Assunto: Explicação sobre o processamento assíncrono com XMLHttpRequest

Time, bom dia!

Vamos realizar um projeto de implementação do XMLHttpRequest, esse objeto permite que o navegador faça requisições HTTP e se comunique de forma assíncrona com o servidor usando JavaScript. Utilizando o XMLHttpRequest não precisamos recarregar a página para buscar informações do servidor, tornando as páginas web mais dinâmicas e melhorando a experiência do usuário.

Segue o passo a passo como a requisição é realizada pelo **XMLHttpRequest:**

- Inicia-se o ciclo com a ação do usuário, por exemplo, clicando em um botão ou link que executa uma função JavaScript.

- A seguir, em JavaScript, um objeto XMLHttpRequest é criado para gerenciamento da requisição. Para a implementação é utilizado o construtor new XMLHttpRequest():

Exemplo : var xhttp = new XMLHttpRequest():

- Podemos utilizar o método *open()* para configurar a requisição. Nesse momento é definido o tipo da requisição, a URL do servidor e se a requisição será assíncrona ou síncrona (true ou false).

Exemplo: xhttp.*open*(“GET”, “https://api.exemplo.com/”, true);

- Será definida uma função de callback para executar quando a resposta do servidor chegar, poderá ser feita utilizando as propriedades *onreadystatechange* ou *onload* do objeto XMLHttpRequest. Essa função irá verificar o estado da requisição (readyState) e o status da resposta (status), para tratar os dados recebidos.

- A requisição será enviada para o servidor usando o método *send().*

Exemplo: xhttp.*send()*

- Em seguida, a requisição será processada pelo servidor, a página continuará funcionando normalmente. Isso é possível devido a configuração assíncrona da requisição (*true* no método *open()*).

- No momento em que a resposta do servidor chegar, será acionada a função de callback definida anteriormente. Podemos utilizar as propriedades *responseText* ou *responseXML* para que a resposta seja acessada.

Exemplos:
var resposta = xhttp.*responseText;* - Para resposta em formato de texto.
var respostaXML = xhttp.*responseXML;* - Para resposta em formato XML.

Por fim, a resposta será tratada e utilizada para a atualização da interface do usuário. Utilizando a função abaixo como exemplo, os dados recebidos podem ser utilizados em uma tabela ou listagem de produtos.

Exemplo:

```
function atualizarInterface(produtos) {
    var listaProdutos = document.getElementById("lista-produtos");
    listaProdutos.innerHTML = ""; // limpa a lista antes de adicionar novos itens

    produtos.forEach(function(produto) {
        var item = document.createElement("li");
        item.textContent = produto.nome + " - R$ " + produto.preco;
        listaProdutos.appendChild(item);
    });
}
```

Então recapitulando todo ciclo:

1. **Ação do Usuário:** Interação do usuário com a interface.
2. **Criação do XMLHttpRequest:** Criação do objeto XMLHttpRequest.
3. **Configuração da Requisição:** Utilização do método *open()* para configuração da requisição.
4. **Definição do Callback:** É definida a função de callback para tratamento da resposta.
5. **Envio da Requisição:** Será enviada a requisição ao servidor utilizando o *send()*.
6. **Processamento Assíncrono:** Enquanto a requisição está sendo processada, a página continuará funcionando.
7. **Recebimento da Resposta:** O servidor recebe e trata a resposta.
8. **Atualização da Interface:** A interface do usuário será atualizada com os dados processados.

Com isso nosso projeto estará finalizado, em breve encaminharemos outro e-mail com a explicação de como será feita a implementação de forma colaborativa, utilizando a plataforma Github.

Qualquer dúvida estamos à disposição.

Att,
Mônica Rossi e Nicole Rocha.