

Project Description: Backend Development of a Banking Microservice

This document provides an overview of an exam project focused on backend development for backend developers. The goal of the project is to develop the backend part of a banking microservice, utilizing various technologies and frameworks such as Spring Boot, MySQL, Liquibase, Mockito, Swagger, AOP, JaCoCo, MapStruct, Hibernate, JPA, and SLF4J.

Objective:

The main objective of this project is to create a robust backend microservice for banking operations without a frontend component. The microservice will expose a RESTful API to perform various banking transactions and manage customer accounts.

Technologies:

Spring Boot: Used for application development and streamlined setup and deployment.

MySQL: Utilized as the database for storing application data.

Liquibase: Employed for managing database migrations.

Mockito: Utilized for creating mock objects and performing unit testing.

Swagger: Used for documenting the API and generating interactive documentation.

AOP (Aspect-Oriented Programming): Implemented for cross-cutting concerns such as logging.

JaCoCo: Utilized for measuring code coverage with unit tests.

MapStruct: Employed for object mapping between different layers of the application.

Hibernate and JPA: Utilized for database interaction and object-relational mapping.

SLF4J: Employed for logging and event registration within the application.

Functionality:

The microservice will implement CRUD operations for various entities related to banking operations. This includes:

Account Management: Creating, updating, and deleting customer accounts.

Transaction Processing: Performing transactions such as transfers between accounts.

Balance Inquiry: Retrieving account balances and transaction history.

Customer Management: Managing customer information and profiles.

Security:

The application will implement security measures to restrict access to authorized users. This will include requiring an "admin" username for accessing sensitive operations like CRUD functionality and performing POST, DELETE, and PUT requests.

Data Initialization:

To facilitate testing and development, a class named "InitDataBase" will be created programmatically to add sample data for accounts, transactions, and customer profiles.

The successful completion of this project will result in a fully functional backend microservice for banking operations, built using industry-standard technologies and best practices.