

Equipe, bom dia!

Como primeiro desafio em equipe gostaria que a nossa comunicação fosse a clara, pois acredito que a comunicação é a chave para o sucesso deste projeto.

Para que fique claro a todos o que pretendo fazer na parte de processamento de dados gostaria de deixar uma breve explicação do que pensei para o projeto.

No projeto em questão, vamos utilizar o objeto XMLHttpRequest(XHR). Para os que não tem um conhecimento tão claro, é um objeto do JavaScript que permite que a página envie e receba os dados do servidor sem precisar a página inteira, ou seja, as páginas serão atualizadas dinamicamente, trazendo uma experiência mais fluida e rápida para o usuário.

O ciclo acontece da seguinte forma: Primeiro o usuário realiza uma ação na página, como clicar em um botão qualquer.

Essa ação irá disparar o código JavaScript que irá criar a requisição assíncrona.

Então o JavaScript irá criar um novo objeto XHR que será responsável por enviar a requisição para o servidor e receber a resposta.

Mas antes de enviar, precisamos dar algumas instruções para o XHR, para que ele receba de forma mais eficaz e limpa. Essas instruções seriam o método (GET,POST,PUT etc...), a URL da API que queremos acessar e a parte da requisição assíncrona, que quase sempre será true.

Nesta etapa também podemos definir headers, como Authorization ou Content-Type, são informações adicionais, porém que fazem toda a diferença na requisição.

Então, iremos definir o que irá acontecer quando a resposta chegar, como a requisição é assíncrona, o código não precisa esperar o servidor responder, a página vai continuar funcionando normalmente.

Então neste caso iremos definir funções chamadas event handlers, ou seja, as funções que o JavaScript executa automaticamente quando certo evento acontece.

Depois de definir e configurar os event handlers, enviamos a requisição, então o navegador envia os dados para o servidor e continua executando tarefas normalmente, pois nossa requisição é assíncrona, ou seja, a página não trava.

Quando o servidor processar a requisição irá enviar os dados de volta como uma resposta, neste caso acontecerá o seguinte:

O XHR dispara o evento onload ou onerror dependendo do sucesso ou da falha

O JavaScript pega os dados, normalmente em formato JSON que são mais fáceis e leves de trabalhar e os processa.

Por fim atualizamos a página dinamicamente, ou seja, sem precisar recarregar a página inteira, dando a fluidez que queríamos.

É bom lembrar que também poderíamos cancelar requisições desnecessárias caso o usuário fizesse uma ação que a tornasse desnecessária, usando o `xhr.abort`

Desta forma, iremos entregar uma experiência mais rápida para o usuário, sem recarregar a página, também terá uma redução baixa de consumo de dados pois apenas partes da página são atualizadas. Além de ter o controle sobre os cancelamentos e tratativas de erros e requisições desnecessárias.

Então Equipe, acredito que este é o caminho completo para implementarmos a requisição assíncrona com XHR, desta forma o projeto será entregue de forma organizada e mantendo uma alta qualidade, qualquer dúvida estou a disposição para discutirmos sobre o projeto.

Abraços,

Gabriel dos Santos Faria

Arquiteto de Interoperabilidade Web