

Assunto: Implementação do Processamento Assíncrono de Dados com XMLHttpRequest

Olá Equipe,

Espero que todos estejam bem! Como parte do nosso novo projeto de interoperabilidade web, vamos implementar o processamento assíncrono de dados utilizando o objeto XMLHttpRequest para troca de requisições na página principal do e-commerce. Abaixo, descrevo o passo a passo de como essa requisição será realizada:

1) Ação do Usuário:

O processo inicia quando o usuário realiza uma ação na interface (exemplo: clicar em um botão para buscar informações do produto).

2) Criação do Objeto XMLHttpRequest:

- No JavaScript, criamos uma instância do objeto XMLHttpRequest para gerenciar a requisição assíncrona.

```
var xhr = new XMLHttpRequest();
```

3) Configuração da Requisição:

Configuramos a requisição com o método HTTP (GET, POST, etc.) e a URL da API que queremos acessar:

```
xhr.open('GET', 'https://api.ecommerce.com/produtos', true);
```

O terceiro parâmetro (true) indica que a requisição será assíncrona.

4) Envio da Requisição:

Enviamos a requisição para o servidor:

```
xhr.send();
```

5) Tratamento da Resposta:

Definimos uma função de callback que será executada quando a requisição for concluída:

```
xhr.onload = function() {  
    if (xhr.status >= 200 && xhr.status < 300) {  
        // Requisição bem-sucedida  
        var data = JSON.parse(xhr.responseText);  
        console.log(data);  
        // Aqui você pode manipular os dados recebidos, como  
        atualizar a UI  
    }  
}
```

```
    } else {  
        // Tratamento de erros  
        console.error('Erro na requisição:', xhr.statusText);  
    }  
};
```

6) Tratamento de Erros:

Podemos também adicionar um tratamento para erros de rede:

```
xhr.onerror = function() {  
    console.error('Erro de rede ao tentar realizar a  
    requisição.');
```

7) Atualização da Interface do Usuário:

Com os dados recebidos, atualizamos a interface do usuário de forma dinâmica, sem precisar recarregar a página.

Esse ciclo permite que a página continue responsiva enquanto aguarda a resposta da API, proporcionando uma melhor experiência ao usuário.

```
function displayProducts(products) {  
    var productList = document.getElementById('product-list');  
    productList.innerHTML = ''; // Limpa a lista atual  
    products.forEach(function(product) {  
        var item = document.createElement('li');  
        item.textContent = product.name;  
        productList.appendChild(item);  
    });  
}
```

Qualquer dúvida, estou à disposição para esclarecimentos.

Abraços,

Davi Faian

Desenvolvimento Web - Back End - UVV