Lab 1.1

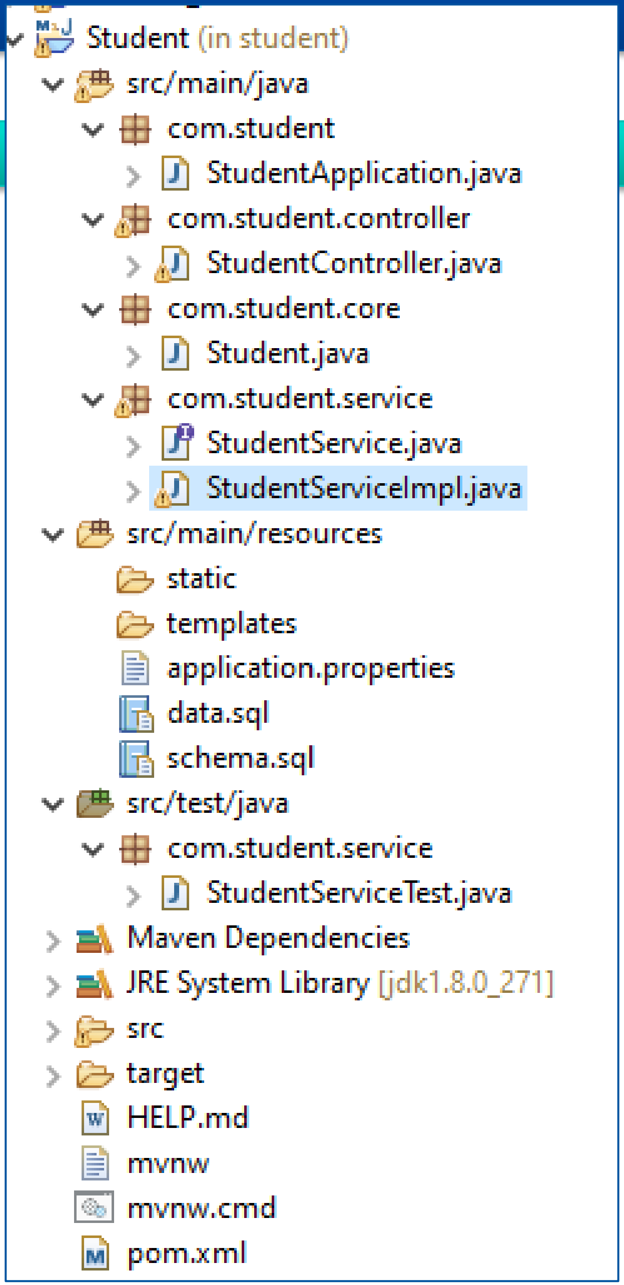
Utilize and Injected EntityManager

In the setup directory for your course Lab01.1contains a maven project that we will work in. Open the project in eclipse. We can see that this is a Spring Boot Project with the necessary dependencies already attached to it via the maven pom. You may have to do a maven clean and build for the all the dependencies to resolve.

We are using a n H2 embedded database that will be populated at application start up time and then dropped. Great for testing purposes. It is populated from the schema.sql and data.sql file that you will find under the resources directory of your project

If you open the class Student, you will see that it is already annotated as a JPA Entity

You also have a test class under src/test/java



```
Student (in student)
  src/main/java
    com.student
      StudentApplication.java
    com.student.controller
      StudentController.java
    com.student.core
      Student.java
    com.student.service
      StudentService.java
      StudentServiceImpl.java
  src/main/resources
    static
    templates
    application.properties
    data.sql
    schema.sql
  src/test/java
    com.student.service
      StudentServiceTest.java
  Maven Dependencies
  JRE System Library [jdk1.8.0_271]
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

Lab 1.1

Delegate to the EntityManager and Database

In the class StudentServiceImpl, there are several TODO statements. You are to Inject in an EntityManager using @PersistenceContext

Use this EntityManager to in the two methods to;

- Get an individual student using find find(Student.class, id);
- Create a query, "SELECT student FROM Student student", Student.class" and from that query execute the getResultList method to bring back the Collection of students

You can test using the provided JUnit test class as previously mentioned, this SpringBootTest test class actually launches the Spring Boot application ensuring all dependencies for JPA are in place without the need for further configuration, however, currently our application has no web capabilities because the application.properties class has omitted them with the entry "spring.main.web-application-type=none"