Web API Design with Spring Boot Week 16 Coding Assignment

Points possible: 75

URL to GitHub Repository:

https://github.com/lopeze25/testrepo

In Week18FinalProject

URL to Public Link of your Video:

https://drive.google.com/file/d/1W4A_BPGyo9SLZnjwdDhRntlB2zK4A-ZB/view

---

Instructions:

Create a GitHub Repository for your Final Project

- 1. Push all of your code to your Final Project Repository (Repo).
- 2. Push any relevant . SQL files to your Repo.
- 3. Push a PDF of your ERD to your Repo.
- 4. Include any additional documentation for your project.
- To submit your project:
- - Click Add Submission Button
- - Add your Final Project GitHub Repo URL to Text Box
- - Add your Public Video Showcase Link to Text Box after the GitHub repo.
- REQUIRED: PUBLIC GitHub Repo link with everything listed above, & PUBLIC link to Video!

Create a Showcase Video (Public Link)

- Create a video, up to five minutes max, showing and explaining how your project works with an emphasis on the portions you contributed. This video should be done using screen share and voice-over. This can easily be done using Zoom, although you don't have to use Zoom, it's just what we recommend. You can create a new meeting, start screen sharing, and start recording. This will create a video recording on your computer.
- UPLOAD that video record to a publicly accessible site, such as YouTube, Dropbox, or Google Drive. MAKE SURE THE LINK YOU SHARE IS PUBLIC. If it is not accessible by your grader, your project will be graded based on what they can access.
- To submit your project:
- - Click Add Submission Button
- - Add your Final Project GitHub Repo URL to Text Box
- - Add your Public Video Showcase Link to Text Box after the GitHub repo.
- REQUIRED: PUBLIC GitHub Repo link with everything listed above, & PUBLIC link to Video!

Web API Design with Spring Boot Week 16 Coding Assignment

Requirements Reminder:

- 1-person Project:
- Database design that contains at least 3 entities and 3 tables.
- Contains all CRUD operations (Create, Read, Update & Delete).
- Each entity should have CRUD operations, with one having all 4 CRUD operations (Create, Read, Update & Delete).
- Contains at least 1 one-to-many relationship.
- Contains at least 1 many-to-many relationship with one or more CRUD operations on this relationship.

Proposal Template:

Project Participants:

Ezekiel Lopez

Title:

Welding Sales API

Executive Summary:

The Welding Sales API focuses on tracking a welding business's orders and inventory. Given materials representing the metal that the order will use or be performed, equipment, and procedures, a welding business can send Get, Post, and Update requests to keep track of their inventory through quantity and each individual table. The Welding Sales API can also calculate the price of a completed order by calculating hourly wages and the order price (adding together equipment and material expenses).

Initial Features:

Entities: Orders, Materials, Equipment, Procedure, Completed, Worklog, Images, Order Equipment

UPDATE an Order

CREATE/POST a new Order

GET an Order

DELETE an Order

GET the order-equipment for an order

GET a procedure

GET a material

GET an equipment

CREATE/POST an image

GET an image

Stretch Goals (to be completed if time allows, or after graduation):

Short-Term: Finalize the completed and worklog tables to calculate the price. Refactor/ Refine functionality.

Mid-Term: Implement new features for Users, Admins, and Employees. Refactor/Refine functionality.

Long-term: Add Frontend feature. Refactor/ Refine functionality.