Lab 2.3

Expose a Repository as a REST service and use the HAL Explorer

In your lab setup directory for this lab, replace your Student class which the Student class that you find there, also bring into the same package as Student the class Course

Annotate tour StudentRepository with; @RepositoryRestResource(collectionResourceRel = "student", path = "enrollments")

From your setup directory bring into the com.student.repository package the interface CourseRepository. Annotate this second interface with @RepositoryRestResource(collectionResourceRel = "course", path = "courses")

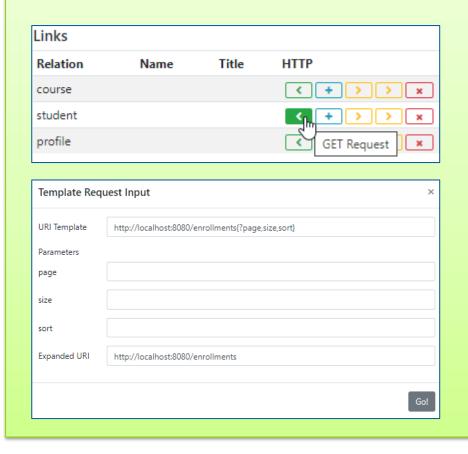
Relaunch your Spring Boot Application

Navigate to http://localhost:8080/

You will be redirected to the HAL Explorer at http://localhost:8080/explorer/index.html#uri=/

The HAL Explorer

 Click on the Student "<" GET button and then on the resultant dialog click "GO"



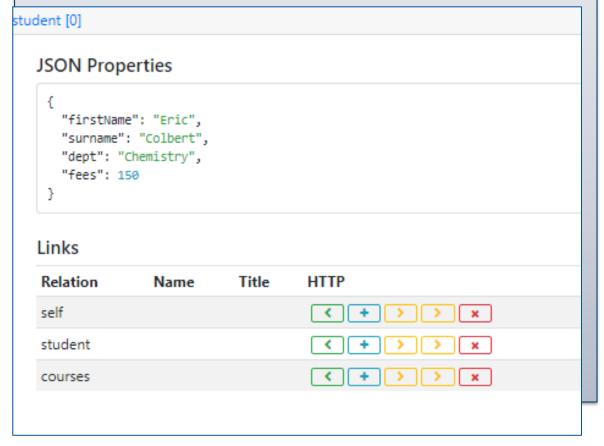
 You will see a list of Students on the right hand side in formatted json

```
"student": [
    "firstName": "Eric",
    "surname": "Colbert",
    "dept": "Chemistry",
    "fees": 150,
    " links": {
      "self": {
       "href": "http://localhost:8080/enrollments/1"
      "student": {
       "href": "http://localhost:8080/enrollments/1"
      "courses": {
       "href": "http://localhost:8080/enrollments/1/courses"
```

Lab 2.3

Navigation JPA Relationships

On the left had side we see a GUI representation of these Student json data structures, select the first one; then click the courses "<" GET button



 This illustrates the courses for that student, we traversed the @OneToMany Relationship.
 Select the fist course link. If you click the student "<" GET button you will go back to the Student details, traversing the @ManyToOne relationship

Links			
Relation	Name	Title	НТТР
self			< + > > x
Embedded F	Resources		
course			
course [0]			
JSON Properties { "title": "French History", "location": "University of Texas" } Links			
Relation	Name	Title	НТТР
self			< + > > x
course			< + > > x
student			*
			GET Request