

DESCRIPTIVO DO DESAFIO

objetivo: Criar um projeto Spring Boot para permitir o cadastramento, consulta, exclusão, lista, pesquisa e alteração de Clientes seguindo a modelagem fornecida.

Este documento é uma versão simplificada de apresentação da resolução do desafio e, nele, você encontrará:


1. Funcionamento
2. Organização dos Pacotes
3. Observações Gerais

Você poderá acessar o código da resolução através do link



https://github.com/patrick-tobias/022023_d-muralis_java-spring

FUNCIONAMENTO

Nesta seção, você encontrará capturas de telas, como a aplicação desenvolvida responde às requisições HTTP.

Inicialização e População dos dados	<div><pre> :: Spring Boot :: (v3.2.0) 2023-12-03T12:14:33.416-03:00 INFO 5890 --- [main] b.c.d.jspring.JspringApplication : Starting JspringApplication using Java 2023-12-03T12:14:33.422-03:00 INFO 5890 --- [main] b.c.d.jspring.JspringApplication : No active profile set, falling back to 2023-12-03T12:14:36.018-03:00 INFO 5890 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (ht 2023-12-03T12:14:36.055-03:00 INFO 5890 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat] 2023-12-03T12:14:36.056-03:00 INFO 5890 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat 2023-12-03T12:14:36.323-03:00 INFO 5890 --- [main] o.a.c.c.C.[Tomcat].[/] : Initializing Spring embedded WebAppli 2023-12-03T12:14:36.327-03:00 INFO 5890 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initializ 2023-12-03T12:14:37.273-03:00 INFO 5890 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) wi 2023-12-03T12:14:37.290-03:00 INFO 5890 --- [main] b.c.d.jspring.JspringApplication : Started JspringApplication in 5.364 s Bem-vindo(a) ***** Digite uma quantidade de contatos para popular o banco de dados 5 Banco populado com sucesso!</pre></div>
HTTP GET (/clientes)	<div><div>GET</div><div>http://localhost:8080/clientes</div></div>
saída da requisição GET(/clientes)	<div><div>Body Cookies Headers (5) Test Results</div><div>Status: 200 OK Time: 13 ms Size: 802 B Save Response</div><div><div>Pretty Raw Preview Visualize JSON</div><pre> 1 [2 { 3 "id": 2, 4 "nome": "Charlene Beatty", 5 "dataCadastro": "03/12/2023", 6 "contato": { 7 "id": 1, 8 "tipo": "email", 9 "texto": "watsica@fatec.sp.gov.br" 10 } 11 }, 12 { 13 "id": 4,</pre></div></div>

<p>HTTP POST(/clientes)</p>	<div> <div>POST</div> <div>http://localhost:8080/clientes</div> </div> <div> <div>Params</div> <div>Authorization</div> <div>Headers (8)</div> <div>Body</div> <div>Pre-request Script</div> <div>Tests</div> <div>Settings</div> </div> <div> <div>none</div> <div>form-data</div> <div>x-www-form-urlencoded</div> <div>raw</div> <div>binary</div> <div>JSON</div> </div> <pre> 1 "id": 0, 2 "nome": "João da Silva Santos", 3 "dataCadastro": "03/12/2023", 4 "contato": { 5 "id": 0, 6 "tipo": "email", 7 "texto": "silvasantos@fatec.sp.gov.br" 8 } 9 10 </pre>
<p>saída da requisição POST(/clientes)</p>	<div> <div>Body</div> <div>Cookies</div> <div>Headers (4)</div> <div>Test Results</div> <div>Status: 200 OK</div> <div>Time: 169 ms</div> <div>Size: 123 B</div> <div>Save Response</div> </div> <div> <div>Pretty</div> <div>Raw</div> <div>Preview</div> <div>Visualize</div> <div>Text</div> </div>
<p>HTTP PUT(/clientes)</p>	<div> <div>PUT</div> <div>http://localhost:8080/clientes</div> </div> <div> <div>Params</div> <div>Authorization</div> <div>Headers (8)</div> <div>Body</div> <div>Pre-request Script</div> <div>Tests</div> <div>Settings</div> </div> <div> <div>none</div> <div>form-data</div> <div>x-www-form-urlencoded</div> <div>raw</div> <div>binary</div> <div>JSON</div> </div> <pre> 1 "id": 11, 2 "nome": "João da Silva Santos Oliveira", 3 "dataCadastro": "03/12/2023", 4 "contato": { 5 "id": 0, 6 "tipo": "email", 7 "texto": "silvasantosoliveira@fatec.sp.gov.br" 8 } 9 10 </pre>
<p>saída da requisição PUT(/clientes)</p>	<div> <div>Body</div> <div>Cookies</div> <div>Headers (4)</div> <div>Test Results</div> <div>Status: 200 OK</div> <div>Time: 15 ms</div> <div>Size: 123 B</div> <div>Save Response</div> </div> <div> <div>Pretty</div> <div>Raw</div> <div>Preview</div> <div>Visualize</div> <div>Text</div> </div>

<p>HTTP DELETE</p> <p>(/clientes/id)</p>	 <p>The image shows the Postman interface for a DELETE request. The top bar displays the method 'DELETE' and the URL 'http://localhost:8080/clientes/2'. Below the bar are tabs for 'Params', 'Authorization', 'Headers (6)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Body' tab is selected, showing radio buttons for 'none', 'form-data', 'x-www-form-urlencoded', 'raw', and 'binary'. The 'none' option is selected.</p>
<p>saída da requisição</p> <p>DELETE</p> <p>(/clientes/id)</p>	 <p>The image shows the Postman interface for the response of the DELETE request. The top bar displays the status '200 OK', time '19 ms', and size '123 B'. Below the bar are tabs for 'Body', 'Cookies', 'Headers (4)', and 'Test Results'. The 'Body' tab is selected, showing buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'Text'. The 'Pretty' button is selected.</p>

observação: Ferramenta utilizada - **Postman**

ORGANIZAÇÃO DOS PACOTES

Nesta seção, você encontrará a estratégia de organização das pastas e subpastas da aplicação resolução do desafio.

Estrutura de organização, esquema de pastas

jspring [jspring master]

src/main/java

src/main/resources

src/test/java

src/test/resources

JRE System Library [Java SE 21.0.1 [21.0.1]]

Maven Dependencies

src

main

java

br

com

desafiomuralis

jspring

comum

controle

dto

ClienteDTOEntrada.java

ClienteDTOSaida.java

ContatoDTO.java

ClienteController.java

modelo

casodeuso

AdicionarCliente.java

AtualizarCliente.java

PesquisarClientes.java

RemoverCliente.java

negocio

Cliente.java

Contato.java

persistencia

banco

Banco.java

Clientes.java

ClientesLocalMemo.java

JspringApplication.java

resources

test

target

HELP.md

mvnw

mvnw.cmd

pom.xml

OBSERVAÇÕES GERAIS

Nesta seção, você encontrará o raciocínio executado para a tomada de decisão durante a realização do desafio.

Banco de Dados Simulado

A iniciativa de incluir um **banco de dados simulado** surgiu da dúvida gerada ao ler o desafio. O desafio não incluía conexão e persistência em banco de dados.

Através de Listas, foi possível simular um banco de dados e suas operações CRUD.

Testes de Funcionamento

A resolução do desafio sugeriu a construção de uma API Restful. Com a finalidade de simular requisições HTTP, utilizei a ferramenta **Postman**.

Interface Repositório

Utilizar interfaces na classe que implementam e simulam o “Banco de Dados” foi uma decisão tomada em teoria, pela facilidade de substituição do Banco de Dados e manutenção da troca.

Métodos separados em Casos de Uso

Utilizar classes que representam um caso de uso do CRUD foi uma decisão tomada com a preocupação em não sobrecarregar classes de negócio com métodos que fogem de seu controle. Assim, Cliente é responsável por gerenciar seus próprios, e somente os seus próprios, atributos.