

Projeto Cliente – Teste UOL

Este repositório contém uma aplicação Spring Boot contemplando toda a parte do backend do que seria uma aplicação CRUD para clientes. Utiliza de serviços de geolocalização de IP e clima para recuperar informações e associar aos clientes cadastrados.

APIs utilizadas:

IP Vigilante: <https://ipvigilante.com> – geolocalização por IP

MetaWeather API: <https://www.metaweather.com/api/> - consulta de clima por latitude, longitude, cidades etc.

Amazon Check IP: <http://checkip.amazonaws.com/> - verificação de IP da máquina.

O projeto

O projeto foi desenvolvido no IntelliJ IDEA. Esta IDE tem uma ótima integração com o Spring e Maven, assim como inteligência de código referência. Para a geração do projeto, foi utilizada a ferramenta Spring Initializr (integrada com a IDE).

Spring Boot

Para a aplicação em si. Roda o servidor de forma que outras configurações de servidores externas não sejam necessárias, assim temos uma aplicação que rodaria em qualquer máquina.

H2 database - Banco de dados

Banco de dados "in-memory". Configuração rápida, confiável e suave integração com o Spring Data (JPA).

Para acessar o banco de dados na aplicação:

{urlBase}/projeto-cliente/banco

Usuário: adm

Senha: uol

Exemplo (local): <http://localhost:8080/projeto-cliente/banco>

Maven

Gerenciamento de dependências e build do projeto. O projeto maven foi gerado a partir da ferramenta Spring Initializr, citada anteriormente.

Swagger

Documentação de toda parte REST da aplicação. Além de documentar, é possível testar os controladores (métodos) de toda a aplicação através de uma interface amigável no contexto: /swagger-ui.html.

Iniciando

Necessário: Maven.

Para executar a aplicação há três maneiras:

1. IDE

Para iniciar direto na IDE basta baixar o repositório em sua máquina e importar em seu ambiente. Executar a aplicação como uma aplicação normal Java pela classe principal.

2. Maven

Com o repositório baixado, executar o comando: mvn spring-boot:run

3. Deploy do WAR em servidor

Se preferir rodar em um servidor (para fins de Produção por exemplo), o Maven gera o .war do projeto. Para isso, executar o seguinte comando: mvn clean install O .war da aplicação estará disponível em /target (raiz do projeto).

Acesso

{urlBase}/projeto-cliente

Exemplo (local): <http://localhost:8080/projeto-cliente>

Testes dos métodos REST

Para documentação e teste dos métodos backend, nesta aplicação, foi implementado o framework Swagger. Para ajudar, utiliza-se de uma interface gráfica leve e intuitiva.

Acesso:

{urlBase}/projeto-cliente/swagger-ui.html

Exemplo (local): <http://localhost:8080/projeto-cliente/swagger-ui.html>

Documentação completa

Documentação da API: Se referir ao documento API_Docs_ProjetoCliente

Especificação do projeto: Se referir ao documento TEST_PLATCORP_V1

Toda a documentação se encontra na pasta docs do projeto.