## What happens in the controller method?

The getCountryIndia() method performs three main actions:

**Load Context**: It creates a Spring ApplicationContext by loading the bean definitions from the country.xml file located in the classpath.

**Retrieve Bean**: It asks the context for the bean with the ID "in" and tells Spring that it expects an object of the Country class. Spring finds the bean definition, creates an instance of the Country class, and populates its properties (code and name) as defined in the XML.

**Return Object**: The method returns the fully configured Country object.

## How is the bean converted into a JSON response?

The conversion from a Java object to a JSON string is handled automatically by Spring Boot[1](https://spring.io/guides/gs/rest-service" \t "https://www.perplexity.ai/search/_blank).

The @RestController annotation on the CountryController class is key. It tells Spring that the return value of a method should be written directly into the HTTP response body[2](https://stackoverflow.com/questions/44839753/returning-json-object-as-response-in-spring-boot" \t "https://www.perplexity.ai/search/_blank).

Because your project includes the spring-boot-starter-web dependency, it comes with the **Jackson** library, which is a powerful tool for converting Java objects to and from JSON[3](https://stackoverflow.com/questions/23373345/replying-with-a-json-for-a-spring-restful-web-service" \t "https://www.perplexity.ai/search/_blank).

When your method returns the Country object, Spring's message converters detect Jackson on the classpath and use it to **serialize** the Country object into a JSON string, which is then sent as the HTTP response[4](https://www.baeldung.com/spring-boot-json" \t "https://www.perplexity.ai/search/_blank).

## Viewing HTTP Headers

The HTTP headers provide important metadata about the response.

**In Chrome Developer Tools**:

Press **F12** to open Developer Tools and go to the **Network** tab.

Access http://localhost:8083/country

Click on the country request in the list

In the **Headers** tab, under **Response Headers**, you will see Content-Type: application/json. This header is crucial because it tells the client that the body of the response is formatted as JSON.

**In Postman**:

After sending your GET request to http://localhost:8083/country.

Look at the response section and click the **Headers** tab

Postman displays a clear, easy-to-read list of all response headers, including the Content-Type: application/json header, confirming the server sent JSON data as expected.