

# APIs web do lado do cliente

Samuel Amaro

15 de maio de 2021

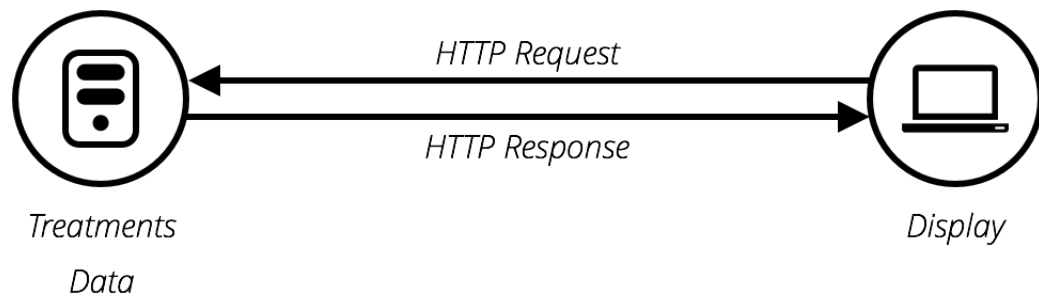
## Buscando dados do Servidor

Outra tarefa muito comum em sites e aplicativos modernos é recuperar itens de dados individuais do servidor para atualizar seções de uma página da Web sem ter que carregar uma página nova inteira. Esse detalhe aparentemente pequeno teve um grande impacto no desempenho e no comportamento dos sites, portanto, neste artigo, explicaremos o conceito e examinaremos as tecnologias que o tornam possível, como XMLHttpRequest e a API Fetch.

**Objetivo:** Aprender como obter dados do servidor e usá-los para atualizar o conteúdo de uma página da web.

## Qual é o problema aqui ?

Originalmente, o carregamento da página na web era simples - você enviaria uma solicitação de um site para um servidor e, desde que nada desse errado, os ativos que formaram a página da web seriam baixados e exibidos em seu computador.



## Arquitetura antiga do site da web

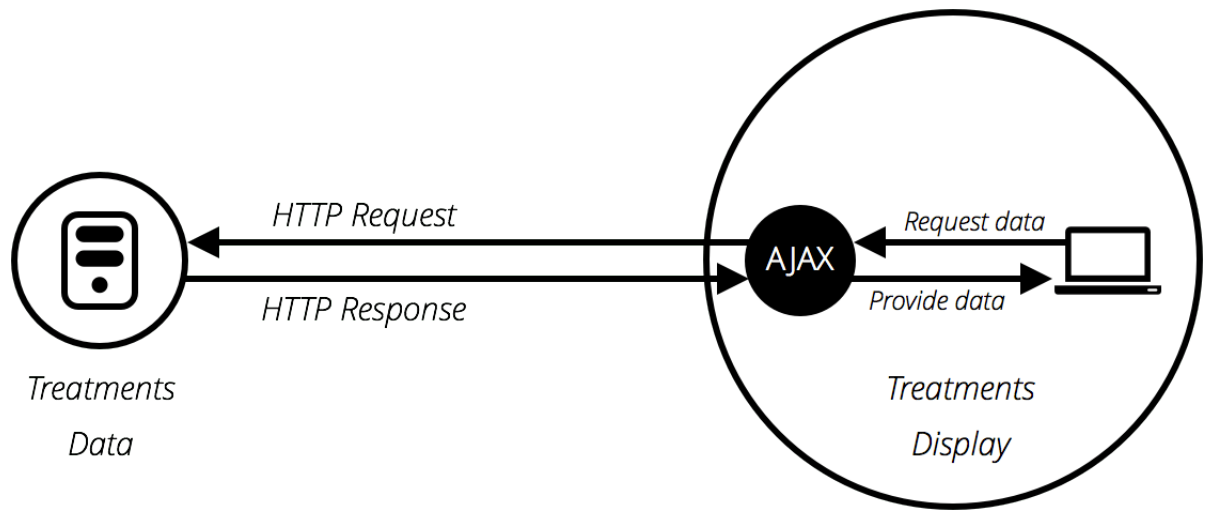
O problema com este modelo é que sempre que você deseja atualizar qualquer parte da página, por exemplo, para exibir um novo conjunto de produtos ou carregar uma nova página, você precisa carregar a página inteira novamente. Isso é extremamente desperdício e resulta em uma experiência do usuário insatisfatória, especialmente à medida que as páginas ficam maiores e mais complexas.

## Entrar em Ajax

Isso levou à criação de tecnologias que permitem que páginas da web solicitem pequenos blocos de dados (como HTML , XML , JSON ou texto simples) e os exibam apenas quando necessário, ajudando a resolver o problema descrito acima.

Isso é obtido usando APIs como XMLHttpRequest ou - mais recentemente - a API Fetch. Essas tecnologias permitem que as páginas da web tratem diretamente de fazer solicitações HTTP para recursos específicos disponíveis em um servidor e formatar os dados resultantes conforme necessário antes de serem exibidos.

**Nota:** Nos primeiros dias, essa técnica geral era conhecida como **Asynchronous JavaScript and XML (Ajax)**, porque tendia a ser usada **XMLHttpRequest** para solicitar dados XML. Normalmente, esse não é o caso hoje em dia (é mais provável que você use **XMLHttpRequest** ou **Fetch** para solicitar **JSON**), mas o resultado ainda é o mesmo e o termo "Ajax" ainda é frequentemente usado para descrever a técnica.



Arquitetura moderna de site da web

O modelo Ajax envolve o uso de uma API da web como proxy para solicitar dados de maneira mais inteligente, em vez de apenas fazer o navegador recarregar a página inteira. Vamos pensar sobre o significado disso:

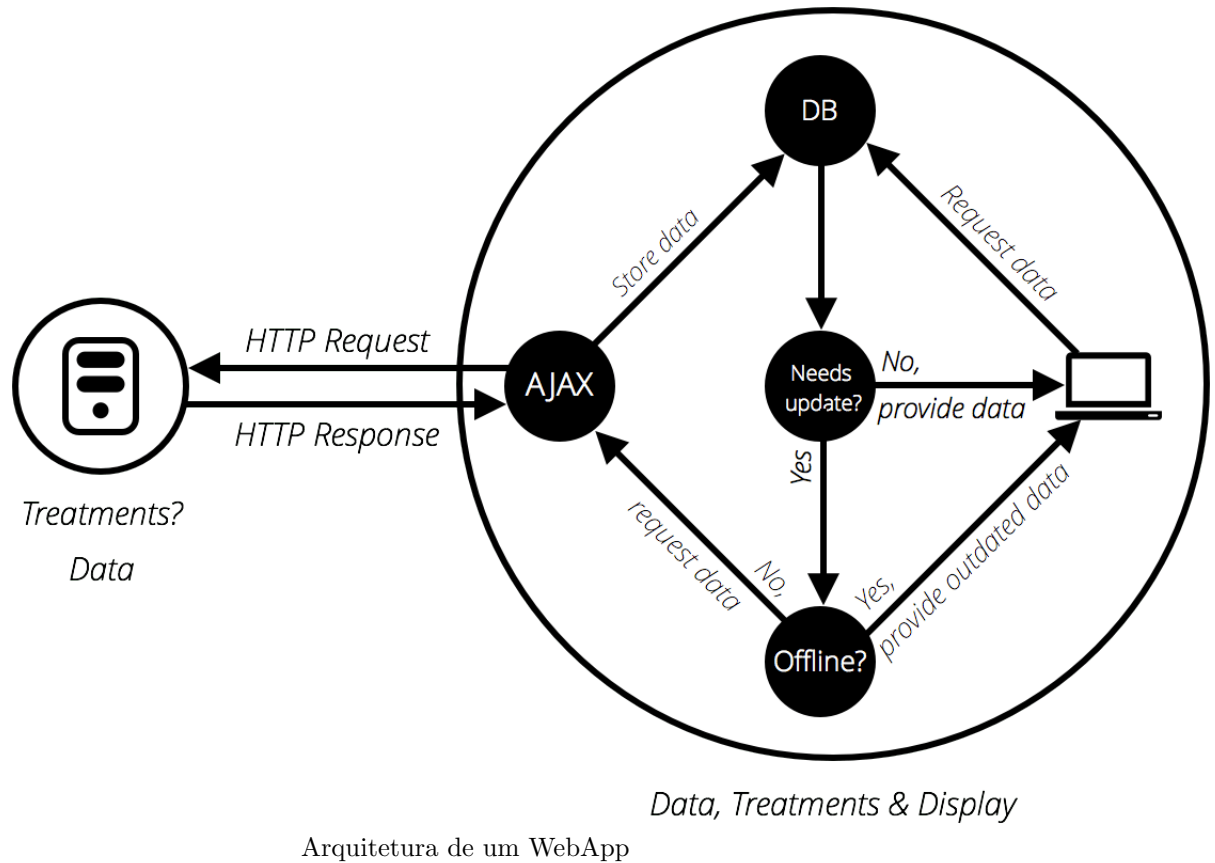
1. Vá para um de seus sites ricos em informações favoritos, como Amazon, YouTube, CNN, etc., e carregue-o.
2. Agora procure por algo, como um novo produto. O conteúdo principal mudará, mas a maioria das informações ao redor, como cabeçalho, rodapé, menu de navegação, etc., permanecerão as mesmas.

Isso pe realmente bom porque:

- As atualizações de página são muito mais rápidas e você não precisa esperar a atualização da página, o que significa que o site parece mais rápido e ágil.
- Menos dados são baixados em cada atualização, o que significa menos largura de banda desperdiçada. Isso pode não ser um problema tão grande em um desktop em uma conexão de banda larga, mas é um grande problema em dispositivos móveis e em países em desenvolvimento que não têm serviço de Internet rápida onipresente.

Para acelerar ainda mais, alguns sites também armazenam ativos e dados no computador do usuário quando são solicitados pela primeira vez, o que significa que nas visitas subsequentes eles usam as versões locais em vez

de baixar novas cópias sempre que a página é carregada pela primeira vez. O conteúdo só é recarregado do servidor depois de atualizado.



## Uma solicitação Ajax básica

Vejamos como essa solicitação é tratada, usando `XMLHttpRequest` e `Fetch`. Para esses exemplos, solicitaremos dados de alguns arquivos de texto diferentes e os usaremos para preencher uma área de conteúdo.

Esta série de arquivos funcionará como nosso banco de dados falso; em um aplicativo real, estaríamos mais propensos a usar uma linguagem do lado do servidor como PHP, Python ou Node para solicitar nossos dados de um banco de dados. Aqui, no entanto, queremos mantê-lo simples e nos concentrar na parte do lado do cliente.