Uso do fetch no JavaScript para Consumo de APIs

Aprendendo a buscar dados com JavaScript e a API ViaCEP

Prof: Me. Guilherme Villaca

O que é AJAX?

AJAX (Asynchronous JavaScript and XML)

Técnica para carregar dados de um servidor sem recarregar a página

Utiliza XMLHttpRequest ou fetch para comunicação com APIs Hoje

em dia, fetch e async/await são preferidos para requisições AJAX

O que é uma API?

API (Application Programming Interface) é uma ponte entre sistemas No

contexto web, é um serviço que retorna dados no formato JSON ou XML

Exemplo: ViaCEP retorna informações de endereço ao fornecer um CEP

Introdução ao fetch

- fetch() é uma função moderna para fazer requisições HTTP em JavaScript
- Usa Promises para lidar com respostas assíncronas
- Sintaxe básica:

```
fetch('https://viacep.com.br/ws/01001000/json/')
 .then(response => response.json())
 .then(data => console.log(data))
 .catch(error => console.error('Erro:', error));
```

- O método .json() converte a resposta para um objeto JavaScript

fetch com async/await

- async/await simplifica o uso de Promises, tornando o código mais legível.

- Exemplo:

```
async function buscarCEP(cep) {
 try {
 let response = await fetch(`https://viacep.com.br/ws/${cep}/json/`);
 let data = await response.json();
 console.log(data);
 } catch (error) {
 console.error("Erro ao buscar CEP:", error);
 }
}
buscarCEP('01001000');
```

Integrando com a Página

- Criamos um input para o usuário inserir o CEP
- Um botão para acionar a busca
- Uma div para exibir os resultados

```
<input type="text" id="cep" placeholder="Digite o CEP">

<button onclick="buscarCEP()">Buscar</button> <div

id="resultado"></div>
```

Integrando com a Página

```
async function buscarCEP() {
 let cep = document.getElementById("cep").value;
 let response = await fetch(`https://viacep.com.br/ws/${cep}/json/`);
let data = await response.json();
 document.getElementById("resultado").innerHTML = `
<p><strong>Endereço:</strong> ${data.logradouro}</p>
<p><strong>Bairro:</strong> ${data.bairro}</p>
```

```
<p><strong>Cidade:</strong> ${data.localidade} - ${data.uf}</p>  `;
}
```

Tratamento de Erros

- Validar se o CEP tem 8 dígitos
- Exibir mensagens caso o CEP não seja encontrado

```
async function buscarCEP() {
 let cep = document.getElementById("cep").value;
if (cep.length !== 8 || isNaN(cep)) {
 alert("CEP inválido!");
 return;
 }
 try {


 } catch (error) {
 console.error("Erro ao buscar CEP:", error);  }
}
```

```
let response = await
fetch(`https://viacep.com.br/ws/${cep}/json/`);
let data = await response.json();
if (data.erro) {
 alert("CEP não encontrado!");
 return;
}
document.getElementById("resultado").innerHTML = `
<p><strong>Endereço:</strong> ${data.logradouro}</p>
<p><strong>Bairro:</strong> ${data.bairro}</p>
<p><strong>Cidade:</strong> ${data.localidade} -
${data.uf}</p>
 `;
```

Conclusão

- fetch facilita o consumo de APIs com JavaScript
- async/await torna o código mais limpo e legível
- Tratamento de erros melhora a experiência do usuário ● A API ViaCEP é uma forma prática de aprender sobre requisições HTTP