

Lab 6.2 A Java Client

Posting and Getting data – THERE IS START UP CODE IN THIS LAB

In your IDE, open the client project to be found in this lab's setup directory

In the class, StudentControllerTest, in src/test/java and the Complete the method void testAdd()

Use a new RestTemplate and use the postForEntity method, this takes the following arguments

- URL,
- new HttpEntity<Student>(student, headers),
- String.class

Then use the returned ResponseEntity to get the first header called "location", this gives us the url of the newly inserted Student

Then use the templates getForEntity method to retrieve the newly inserted Student with the arguments

- "http://localhost:8081" + url from the retrieved header
- Student.class);
- Retrieve the body from the returned ResponseEntity and print it out

Lab 6.2 A Java Client

You can actually test for both positive and negative outcomes

```
void testAdd() {
    Student student = new Student(0, "Susan", "Doubtfire", "French", 75.00);
    ResponseEntity<String> responseEntity = new RestTemplate().postForEntity(URL, new
        HttpEntity<Student>(student, headers), String.class);
    String locationUrl = responseEntity.getHeaders().get("location").get(0);
    ResponseEntity<Student> response = new RestTemplate().getForEntity("http://localhost:8081" + locationUrl,
        Student.class);
    System.out.println(response.getBody());
}

@Test
void testAddNegative() {
    Student student = new Student(0, null, "Doubtfire", "French", 75.00);
    Assertions.assertThrows(HttpClientErrorException.class, () -> {
        ResponseEntity<?> responseEntity = new RestTemplate().postForEntity(URL, new
            HttpEntity<Student>(student, headers), String.class);
        assertEquals(HttpStatus.BAD_REQUEST, responseEntity.getStatusCode());
    });
}
```

And maybe test some of the @GetMapping endpoints

```
@Test
void testGetAll() {
    ResponseEntity<Collection<Student>> responseEntity =
        new RestTemplate().exchange(URL, HttpMethod.GET, null, new
        ParameterizedTypeReference<Collection<Student>>() {
        });
    assertThat(responseEntity.getStatusCode(), equalTo(HttpStatus.OK));
    responseEntity.getBody().forEach(p-> {
        System.out.println(p);
    });
}

@Test
void testGetOnePathParam() {
    ResponseEntity<Student> responseEntity =
        new RestTemplate().exchange(URL +("/{id}", HttpMethod.GET, null, Student.class, 3);
    assertThat(responseEntity.getStatusCode(), equalTo(HttpStatus.OK));
    System.out.println(responseEntity.getBody());
}
```

Lab 6.2 A Java Client

Launch your Service Project

Run your JUnit in your Client Project against the running Service project

