

Expeditors Backend Academy Labs

Introduction

This document contains the labs for the Expeditors Backend Academy.

The instructions are split up into **Classwork** and **Homework**. The Classwork we will do in class together. You should do the Homework in the **Homework** module. Create a new package for each week's homework if it makes sense. I will go through an example before we start.

Week 15 – Backend Capstone 2

Objectives

1. Apply the database and JPA learning from the last modules to have your application interact with a database.
2. Expand the functionality of your Track Price Service
3. Protect your application with Spring Security.

Tasks

Track Service

1. Design a database for your application.
 - a. Again, you will do this together with your group.
2. Build your design
 - a. And again, the daily morning scrum should be a part of this exercise.
 - b. Write the SQL to create and populate the database.
 - c. Convert your application to use JPA to interact with the database.
 - d. Write any new tests that are necessary. And, of course, make sure all your old tests still work.
3. Set up your application such that you can switch the database you will use for any particular run. You will have to setup another database to make this work. H2 is an always available option with Spring Boot.

Pricing Service

1. If you didn't make one the last time, you should make one now. Look at the instructions for the last Capstone

2. Add functionality to the Track Price service to be able to change the lower and upper limit of the prices it shows.
3. Add code to allow the limits to be read and written remotely using a REST API

Security

1. Once you have it working, add Basic Http Security to the Pricing service.
2. Create at least two users, one with privileged access (e.g. ADMIN) and the other with regular access (e.g. USER).
3. Configure your Security Filter Chain so
 - a. Only privileged users can change the limits
 - b. All users have to be authenticated.
4. You will now have to also change the Rest Client code in your Track application to include a Basic Authorization header when it calls the pricing service.
5. Once you have that working, you can add Basic Security to the Track application.

Optional: May require some self study

1. Add SSL to your Pricing app.
 - a. May require some self study.
 - b. It will also require changes in the Rest Client you use to call the pricing service.
2. And now that you have come this far, you might as well add SSL to your Track application as well.

Optional plusplus

1. Use the MusicInfoExtractor or a similar program to feed “real” Track data to your application. We have not seen this yet, we will. And it will be purely for fun if you want to use it.

Deliverables

1. A working Spring Boot REST application that is talking to a database and has a security layer to prevent unauthorized access.
2. A suite of JUnit tests that provides as large coverage as possible. Aim for the high 80's, specially for controller and service code.
3. A presentation of the final project.