

Course code: PGR209
Course name: Backend programming
Assessment type: mappeksamen i gruppe (2 studenter)
Start date: 28.11.2023
End date: 19.12.2023

Plagiarism detection:

All exam papers, including bachelor's and master's theses will be checked for plagiarism. You can read more about this in the guidelines on citing sources (referencing), plagiarism, and formal requirements for submitting assignments.

General information:

Start uploading your exam paper/file ahead of time, as it may take a long time to upload

Exam papers that are not handed in on Wiseflow by the specified time of the submission date will not be proceeded to assessment

Before submitting, remember to check that all files can be opened and that every file is included. It may be a good

idea to check the saved files on several machines before submitting in Wiseflow.

Incorrect file format or lacking documents may result in the submission not being passed or assessed.

The total file size of the entire exam paper/file must not exceed 5GB in zip format or 5MB in PDF format

Assignment:

You will build a rest Api using Java and Spring Boot.

You must use Spring Data JPA and H2 in-memory database to persist all data.

You must use Repositories, Services, and RestControllers

You should implement pagination for all endpoints that return lists

You may implement Flyway, but it is not required

You should have the following tests, making use of @SpringBootTest, @DataJpaTest where necessary:

Unit tests of services and business logic using MockBean

Integrations tests

End-to-end tests using MockMvc

You are building an order system for a factory that sells machines.

The domain objects are as follows:

Customer

Address

Order

Machine

Subassembly

Part

A customer can have many addresses

An address has one or more customers

A customer can have many orders

An order has one or more machines

A machine has one or more subassemblies

A subassembly has one or more parts

Customer must know about addresses, and addresses must know customers they belong to

Order must know about customers, and customers must know all their orders

Order must know about machines, but machines do not need to know what orders they are part of

Machines must know what subassemblies they require, subassemblies do not need to know what machines they are a part of

Subassemblies must know about parts, parts do not need to know what machines they are part of

Controllers, Services, Repositories should be implemented for all domain objects, implementing the following functionality:

- Get one by id
- Get all with pagination
- Create one
- Delete one
- Update one

Additional functionality can be added, such as:

- Create a customer, and add an address to it
- Create an address and add it to a customer
- Add an address to a customer

Delivery:

You should deliver a zip file with your complete code. DO NOT INCLUDE THE /TARGET DIRECTORY. Include a readme file in the project root. This file should include any instructions needed to run the application.

This readme should also include anything you were unable to get working, and what you tried to do to fix it.

Do not include names of the group members.

Grading:

F - That application does not include at least one model, repository, service, controller, or the application does not run.

E - The application implements partial functionality

D - The application implements complete functionality, but no tests

C - The application implements complete functionality, some additional functionality and includes unit tests

B - The application implements complete functionality, some additional functionality and includes both unit and integration tests

A - The application implements complete functionality, some additional functionality and includes unit, integration, and end-to-end tests