

htmx Quick Reference

Attributes, triggers, swapping, events, extensions

Basics

Installation

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

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

How It Works

HTML-driven	Issue AJAX from HTML attributes, no JS needed
Swap content	Server returns HTML fragments, htmx swaps them in
Progressive	Enhances standard HTML; degrades gracefully
REST-friendly	Works with any server-side framework

Attributes

Core Request Attributes

hx-get="/url"	Issue GET request to URL
hx-post="/url"	Issue POST request to URL
hx-put="/url"	Issue PUT request to URL
hx-patch="/url"	Issue PATCH request to URL
hx-delete="/url"	Issue DELETE request to URL

Behavior Attributes

hx-trigger	Event that triggers the request
hx-target	Element to swap content into
hx-swap	How to swap the response content
hx-select	Select a subset of the response HTML
hx-vals	Add extra values to the request
hx-confirm	Show confirm dialog before request
hx-disable	Disable htmx processing on element

Triggers

Trigger Syntax

```
<!-- 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" />
```

Trigger Modifiers

changed	Only fire if value changed
delay:Ns	Wait N seconds before firing
throttle:Nms	Throttle to once per N ms
once	Trigger only the first time
from:selector	Listen on a different element
every Ns	Poll every N seconds
load	Fire on element load
revealed	Fire when element scrolls into view

Targets

Target Selection

```
<!-- 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>
```

Target Keywords

this	The element itself (default)
closest <sel>	Closest ancestor matching selector
find <sel>	First descendant matching selector
next <sel>	Next sibling matching selector
previous <sel>	Previous sibling matching selector

Swapping

Swap Strategies

innerHTML	Replace inner content (default)
outerHTML	Replace entire target element
afterbegin	Prepend inside target
beforeend	Append inside target
beforebegin	Insert before target
afterend	Insert after target
delete	Remove target element
none	No swap, just fire events

Swap Modifiers

```
<!-- 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

Request Headers (sent by htmx)

HX-Request	Always "true" for htmx requests
HX-Target	ID of the target element
HX-Trigger	ID of the triggered element
HX-Trigger-Name	Name of the triggered element
HX-Current-URL	Current URL of the browser
HX-Prompt	User response from hx-prompt

Response Headers (sent by server)

HX-Redirect	Client-side redirect to URL
HX-Refresh	Full page refresh if "true"
HX-Retarget	Override hx-target with CSS selector
HX-Reswap	Override hx-swap strategy
HX-Trigger	Trigger client-side events
HX-Push-Ur1	Push URL into browser history

Events

Lifecycle Events

htmx:configRequest	Before request; modify params/headers
htmx:beforeRequest	Just before AJAX call
htmx:afterRequest	After request completes
htmx:beforeSwap	Before content is swapped in
htmx:afterSwap	After content is swapped in
htmx:afterSettle	After DOM settles from swap
htmx:responseError	Server returned error status

Listening to Events

```
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();  
});
```

Extensions

Using Extensions

```
<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>
```

Popular Extensions

json-enc	Encode request body as JSON
loading-states	Manage loading state CSS classes
head-support	Merge <head> tags from responses
preload	Preload links on mousedown/hover
sse	Server-Sent Events support
ws	WebSocket support
response-targets	Different targets for error responses

Indicators

Loading Indicator

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

Indicator CSS

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

htmx adds htmx-request class to element during requests

Disable During Request

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

hx-disabled-elt adds disabled attribute during request

Common Patterns

Active Search

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

Infinite Scroll

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

htmx Quick Reference

Delete with Confirmation

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

Inline Editing

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