INSTITUTO FEDERAL
São Paulo

AJAX – Requisições assíncronas

PROF. DR. EDER PANSANI
epansani@gmail.com / epansani@ifsp.edu.br

Conteúdo da aula

INSTITUTO FEDERAL
São Paulo

Nesta aula aprenderemos um pouco sobre as **requisições assíncronas** e a tecnologia **AJAX**.

INSTITUTO FEDERAL
São Paulo

Introdução

Introdução

INSTITUTO FEDERAL
São Paulo

AJAX é o acrônimo de ***Asynchronous JavaScript and XML***, é uma **técnica** de desenvolvimento Web que permite a criação de aplicações mais interativas.

Introdução

INSTITUTO FEDERAL
São Paulo

De acordo com o W3Schools,

AJAX é o sonho de qualquer desenvolvedor, porque você pode:

- Ler dados de um servidor web após o carregamento da página
- Atualizar uma página da web sem recarregar a página
- Envie dados para um servidor web - em segundo plano

Introdução

INSTITUTO FEDERAL
São Paulo

Um dos principais **objetivos** é tornar as **respostas** das **páginas Web mais rápidas** pela troca de pequenas quantidades de informações com o servidor Web, nos bastidores.

Introdução

INSTITUTO FEDERAL
São Paulo

AJAX é uma técnica de desenvolvimento web que permite **atualizar partes específicas de uma página web sem** a necessidade de **recarregar** a **página inteira**.

Introdução

INSTITUTO FEDERAL
São Paulo

Isso é possível através da comunicação **assíncrona** entre o navegador e o servidor web, o que significa que os dados podem ser enviados e recebidos em **segundo plano**, sem interromper ou bloquear a interação do usuário com a página.

Introdução

INSTITUTO FEDERAL
São Paulo

O AJAX é uma forma de utilizar em **conjunto** algumas tecnologias, incluindo HTML, CSS, JavaScript, DOM, XML, XSLT e o mais importante: objeto ***XMLHttpRequest***.

Introdução

INSTITUTO FEDERAL
São Paulo

Todas essas tecnologias já foram discutidas, com exceção da XSLT (*Extensible Stylesheet Language for Transformation*), uma linguagem para transformação de documentos XML.

INSTITUTO FEDERAL
São Paulo

Histórico

Histórico

O termo AJAX foi introduzido por **Jesse James Garrett**, presidente e fundador da empresa Adaptive Path, no artigo "***Ajax: A New Approach to Web Applications***", em 18 de fevereiro de 2005.

Histórico

O artigo não apresentou nenhuma técnica revolucionária de desenvolvimento para Web, mas **introduziu um novo termo simples e representativo** para um conjunto de tecnologias que já estavam sendo utilizadas.

Histórico

INSTITUTO FEDERAL
São Paulo

Além disto, Garret explicou de forma clara e sucinta **como essas tecnologias melhoravam a experiência dos usuários** com aplicações Web, inclusive citando aplicações pioneiras como *Google Suggest* e *Google Maps*.

Histórico

Ajax: A New Approach to Web Applications:

https://designftw.mit.edu/lectures/apis/ajax_adaptive_path.pdf

Leitura recomendada!!

INSTITUTO FEDERAL
São Paulo

Vantagens

Vantagens

INSTITUTO FEDERAL
São Paulo

A utilização do Ajax em aplicações web em relação às abordagens tradicionais de carregamento de página pode ser muito benéfica, apresentando como vantangens:

Vantagens

INSTITUTO FEDERAL
São Paulo

Melhoria na Experiência do Usuário: Com o uso de Ajax, é possível criar interfaces web mais dinâmicas e interativas, onde as atualizações de conteúdo ocorrem de forma suave e rápida, sem a **necessidade de recarregar a página inteira**. Isso resulta em uma experiência mais fluida e responsiva para os usuários.

Vantagens

Redução da Latência Percebida: Ao permitir a atualização de conteúdo em segundo plano, Ajax pode reduzir a percepção de latência por parte dos usuários, pois eles **não precisam** esperar o carregamento completo da página para ver as mudanças.

Vantagens

INSTITUTO FEDERAL
São Paulo

Economia de Largura de Banda: Como apenas partes específicas da página são atualizadas, em vez da página inteira, o uso de Ajax pode resultar em uma economia significativa de largura de banda, especialmente em aplicações web com muitas interações do usuário.

Vantagens

INSTITUTO FEDERAL
São Paulo

Desenvolvimento de Aplicações Web Mais Interativas: Ajax é fundamental para o desenvolvimento de aplicações web mais avançadas e interativas, como clientes de e-mail em tempo real, feeds de redes sociais dinâmicos, sistemas de chat em tempo real e aplicativos de colaboração online.

Vantagens

INSTITUTO FEDERAL
São Paulo

Integração com Serviços Web: Ajax facilita a integração de páginas web com serviços web, como APIs RESTful, permitindo que os dados sejam recuperados e atualizados de forma assíncrona, sem interrupções na experiência do usuário.

INSTITUTO FEDERAL
São Paulo

Requisições assíncronas

Requisições assíncronas

INSTITUTO FEDERAL
São Paulo

Requisições assíncronas referem-se a **solicitações** de dados feitas por um **cliente** (geralmente um navegador web) a um **servidor**, onde a **execução do código do cliente não é bloqueada** enquanto **aguarda** a **resposta do servidor**.

Requisições assíncronas

INSTITUTO FEDERAL
São Paulo

Isso significa que o cliente pode continuar a executar outras tarefas enquanto espera pela resposta da requisição.

Requisições assíncronas

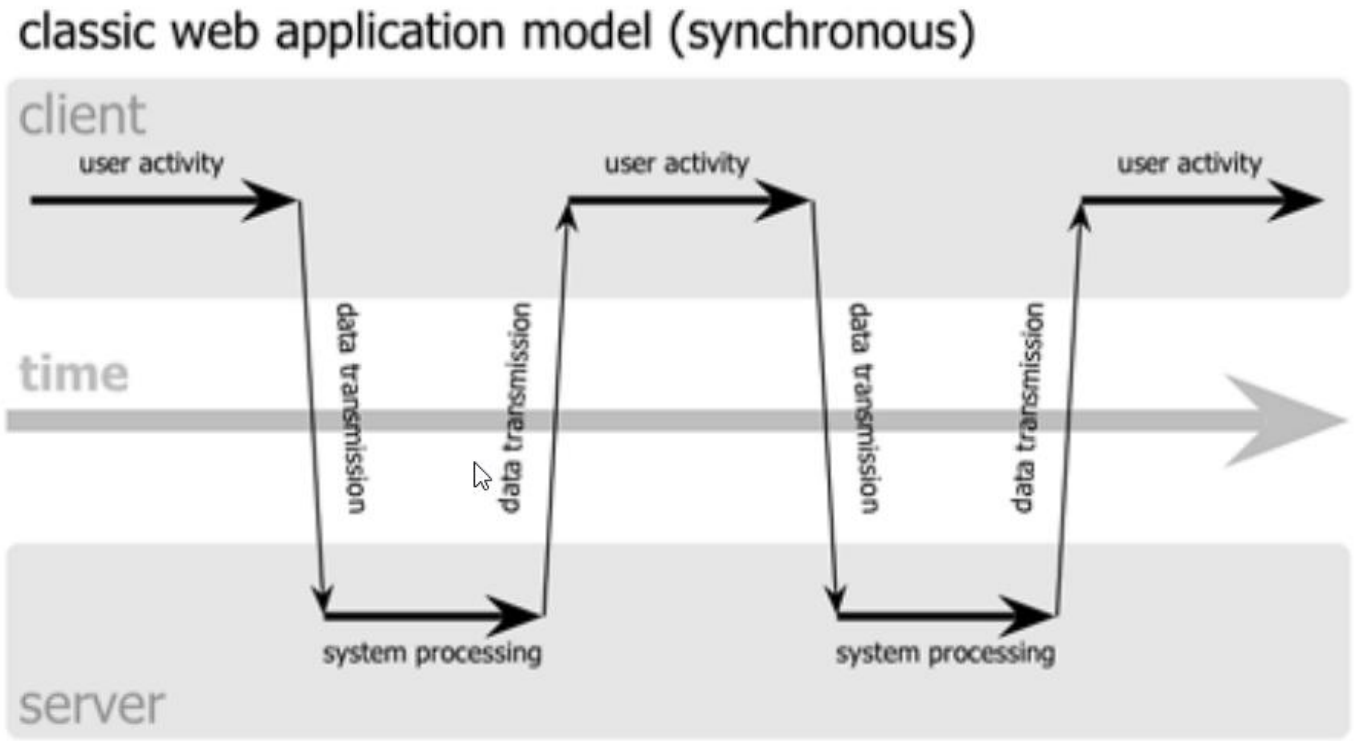
INSTITUTO FEDERAL
São Paulo

Requisições Síncronas: Nesse tipo de requisição, o cliente faz uma solicitação ao servidor e **aguarda** até que a resposta seja recebida antes de continuar a executar outras tarefas. Durante esse período de espera, a interface do usuário pode ficar congelada ou não responsiva.

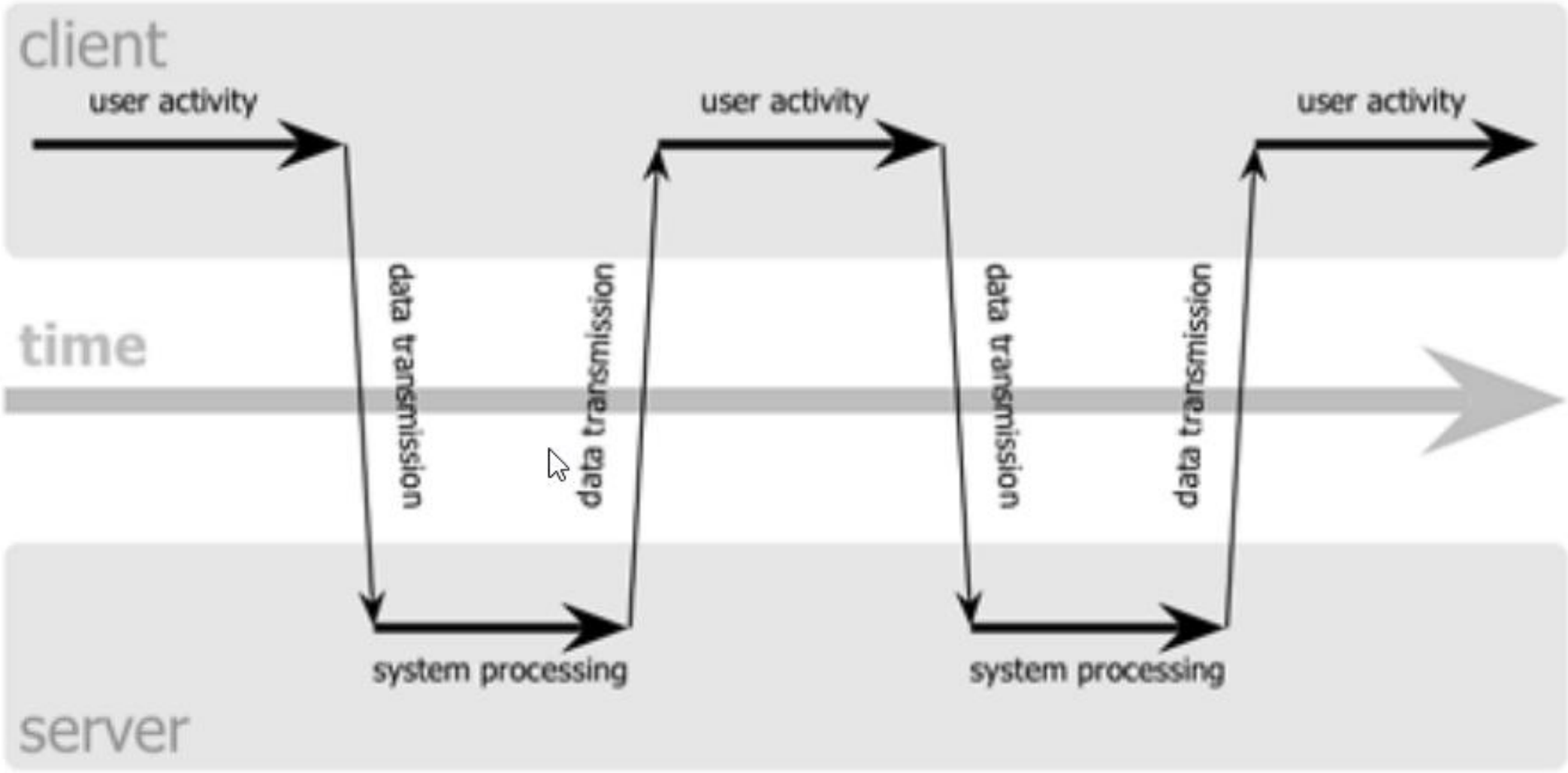
Requisições assíncronas

INSTITUTO FEDERAL
São Paulo

Requisições Síncronas:



classic web application model (synchronous)



Requisições assíncronas

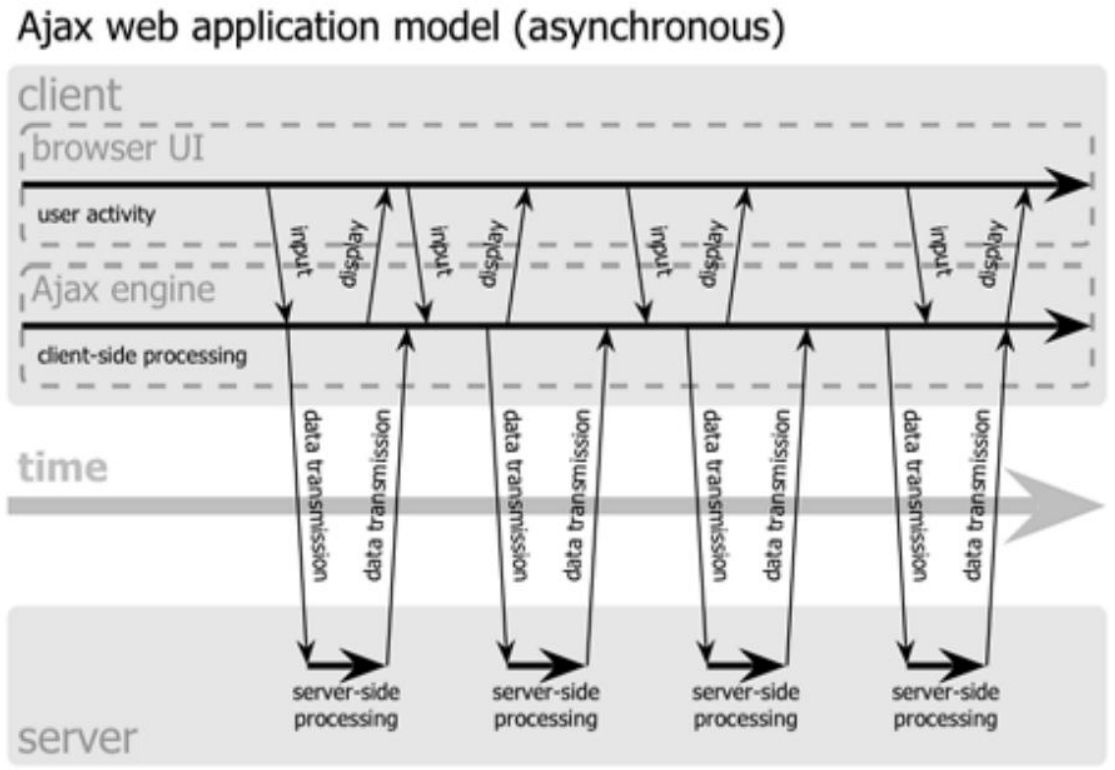
INSTITUTO FEDERAL
São Paulo

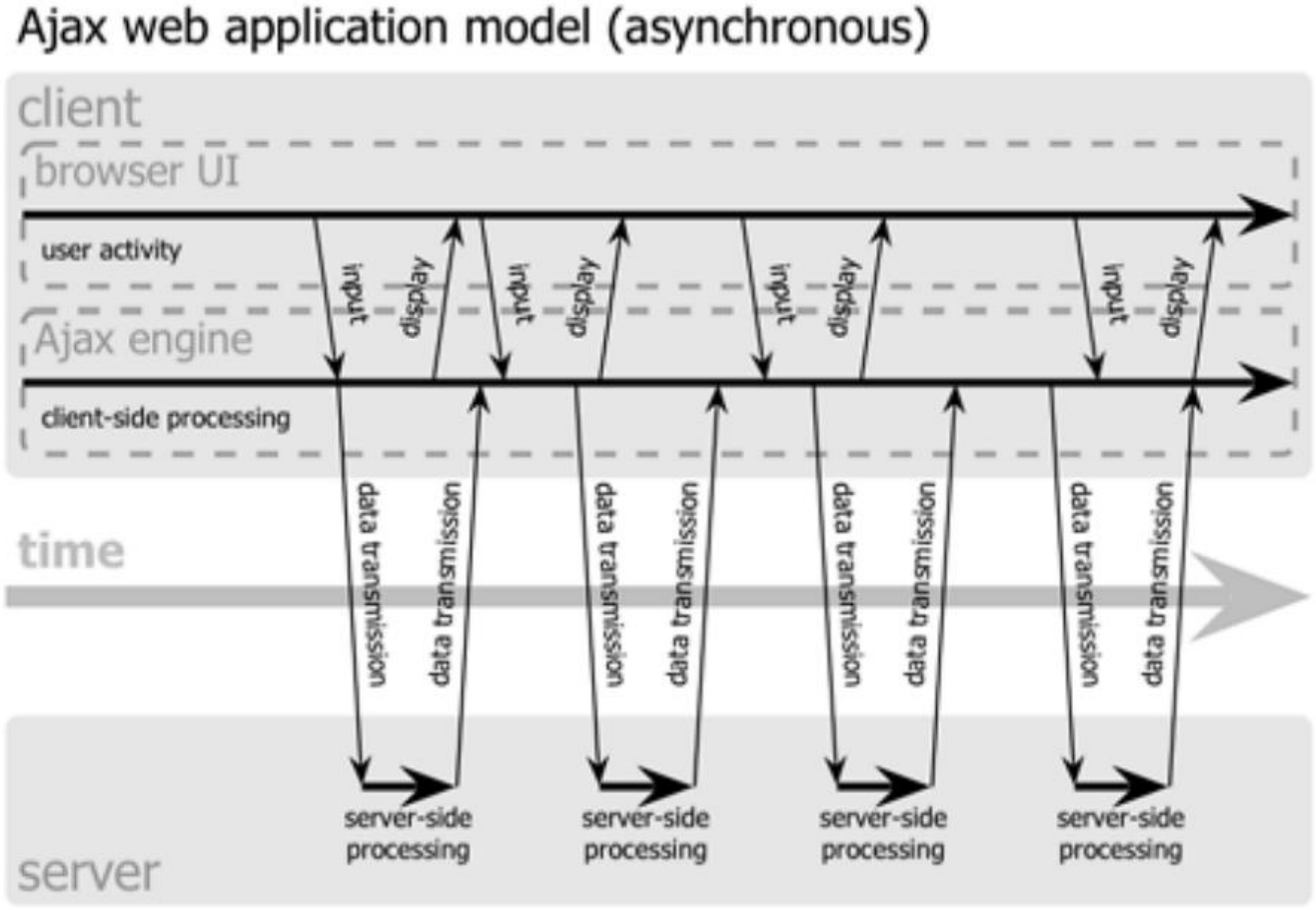
Requisições Assíncronas: Com requisições assíncronas, o cliente faz uma solicitação ao servidor, mas continua a executar outras tarefas sem esperar pela resposta imediata. Quando a resposta do servidor estiver pronta, o cliente será notificado e poderá processar os dados recebidos.

Requisições assíncronas

INSTITUTO FEDERAL
São Paulo

Requisições Assíncronas:




Ajax web application model (asynchronous)
client
browser UI
user activity
input
display
Ajax engine
client-side processing
data transmission
time
server-side processing
server

INSTITUTO FEDERAL
São Paulo

O objeto *XMLHttpRequest*

O objeto *XMLHttpRequest*

INSTITUTO FEDERAL
São Paulo

Ainda segundo o W3Schools, **a pedra angular do AJAX é o objeto XMLHttpRequest**.

https://www.w3schools.com/js/js_ajax_http.asp

O objeto *XMLHttpRequest*

INSTITUTO FEDERAL
São Paulo

O objeto ***XMLHttpRequest*** *(XHR)* é uma **API** padrão do navegador que **permite fazer requisições HTTP assíncronas** a um servidor web a partir de scripts JavaScript em uma página web.

O objeto *XMLHttpRequest*

INSTITUTO FEDERAL
São Paulo

Para criar uma instância do objeto XMLHttpRequest, geralmente se usa o construtor ***new XMLHttpRequest()***. Esse objeto representa uma única requisição HTTP.

```
variable = new XMLHttpRequest();
```

O objeto *XMLHttpRequest*

INSTITUTO FEDERAL
São Paulo

O XHR opera de forma **assíncrona por padrão**, o que significa que o navegador não fica bloqueado enquanto aguarda a resposta do servidor. Isso permite que outras tarefas na página continuem executando enquanto a requisição está em andamento.

O objeto *XMLHttpRequest*

INSTITUTO FEDERAL
São Paulo

O XHR **suporta vários métodos HTTP**, incluindo GET, POST, PUT, DELETE, entre outros. O método adequado é escolhido de acordo com o tipo de operação que se deseja realizar.

O objeto *XMLHttpRequest*

INSTITUTO FEDERAL
São Paulo

Após **criar uma instância do XHR**, é necessário **configurá-la** adequadamente antes de enviar a requisição. Isso pode incluir a definição do método HTTP, o URL de destino da requisição e, opcionalmente, o corpo da requisição (para métodos como POST).

O objeto *XMLHttpRequest*

INSTITUTO FEDERAL
São Paulo

Quando a **resposta** do servidor é recebida, o XHR invoca uma função de retorno de chamada (*callback*) para lidar com a resposta. Essa função pode ser definida usando a propriedade *onreadystatechange* do objeto XHR.

O objeto *XMLHttpRequest*

INSTITUTO FEDERAL
São Paulo

A **resposta** do servidor pode ser recuperada do objeto XHR e manipulada de várias maneiras. Dependendo do tipo de resposta (como texto, XML, JSON), é possível processar os dados recebidos e atualizar dinamicamente a página web com eles.

O objeto *XMLHttpRequest*

INSTITUTO FEDERAL
São Paulo

O XHR ainda dispara uma série de **eventos durante o ciclo de vida da requisição**, como eventos de progresso, eventos de carregamento e eventos de erro. Esses eventos podem ser usados para **monitorar o estado da requisição** e realizar ações apropriadas conforme necessário.

O objeto *XMLHttpRequest*

INSTITUTO FEDERAL
São Paulo

Exemplo,

link

```
// Create an XMLHttpRequest object
const xhttp = new XMLHttpRequest();

// Define a callback function
xhttp.onload = function() {
  // Here you can use the Data
}

// Send a request
xhttp.open("GET", "ajax_info.txt");
xhttp.send();
```

INSTITUTO FEDERAL
São Paulo

Acesso entre domínios

Acesso entre domínios

INSTITUTO FEDERAL
São Paulo

Por motivos de **segurança**, os **navegadores modernos não permitem acesso entre domínios**.

Isso significa que tanto a página web quanto o arquivo que ela tenta carregar **devem** estar localizados no mesmo servidor.

Acesso entre domínios

INSTITUTO FEDERAL
São Paulo

Todos os exemplos em W3Schools abrem arquivos localizados no domínio W3Schools.

Se você quiser usar o exemplo acima em uma de suas próprias páginas da web, os arquivos carregados deverão estar localizados em seu próprio servidor.

INSTITUTO FEDERAL
São Paulo

Segurança

Segurança

INSTITUTO FEDERAL
São Paulo

Ao usar Ajax em aplicações web, é essencial estar ciente das **questões de segurança associadas a essa técnica**.

Dois dos ataques mais comuns relacionados ao uso de Ajax são CSRF (*Cross-Site Request Forgery*) e XSS (*Cross-Site Scripting*).

Segurança

INSTITUTO FEDERAL
São Paulo

CSRF (*Cross-Site Request Forgery*): é um tipo de ataque em que um invasor induz um usuário autenticado a executar ações indesejadas em um aplicativo web no qual ele está autenticado. Isso é feito enganando o usuário para que ele faça uma solicitação HTTP, geralmente através de um formulário, que executa uma ação específica (como transferência de fundos, alteração de configurações, etc.) no aplicativo sem o conhecimento do usuário.

Segurança

INSTITUTO FEDERAL
São Paulo

No contexto de Ajax, os ataques CSRF podem ser realizados enviando solicitações maliciosas para *endpoints* de API protegidos por sessões autenticadas. Se o aplicativo não implementar medidas de proteção adequadas, um invasor pode explorar a sessão autenticada do usuário para realizar ações indesejadas em seu nome.

Segurança

INSTITUTO FEDERAL
São Paulo

Para mitigar ataques CSRF em solicitações Ajax, é importante **implementar tokens CSRF** ou outras técnicas de proteção, como a verificação de origem (*Origin Header*), para garantir que as solicitações sejam originadas do próprio aplicativo e não de sites maliciosos.

Segurança

INSTITUTO FEDERAL
São Paulo

XSS (Cross-Site Scripting) é um tipo de vulnerabilidade que permite que um invasor **injete scripts maliciosos em páginas web visualizadas por outros usuários**. Esses scripts podem ser usados para roubar informações confidenciais, redirecionar usuários para sites maliciosos, ou mesmo tomar controle sobre a sessão do usuário.

Segurança

INSTITUTO FEDERAL
São Paulo

Com o uso de Ajax, os ataques XSS podem ser ainda mais perigosos, pois os dados dinâmicos recebidos do servidor e injetados na página podem conter código JavaScript malicioso.

Segurança

INSTITUTO FEDERAL
São Paulo

Para mitigar ataques XSS em solicitações Ajax, é importante validar e **sanitizar** todos os dados recebidos do servidor antes de exibi-los na página. Além disso, é crucial usar técnicas como *Content Security Policy* (CSP) para restringir quais recursos externos podem ser carregados e executados na página, limitando assim a capacidade de um invasor de injetar scripts maliciosos.

INSTITUTO FEDERAL
São Paulo

Bibliotecas Ajax

Bibliotecas Ajax

INSTITUTO FEDERAL
São Paulo

Como discutido pra utilizar AJAX é necessário apenas fazer uso do objeto *XMLHttpRequest*, no entanto, é possível fazer isso de forma mais fácil usando bibliotecas e/ou *frameworks* Ajax.

Bibliotecas Ajax

INSTITUTO FEDERAL
São Paulo

Bibliotecas Ajax são conjuntos de código pré-escrito em JavaScript que oferecem funcionalidades e abstrações para facilitar o desenvolvimento de aplicações web que fazem uso de Ajax.

Bibliotecas Ajax

INSTITUTO FEDERAL
São Paulo

As bibliotecas Ajax geralmente incluem métodos e utilitários que **encapsulam a complexidade do objeto XMLHttpRequest (XHR)** e fornecem uma interface mais simples e amigável para fazer requisições HTTP, lidar com respostas do servidor e atualizar dinamicamente o conteúdo da página com os dados recebidos.

Bibliotecas Ajax

INSTITUTO FEDERAL
São Paulo

Existem várias bibliotecas e *frameworks* que facilitam o desenvolvimento com Ajax e oferecem uma variedade de recursos para simplificar o processo de fazer requisições assíncronas e lidar com respostas do servidor.

Bibliotecas Ajax

INSTITUTO FEDERAL
São Paulo

Dentre elas destacam-se a jQuery, Axios, Fetch API, Angular, React, Vue.js, Ember.js entre outras.

Bibliotecas Ajax

INSTITUTO FEDERAL
São Paulo

Destaca-se nestas a ***Fetch API***, que é uma API nativa do navegador para fazer requisições HTTP assíncronas. Ela fornece uma alternativa moderna ao objeto ***XMLHttpRequest***, oferecendo uma sintaxe mais limpa e baseada em promessas. É suportada na maioria dos navegadores modernos e pode ser usada sem a necessidade de bibliotecas externas.

Bibliotecas Ajax

INSTITUTO FEDERAL
São Paulo

Já Angular, React, Vue.js, Ember.js e outras desta “categoria” são frameworks completos e mais robustos mais indicados para a criação de aplicações maiores e não somente para lidar com Ajax.

Bibliotecas Ajax

INSTITUTO FEDERAL
São Paulo

Sobrando assim o jQuery, que em minha opinião é o mais adequado para o uso e implementação de Ajax.

https://api.jquery.com/category/ajax/

https://www.w3schools.com/jquery/jquery_ref_ajax.asp

Bibliotecas Ajax

INSTITUTO FEDERAL
São Paulo

A sintaxe da implementação Ajax com jQuery é bem mais simples:

```
$("button").click(function(){
  $.ajax({url: "demo_test.txt", success: function(result){
    $("#div1").html(result);
  }});
});
```

Bibliotecas Ajax

INSTITUTO FEDERAL
São Paulo

Reveja o código de JavaScript padrão para implementação básica do Ajax:

```
// Create an XMLHttpRequest object
const xhttp = new XMLHttpRequest();

// Define a callback function
xhttp.onload = function() {
  // Here you can use the Data
}

// Send a request
xhttp.open("GET", "ajax_info.txt");
xhttp.send();
```

Bibliotecas Ajax

INSTITUTO FEDERAL
São Paulo

Há ainda facilidades como a disponibilidade de métodos GET e POST e suas variações para obtenção de dados em formato JSON.

Method	Description
$.ajax()	Performs an async AJAX request
$.ajaxPrefilter()	Handle custom Ajax options or modify existing options before each request is sent and before they are processed by $.ajax()
$.ajaxSetup()	Sets the default values for future AJAX requests
$.ajaxTransport()	Creates an object that handles the actual transmission of Ajax data
$.get()	Loads data from a server using an AJAX HTTP GET request
$.getJSON()	Loads JSON-encoded data from a server using a HTTP GET request
$.parseJSON()	Deprecated in version 3.0, use JSON.parse() instead. Takes a well-formed JSON string and returns the resulting JavaScript value
$.getScript()	Loads (and executes) a JavaScript from a server using an AJAX HTTP GET request
$.param()	Creates a serialized representation of an array or object (can be used as URL query string for AJAX requests)
$.post()	Loads data from a server using an AJAX HTTP POST request
ajaxComplete()	Specifies a function to run when the AJAX request completes
ajaxError()	Specifies a function to run when the AJAX request completes with an error
ajaxSend()	Specifies a function to run before the AJAX request is sent

Bibliotecas Ajax

INSTITUTO FEDERAL
São Paulo

Aqui um exemplo enviando dados para um servidor e mostrando uma mensagem quando a ação for completa:

```
1 | $.ajax({
2 |   method: "POST",
3 |   url: "some.php",
4 |   data: { name: "John", location: "Boston" }
5 | })
6 |   .done(function( msg ) {
7 |     alert( "Data Saved: " + msg );
8 |   });
```

INSTITUTO FEDERAL
São Paulo

Referências

Referências

INSTITUTO FEDERAL
São Paulo

Referências consultadas:

https://www.devmedia.com.br/o-que-e-o-ajax/6702

https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Client-side_web_APIs/Fetching_data

Referências

INSTITUTO FEDERAL
São Paulo

Referências consultadas:

https://www.w3schools.com/js/js_ajax_intro.asp

https://www.jetersonlordano.com.br/javascript/aprenda-a-fazer-requisicoes-assincronas-com-ajax-em-javascript-puro

INSTITUTO FEDERAL
São Paulo

Dúvidas?

Perguntas?

Sugestões?