**REST - Country Web Service**

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

**1. What happens in the controller method?**  
 When you send a request like GET http://localhost:8083/api/users/42, Spring’s **DispatcherServlet** first receives that HTTP call. It matches the URL and method to your @GetMapping("/{id}") handler in UserController. Spring extracts the path variable id (in this case, 42) and invokes your method signature public User getUser(@PathVariable Long id). Inside this method you might call something like User user = userService.findById(id); which fetches the User object from the database. Because the class is annotated with @RestController, Spring automatically understands that your returned User bean should be written directly into the HTTP response body rather than forwarded to a view.

**2. How the bean is converted into a JSON response**  
 Once your controller returns the User object, Spring’s **content‐negotiation** mechanism examines the client’s Accept header (for example, Accept: application/json). Recognizing that JSON is requested, Spring delegates serialization to its Jackson‐based MappingJackson2HttpMessageConverter. Jackson introspects the User class—looking at its getters or fields—and produces a JSON string. For example, if your User bean has id = 42, firstName = "Alice", lastName = "Smith", and email = "alice@example.com", Jackson will emit:

{"id":42,"firstName":"Alice","lastName":"Smith","email":"alice@example.com"}

Spring then sets the HTTP response header Content-Type: application/json;charset=UTF-8 and writes that JSON text into the response body.

**3. Example of HTTP headers in the browser’s Network tab**  
 After opening Developer Tools (e.g., F12 in Chrome) and selecting the **Network** panel, trigger your API call. Click on the request entry, then the **Headers** tab. You will see sections like:

* **General**
  + Request URL: http://localhost:8083/api/users/42
  + Request Method: GET
  + Status Code: 200 OK
* **Response Headers**
  + Content-Type: application/json;charset=UTF-8
  + Content-Length:  ninety bytes (actual number may vary)
  + Date: Sun, 13 Jul 2025 06:30:00 GMT+0530
  + Server: Apache-Coyote/1.1
* **Request Headers**
  + Accept: application/json
  + User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) …

**4. Example of HTTP headers in Postman**  
In Postman, after sending GET http://localhost:8083/api/users/42, switch to the **Headers** tab beneath the response body. You will see two columns without a table:

**Response Headers**

Content-Type: application/json;charset=UTF-8

Content-Length: 87

Date: Sun, 13 Jul 2025 06:30:00 GMT+0530

Server: Apache-Coyote/1.1

**Request Headers**

Accept: application/json

Host: localhost:8083

User-Agent: PostmanRuntime/7.29.0