**Exercise 10: Online Bookstore - Configuring Content Negotiation**

**Business Scenario:**  
Your online bookstore needs to support multiple media types (JSON and XML) for its RESTful services. This will allow clients to request responses in different formats based on their needs.

**1. Content Negotiation Configuration**

**Objective:**  
Configure Spring Boot to handle different media types in responses.

**Steps:**

1. **Create a Configuration Class:**
   * WebConfig.java: Define the configuration for content negotiation.
   * **Explanation:**
     + favorParameter(true): Allows the use of a query parameter (e.g., ?mediaType=xml) to specify the desired media type.
     + parameterName("mediaType"): Specifies the parameter name used for content negotiation.
     + ignoreAcceptHeader(false): Ensures that the Accept header in the HTTP request is considered for content negotiation.
     + defaultContentType(MediaType.APPLICATION\_JSON): Sets JSON as the default media type.
     + mediaType("xml", MediaType.APPLICATION\_XML): Maps the "xml" media type to XML.
     + mediaType("json", MediaType.APPLICATION\_JSON): Maps the "json" media type to JSON.

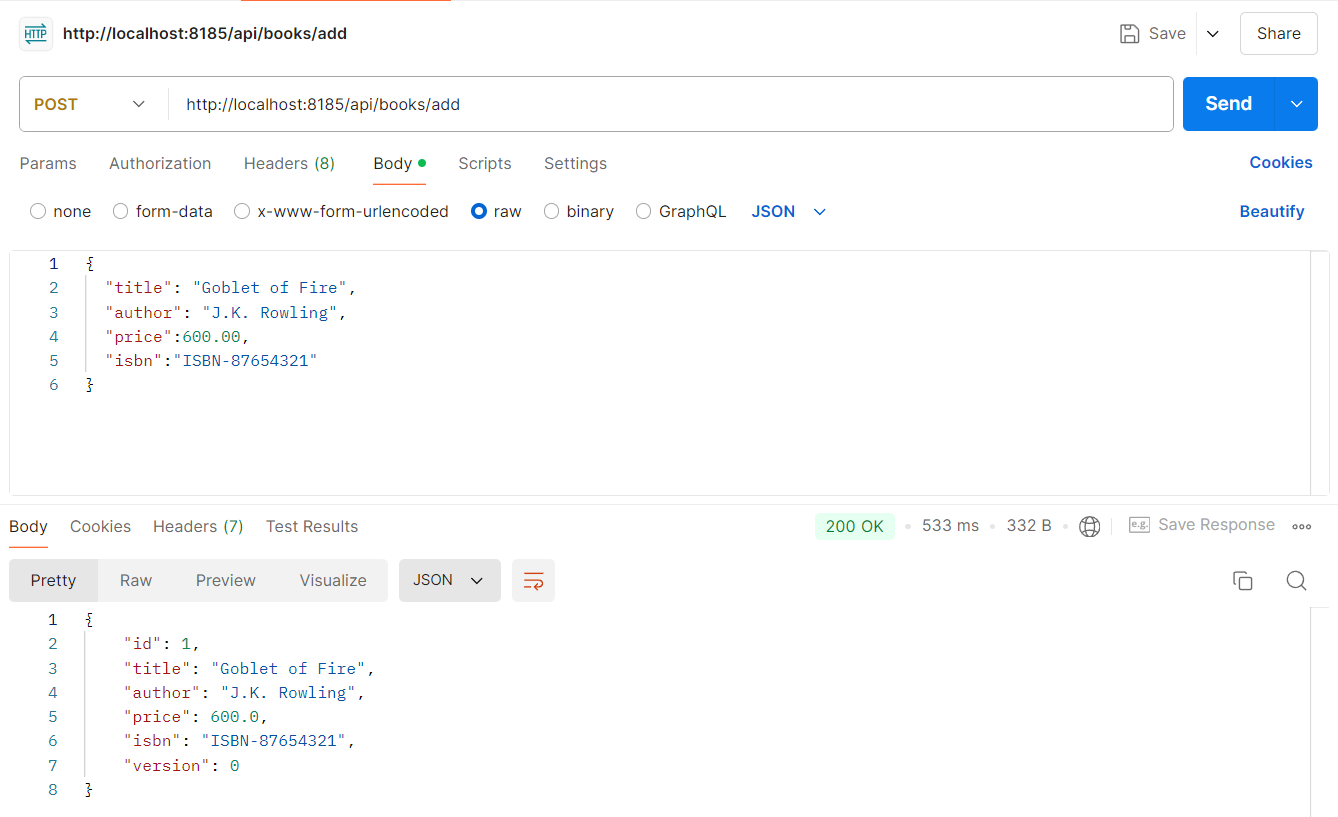
**2. Handling Media Types in Controllers**

**Objective:**  
Ensure that the RESTful services can produce and consume different media types based on the Accept header or query parameters.

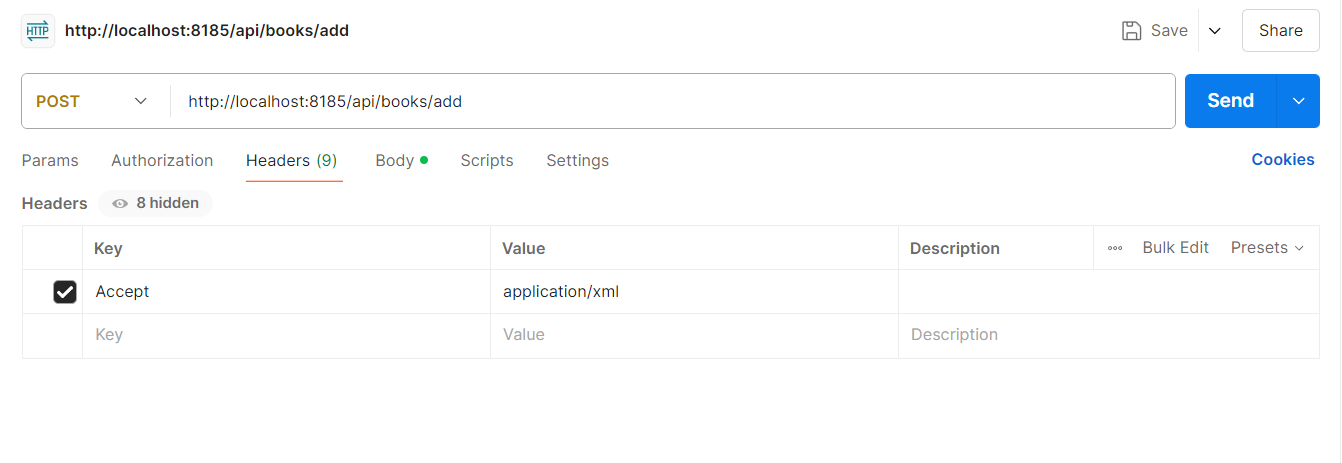
**Steps:**

1. **Update Controller Methods:**
   * BookController.java: Define how different media types are handled in request and response bodies.
   * **Explanation:**
     + produces = { MediaType.APPLICATION\_JSON\_VALUE, MediaType.APPLICATION\_XML\_VALUE }: Specifies the media types the endpoint can produce.
     + consumes = { MediaType.APPLICATION\_JSON\_VALUE, MediaType.APPLICATION\_XML\_VALUE }: Specifies the media types the endpoint can consume.
2. **Testing:**

**Adding Book(By Default JSON):**



In Header, providing Key and Value for XML format:



XML Format:

