

# 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

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

## Attributes

### Core Request Attributes

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

### Behavior Attributes

<b>hx-trigger</b>	Event that triggers the request
<b>hx-target</b>	Element to swap content into
<b>hx-swap</b>	How to swap the response content
<b>hx-select</b>	Select a subset of the response HTML
<b>hx-vals</b>	Add extra values to the request
<b>hx-confirm</b>	Show confirm dialog before request
<b>hx-disable</b>	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

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

<b>this</b>	The element itself (default)
<b>closest &lt;sel&gt;</b>	Closest ancestor matching selector
<b>find &lt;sel&gt;</b>	First descendant matching selector
<b>next &lt;sel&gt;</b>	Next sibling matching selector
<b>previous &lt;sel&gt;</b>	Previous sibling matching selector

## Swapping

### Swap Strategies

<b>innerHTML</b>	Replace inner content (default)
<b>outerHTML</b>	Replace entire target element
<b>afterbegin</b>	Prepend inside target
<b>beforeend</b>	Append inside target
<b>beforebegin</b>	Insert before target
<b>afterend</b>	Insert after target
<b>delete</b>	Remove target element
<b>none</b>	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)

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

### Response Headers (sent by server)

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

## Events

### Lifecycle Events

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

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