

## Content Negotiation

### 👉 What is Content Negotiation?

Content negotiation allows clients and servers to communicate in different formats like JSON or XML based on what they prefer or support. This happens through HTTP headers that specify the desired format.

### 👉 How to Request XML Instead of JSON

When using Postman to get job posts:

- By default, Spring Boot returns data in JSON format
- To request XML format instead:
  - Go to the Headers tab in Postman
  - Add a header with key **Accept** and value **application/xml**
  - Send your request

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/jobPosts`. In the Headers section, there is a row for the `Accept` header with the value `application/xml`. The response status is shown as `406 Not Acceptable`.

### 👉 Understanding Error 406

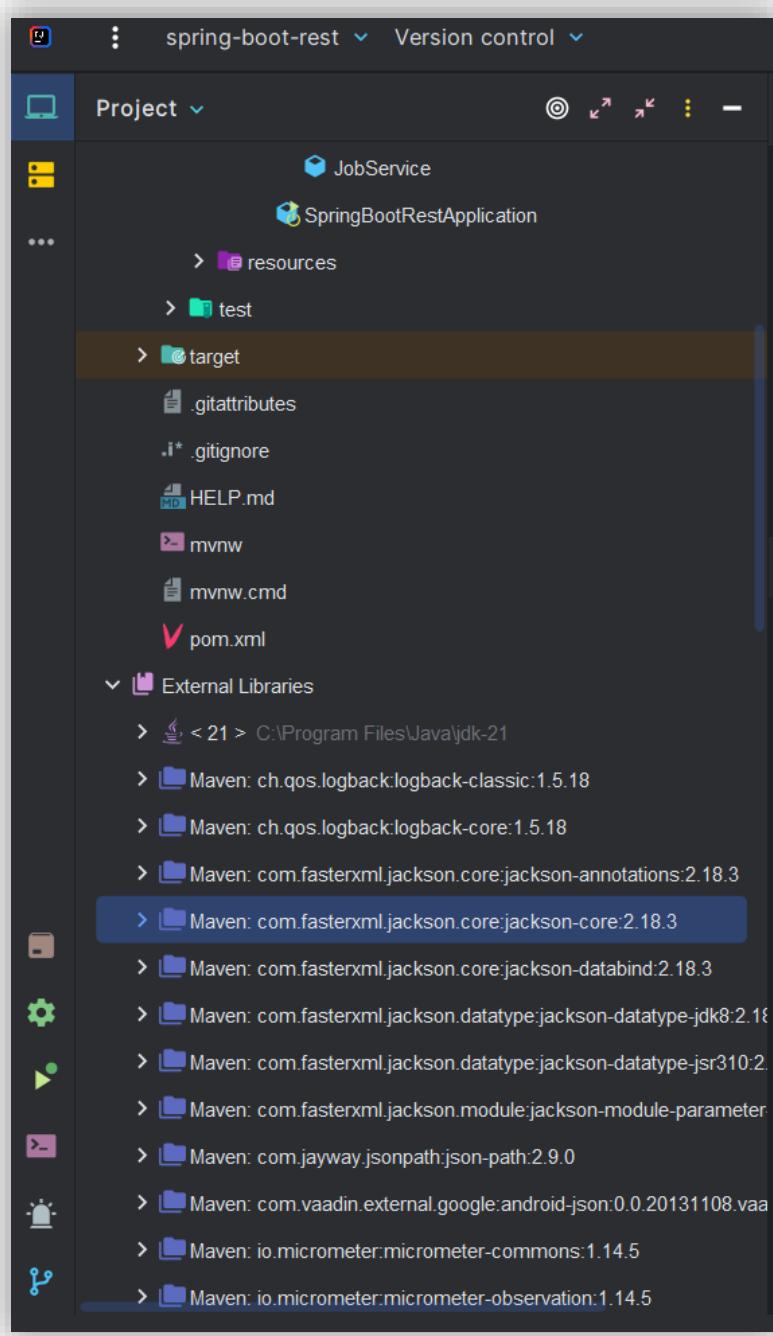
The 406(Not Acceptable) error occurs because:

- The backend returns a List of JobPost objects

```
● ● ●  
@GetMapping("jobPosts")  
public List<JobPost> getAllJobs() {  
    return service.getAllJobs();  
}
```

- Spring Boot needs a library to convert Java objects to XML

- By default, Spring only has Jackson configured for JSON conversion



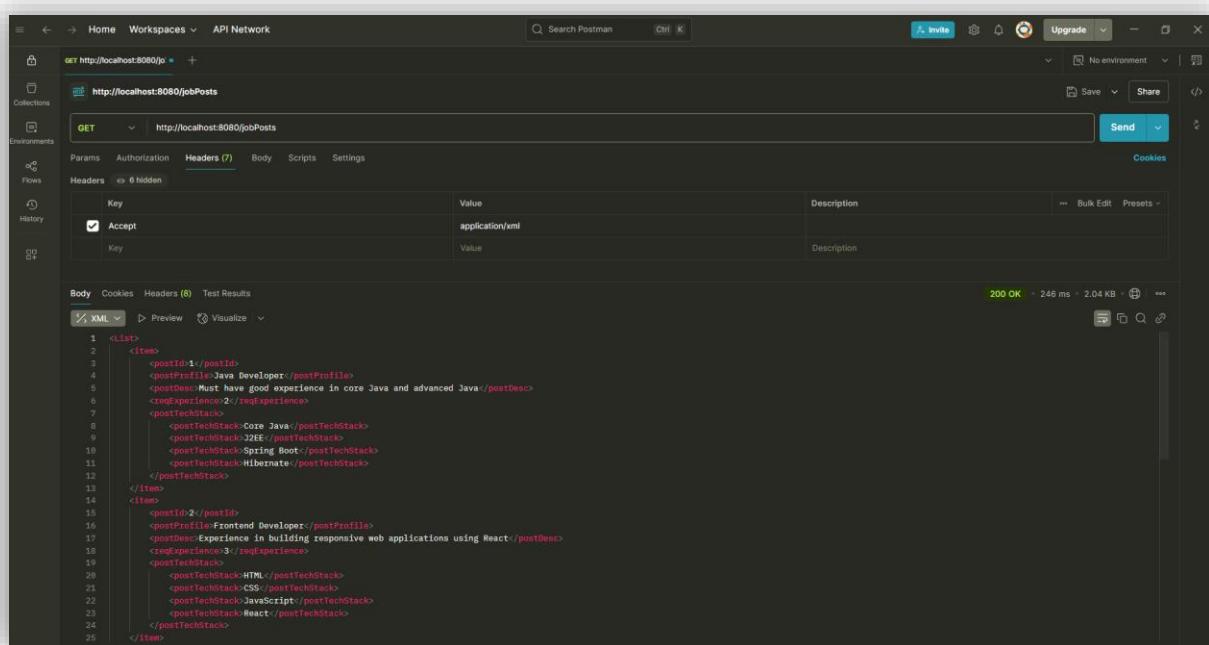
## 👉 Adding XML Support

To enable XML responses:

- Add the Jackson XML formatter dependency to your pom.xml:

```
<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.dataformat/jackson-dataformat-xml -->
<dependency>
    <groupId>com.fasterxml.jackson.dataformat</groupId>
    <artifactId>jackson-dataformat-xml</artifactId>
    <version>2.18.3</version>
</dependency>
```

- Reload your project
  - Now your requests with the `Accept : application/xml` header will return XML responses
  - Now we can see the response is in the XML format



## 👉 Restricting Content Types in Controllers

We can control which formats your endpoints support using the `produces` and `consumes` attributes.

### For Responses

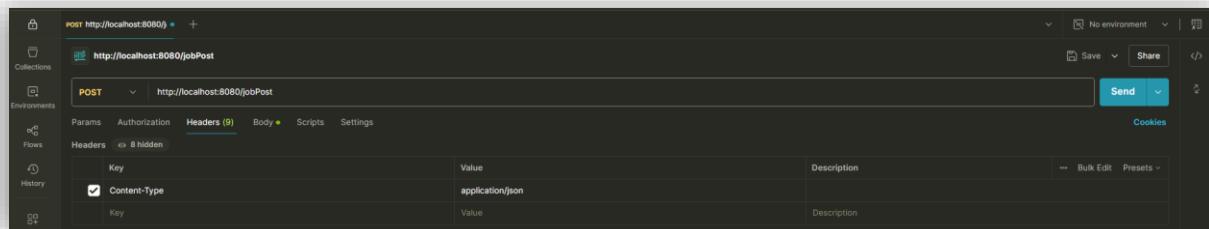
```
@GetMapping(path="jobPosts", produces = {"application/json"}) //content negotiation
public List<JobPost> getAllJobs() {
    return service.getAllJobs();
}
```

- This controller method will only produce JSON responses, even if XML is requested.

### For Requests

When sending data to the server:

- Set the `Content-Type` header to specify the format you're sending  
Example: `Content-Type : application/json`



- Configure your controller to only accept specific formats:

```
@PostMapping(path="/jobPost", consumes = "application/json") //content negotiation
public JobPost addJob(@RequestBody JobPost jobPost){
    service.addJob(jobPost);
    return service.getJob(jobPost.getPostId());
}
```

## 👉 Remember

- Content negotiation happens through HTTP headers
- `Accept` header: what format the client wants to receive
- `Content-Type` header: what format the client is sending
- Spring Boot needs proper libraries to support different formats
- Use `produces` to restrict response formats
- Use `consumes` to restrict request formats