Aplicação Consumidora Frontend CRUD Produto Parte 2

Prof. Pedro Toledo

Agenda

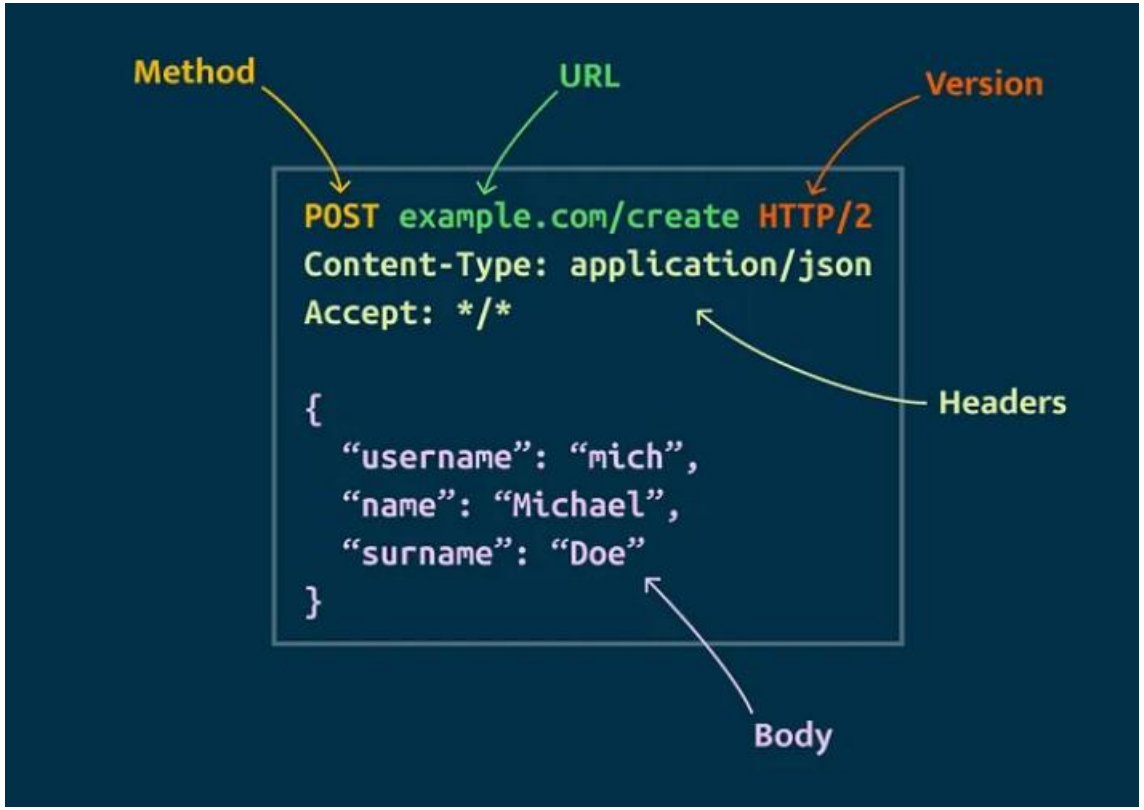
- Compreender a anatomia do “HTTP REQUEST”
- Implementar uma aplicação front-end para cadastrar um produto
- Implementar uma aplicação Front-end listar todos os produtos
- Desafios

Anatomia do “HTTP REQUEST”

- Uma requisição HTTP é composta por cinco partes principais:
  - Método
  - Path
  - versão do HTTP
  - cabeçalhos (headers)
  - corpo (body)

- Cada parte desempenha um papel crucial na definição e no processamento da requisição.

Anatomia do HTTP POST REQUEST

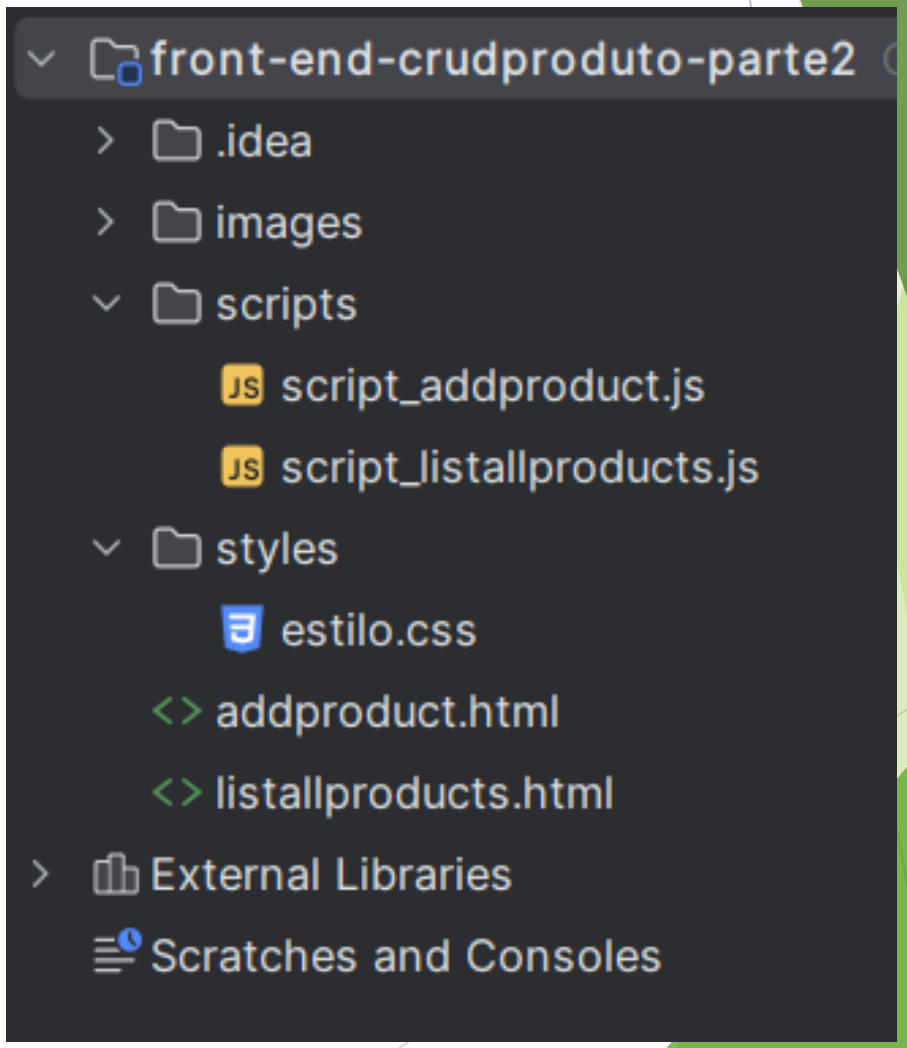


Anatomia do HTTP REQUEST

```
POST "localhost:8080/produto/add" HTTP/1.1
User-Agent: Chrome/130.0.0.0
Content-Type: application/json
Content-Length: 32
{
    "descricao": "CPU i7 10 gen",
    "preco": 1990.0
}
```

Aplicação front-end: exemplo HTML, CSS e JS

- Usando o IntelliJ (ou outra IDE), crie um projeto com a estrutara ilustrada pela figura ao lado. Observe os nomes dos diretórios e dos arquivos.
- Esse projeto precisa ser colocado em diretório de maneira que fique disponível pelo serviço HTTP.
- Nas máquinas do laboratório, pode ser utilizando, por exemplo, através de um navegador (Chrome, Edge ou outro)



Aplicação front-end: cadastrar um produto

- Crie o arquivo “addproduct.html” como ilustrado na figura ao lado.
- Atenção para seu código e para sua localização dentro do projeto.

```
<!doctype html>
<html lang="pt-br">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport"
              content="width=device-width, user-scalable=no, initial-scale=1.0,
                                          maximum-scale=1.0, minimum-scale=1.0">
        <meta http-equiv="X-UA-Compatible" content="ie=edge">
        <title>App Web Consumidor API CRUD Produtos</title>
        <link rel="stylesheet" href="styles/estilo.css">
        <script type="text/javascript" src="scripts/script_addproduct.js"></script>
    </head>
    <body>
        <h1>Consumidor API Add Product</h1>
        <form class="formadd" id="formadd" name="formadd">
            <div>
                <label>Descrição</label>
                <input type="text" name="descricao" id="descricao">
            </div>
            <div>
                <label>Preço</label>
                <input type="text" name="preco" id="preco">
            </div>
            <div>
                <button type="submit"  onclick="addProduct()">Cadastrar</button>
            </div>
        </form>
    </body>
</html>
```

Aplicação front-end: cadastrar um produto

- Crie o arquivo “script_addproducts.js” como ilustrado na figura ao lado.
- Atenção para seu código e para sua localização dentro do projeto.
- Repare como a função FETCH utiliza o endpoint.

```
function clearTextFields(){
    document.getElementById("descricao").value = "";
    document.getElementById("preco").value = "";
}

async function addProduct(){
    const formE1 = document.querySelector("#formadd");
    const formData = new FormData(formE1);
    const product = Object.fromEntries(formData);
    const url = "http://localhost:8080/produto/add";
    const option = {
        method: 'POST',
        headers:{'Content-Type': 'application/json'},
        body: JSON.stringify(product)
    }
    const result = await fetch(url, option);
    if(result.status === 201){
        clearTextFields();
        alert('Cadastrado com sucesso');
    }
    else{
        alert('Erro ao cadastrar');
    }
}
```

Aplicação front-end: listar todos os produtos

- Crie o arquivo “listallproducts.html” como ilustrado na figura ao lado.
- Atenção para seu código e para sua localização dentro do projeto.

```
<!doctype html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
          content="width=device-width, user-scalable=no, initial-scale=1.0,
                                              maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>App Web Consumidor API CRUD Produtos</title>
    <link rel="stylesheet" href="styles/estilo.css">
    <script type="text/javascript" src="scripts/script_listallproducts.js"></script>
</head>
<body>
    <div class="mydiv-1">
        <div id="loading">
            <span>Buscando produtos...</span>
        </div>
        <h1>Consumidor API ListAll Products</h1>
        <table>
            <tbody id="products"></tbody>
        </table>
    </div>
</body>
</html>
```

Aplicação front-end: listar todos os produtos

- Crie o arquivo “script_listallproducts.js” como ilustrado na figura ao lado.
- Atenção para (PRIMEIRA PARTE) seu código e para sua localização dentro do projeto.

```
function clearLoading(){
    document.getElementById("loading").style.display = "none";
}

function showProducts(products){
    let tab = ` <thead>
        <th>id</th>
        <th>Descrição</th>
        <th>Preço</th>
        <th>Editar</th>
        <th>Remover</th>
    </thead>`;

    for(let product of products){
        tab += `
            <tr>
                <td>${product.id}</td>
                <td>${product.descricao}</td>
                <td>${product.preco}</td>
                <td><img src="images/edit01.png" width="20" height="15"></td>
                <td><img src="images/trash01.png" width="20" height="15"></td>
            </tr>
        `;
    }
    document.getElementById("products").innerHTML = tab;
}
```

Aplicação front-end: listar todos os produtos

- Crie o arquivo “script_listallproducts.js” como ilustrado na figura ao lado.
- Atenção para (SEGUNDA PARTE) seu código e para sua localização dentro do projeto.

```
27
28  async function listAllProducts(){
29      const url = "http://localhost:8080/produto/listall";
30      const dados = await fetch(url, {method: "GET"});
31      if(dados.status === 200){
32          const products = await dados.json();
33          if(products){
34              clearLoading();
35          }
36          showProducts(products);
37      }
38  }
39
40  listAllProducts();
41
```

Executar

- Executar a aplicação back-end.
- Executar a aplicação front-end.

Desafio

1. Qual é a anatomia do HTTP RESPONSE?
2. Capacitar o projeto front-end a utilizar as demais operações disponíveis no back-end.

Referências

- https://medium.com/@httpiness/getting-started-with-http-for-complete-beginners-146913d477fb