

Lab 2.3
















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

- 1 In your lab setup directory for this lab, replace your Student class with the Student class that you find there, also bring into the same package as Student the class Course
- 2 Annotate your StudentRepository with; `@RepositoryRestResource(collectionResourceRel = "student", path = "enrollments")`
- 3 From your setup directory bring into the `com.student.repository` package the interface `CourseRepository`. Annotate this second interface with `@RepositoryRestResource(collectionResourceRel = "course", path = "courses")`
- 4 Relaunch your Spring Boot Application
- 5 Navigate to <http://localhost:8080/>
- 6 You will be redirected to the HAL Explorer at <http://localhost:8080/explorer/index.html#uri=/>

Lab 2.3

The HAL Explorer

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

Links			
Relation	Name	Title	HTTP
course			    
student			    
profile			    

Template Request Input

URI Template

Parameters

page

size

sort

Expanded URI

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

student [0]

JSON Properties

```
{
  "firstName": "Eric",
  "surname": "Colbert",
  "dept": "Chemistry",
  "fees": 150
}
```

Links

Relation	Name	Title	HTTP
self			< + > > x
student			< + > > x
courses			< + > > x

- This illustrates the courses for that student, we traversed the @OneToMany Relationship. Select the first 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	HTTP
self			< + > > x

Embedded Resources

course

course [0]

JSON Properties

```
{
  "title": "French History",
  "location": "University of Texas"
}
```

Links

Relation	Name	Title	HTTP
self			< + > > x
course			< + > > x
student			< + > > x

GET Request