HTMX and ASP.NET Core (I don’t know how to React)

Maarten Balliauw

🦋 @maartenballiauw.be

https://blog.maartenballiauw.be/

Duende Software

Let’s take a step back…

- What do we do as developers?
  - Build applications
  - Provide a great user experience
- Do we *really* need React/Angular/Vue/… for that?
  - Pick a framework
  - Node, npm (or yarn), asset pipelines (gulp, grunt, vite, …)
  - So many options!

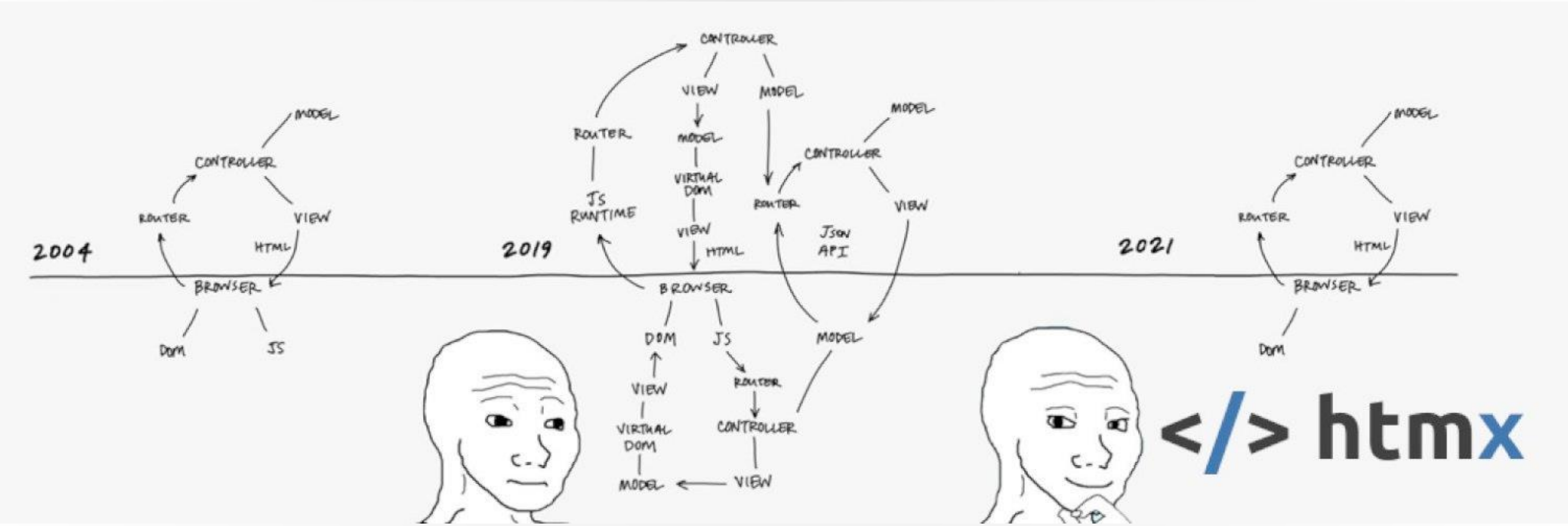
Going further…

- Do we *really* need SPAs?
  - Manage state on client and server
  - Identity and authentication
  - Manage routing
  - API design with status codes, hypermedia, paging, POST or PUT?
  - “HTML as JSON, minus the markup, plus complexity”

But the user experience!

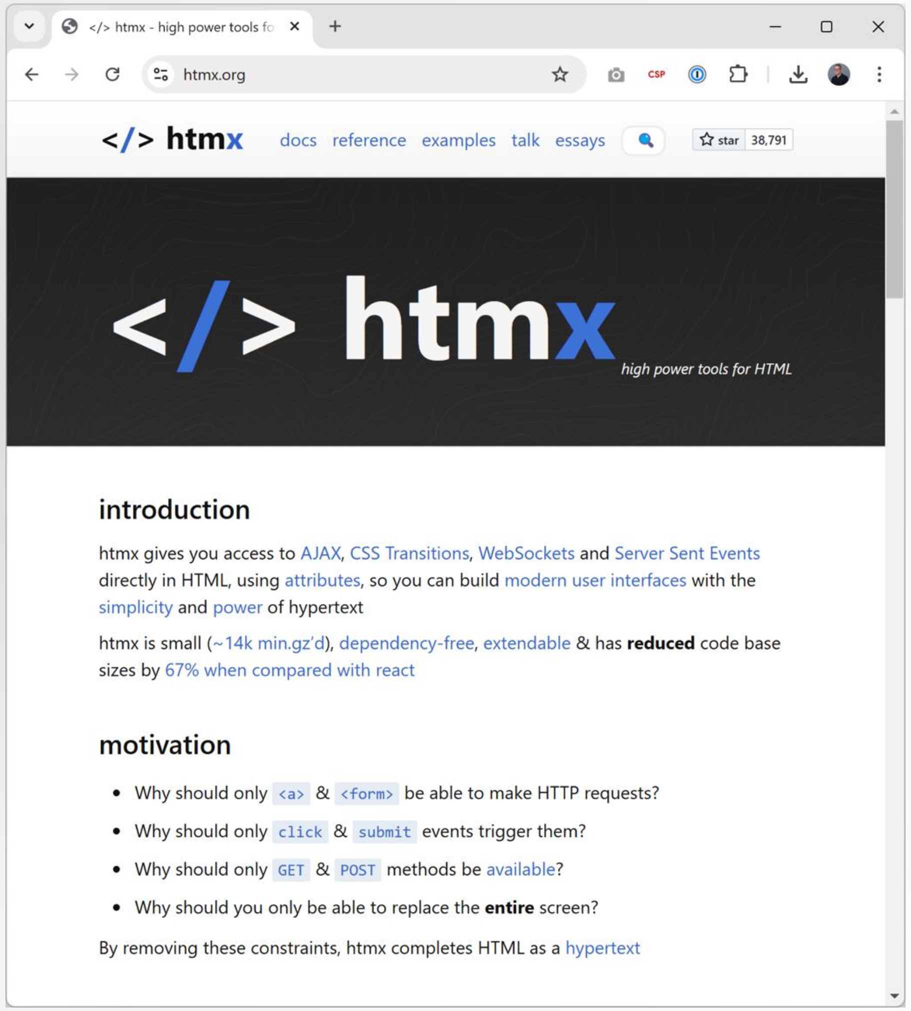
- What do we even want from it?
  - Nice URLs?
  - Browser history?
  - Page transitions?
  - Open in new tab that actually works?
- Performance!
  - Fast rendering? Critical Rendering Path
  - Small payloads?
  - Ajax?
- **We have all that already!**
  Browser, HTTP request/response, HTML/CSS/JS
  Hypermedia

But the user experience!



HTMX

https://htmx.org by Carson Gross



introduction

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

motivation

- Why should only `<a>` & `<form>` be able to make HTTP requests?
- Why should only `click` & `submit` events trigger them?
- Why should only `GET` & `POST` methods be available?
- Why should you only be able to replace the **entire** screen?

By removing these constraints, htmx completes HTML as a hypertext

HTMX

https://htmx.org by Carson Gross

~49kb (v2.0.3)

~14kb (gzipped)

1 JavaScript file

```
<script src="htmx.min.js"></script>
```

HTMX

https://htmx.org by Carson Gross

~49kb (v2.0.3)

~14kb (gzipped)

1 JavaScript file

Attributes, CSS classes, headers, events, JS API

```
<script src="htmx.min.js"></script>

<button hx-get="/hello"
      hx-swap="outerHTML">
   Say hello
</button>

<i class="htmx-indicator spinner"></i>
```

“HTMX is HTML if they kept going”

- Any DOM element can trigger a request
- More than GET and POST
- Any interaction results in HTML & updates UI
- State management is server-side

Hello, World

A quick look at HTMX

HTMX attributes

- HTTP request
  - hx-get, hx-post, hx-put, hx-patch, hx-delete
- Target
  - hx-target takes a CSS selector
  - closest, next, previous modifiers
- Request header
  - HX-Request

More attributes

Triggers and events

More HTMX attributes!

- HTTP request
- Target
- Request header
- Event trigger
  - hx-trigger on change, submit, click, mouse enter, ...
  - Special events: load, revealed, intersect, every 2s
- Event modifier
  - once, changed, delay:1s, throttle:1s, from:selector

State management

Back to the server side

State management

- Your responsibility
- Static fields, cookies, sessions, database, …

Headers

Metadata for request/response

Request headers

- HX-Request – Available on every HTMX request
- HX-Target, HX-Trigger, HX-Trigger-Name – Event info
- HX-Current-Url – Current page URL
- …
- https://htmx.org/reference/#request_headers

Response headers

- HX-Location, HX-Redirect – Client-side redirect
- HX-Push-Url – Add URL to history stack
- HX-Refresh – Refresh page
- HX-Reswap, HX-Retarget, HX-Reselect – Pick DOM element
- HX-Trigger, … – Trigger event on client
- …
- https://htmx.org/reference/#response_headers

Htmx.net

Making the server-side better

Htmx.net

- `dotnet add package Htmx`
  - Utilities on Request/Response, e.g. to set headers
- `dotnet add package Htmx.TagHelpers`
  - Tag helpers to build page/controller/action attributes

There is more in HTMX…

- CSS Transitions
- Confirmations/prompts
- Boosting
- Web Sockets and Server Sent Events (SSE)
- History support
- Extensions
- Security
- Configuration options

Creating a typeahead

ASP.NET Core is great! Especially with a sprinkle of HTMX

ASP.NET Core / Razor views

- Seen in the demo:
  - View with layout – full HTML
  - Partial (no layout) – partial HTML
- But also:
  - View components – partial HTML with server-side logic
  - Tag helpers – server-side logic

Validation

Server-side validation, client-side feel

Side step: Hyperscript

- https://hyperscript.org/
- Easy and approachable language for DOM event handling
- Good companion to HTMX

Validation

- All server-side
  - Easy to do more complex validations
  - Use validation framework of choice
    - (e.g. FluentValidation)
  - No need to duplicate validation logic
    - (e.g. with jQuery Unobtrusive Validation)

Out-of-band swaps

Update other parts of the UI

Out-of-band swaps

- Use hx-swap-oob="true" with element id
    - Easy! But cumbersome in complex layouts.
- Use HTMX triggers
    - HX-Trigger response header
    - Set hx-trigger="custom-event from:body"
    - ⚠️ Do not forget event bubbling in HTMX needs from:body

Modals & prompting

Adding in-page dialogs

Modals and prompting

- Modal
  - Add hx-target to a placeholder dialog element (e.g. using Bootstrap)
  - Render dialog content
  - (similar flow with e.g. Bootstrap tabs)
- Confirm
  - Add hx-confirm to show a message
  - Customizable with htmx:confirm event handler
- Prompt
  - Add hx-prompt to use browser prompt
  - Prompt response in HX-Prompt header
  - No custom option (use your own logic with a custom modal)

Shopping cart

(thanks Khalid Abuhakmeh)

Shopping cart

- Product listing
  - Handles search/category
  - Renders partial + pushes URL into history
- Product details/options modal
  - Targets modal container element
  - Uses OOB swaps for cart button updates
- Update shopping cart
  - Targets modal container element
  - Uses hx-vals to send product/option combination, compatible with model binding

Conclusion

- No need for complex client-side SPA
- Build server-side apps with client-side feel
- HTMX
  - Any DOM element can trigger a request
    - Use attributes to specify behavior
  - Any interaction results in HTML & updates UI
    - Use optional headers for extra features
- ASP.NET Core
  - View, partial view, view component, tag helpers
  - Strong model binding features
  - Htmx.net

Thank you!

Maarten Balliauw

🦋 @maartenballiauw.be

https://blog.maartenballiauw.be/

Duende Software