

Everton Mendonça de Jesus

(71) 98118-4851
evertonmj@gmail.com

DOCK - Manual de execução do teste prático

15 de fevereiro de 2021

Visão geral

Esta aplicação foi desenvolvida seguindo os requisitos definidos pela DOCK. A aplicação foi desenvolvida em Java utilizando o framework Spring Boot. O banco de dados utilizado é o MySQL.

Tech Stack

Sistema operacional: Ubuntu Linux 20.04

Linguagem utilizada: Java v1.8

Framework: Spring Boot v2.4.2

IDE: Spring Tool Suite 4 v4.9.0

Banco de dados: MySQL v8.0.23

Outras bibliotecas:

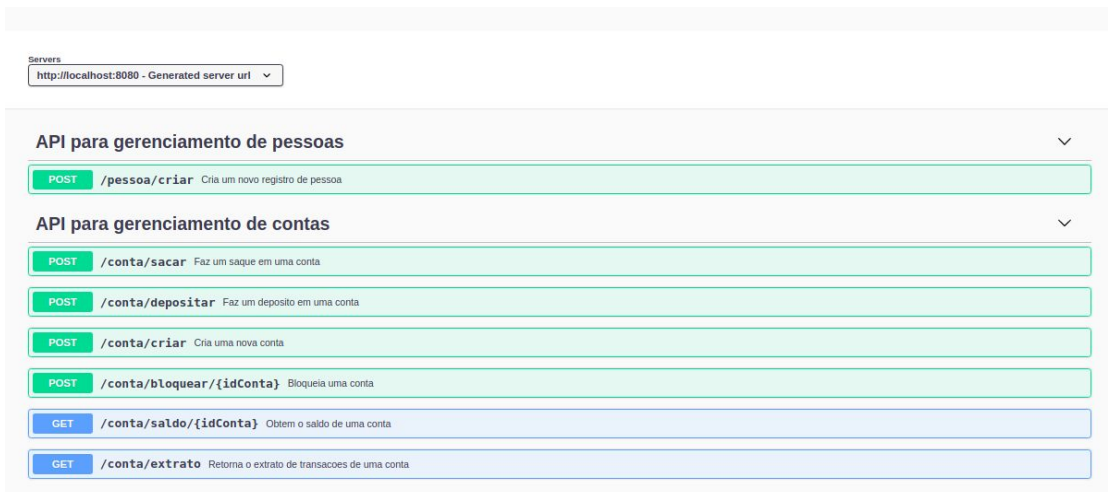
- Esta aplicação utiliza Lombok para reduzir quantidade de código boilerplate. Para instalar no ambiente, só seguir as instruções deste link:
<https://dicasdejava.com.br/como-configurar-o-lombok-no-eclipse/>

Arquivos

1. Todos os arquivos auxiliares foram disponibilizados no repositório github (<https://github.com/evertomj/desafio-dev-api-rest>) junto ao código-fonte do projeto.
 - a. No diretório "SQL" foi disponibilizado o dump do banco de dados MySQL.
 - b. No diretório "DOCS" foram disponibilizados arquivos de documentação da aplicação, sendo estes:
 - i. api-swagger.json - arquivo JSON da documentação da API da aplicação
 - ii. Processo-DOCK.postman_collection.json - Exportação dos endpoints de teste para o POSTMAN. Só importar novamente no Postman e testar a aplicação :)
 - iii. Proposta-Arquitetura-AWS.pdf - Desenho de arquitetura proposta da aplicação utilizando o ambiente da AWS
 - iv. Este arquivo de documentação da aplicação

Documentação da API

Além do arquivo DOCS/api-docs.json disponibilizado, também é possível acessar a documentação da API diretamente pelo browser em interface amigável. Para fazer isso, após o deploy da aplicação basta acessar pelo browser: **http://<URL>/swagger-ui/**.



Configurações do Banco de Dados MySQL

Como mencionado anteriormente, o dump do banco de dados com criação de tabelas e população das tabelas está no script SQL dentro do diretório "SQL".

As configurações do banco de dados devem ser feitas no arquivo de properties (src/main/resources/application.properties)

Execução da aplicação

A aplicação foi desenvolvida para ser executada diretamente ou através de container docker. Para isso existem dois shell-scripts na raiz da aplicação:

1. Diretamente através do script: **start-local.sh**
2. Através do container docker: **start-docker.sh**

Padrões de Projeto

Durante a execução utilizei os seguintes padrões:

1. Controller - Padrão de projeto utilizado para fornecer um handler centralizado de acesso aos requests da aplicação. Para organizar melhor o código, criei um controller para conter os endpoints de Pessoa e outro para os endpoints de Conta
2. Repository - Utilizado para abstrair o acesso a camada de dados tornando transparente para a camada de negócio o seu acesso. Além disso, o SpringBoot framework facilita este processo implementando as operações padrão CRUD ao estender a interface `CrudRepository<T, ID>`, agilizando o desenvolvimento.

Melhorias

1. Autenticação na API utilizando Bearer Authentication
2. Colocar algumas das regras de negócio presente no controller em uma camada Service da aplicação
3. Validação de todos os campos de entrada no request, tanto de tipo quanto de formato.