Referência Rápida htmx

Atributos, triggers, swap, eventos, extensões

Básico

Instalação

```
<script src="https://unpkg.com/htmx.org@2"></script>

<!-- Or via npm -->
npm install htmx.org
```

Como Funciona

HTML-driven	Emitir AJAX de atributos HTML, sem JS necessário
Swap content	Servidor retorna fragmentos HTML, htmx os troca
Progressive	Aprimora HTML padrão; degrada graciosamente
REST-friendly	Funciona com qualquer framework server-side

Atributos

Atributos de Requisição Principais

hx-get="/url"	Emitir requisição GET para URL
hx-post="/url"	Emitir requisição POST para URL
hx-put="/url"	Emitir requisição PUT para URL
hx-patch="/url"	Emitir requisição PATCH para URL
hx-delete="/url"	Emitir requisição DELETE para URL

Atributos de Comportamento

hx-trigger	Evento que aciona a requisição
hx-target	Elemento para trocar o conteúdo
hx-swap	Como trocar o conteúdo de resposta
hx-select	Selecionar subconjunto do HTML de resposta
hx-vals	Adicionar valores extras à requisição
hx-confirm	Mostrar diálogo de confirmação antes da requisição
hx-disable	Desabilitar processamento htmx no elemento

Triggers

Sintaxe de Trigger

```
<!-- Default: natural event (click for buttons) -->
<button hx-get="/data">Load</button>

<!-- Custom trigger event -->
<div hx-get="/news" hx-trigger="every 5s">Feed</div>

<!-- Multiple triggers -->
<input hx-get="/search" hx-trigger="keyup changed delay:300ms" />
```

Modificadores de Trigger

changed	Acionar somente se o valor mudou
delay:Ns	Aguardar N segundos antes de acionar
throttle:Nms	Limitar a uma vez por N ms
once	Acionar apenas na primeira vez
from:selector	Ouvir em um elemento diferente
every Ns	Fazer polling a cada N segundos
load	Acionar no carregamento do elemento
revealed	Acionar quando o elemento rola para a visualização

Targets

Seleção de Target

```
<!-- Target another element -->
<button hx-get="/data" hx-target="#result">Load</button>
<div id="result"></div>

<!-- Target closest ancestor -->
<button hx-get="/row" hx-target="closest tr">Update</button>

<!-- Target with CSS selector -->
<button hx-get="/info" hx-target="next .output">Go</button>
```

Palavras-chave de Target

this	O próprio elemento (padrão)
closest <sel>	Ancestral mais próximo que corresponde ao seletor
find <sel>	Primeiro descendente que corresponde ao seletor
next <sel>	Próximo irmão que corresponde ao seletor
previous <sel>	Irmão anterior que corresponde ao seletor

Swap

Estratégias de Swap

innerHTML	Substituir conteúdo interno (padrão)
outerHTML	Substituir todo o elemento alvo
afterbegin	Inserir no início do target
beforeend	Adicionar ao final do target
beforebegin	Inserir antes do target
afterend	Inserir após o target
delete	Remover elemento alvo
none	Sem swap, apenas acionar eventos

Modificadores de Swap

```
<!-- Swap with transition delay -->
<div hx-get="/data" hx-swap="innerHTML swap:300ms">

<!-- Settle delay for CSS transitions -->
<div hx-get="/data" hx-swap="innerHTML settle:500ms">

<!-- Scroll to top after swap -->
<div hx-get="/page" hx-swap="innerHTML scroll:top">
```

Headers

Headers de Requisição (enviados pelo htmx)

HX-Request	Sempre "true" para requisições htmx
HX-Target	ID do elemento alvo
HX-Trigger	ID do elemento acionador
HX-Trigger-Name	Nome do elemento acionador
HX-Current-URL	URL atual do navegador
HX-Prompt	Resposta do usuário ao hx-prompt

Headers de Resposta (enviados pelo servidor)

HX-Redirect	Redirecionamento client-side para URL
HX-Refresh	Atualização completa da página se "true"
HX-Retarget	Substituir hx-target por seletor CSS
HX-Reswap	Substituir estratégia hx-swap
HX-Trigger	Acionar eventos client-side
HX-Push-Url	Inserir URL no histórico do navegador

Eventos

Eventos de Ciclo de Vida

htmx:configRequest	Antes da requisição; modificar params/headers
htmx:beforeRequest	Logo antes da chamada AJAX
htmx:afterRequest	Após a requisição ser concluída
htmx:beforeSwap	Antes do conteúdo ser trocado
htmx:afterSwap	Após o conteúdo ser trocado
htmx:afterSettle	Após o DOM se estabilizar após o swap
htmx:responseError	Servidor retornou status de erro

Ouvindo Eventos

```
document.body.addEventListener("htmx:afterSwap", (e) => {
  console.log("Swapped:", e.detail.target);
});

// Cancel a request
document.body.addEventListener("htmx:configRequest", (e) => {
  if (!confirm("Proceed?")) e.preventDefault();
});
```

Extensões

Usando Extensões

```
<script src="https://unpkg.com/htmx-ext-json-enc"></script>

<div hx-ext="json-enc">
  <form hx-post="/api/data">
    <input name="title" />
    <button>Submit as JSON</button>
  </form>
</div>
```

Extensões Populares

json-enc	Codificar corpo da requisição como JSON
loading-states	Gerenciar classes CSS de estado de carregamento
head-support	Fundir tags <head> das respostas
preload	Pré-carregar links no mousedown/hover
sse	Suporte a Server-Sent Events
ws	Suporte a WebSocket
response-targets	Diferentes targets para respostas de erro

Indicadores

Indicador de Carregamento

```
<button hx-get="/slow" hx-indicator="#spinner">
  Load Data
</button>
<span id="spinner" class="htmx-indicator">Loading...</span>
```

CSS do Indicador

```
.htmx-indicator { display: none; }
.htmx-request .htmx-indicator { display: inline; }
.htmx-request.htmx-indicator { display: inline; }
```

htmx adiciona a classe htmx-request ao elemento durante requisições

Desabilitar Durante Requisição

```
<button hx-post="/submit" hx-disabled-elt="this">
  Submit
</button>
```

hx-disabled-elt adiciona o atributo disabled durante a requisição

Padrões Comuns

Busca Ativa

```
<input type="search" name="q"
  hx-get="/search"
  hx-trigger="keyup changed delay:300ms"
  hx-target="#results" />
<div id="results"></div>
```

Scroll Infinito

```
<tr hx-get="/rows?page=2"
    hx-trigger="revealed"
    hx-swap="afterend">
  <td>Loading more...</td>
</tr>
```

Referência Rápida htmx

Excluir com Confirmação

```
<button hx-delete="/item/42"
        hx-confirm="Delete this item?"
        hx-target="closest .item"
        hx-swap="outerHTML">
  Delete
</button>
```

Edição Inline

```
<div hx-get="/edit/1" hx-trigger="click"
     hx-swap="outerHTML">
  Click to edit this text
</div>
```