HTMX 快速参考

属性、触发器、内容替换、事件、扩展

基础

安装

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

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

工作原理

`HTML-driven`	通过 HTML 属性发起 AJAX，无需写 JS
`Swap content`	服务端返回 HTML 片段，htmx 替换到页面
`Progressive`	增强标准 HTML；优雅降级
`REST-friendly`	兼容任意服务端框架

属性

核心请求属性

`hx-get="/url"`	向 URL 发起 GET 请求
`hx-post="/url"`	向 URL 发起 POST 请求
`hx-put="/url"`	向 URL 发起 PUT 请求
`hx-patch="/url"`	向 URL 发起 PATCH 请求
`hx-delete="/url"`	向 URL 发起 DELETE 请求

行为属性

`hx-trigger`	触发请求的事件
`hx-target`	将响应内容替换到哪个元素
`hx-swap`	如何替换响应内容
`hx-select`	从响应 HTML 中选取子集
`hx-vals`	向请求添加额外参数
`hx-confirm`	请求前显示确认对话框
`hx-disable`	禁用元素上的 htmx 处理

触发器

触发器语法

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

触发器修饰符

`changed`	仅值变化时触发
`delay:Ns`	等待 N 秒后触发
`throttle:Nms`	每 N 毫秒最多触发一次
`once`	仅触发一次
`from:selector`	监听另一个元素的事件
`every Ns`	每 N 秒轮询一次
`load`	元素加载时触发
`revealed`	元素滚动进入视口时触发

目标

目标选择

```
<!-- 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>
```

目标关键字

`this`	元素自身（默认）
`closest <sel>`	匹配选择器的最近祖先
`find <sel>`	第一个匹配选择器的后代
`next <sel>`	匹配选择器的下一个兄弟
`previous <sel>`	匹配选择器的上一个兄弟

替换策略

替换方式

`innerHTML`	替换内部内容（默认）
`outerHTML`	替换整个目标元素
`afterbegin`	在目标内部开头插入
`beforeend`	在目标内部末尾追加
`beforebegin`	在目标前插入
`afterend`	在目标后插入
`delete`	移除目标元素
`none`	不替换，仅触发事件

替换修饰符

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

请求头

请求头（htmx 发送）

`HX-Request`	htmx 请求始终为 "true"
`HX-Target`	目标元素的 ID
`HX-Trigger`	触发元素的 ID
`HX-Trigger-Name`	触发元素的 name
`HX-Current-URL`	浏览器当前 URL
`HX-Prompt`	hx-prompt 的用户输入

响应头（服务端发送）

`HX-Redirect`	客户端重定向到 URL
`HX-Refresh`	值为 "true" 时整页刷新
`HX-Retarget`	覆盖 hx-target，用 CSS 选择器
`HX-Reswap`	覆盖 hx-swap 策略
`HX-Trigger`	触发客户端事件
`HX-Push-Url`	将 URL 推入浏览器历史

事件

生命周期事件

`htmx:configRequest`	请求前；可修改参数/头
`htmx:beforeRequest`	AJAX 调用前
`htmx:afterRequest`	请求完成后
`htmx:beforeSwap`	内容替换前
`htmx:afterSwap`	内容替换后
`htmx:afterSettle`	DOM 从替换中稳定后
`htmx:responseError`	服务器返回错误状态

监听事件

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

扩展

使用扩展

```
<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>
```

常用扩展

`json-enc`	将请求体编码为 JSON
`loading-states`	管理加载状态 CSS 类
`head-support`	合并响应中的 <head> 标签
`preload`	鼠标按下/悬停时预加载链接
`sse`	Server-Sent Events 支持
`ws`	WebSocket 支持
`response-targets`	错误响应使用不同目标

加载指示器

加载指示器

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

指示器 CSS

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

请求期间 htmx 会给元素添加 htmx-request 类

请求期间禁用

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

hx-disabled-elt 在请求期间添加 disabled 属性

常用模式

实时搜索

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

无限滚动

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

带确认的删除

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

内联编辑

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