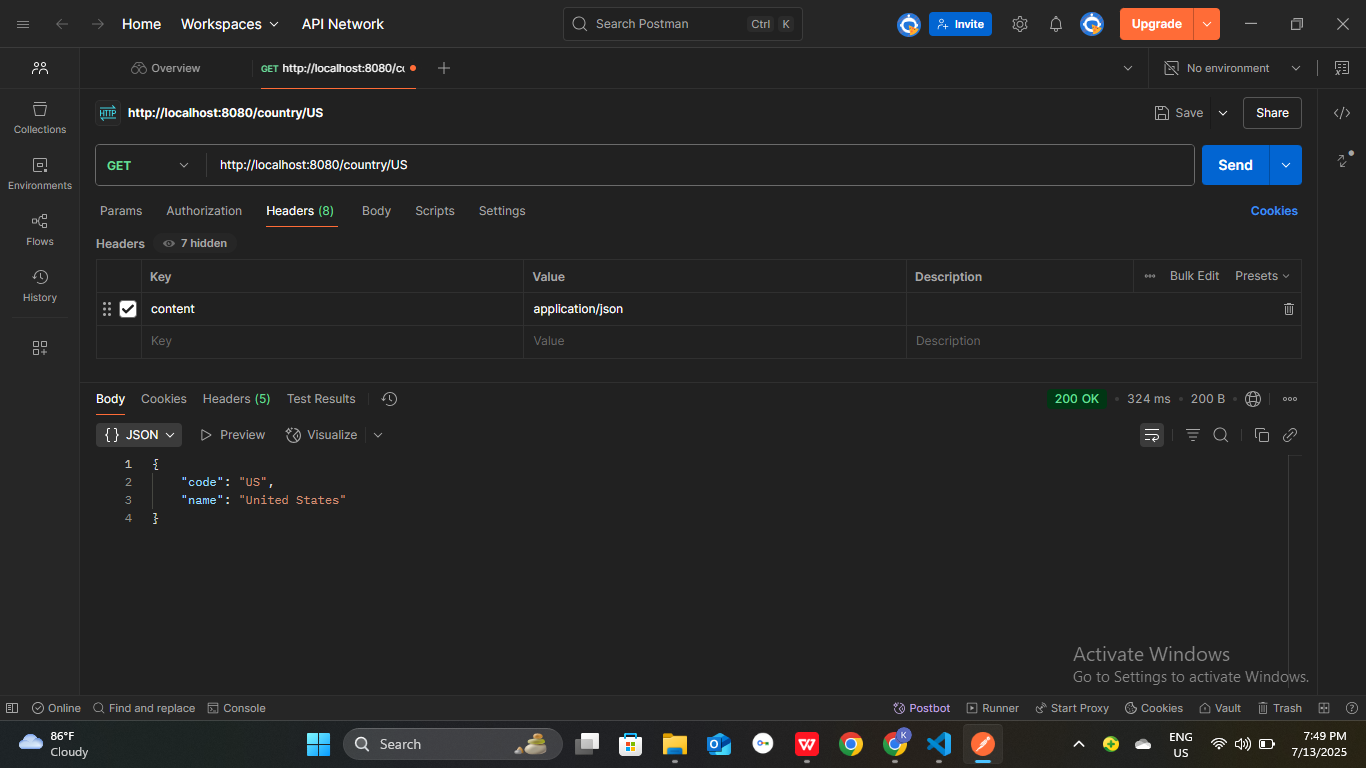
### **4. Output Behavior (Functional Flow)**

* A GET request to /countries/in, /countries/US, /countries/de, etc., returns the matching country’s code and name.
* The match is case-insensitive.
* If found, a JSON response is returned with HTTP status 200 OK.
* If not found, null is returned or handled depending on error handling setup (optional improvement).
* The browser’s Developer Tools (Network tab) and Postman both show response headers like Content-Type: application/json.

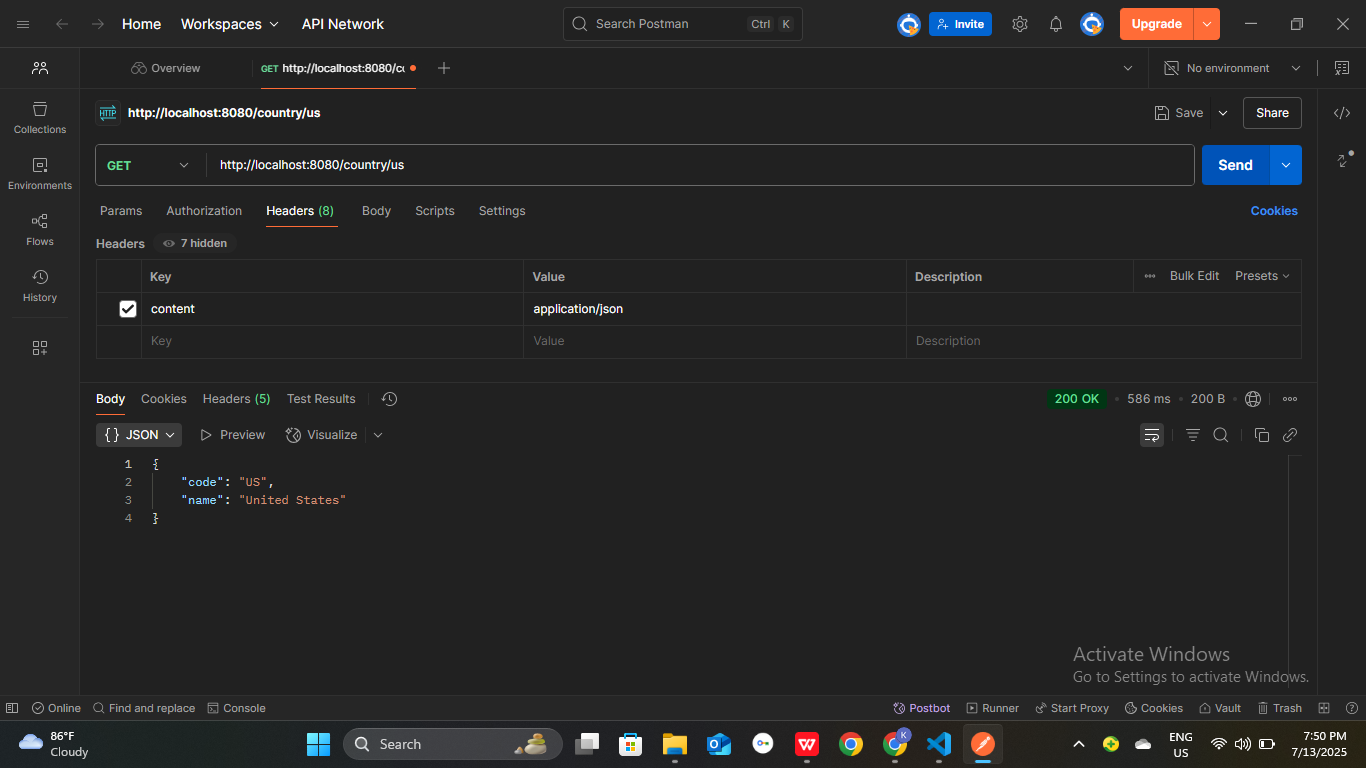
### **5. Testing & Verification**

* The endpoint was tested on http://localhost:8083/countries/in in both:  
  + A web browser (Network tab used to inspect headers and payload).
  + Postman (Headers tab used to verify HTTP response metadata).
* The expected output JSON is successfully returned based on the provided country code.

**Output:** Uppercase

****

**Lowercase:**

****