**Projeto Cliente (Teste UOL)**

**Documentação da API**

Para auxiliar na documentação e testes da API, a aplicação utiliza-se da ferramenta Swagger.

Acesse: /*projeto-cliente/swagger-ui.html* ou acesse o link na home da aplicação.

**Métodos Cliente**

**Criar um cliente**

**URL: */cliente/inserir?nomeCliente={nome}***

**Parâmetros**

***nomeCliente*: O nome do cliente que deseja inserir na base.**

**Exemplos**

* ***/cliente/inserir?nomeCliente=Felipe***
* ***/cliente/inserir?nomeCliente=Renato%20Pereira***

**Resposta**

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descrição** |
| **id** | **long** | **ID do cliente inserido.** |
| **nome** | **string** | **Nome do cliente.** |
| **clienteGeo** |  |  |
| **id** | **long** | **ID dos dados de geolocalização.** |
| **cidade** | **string** | **Cidade recuperada no cadastro do cliente.** |
| **temperaturaMinima** | **float** | **Temperatura mínima registrada na requisição.** |
| **temperaturaMaxima** | **float** | **Temperatura máxima registrada na requisição.** |
| **temperaturaAtual** | **float** | **Temperatura atual registrada na requisição.** |

**Erros**

***400 (Bad Request)* – Nome do cliente inválido (nulo ou vazio).**

***417 (Expectation Failed)* – Problemas nos serviços de geolocalização.**

**Alterar um cliente**

**URL: */cliente/atualizar?idCliente={id}&novoNome={nome}***

**Parâmetros**

***idCliente*: ID do cliente que deseja atualizar.**

***nomeCliente*: O nome atualizado do cliente.**

**Exemplos**

* ***/cliente/atualizar?idCliente=1&novoNome=Felipe***
* ***/cliente/atualizar?idCliente=2&novoNome=Joselino***

**Resposta**

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descrição** |
| **id** | **long** | **ID do cliente inserido.** |
| **nome** | **string** | **Nome do cliente.** |
| **clienteGeo** |  |  |
| **id** | **long** | **ID dos dados de geolocalização.** |
| **cidade** | **string** | **Cidade recuperada no cadastro do cliente.** |
| **temperaturaMinima** | **float** | **Temperatura mínima registrada na requisição.** |
| **temperaturaMaxima** | **float** | **Temperatura máxima registrada na requisição.** |
| **temperaturaAtual** | **float** | **Temperatura atual registrada na requisição.** |

**Erros**

***400 (Bad Request)* – Nome do cliente inválido (nulo ou vazio).**

***404 (Not Found)* – Não encontrado um cliente com o ID informado.**

**Consultar um cliente (GET)**

**URL: */cliente/{idCliente}***

**Parâmetros**

***idCliente*: ID do cliente que deseja consultar.**

**Exemplos**

* ***/cliente/1***
* ***/cliente/2***

**Resposta**

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descrição** |
| **id** | **long** | **ID do cliente inserido.** |
| **nome** | **string** | **Nome do cliente.** |
| **clienteGeo** |  |  |
| **id** | **long** | **ID dos dados de geolocalização.** |
| **cidade** | **string** | **Cidade recuperada no cadastro do cliente.** |
| **temperaturaMinima** | **float** | **Temperatura mínima registrada na requisição.** |
| **temperaturaMaxima** | **float** | **Temperatura máxima registrada na requisição.** |
| **temperaturaAtual** | **float** | **Temperatura atual registrada na requisição.** |

**Erros**

***404 (Not Found)* –** Não foi encontrado um cliente com o ID informado.

**Remover um cliente (GET)**

**URL: */cliente/deletar/{idCliente}***

**Parâmetros**

***idCliente*:** ID do cliente que deseja remover.

**Exemplos**

* ***/cliente/deletar/1***
* ***/cliente/deletar/2***

**Resposta**

***200 (OK)* –** O cliente foi removido.

**Erros**

***404 (Not Found)* –** Não foi encontrado um cliente com o ID informado.

V1.:

Premissas:  
- Serão avaliadas todas as características da solução apresentada  
- Caso não saiba por onde começar, procure algum tutorial de criação de CRUD Rest com Spring Boot  
- Caso tenha dificuldades ou não saiba como continuar, entregue até onde conseguir e explique quais problemas teve  
- Usamos Java em nossos sistemas  
- Considere apenas a parte backend. Não é necessário desenvolver tela/formulário, apenas os endpoints REST  
- Um cliente é composto apenas por nome e idade  
- API aberta de geolocalização por IP https://www.ipvigilante.com/  
- API aberta de clima por geolocalização https://www.metaweather.com/api/  
- Quando executar a busca de clima por geolocalização, caso não exista a cidade especifica de origem, utilize o resultado mais próximo.  
  
Sua tarefa é desenvolver os serviços REST abaixo:  
- Criar um Cliente  
- Alterar um Cliente  
- Consultar um Cliente por id  
- Listar todos os Clientes salvos  
- Remover Cliente por id  
  
Ao criar um cliente, apenas para fins estatísticos e históricos, busque qual a localização geográfica de quem executou a requisição, usando o IP de origem. Com a localização geográfica, consulte qual é a temperatura máxima e mínima do dia da requisição de criação no local do IP de origem. Salve essa informação e a associe ao cliente resultado da requisição de origem.  
  
Tenha em mente que a consulta de Cliente por ID será altamente requisitada.  
  
Junto com o código é necessário entregar uma documentação que contenha os itens abaixo:  
- Como usar os serviços  
- Quais ferramentas foram usadas (e porque essas foram as escolhidas)  
- Qualquer infraestrutura adicional necessária para executar, testar, empacotar e entregar seu projeto  
- Como executar, testar, empacotar e entregar o seu projeto  
- Instruções para como montar o ambiente de produção onde seus serviços devem ser executados (preferencialmente citando ferramentas e processos)