REST – Country Web Service

# 1. What Happens in the Controller?

**Q: What is the purpose and function of the controller method in the Spring Boot REST API?**

Answer:  
The controller method in a Spring Boot REST application serves as the entry point for handling HTTP requests. In the CountryController class, the getCountryIndia() method is responsible for processing requests to the /country endpoint. Here's what happens step-by-step:  
  
1. Annotation: @RestController  
 - Tells Spring that this class is a REST controller. It combines @Controller and @ResponseBody.  
  
2. Annotation: @RequestMapping("/country")  
 - Binds the HTTP request path /country to the getCountryIndia() method.  
  
3. Application Context Creation  
 - ClassPathXmlApplicationContext loads the country.xml Spring configuration.  
  
4. Bean Retrieval  
 - getBean() retrieves the Country bean (with ID 'in') from the Spring context.  
  
5. Return of Bean  
 - The Country object is returned and automatically converted to JSON.  
  
6. Logging  
 - Logs are recorded at the start and end of the method for traceability.

# 2. How Is the Bean Converted to JSON?

**Q: How does Spring Boot convert the Java bean into a JSON response in a REST API?**

Answer:  
The automatic conversion of a Java object (such as Country) to JSON in a Spring Boot application is enabled by the Jackson library, which is included by default in Spring Boot starter dependencies.  
  
1. Automatic Dependency  
 - spring-boot-starter-web includes Jackson (jackson-databind) for JSON processing.  
  
2. Object Mapping via Jackson  
 - Spring Boot uses Jackson’s ObjectMapper to serialize Java objects to JSON.  
  
3. Reflection on Getters  
 - Jackson uses reflection to call public getter methods (getCode(), getName()) to read values.  
  
4. JSON Output Generation  
 - Jackson maps the Java object to a JSON structure:  
 {  
 "code": "IN",  
 "name": "India"  
 }  
  
5. Content-Type Header  
 - Spring sets the Content-Type header to application/json automatically.