feature/servic... 2 branches 0 tags Go to file Add file Code

This branch is 16 commits ahead, 1 commit behind main. Contribute

carlosofpersia Merge branch 'feature/servicos_comerciante' of https://github.com/car... ... c36b6e9 9 hours ago 16 commits

.docker/database	Arquitetura inicial da aplicacao	18 hours ago
documentacao	Documentacao	9 hours ago
src	Testes Unitarios	9 hours ago
target	Testes Unitarios	9 hours ago
.classpath	Arquitetura inicial da aplicacao	18 hours ago
.gitignore	Testes Unitarios	9 hours ago
Dockerfile	Arquitetura inicial da aplicacao	18 hours ago
README.md	Update README.md	9 hours ago
docker-compose.yml	Arquitetura inicial da aplicacao	18 hours ago
pom.xml	Testes Unitarios	9 hours ago

README.md

entrevista-act

Projeto Fluxo Caixa - Docker(JAVA + Mysql)

Projeto visa atender um fluxo simples de caixa de um comerciante onde o usuário possui uma carteira digital;

Listar usuários;

Realizar lançamentos Credito/Debito;

Relatório consolidado diário geral;

Recursos necessários para o desenvolvimento do projeto:

jdk-8.0.372.07-hotspot;

apache-maven-3.9.1;

mysql-connector-java-5.1.7-bin;

Docker;

gitbash;

IntelliJ IDEA Community Edition 2023.1.1;

Padrões utilizados no projeto:

Inversão de Controle (IoC) e Injeção de Dependência (DI);

Padrão MVC;

Padrão Repository;

Padrão DTO (Mapper);

SOLID;

Teste Uniários com Mockito;

Cobertura de testes unitários com JACOCO;

```
\employee-producer\target\site\jacoco\index.html
```

About

Projeto Fluxo Caixa - JAVA

Readme
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Sobre segurança todos os endpoints solicitam **Authorization** e **x-transaction-id**, podendo assim implantar qualquer controle via intercptions (JWT|OAUTH).

Com o ambiente todo configurado, basta ir na raiz do projeto, abrir o gitbash e rodar os seguintes comandos:

mvn clean install

docker compose up --build

- Caso já tenha rodado o docker compose e tenha dado algum problema, deve-se apagar todas as configurações do docker do projeto;
- Caso não tenha o ambiente completo, bastar rodar "docker compose up --build" que deixei o projeto compilado na pasta target;
-

Para testar o projeto segue exemplos práticos via CURL:

1. Listar usuários para obter o ID

```
curl --location --request GET 'http://localhost:8080/v1/person'
--header 'Authorization: XXXX'
--header 'x-transaction-id: xxxx'
```

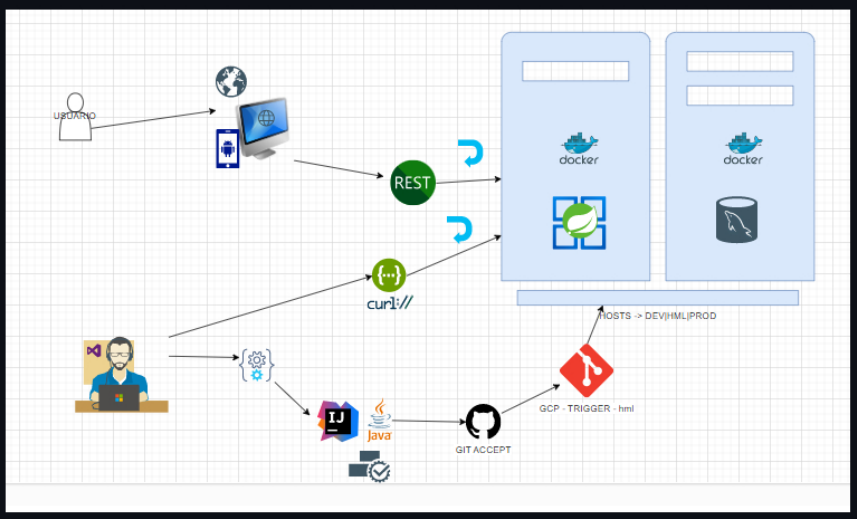
2. Lançar movimentação de valores (C|D)

```
curl --location --request POST 'http://localhost:8080/v1/release'
--header 'Authorization: XXXX'
--header 'x-transaction-id: xxxx'
--header 'Content-Type: application/json'
--data-raw '{ "person": { "id": 2 }, "cd_type_expense": "D", "vl_release": "17.50" }'
```

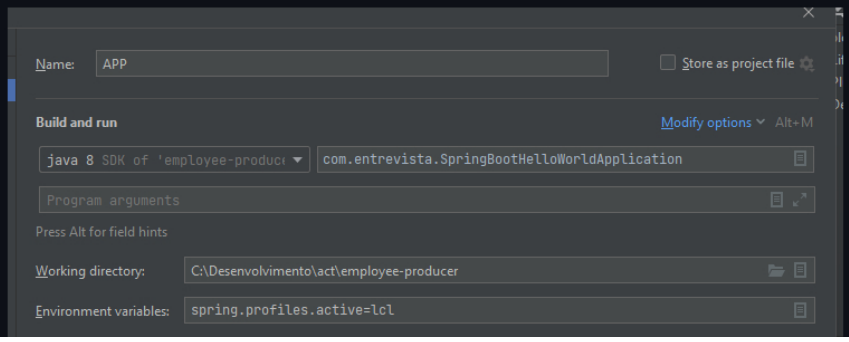
3. Apresentar saldo consolidado do dia:

```
curl --location --request GET 'http://localhost:8080/v1/statement/consolidated/2023-05-11'
--header 'Authorization: XXXX'
--header 'x-transaction-id: xxxx'
```

Desenho processo Containerização:



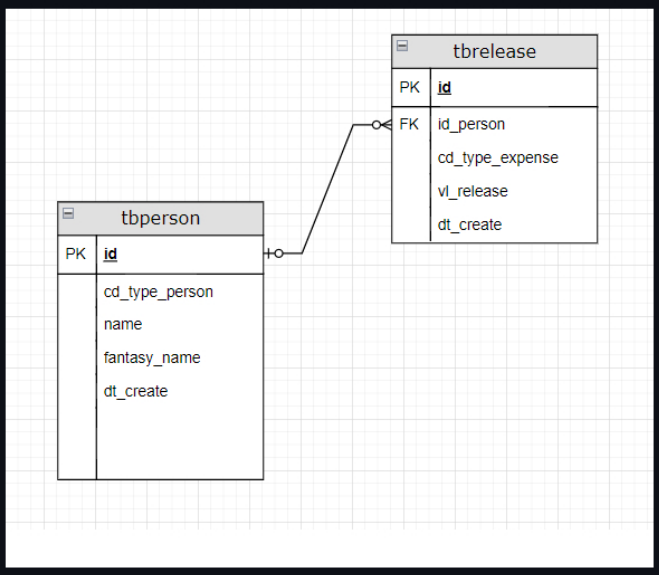
Configuração Intelij:



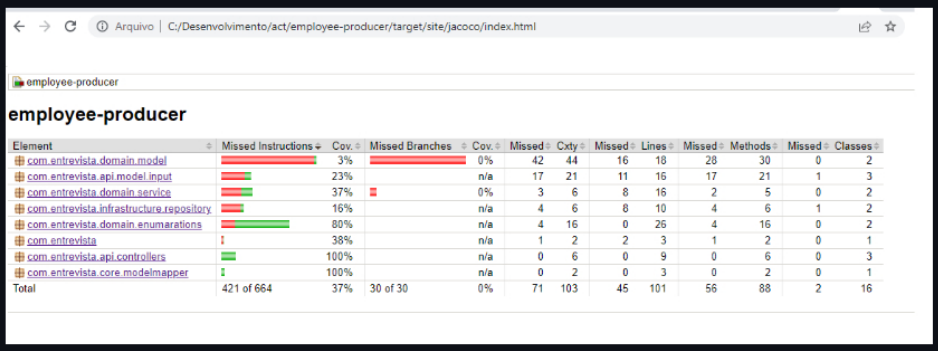
Separate variables with semicolon: VAR=value; VAR1=value1

Open run/debug tool window when started

Tabelas mapeadas:



Cobertura de Testes com JACOCO



Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.entrevista.domain.model		3%		0%	42	44	16	18	28	30	0	2
com.entrevista.api.model.input		23%		n/a	17	21	11	16	17	21	1	3
com.entrevista.domain.service		37%		0%	3	6	8	16	2	5	0	2
com.entrevista.infrastructure.repository		16%		n/a	4	6	8	10	4	6	1	2
com.entrevista.domain.enumarations		80%		n/a	4	16	0	26	4	16	0	2
com.entrevista		38%		n/a	1	2	2	3	1	2	0	1
com.entrevista.api.controllers		100%		n/a	0	6	0	9	0	6	0	3
com.entrevista.core.modelmapper		100%		n/a	0	2	0	3	0	2	0	1
Total	421 of 664	37%	30 of 30	0%	71	103	45	101	56	88	2	16