Hypermedia & HTMX

A Different Approach To Full Stack Development

Jeremie Bornais

Workshop Code Can Be Found Here
github.com/jere-mie/htmx-workshop

github.com/jere-mie/htmx-workshop

Hello!

My Name is Jeremie Bornais

- Software Engineer @ RIIS LLC, Publicis Sapient, Assent, GreenShield
- Former President of the UWindsor Computer Science Society
- Co-Founder of WinHacks & BorderHacks
- Avid Hackathon Enthusiast
- Strongly dislikes React

@jere-mie in/jeremie-bornais

Agenda

github.com/jere-mie/htmx-workshop

Intro & Set-Up	[5 min]
What is “Traditional” Full Stack Development?	[5 min]
What are Hypermedia-Driven Web Applications?	[3 min]
What is HTMX?	[2 min]
Interactive Demo	[40 min]
Q&A, Next Steps	[5 min]

Workshop Code

All Workshop Code Can Be Found at:

github.com/jere-mie/htmx-workshop

Ensure you have Python 3.10+ installed,
or use GitHub Codespaces

github.com/jere-mie/htmx-workshop

Workshop Prerequisites

To get the most out of this workshop, hackers should have the following:

- A basic understanding of web development, including HTML and CSS
- Experience coding in at least one programming language

Nice to haves:

- Experience building a REST API
- An open mind
- A distaste for the current JavaScript ecosystem

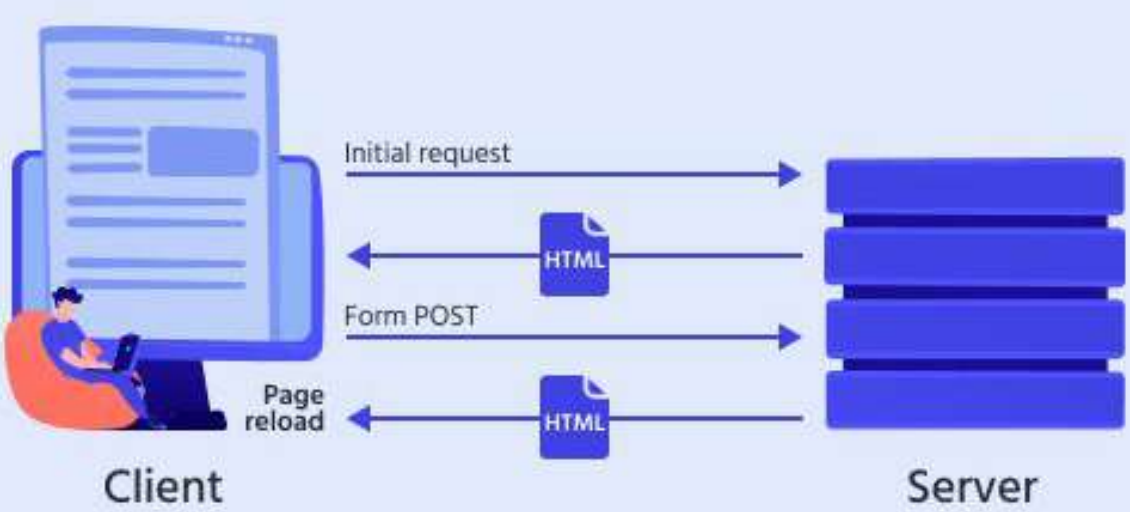
github.com/jere-mie/htmx-workshop

What is “Traditional” Full Stack Development?

Multi Page Applications (MPAs) + Single Page Applications (SPAs)

Multi-Page Applications

Back end directly renders HTML pages for user (Laravel, Django, Ruby on Rails, etc.)

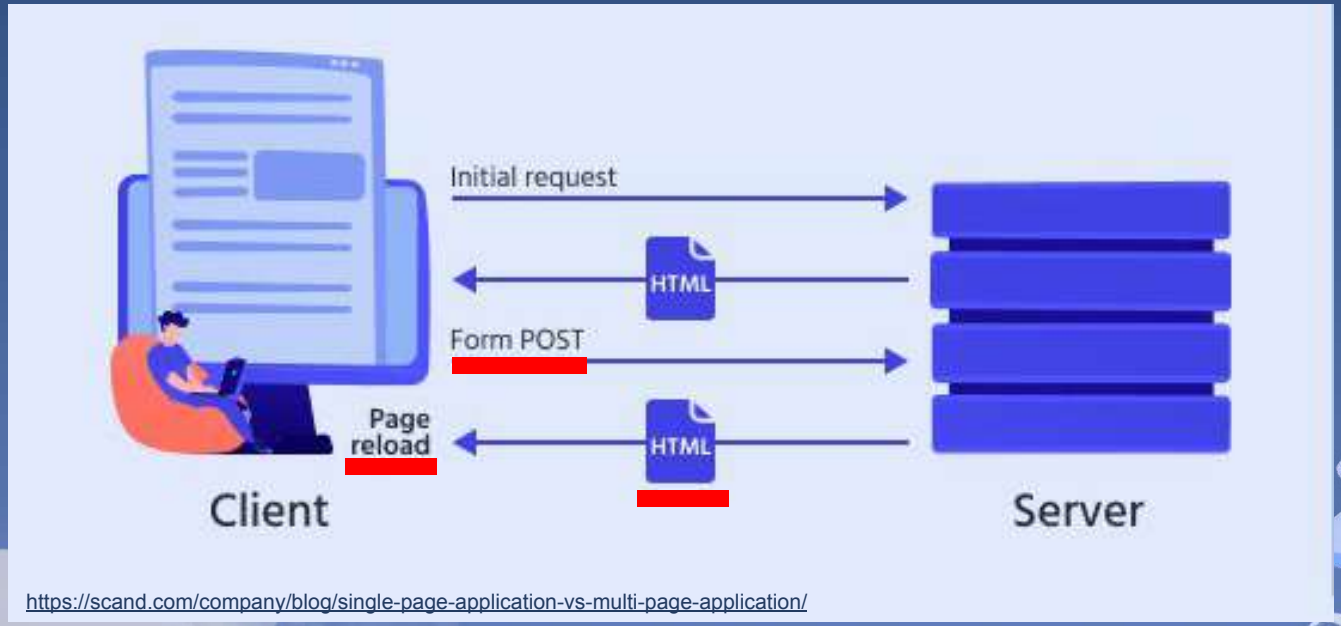


https://scand.com/company/blog/single-page-application-vs-multi-page-application/

github.com/jere-mie/htmx-workshop

Multi-Page Applications

Back end directly renders HTML pages for user (Laravel, Django, Ruby on Rails, etc.)

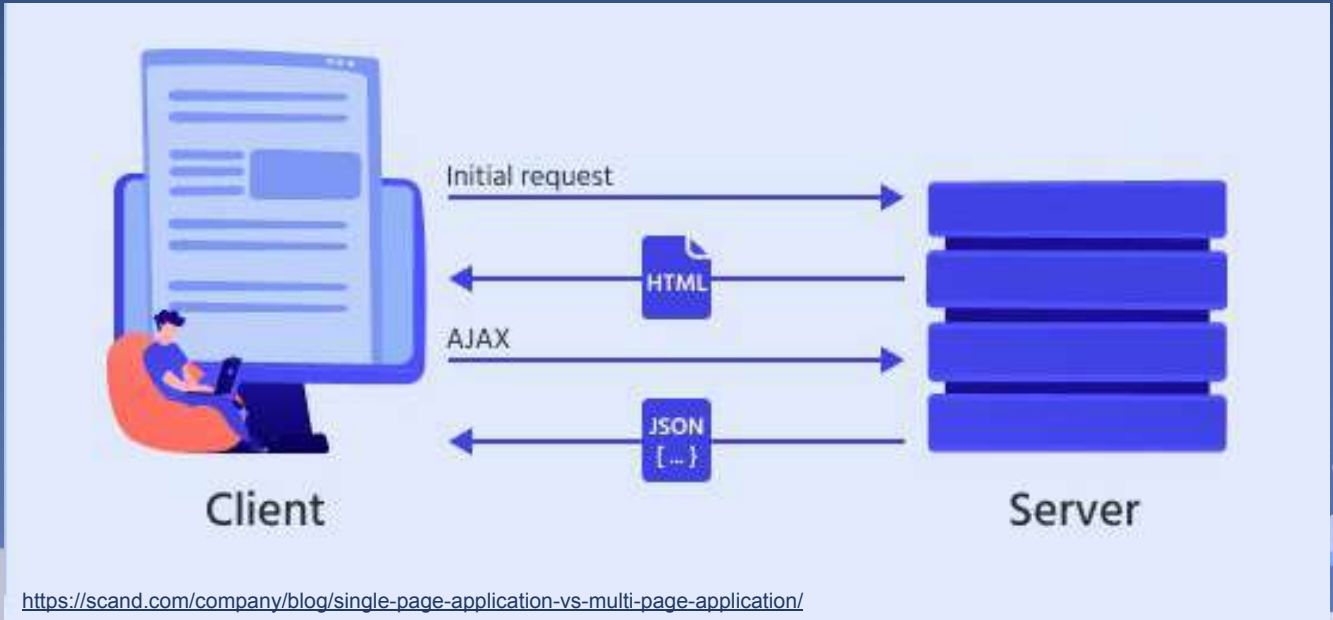


https://scand.com/company/blog/single-page-application-vs-multi-page-application/

github.com/jere-mie/htmx-workshop

Single-Page Applications

Front end calls back end, back end returns JSON, front end turns JSON to HTML (React + Express, Svelte + Flask, Vue + Spring Boot, NextJS, etc.)

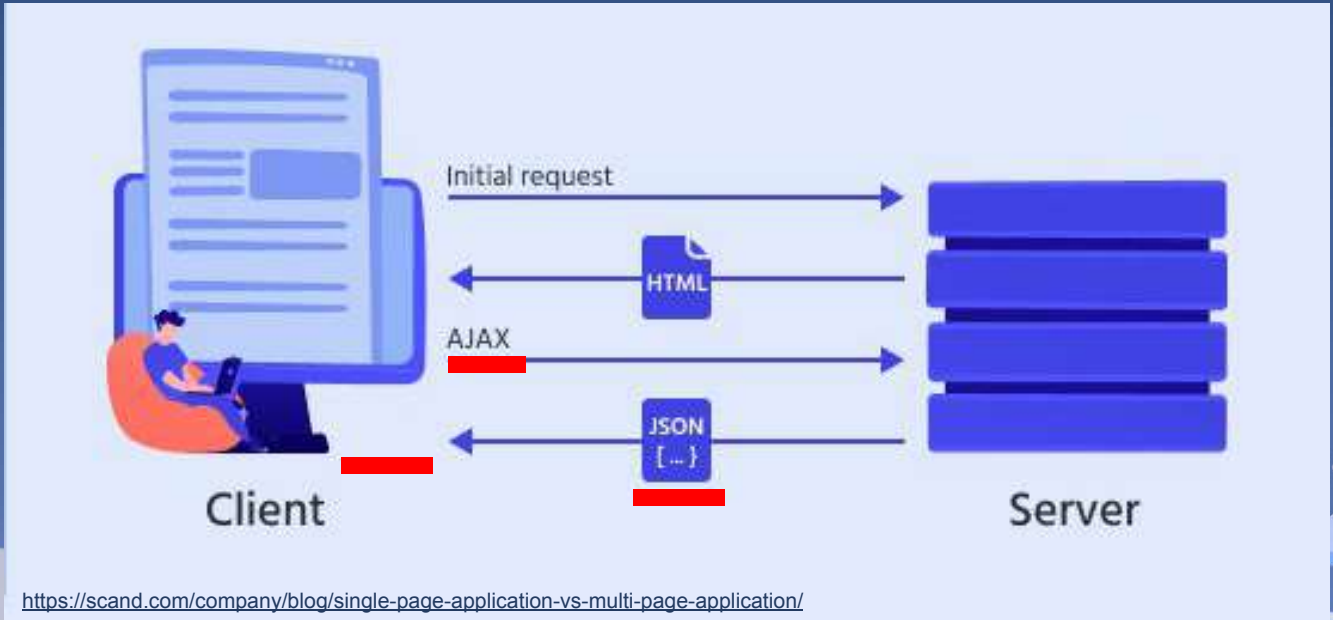


https://scand.com/company/blog/single-page-application-vs-multi-page-application/

github.com/jere-mie/htmx-workshop

Single-Page Applications

Front end calls back end, back end returns JSON, front end turns JSON to HTML (React + Express, Svelte + Flask, Vue + Spring Boot, NextJS, etc.)

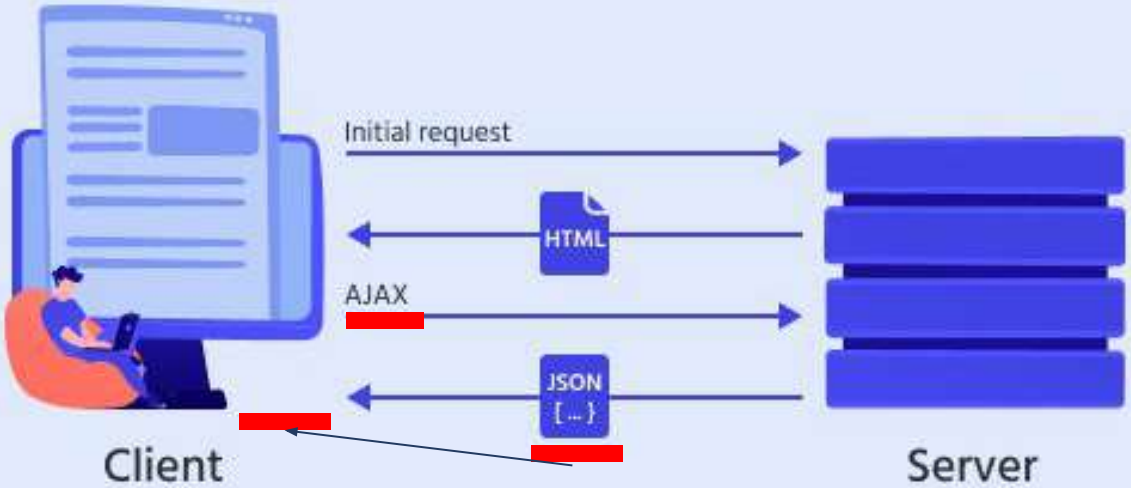


https://scand.com/company/blog/single-page-application-vs-multi-page-application/

github.com/jere-mie/htmx-workshop

Single-Page Applications

Front end calls back end, back end returns JSON, front end turns JSON to HTML (React + Express, Svelte + Flask, Vue + Spring Boot, NextJS, etc.)



Client creates HTML from JSON using JavaScript

https://scand.com/company/blog/single-page-application-vs-multi-page-application/

github.com/jere-mie/htmx-workshop

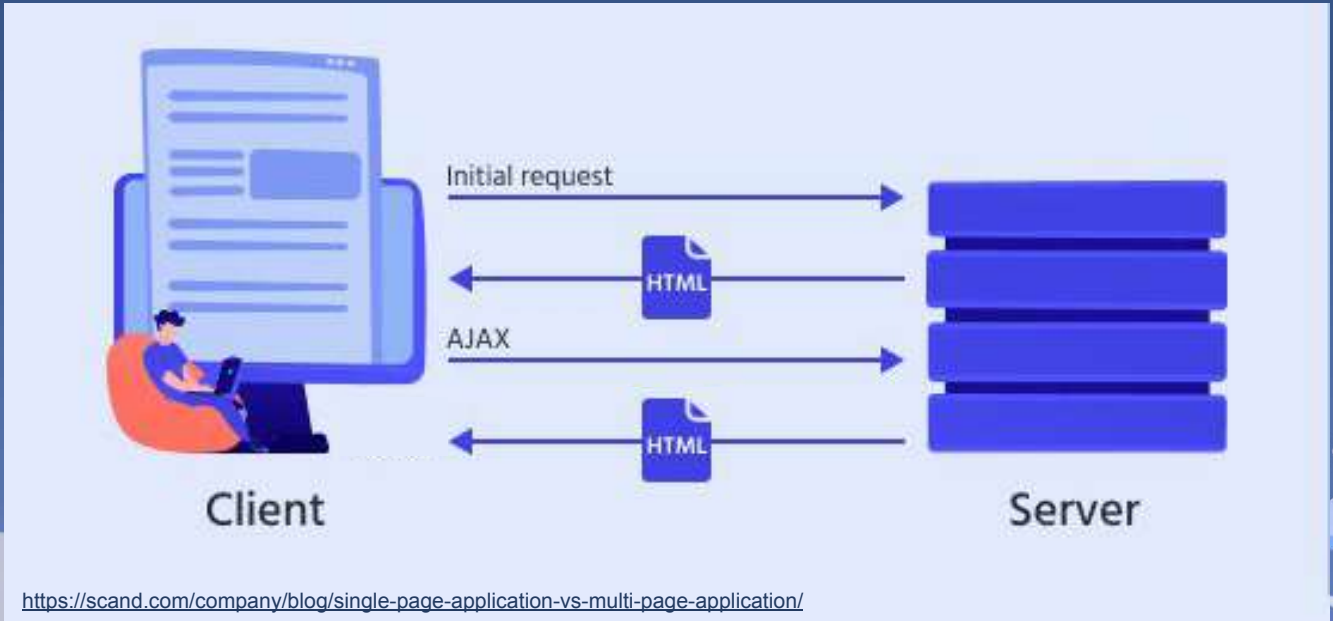
What are Hypermedia-Driven Applications (HDAs)?

The Best of Both Worlds?

github.com/jere-mie/htmx-workshop

Hypermedia-Driven Applications

Front end calls backend, back end sends partial HTML pages (Any Back End Library + HTMX)

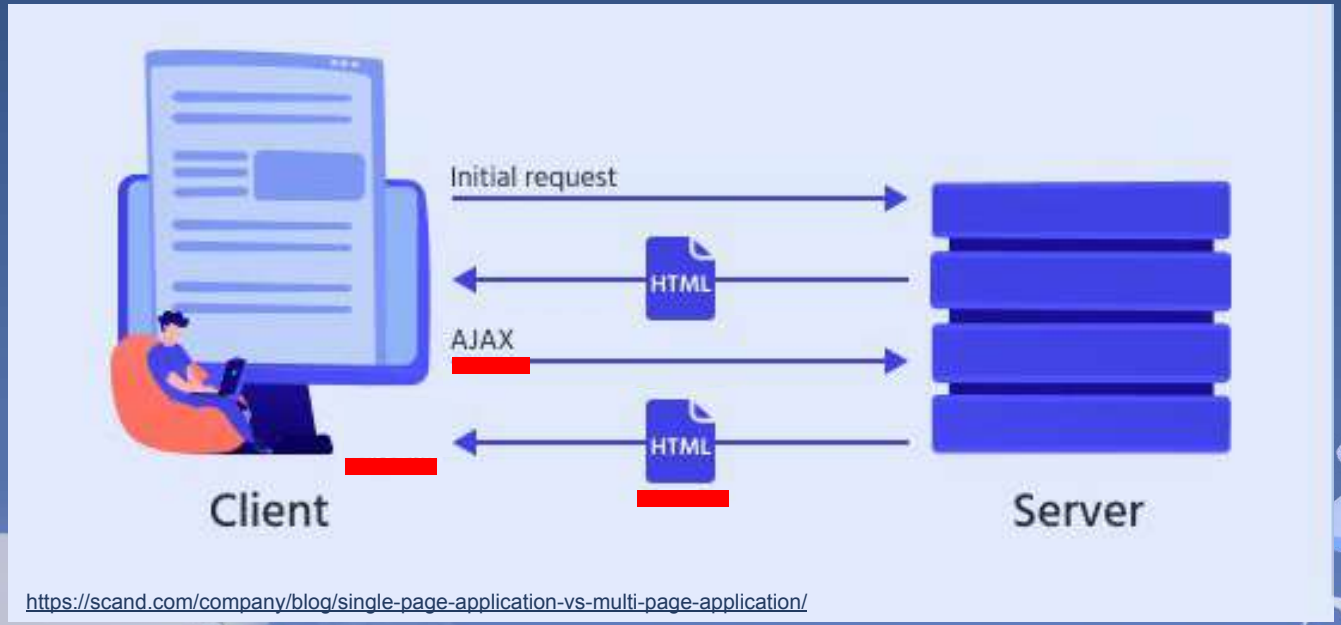


https://scand.com/company/blog/single-page-application-vs-multi-page-application/

github.com/jere-mie/htmx-workshop

Hypermedia-Driven Applications

Front end calls backend, back end sends partial HTML pages
(Any Back End Library + HTMX)

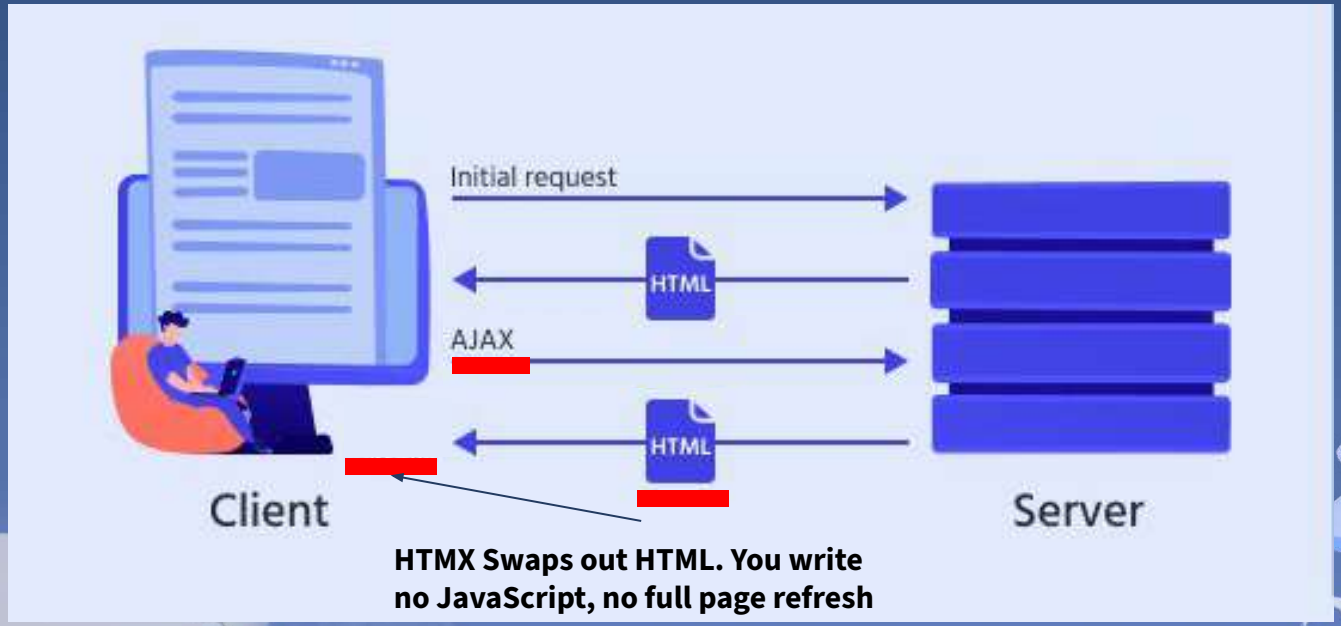


https://scand.com/company/blog/single-page-application-vs-multi-page-application/

github.com/jere-mie/htmx-workshop

Hypermedia-Driven Applications

Front end calls backend, back end sends partial HTML pages
(Any Back End Library + HTMX)



HTMX Swaps out HTML. You write no JavaScript, no full page refresh

github.com/jere-mie/htmx-workshop

Summary

Multi-Page Applications (MPAs)

- **Full-page reloads** for every interaction.

- Slower user experience due to frequent page loads.

Single-Page Applications (SPAs)

- Single-page load, dynamic updates via JavaScript (AJAX).

- **Complex development** with heavy front-end focus.

Hypermedia-Driven Applications (e.g., HTMX)

- Server-rendered HTML with partial updates via AJAX.

- Simple setup with minimal JavaScript.

github.com/jere-mie/htmx-workshop

What is HTMX?

</> htmx

github.com/jere-mie/htmx-workshop

What is HTMX?

(Stolen from HTMX.org)

“htmx gives you access to AJAX, CSS Transitions, WebSockets and Server Sent Events directly in HTML, using attributes, so you can build modern user interfaces with the simplicity and power of hypertext

htmx is small (~14k min.gz’d), dependency-free, extendable & has reduced code base sizes by 67% when compared with react”

github.com/jere-mie/htmx-workshop

What is HTMX?

Rather than using JavaScript to define how we can interact with our webpage, we can encode that information directly within our HTML via hx- attributes.

It is backend-agnostic: as long as your backend can return HTML, you can use HTMX with any programming language!

Remember - HTMX is all about swapping out HTML with other HTML

github.com/jere-mie/htmx-workshop

What is HTMX?

```
<script src="https://unpkg.com/htmx.org@2.0.2"></script>
<!-- have a button POST a click via AJAX -->
<button hx-post="/clicked" hx-swap="outerHTML">
  Click Me
</button>
```

“When a user clicks on this button, issue an AJAX request to /clicked, and replace the entire button with the HTML response”

Interactive Demo

github.com/jere-mie/htmx-workshop

github.com/jere-mie/htmx-workshop

What We'll Be Making

github.com/jere-mie/htmx-workshop

Simple Blogging App

Key Features:

- See all blog articles
- Create new article
- Delete articles
- Editing existing articles

We'll be taking a basic skeleton application and progressively adding new features using HTMX

github.com/jere-mie/htmx-workshop

Coding Time...

github.com/jere-mie/htmx-workshop

Next Steps and Resources

Next Steps and Resources

Feeling nerdy? Like Theory?

- HTMX essays (htmx.org/essays)
- Hypermedia Book (hypermedia.systems)

Practical HTMX Resources

- Simple code snippets (htmx.org/examples)
- HTMX Documentation (htmx.org/docs)

Flask Resources

- My Flask workshop (github.com/jere-mie/flask-workshop)

github.com/jere-mie/htmx-workshop

Thank you!

Any questions?