Assignment # 3–Developing a Spring Boot App with REST capabilities

Due Date: Week 11

Purpose: By finishing this assignment you will practice to:
- Develop, code and test a Spring Boot app for given requirements.
- Implement RESTful web services in Spring Boot App.

References: Read the lecture slides week 8 – 9 and lab exercises. This material provides the necessary information you need to complete the exercises.

Instructions and rules:

-This lab should be completed by every student **individually.**
-You will have to demonstrate your solution in a scheduled lab session and submitting the project **through the assignment drop box on e-centennial**. You must name your Eclipse project according to the following rule:

Your Name_COMP303_AssignementNumber

Example: **John_COMP303_Assignment3**

Each file submitted in the solution should have student name, student and submission date in the top of the code file.

Comments may be necessary inside the functions or predicates but that the names of the functions and predicates and the comments you include to explain each are especially important.

Student must hand in the assignment to e-centennial drop box **before in-class demonstration.**

Description

Students are asked to develop a simple spring boot app to handle partial "Recruitment Agency App" with job, organization and job category information. You should have a spring boot application file named "SpringBootYourNameAssign3Appplication", which starts the Tomcat Apache server on port 8080 (http://localhost:8080) (You may assign different port number to the server).

You should follow the below given instructions while you are working on this exercise:

a. All new components should have an Id with type of integer or long and all other properties are defined based on given table information in the next page.

b. The architecture of the web service is built with the followings:
   i. Rest Controllers: Implements the processing logic of the web service, parsing of parameters and validation of in- and outputs.

ii. Services: Implements the business logic and handles the access to the resources.
iii. HTML files with thymeleaf templates – may implement this in the same or different project.
iv. POM.xml configuration.
v. You are free to add any other files or configuration needed

c. Your app should support the REST web services of partial "Recruitment Agency App" details as shown in the below given table. (You should use a Map or List as a memory storage, but don't need to create a table and no JPA implementation required)

Job	Organization	Category
jobId	orgId	jobCatId
jobCode	orgName	catCode
jobName	address	catName
jobDesc	postalCode	catDesc
pubDate	phoneNo	
numVacancy	email	
	website	

d. You should define entities, services and controllers for each and every information:
Example for Job information:
Job– Component / Entity name
JobService - service name
JobController – REST controller name

e. Your REST service should perform the following requests for all three entities (Job, Organization, and Category).
   a. GET
   b. POST
   c. PUT
   d. DELETE

f. You should use a POSTMAN google chrome browser app to analyze and test your REST services mapping. (Refer recipe _3_2 and SpringBootRestEmployee examples)

g. Design UI pages using HTML files and use thymeleaf template instead of JSP files to implement web interface only for JOB information. To implement this task you can use the same project or different project. (Refer recipe _3_3 and SpringBootThymeEmployee examples)

h. To fulfill an innovation requirements, students may identify a missing feature or functionality and can implement with this app or adding a new feature based on new dependency to this app (jar file for APIs) or any other relevant code implementation.

Demonstration:

Demonstration is mandatory for this assignment. Every students must demonstrate their solutions in lab. There is a **20% off** for the lab demonstration, if a student is failed demonstration in a scheduled lab.

Assessment Rubrics:

Functionalities: Developing Spring Boot REST app with entities, services, controllers and html files with thymeleaf template and set of appropriate dependency in POM.xml files	35 points
Testing: REST services testing with POSTMAN app and showing output in browser. (Demonstration is important to show your testing)	5 points
UI friendliness, use of CSS, and code standards.	5 points
Innovation – Implement any new feature or additional REST service etc.	5 points
Total	**50 points**

Academic honesty (Plagiarism and cheating)

All students must follow the academic honesty policies regarding Plagiarism and cheating on assignments, Quizzes or Tests. Centennial college's Academic Policy will be strictly enforced. To support academic honesty at Centennial College, all academic work submitted by students may be reviewed for authenticity and originality, with utilizing software tools.

For more details, please visit the Academic Honesty site on
https://www.centennialcollege.ca/mycentennial/your-support/academic-support/student-academic-advising/academic-honesty/