

Lab 6.3 Content Negotiation

- Annotate the StudentController method mapped to “/single” in your service project to produce either XML or Json textual responses

```
@GetMapping(path="/single",  
produces= {MediaType.APPLICATION_JSON_VALUE, MediaType.APPLICATION_XML_VALUE})  
public ResponseEntity<Student> getSingleStudent(@RequestParam("id") Optional<Long>  
optional) {  
    return ResponseEntity.ok(studentService.get(optional.orElse(11)));  
}
```

- Make sure that your Student class is annotated with @XmlRootElement, the Json annotation excludes null values in the student when returning a textual representation of the student class.
- Launch your service project.

```
@JsonInclude(Include.NON_NULL)  
@XmlRootElement  
public class Student
```

Lab 6.3 Content Negotiation

- In your CLIENT PROJECT complete the method in the class StudentControllerTest, testGetOneRequestParam()
- Create a header with the desired MediaType of XML

```
HttpHeaders headers = new HttpHeaders();  
headers.add("accept", MediaType.APPLICATION_XML_VALUE);
```

- Trigger your service

```
ResponseEntity<String> responseEntity =  
new RestTemplate().exchange(URL + "/single?id={id}", HttpMethod.GET, new  
HttpEntity<String>(headers), String.class, 3);  
System.out.println(responseEntity.getBody());
```

- Repeat for a Json response, this means changing the Headers
- Repeat where you have both XML and json as an Accept header

```
headers = new HttpHeaders();  
headers.add("accept", "application/xml, application/json");
```