

Roteiro Didático de Aula
Aula 08 – JavaScript
30/09/2020

Assuntos que serão abordados:

AJAX

Construtor

Métodos

Referências

AJAX

AJAX é o acrônimo para JavaScript assíncrono + XML. Não é exatamente uma tecnologia nova, mas um termo empregado em 2005 por Jesse James Garrett para descrever uma nova forma de utilizar em conjunto algumas tecnologias, incluindo HTML ou XHTML, CSS, JavaScript, DOM, XML, XSLT, e o mais importante: objeto XMLHttpRequest.

Quando essas tecnologias são combinadas no modelo AJAX, as aplicações web que a utilizam são capazes de fazer rapidamente atualizações incrementais para a interface do usuário sem recarregar a página inteira do navegador. Isso torna a aplicação mais rápida e sensível às ações do usuário.

Embora a letra X em AJAX corresponda ao XML, atualmente o JSON é mais utilizado que o XML devido às suas vantagens, como ser mais leve e ser parte do JavaScript. Ambos (JSON e XML) são utilizados para obter informações do pacote no modelo AJAX.

XMLHttpRequest é um objeto que fornece funcionalidade ao cliente para transferir dados entre um cliente e um servidor. Ele fornece uma maneira fácil de recuperar dados de um URL sem ter que fazer uma atualização de página inteira. Isso permite que uma página da Web atualize apenas uma parte do conteúdo sem interromper o que o usuário esteja fazendo. XMLHttpRequest é usado constantemente na programação de AJAX.

XMLHttpRequest foi originalmente projetado pela Microsoft e adotado pela Mozilla, Apple e Google. Está sendo padronizado pela WHATWG. Apesar do nome, XMLHttpRequest pode ser usado para recuperar qualquer tipo de dados, e não apenas XML, suportando também, protocolos diferentes de HTTP (incluindo file e ftp).

Para criar uma instância de XMLHttpRequest , basta fazer isso:

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

Construtor

XMLHttpRequest()

O construtor inicia um XMLHttpRequest. Ele deve ser chamado antes de quaisquer outras chamadas de método.

Gecko/Firefox 16 acrescenta um parâmetro não-padrão para o construtor que pode ativar o modo anônimo (veja Bug 692677). Definir o mozAnon bandeira de true eficácia se assemelha a AnonXMLHttpRequest() construtor descrito na especificação XMLHttpRequest que não tenha sido implementado em qualquer navegador ainda (em setembro de 2012).

```
XMLHttpRequest (  
  JSObject objParameters  
);
```

Parâmetros (não-padrão)

objParameters

Há dois sinalizadores que você pode definir:

mozAnon

Boolean: Definir esse sinalizador de true fará com que o navegador para não expor a origem e as credenciais do usuário ao buscar recursos. Mais importante, isto significa que os cookies não será enviado a menos que explicitamente adicionado usando setRequestHeader.

mozSystem

Boolean: Definir esse sinalizador de true . permite fazer conexões entre sites sem a necessidade de o servidor para opt-in usando CORS requer a configuração mozAnon: true . Ou seja, este não pode ser combinada com o envio de cookies ou outras credenciais do usuário. Isso só funciona em privilegiados (revisto) Apps;ele não funciona em páginas da web arbitrários carregados no Firefox.

Métodos

abort()

Aborta o pedido, se já foi enviada.

getAllResponseHeaders()

DOMString getAllResponseHeaders();

Retorna todos os cabeçalhos de resposta como uma string, ou null se nenhuma resposta foi recebida. Nota: Para os pedidos de várias partes, isso retorna os cabeçalhos da parte atual da solicitação, não a partir do canal original.

getResponseHeader()

DOMString? getResponseHeader(DOMString header);

Retorna a string contendo o texto do cabeçalho especificado, ou null se quer a resposta ainda não foi recebida ou o cabeçalho não existe na resposta.

open()

Inicializa um pedido. Este método é para ser usado a partir do código JavaScript; para inicializar um pedido do código nativo, use `openRequest()` em seu lugar.

```
void open(  
    DOMString method,  
    DOMString url,  
    optional boolean async,  
    optional DOMString user,  
    optional DOMString password  
);
```

Parameters

method

O método HTTP para usar, como "GET", "POST", "PUT", "DELETE", etc. ignorado para URLs não-HTTP (S).

url

O URL para o qual enviar a solicitação.

async

Um parâmetro booleano opcional, por padrão `true`, indicando se a operação deve ou não ser executada de forma assíncrona. Se esse valor for `false`, o `send()` método não retorna até que a resposta seja recebida. Se `true`, a notificação de uma transação concluída é fornecida usando ouvintes de evento. Isso deve ser `true` se o `multipart` atributo for `true`, ou uma exceção será lançada.

user

O nome de usuário opcional para usar para fins de autenticação; por padrão, essa é uma sequência vazia.

password

A senha opcional para usar para fins de autenticação; por padrão, essa é uma sequência vazia.

send()

Envia a solicitação. Se o pedido é assíncrono (que é o padrão), este método retorna assim que o pedido for enviado. Se o pedido é síncrono, este método não retorna até a resposta chegar.

```
void send();  
void send(ArrayBuffer data);  
void send(Blob data);  
void send(Document data);  
void send(DOMString? data);  
void send(FormData data);
```

tutorial step by step

<https://www.xul.fr/en-xml-ajax.html>

Referências

Lima, M. (s.d.). *Guia do ES6*. Fonte: Medium: <https://medium.com/@matheusml/o-guia-do-es6-tudo-que-voc%C3%AA-precisa-saber-8c287876325f>

MDN. (s.d.). <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Fun%C3%A7%C3%B5es>. Fonte: MDN web docs.