Author: Abdallah Nazmy

EmployeeStatesMs

The employees states microservice is a maven based springboot api used to alter company employees's state based on certain event triggers sent to the api by the client.

A) EmployeeStatesMs technology/dependencies stack:

1. Spring boot V2.7.1
2. Maven V2.5.7
3. Java 11
4. Postgress database (Main application's database)
5. H2 in-memory database (For the testing environment)
6. Spring boot starter web (For REST API's)
7. Spring JPA (Entity management)
8. Hibernate validator (For validations)
9. Commons-lang3 V3.12.0 (For validations)
10. Spring boot starter test (For testing)
11. Junit Jupiter api V5.9.1 (Testing framework)
12. OpenApi V1.6.4 (Documentation specification)
13. Docker (For dockerizing the application)

B) Entities – Enums - Models

B. Entities

1. Employee

id	Long	Employee's id	
firstName	String	Employee's first name	
lastName	String	Employee's last	
Age	Integer	Employee's age	
phoneNumber	String	Employee's phone number	
startDate	LocalDate	Employee's start date	
employeeState	Enum	Current state of the employee	ADDED, IN_CHECK, APPROVED, ACTIVE
contract	Entity	Contract info of the employee(references the contract id of the employee)	

2. Contract

id	Long	Contract id	
contractType	Enum	Type of the contract	FULL_TIME, PART_TIME, FREELANCE
salary	Integer	Salary for given contract	
creationDate	LocalDate	Contract creation date	

B. Enums

1. EmployeeState

ADDED	IN_CHECK	APPROVE	ACTIVE

2. ContractType

FULL_TIME	PART_TIME	IN_CHECK

3. Event (Event that will trigger change to the employee's state)

BEGIN_CHECK	APPROVE	UNAPPROVE	ACTIVATE

B. Models

1. TriggerRequest

id	Long	Employee id that the event will effect
event	Interface	Event sent in the request will trigger execution of state machine logic to change employee state (will be sent as json in the request)

C) CLONE repository "git clone https://github.com/abdallahnazmy/State-Machine.git"

D) How to run the project from source code:

1. Create a new local database with the name "statedb", username:"postgres" and password: "postgres"
2. Run the Main application class "EmployeeStatesMsApplication" from your ide
3. Open your browser to url http://localhost:8050/state-machine-ui.html
4. FIRST you must send a post request from the contract controller to create contracts that will be assigned to employees when adding new employees
5. Send post request to the employee controller to create new employees and assign contract id (generated from step 4)to the employee data before submitting the request
6. Send get request with employee id to get the employee details (with contract details included in the response)
7. Send put request with the employee id and event name to trigger a change state in the employee's state (Event names are "beginCheck, approve, unapproved, activate") which will return the employee's details with the updated state
8. Note: a postman collection is provided with the project for a simpler visualization of how the request body would look like

E) How to run the project using docker

Note: the dockerized project will use postgress database from a docker image

1. Make sure docker is installed on your local machine
2. Open a terminal and write the command "docker pull postgres" to get a postgres image

NOTE 1: the project contains a dockerfile located at the root of the application, it will specify how to the image of the project will be generated

Note 2: the project also contains a docker-compose.yml file located at src/main/docker. This file contains configuration for the postgres image (ports exposed, environment variables,…etc). Also has the configuration for the API image itself (defining datasources, exposing ports and making the api dependent on the database image service)

3. Run mvn install to create jar file of the project (will be named employee-states-ms.jar) and found in the target folder
4. Run command (from root project path) “docker build -t employee-states-ms.jar .” to build a docker image of the project

Now we’re all set with the project docker image and the postgres image , let’s run the project from docker

5. Open a terminal with the path set to src/main/docker and run the command “docker compose up -d” to run the container from the docker-compose.yml file in detached mode.
6. The project should run successfully, head over to the url specified in C).3 and start testing or you can test from the postman collection