Microservice implementation of a REST API.

Repository pattern used.

The case is a bank system.

User activities with HTTP requests

User can sign-up - POST

User can login - POST

User can open a debit or deposit account - POST

User can list his accounts - GET

User can view one of his account - GET

User can view an accounts transactions - GET

User can view all his transactions - GET

Authentication/Authorization

Other than login and signup functions, all other functions require token.

Token can be granted using login function.

GetServices.Login > LoginService (Authentication)

After getting the token elligibility of a token will be checked when a function wanted to be used. bankApplication.openAccount > HTTP req to getServices. > authorizationController > AuthorizationService grants UserID > DepositService.saveDeposit(UserID)

Microservice structure

DockerNetwork - “bankNetwork”

Business, getServices, db (bankNetwork)

Start algorithm

Docker-compose file creates network banknetwork

Docker-compose file creates image of postgres db and execute SQL queries to create tables and data. Lastly creates a container from it.

Docker-compose file creates image of getServices microservice using environmental variables to be used in getServices.applicationproperties. Lastly creates a container from it.

Docker-compose file creates image of business microservice using environmental variables to be used in business.applicationproperties. Lastly creates a container from it.

GetServices is Authorization/Authentication layer

Banking operations are inside of business layer.

All operations in business are being controlled by GetServices.

Requirements

JDK17

Maven

Docker

Installation

-Docker engine should be working

-Add your email variables to docker-compose file

-If docker-compose up doesnt work with error no jar file found folow the steps below

1-change docker-compose file to only create database container and docker-compose up

2-change getServices.applicationProperties with getServices.resources.application-reset reset

3-getServices mvn clean, mvn install

4-change bankApplication.applicationProperties with bankApplication.resources.application-reset reset

5-bankApplication mvn clean, mvn install

6-delete the db container

6-edit docker-compose file, and both application properties files with backup docker-compose-backup, application-reset

7-docker-compose up

Requirements

JDK17

Maven

Docker

Installation

Docker engine should be working

Add your email variables to docker-compose file

If docker-compose up doesnt work with error no jar file found folow the steps below

1-change docker-compose file to only create database container and docker-compose up

2-change getServices.applicationProperties with getServices.resources.application-reset reset

3-getServices mvn clean, mvn install

4-change bankApplication.applicationProperties with bankApplication.resources.application-reset reset

5-bankApplication mvn clean, mvn install

6-delete the db container

6-edit docker-compose file, and both application properties files with backup docker-compose-backup, application-reset

7-docker-compose up