



`</>` **htmx**

Introdução

Ao HTMX



O que é HTMX?

- **Biblioteca JavaScript Moderna:** Simplifica a adição de interatividade a websites.
- **Atualizações de Página Parciais:** Permite atualizar partes de uma página sem recarregar tudo.
- **Baseado em Atributos HTML:** Usa atributos especiais para adicionar funcionalidades diretamente no HTML.
- **Simplicidade e Acessibilidade:** Fácil de aprender, integra-se bem com HTML existente.

`</> htmx`

HTMX vs SPA (Single Page Applications)

- **SPAs:** Carregam uma única página HTML, atualizada dinamicamente (Ex: React, Angular).
- **HTMX:** Abordagem mais simples para atualizações dinâmicas sem frameworks completos.
- **Uso de Recursos:** SPAs geralmente exigem mais recursos; HTMX é mais leve.
- **Escolha Baseada em Necessidade:** HTMX para simplicidade e eficiência, SPAs para aplicações interativas complexas.

`</>` htmx



Quando usar HTMX?

- **Interatividade em Páginas Estáticas:** Ideal para formulários dinâmicos e conteúdo sob demanda.
- **Atualizações Parciais de Página:** Ótimo para carregar ou atualizar seções específicas da página.
- **Simplicidade sobre Complexidade:** Prefira HTMX em projetos que evitam a complexidade de SPAs.
- **Limitações:** Menos adequado para aplicações que exigem interatividade complexa ou em tempo real.

`</> htmx`

O que é Hypermedia API?

- **APIs Navegáveis como Hipertexto:** Permite que clientes descubram ações e recursos dinamicamente.
- **HTMX e HAPI:** HTMX pode utilizar Hypermedia API para carregar e interagir com conteúdo de forma dinâmica.
- **Flexibilidade e Escalabilidade:** Favorece uma arquitetura adaptável, facilitando manutenção e expansão.
- **Independência Cliente-Servidor:** Permite evolução independente de clientes e servidores.

`</>` **htmx**

Extensão para HTMX e http-server

- Vamos instalar a extensão: **htmx-tags**, ela nos ajudará com o autocomplete dos recursos do HTMX.
- Para instalar o HTTP server, use o comando: **npm install -g http-server**.
- E para executar: vá a uma pasta com um arquivo HTML e digite http-server.

</> htmx

Hello World com HTMX

- Primeiramente é necessário ter uma **API com o backend funcional**, e instalar o HTMX em um arquivo .html, por meio de uma tag script;
- **HTMX para Interatividade:**
 - Utiliza hx-get para solicitar dados ao servidor (Express) e hx-target para definir onde exibir a resposta.
- **Servidor Express:**
 - Responde às solicitações HTMX, retornando conteúdo dinâmico (ex: 'Hello World!').
- **Fluxo de Requisição e Resposta:**
 - Botão HTMX envia solicitação GET, servidor Express processa e retorna resposta, HTMX atualiza o DOM.

`</> htmx`



Repositório do curso

- O repositório do curso pode ser acessado através do link:
https://github.com/matheusbattisti/curso_htmx
- Nele você encontrará todos os arquivos com o conteúdo das aulas.
- Utilize como referência para possíveis erros, e também para copiar código que você não queria criar (não recomendado =D).

</> htmx

Comunidade de alunos

- Utilize o link para entrar no **Discord da Hora de Codar**:
<https://discord.gg/Veq4mvsWwk>
- Lá você encontrará nossos alunos, você pode entrar em contato para **networking** e também tirar dúvidas.
- Além disso, temos canais de: vagas, compartilhamento de perfis (GH e LinkedIn), Sugestões para cursos da HDC e muito mais.
- **Eu** estou lá também =)



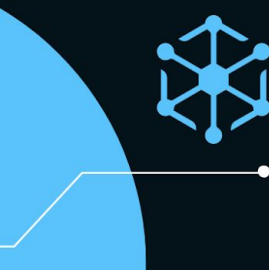
</> htmx



`</>` **htmx**

Introdução

Fim da seção





Conceitos fundamentais

Introdução

Requisição HTTP com HTMX

- **Introdução ao `hx-get` e `hx-post`:**
 - `hx-get` para solicitar e carregar conteúdo dinâmico sem recarregar a página inteira.
 - `hx-post` envia dados de formulário ou interações de usuário ao servidor, facilitando a criação de interfaces dinâmicas.
- **Interatividade Simplificada:**
 - HTMX permite enviar e receber dados de maneira simplificada, melhorando a experiência do usuário com atualizações rápidas e fluidas.
- **Exemplo Prático de Uso:**
 - Botão com `hx-post` que envia uma requisição POST ao servidor, o qual responde e atualiza um elemento específico da página.

`</>` **htmx**



Métodos Put e Delete com HTMX

- **hx-put e hx-delete:**
 - hx-put é usado para enviar dados ao servidor e atualizar informações existentes, ideal para cenários de edição.
 - hx-delete permite enviar uma requisição de deleção ao servidor, útil para remover itens ou dados.
- **Aplicações em Sistemas CRUD:**
 - Esses métodos são cruciais em sistemas CRUD, proporcionando uma maneira eficiente de gerenciar dados com interações simples.
- **Fluxo Simplificado de Requisições:**
 - HTMX facilita as operações de PUT e DELETE, tornando o desenvolvimento mais intuitivo e menos propenso a erros.

`</>` htmx



Gatilhos de Requisição com HTMX

- **Fundamentos do hx-trigger:**
 - Define condições e momentos em que as requisições HTMX são acionadas, como em eventos de clique, mudança de foco ou entrada de texto.
- **Customização de Interatividade:**
 - Permite ajustar precisamente como e quando as ações do usuário resultam em requisições HTTP, enriquecendo a interatividade da página.
- **Exemplos Práticos:**
 - Implementação de um campo de busca que reage a cada tecla pressionada pelo usuário, enviando requisições de forma eficiente e dinâmica.

`</>` htmx



Modificador de Trigger

- **Diversidade de Modificadores:**
 - once para uma ação única, delay para adicionar um atraso, e acionamento de eventos com base em ações de outros elementos.
- **Riqueza em Detalhes de UI/UX:**
 - Tais modificações nos gatilhos oferecem um controle fino sobre a experiência do usuário, permitindo criar páginas web altamente interativas e reativas.

`</>` htmx

Filtros de Trigger

- **Definição de Filtros de Trigger:**
 - Filtros de trigger no HTMX controlam a execução de requisições baseadas em condições específicas nos elementos.
 - Permitem uma interação mais refinada e controlada, evitando requisições desnecessárias.
- **Diversidade de Aplicações:**
 - Utilizados em formulários, interações de usuário e outros elementos dinâmicos da página.
 - Exemplos comuns incluem condicionais com base na presença de texto, mudanças em campos de formulário ou estados específicos de um elemento.
- **Exemplos de Filtros de Trigger:**
 - `keyup changed`: Aciona a requisição apenas se houver mudanças no valor do campo ao soltar a tecla.
 - `click [condição]`: Aciona a requisição com base em uma condição específica ao clicar.

`</>` **htmx**



Explorando Eventos Especiais no HTMX

- **Diversidade de Eventos Especiais:**
 - load: Carrega conteúdo assim que o elemento é carregado.
 - revealed: Aciona quando o elemento se torna visível na tela.
 - intersect: Interage com a API de Intersection Observer para acionar eventos.
 - threshold: Define um limiar para o evento intersect.
- **Aplicações Interativas:**
 - Criação de páginas web que respondem dinamicamente à navegação e visualização do usuário.
- **Exemplos e Demonstrações:**
 - Uso de load para carregamento inicial, revealed e intersect para carregamento dinâmico baseado na posição de scroll.

`</>` htmx



Implementando Polling com HTMX

- **Polling no HTMX:**
 - Técnica para atualizar periodicamente um conteúdo ou verificar mudanças no servidor.
- **Uso de hx-poll:**
 - Atributo hx-poll para definir intervalos de tempo para requisições automáticas.
- **Cenários de Uso:**
 - Ideal para atualizações de status, feeds de notícias, ou monitoramento de dados em tempo real.
- **Práticas Recomendadas:**
 - Balanceamento entre frequência de polling e carga no servidor, garantindo eficiência e responsividade.

`</>` htmx



Criando Indicadores de Progresso com HTMX

- **Importância dos Indicadores de Progresso:**
 - Melhora a experiência do usuário ao fornecer feedback visual durante requisições HTTP.
- **Implementação com HTMX:**
 - Uso de hx-indicator para mostrar e esconder elementos durante o processamento de requisições.
- **Exemplos Práticos:**
 - Indicadores de carregamento, barras de progresso, ou mensagens de status durante ações de longa duração.

`</> htmx`

Enviando Dados com HTMX

- **Utilização de Formulários HTMX:**
 - Criação de formulários com HTMX para enviar dados ao servidor de forma assíncrona.
- **Configuração de Formulário com hx-post:**
 - Atributo hx-post para especificar a URL de destino da submissão do formulário.

`</> htmx`

Mensagens de Confirmação com HTMX

- **Uso de `hx-confirm`:**
 - Atributo `hx-confirm` para adicionar uma camada de confirmação antes de executar ações.
- **Interação do Usuário:**
 - Criação de diálogos de confirmação para ações críticas como deletar ou alterar dados importantes.

`</>` htmx

Upload de Arquivos

- **Configuração de Upload de Arquivos:**
 - Criar formulário no HTML com enctype para multipart/form-data.
 - Atributo hx-post para a rota de destino.
- **Processamento no Backend:**
 - Adicionar algum pacote como o Multer para facilitar o recebimento e tratamento de imagem.

`</>` htmx

Sincronização de Requisições

- **Conceito de Sincronização:**
 - Sincronização de requisições garante uma experiência de usuário coesa, onde múltiplas ações interdependentes são gerenciadas eficientemente.
- **Uso de `hx-sync`:**
 - O atributo `hx-sync` permite sincronizar requisições de diferentes elementos, garantindo que ações relacionadas ocorram em uma ordem específica.

`</>` htmx

Cancelando Requisições

- **Importância do Cancelamento:**
 - Permitir que usuários cancelem operações em andamento melhora a interatividade e o controle do usuário sobre a aplicação.
- **Técnica de Cancelamento com HTMX:**
 - Precisamos criar um elemento que ativa um evento especial que é o `htmx:abort`.
 - O evento será chamado através do método `trigger` do objeto `htmx`, que está disponível para ser executado via JavaScript.

`</>` htmx





Conceitos fundamentais

Conclusão



Carregamento de conteúdo

Introdução

Selecionando Alvos com hx-target

- **Fundamentos do hx-target:**
 - hx-target define onde o conteúdo retornado será carregado na página.
- **Seleção por ID e Classe:**
 - Uso de IDs (#id) para alvos únicos e classes (.class) para elementos com classe.
- **Exemplos Práticos:**
 - Demonstração de como carregar conteúdo em diferentes partes da página, utilizando IDs e classes.

`</>` htmx

Troca de Conteúdo com HTMX

- **Conceito de Content Swapping:**
 - `hx-swap` controla como o conteúdo retornado é inserido no elemento alvo.
- **Swap Padrão com HTMX:**
 - Por padrão, `hx-swap` substitui o conteúdo interno (`inner`) do elemento alvo.
- **Vantagens do Content Swapping:**
 - Permite atualizar seletivamente partes da página sem necessidade de recarregar toda a página.

`</>` **htmx**

hx-target: Especifica onde o conteúdo retornado será inserido na página.

hx-swap: Define como o conteúdo será inserido no elemento alvo (`hx-target`).



Explorando as Opções de hx-swap

- **Diversidade no hx-swap:**
 - hx-swap oferece opções como afterbegin, beforebegin, beforeend, afterend para controle preciso do posicionamento do conteúdo.
- **Usos de afterbegin e beforebegin:**
 - afterbegin: insere conteúdo logo após a abertura do alvo.
 - beforebegin: coloca conteúdo imediatamente antes do alvo.
- **Usos de beforeend e afterend:**
 - beforeend: adiciona conteúdo antes do fechamento do alvo.
 - afterend: insere conteúdo após o fechamento do alvo.

`</>` htmx



Subconjuntos da Resposta com hx-select

- **Seleção de Conteúdo com hx-select:**
 - hx-select permite escolher um subconjunto específico do HTML retornado para ser inserido no alvo.
- **Controle Fino do Conteúdo Carregado:**
 - Use hx-select para extrair apenas os elementos relevantes da resposta do servidor.
- **Casos de Uso:**
 - Muito útil em situações onde apenas uma parte da resposta do servidor é necessária, como listas ou itens específicos.

`</>` htmx

Explorando hx-params

- **Customização de Parâmetros com hx-params:**
 - hx-params oferece controle avançado sobre os parâmetros enviados em uma requisição HTMX.
- **Adicionando Parâmetros Dinâmicos:**
 - Como adicionar e excluir parâmetros dinamicamente de requisições, adequando-se a diferentes necessidades.

`</>` htmx

Manipulando Respostas JSON com HTMX

- **Introdução a Respostas JSON:**
 - HTMX pode manipular respostas JSON para atualizar a UI de maneira dinâmica e eficiente.
- **Utilização de JSON em HTMX:**
 - Como usar hx-get para solicitar e tratar dados JSON do servidor.

`</>` htmx



Implementando Lazy Loading com HTMX

- **Conceito de Lazy Loading:**
 - Técnica de carregar conteúdo conforme necessário, melhorando a performance da página.
- **Aplicação com HTMX:**
 - Uso de atributos HTMX para carregar imagens, vídeos ou partes de texto sob demanda.
- **hx-trigger e hx-swap:**
 - Configuração de gatilhos como revealed ou load para implementar o lazy loading.

`</> htmx`



Carregamento de conteúdo

Conclusão



`</>` **htmx**

Eventos

Introdução



Eventos de Carregamento de Página no HTMX

- Introdução aos Eventos de Carregamento:
 - HTMX dispara eventos específicos durante o carregamento da página que podem ser usados para inicializar ou configurar a página.
- Uso do `htmx:load`:
 - O evento `htmx:load` é útil para executar scripts ou inicializações após o carregamento da página.

`</>` htmx

Ações Antes e Depois de um Swap

- **Entendendo Eventos de Swap:**
 - HTMX fornece eventos para executar ações antes e depois de swaps, como `htmx:beforeSwap` e `htmx:afterSwap`.
- **Customização de Swaps:**
 - Como usar esses eventos para modificar o comportamento padrão de swaps ou executar ações adicionais.

</> htmx

Ações Antes e Depois de um Request

- **Eventos de Requisições HTTP:**
 - Exploração dos eventos `htmx:beforeRequest` e `htmx:afterRequest`, que ocorrem antes e depois de uma requisição HTTP.
- **Interceptação e Modificação de Requisições:**
 - Uso desses eventos para modificar requisições antes de serem enviadas ou para processar a resposta.

`</>` htmx

Tratando Falha na Requisição

- **Gestão de Erros com HTMX:**
 - Entenda o manejo de eventos de erro usando `htmx:onError`.
- **Reação a Falhas de Requisições:**
 - Implemente mecanismos de feedback ao usuário em caso de falhas de comunicação com o servidor.

`</>` htmx

Interceptando Request

- **Eventos para Manipulação de Requisições:**
 - Explore o uso de `htmx:configRequest` para interceptar e modificar detalhes da requisição.
- **Customização de Requisições:**
 - Adicione cabeçalhos personalizados e modifique parâmetros de requisição antes do envio.

`</>` htmx

Conhecendo o HTMX Logger

- **Visão Geral do HTMX Logger:**
 - O HTMX Logger é uma ferramenta integrada que oferece insights detalhados sobre o funcionamento interno do HTMX.
- **Ativação do Logger:**
 - Criar uma função anônima na propriedade logger do objeto htmx;
- **Recursos do Logger:**
 - Rastreamento de Eventos: Observa eventos HTMX em tempo real, como requisições enviadas, respostas recebidas e eventos de swap.
 - Detalhamento de Erros: Identifica e registra erros, facilitando a depuração de problemas comuns.

`</>` htmx

Aprimorando Formulários com HTMX

- **HTMX em submit e change:**
 - Utilize `hx-post` em botões ou elementos de submit para enviar dados do formulário. Em campos de entrada, use `hx-get` com `hx-trigger="change"` para requisições automáticas ao modificar o campo.
- **Interação com focus e blur:**
 - Adicione `hx-get` com `hx-trigger="focus"` em campos de entrada para carregar dados ou informações ao focar e `hx-trigger="blur"` para ações quando o campo perde o foco.

`</>` **htmx**

Eventos Customizados e Acionadores HTMX

- **Disparando Eventos Customizados:**

- Crie um evento customizado em JavaScript com `new CustomEvent("nomeDoEvento")` e use `htmx.trigger(elemento, "nomeDoEvento")` para dispará-lo, acionando ações HTMX.

- **Captura e Resposta a Eventos:**

- Configure um ouvinte de evento em JavaScript para reagir ao evento customizado e realizar ações como enviar requisições HTMX ou atualizar a interface do usuário.

`</>` htmx





`</>` **htmx**

Eventos

Conclusão

