




Why Overcomplicate When HTML Will Do?



Jonathan Knapp
Coffee and Code



July 26-28

at Hyland
Westlake, Ohio



clevelandgivecamp.org

Me

Independent consultant + contractor at Coffee and Code.

Working on web-based projects since 2006.

COFFEE
AND
CODE

My Work Goals

Constantly provide value in the short term.

Make the long term reasonably painless.

When possible, reduce the amount of code needed to run an application.

Complexity Budget

Hosting

Caching

Infrastructure

Code quality

User experience

Design and functionality

Ever-changing business requirements

We have made our jobs harder
than necessary.

Allure of the Unknown

We crave knowledge.

We want to iterate and evolve.

We constantly seek greener pastures.

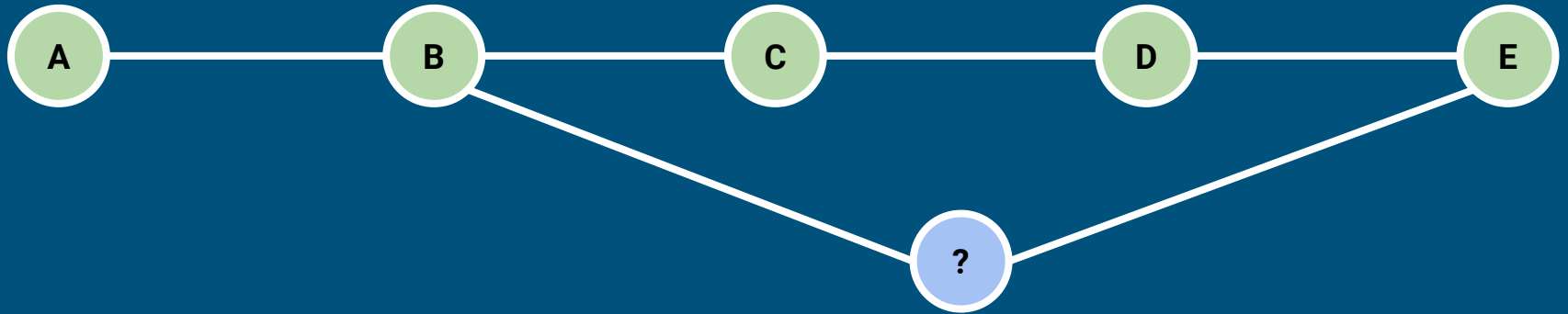
New is “Cool”

Newer *has to* mean better.

We want to stay relevant.

Developer advocate is a job title.

Technological Advancement



People are scared to ask “Is this too complex?” because it sounds an awful lot like “I’m not smart enough to understand this.”

- Carson Gross, Adam Stepinski, Deniz Akşimşek in [Hypermedia Systems](#)

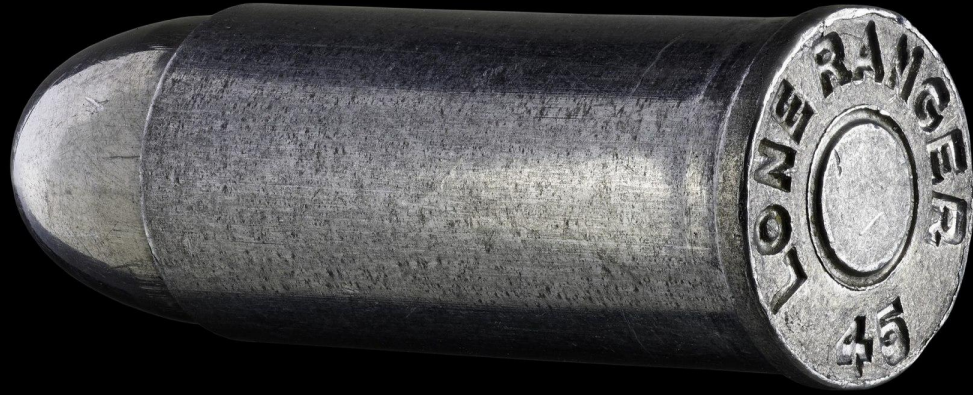
My Presentation Goals

Rethink how we can build modern web applications.

Reduce number of dependencies and technical knowledge needed.

Help development and release functionality faster.

Understand that *code is a liability*.



“the effectiveness of real silver bullets compared to lead ones is not entirely known.”

- https://en.wikipedia.org/wiki/Silver_bullet

How do we do better?

How do we do better?

1. Shared knowledge
2. Start simple
3. Use technologies that support our goals

1. Shared Knowledge

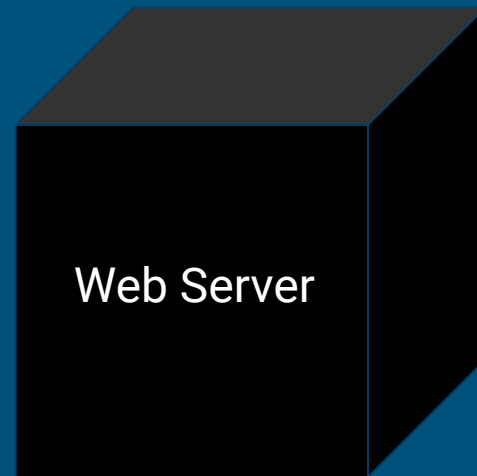
Black Box



Requests



Responses



Using DataTables

Examples

Basic initialisation

Advanced initialisation

Styling

Data sources

API

Ajax

Server-side

Plug-ins

Manual

Reference

Extensions

Plug-ins

More help

Blog

Forums

Support

FAQs

Get DataTables

Download

Purchase

Zero configuration

DataTables has most features enabled by default, so all you need to do to use it with your own tables is to call the construction function: `new DataTable('#myTable')` for plain Javascript or `$('#myTable').dataTable()` when using jQuery.

Searching, ordering and paging goodness will be immediately added to the table, as shown in this example.

10 ▾ entries per page

Search:

Name	Position	Office	Age	Start date	Salary
Airi Satou	Accountant	Tokyo	33	2008-11-28	\$162,700
Angelica Ramos	Chief Executive Officer (CEO)	London	47	2009-10-09	\$1,200,000
Ashton Cox	Junior Technical Author	San Francisco	66	2009-01-12	\$86,000
Bradley Greer	Software Engineer	London	41	2012-10-13	\$132,000
Brenden Wagner	Software Engineer	San Francisco	28	2011-06-07	\$206,850
Brielle Williamson	Integration Specialist	New York	61	2012-12-02	\$372,000
Bruno Nash	Software Engineer	London	38	2011-05-03	\$163,500
Caesar Vance	Pre-Sales Support	New York	21	2011-12-12	\$106,450
Cara Stevens	Sales Assistant	New York	46	2011-12-06	\$145,600
Cedric Kelly	Senior Javascript Developer	Edinburgh	22	2012-03-29	\$433,060

Name

Position

Office

Age

Start date

Salary

Showing 1 to 10 of 57 entries

« < 1 2 3 4 5 6 > »

Javascript

HTML

CSS

Comments (0)

The Javascript shown below is used to initialise the table shown in this example:

```
1 new DataTable('#example', {
2   layout: {
3     bottomEnd: {
4       paging: {
5         boundaryNumbers: false
6       }
7     }
8   }
9 });
```

In addition to the above code, the following Javascript library files are loaded for use in this example:

Standards!

HTTP
HTML
CSS

JavaScript standard library

HTML

Living Standard — Last Updated 19 April 2024



[One-Page Version](#)
html.spec.whatwg.org

[Multipage Version](#)
/multipage

[Version for Web Devs](#)
/dev

[PDF Version](#)
/print.pdf

[Translations](#)
日本語・簡体中文

[FAQ](#)
on GitHub

[Chat](#)
on Matrix

[Contribute on GitHub](#)
whatwg/html repository

[Commits](#)
on GitHub

[Snapshot](#)
as of this commit

[Twitter Updates](#)
@htmlstandard

[Open Issues](#)
filed on GitHub

[Open an Issue](#)
whatwg.org/newbug

[Tests](#)
web-platform-tests/html/

[Issues for Tests](#)
ongoing work

Table of contents

- 1 Introduction
- 2 Common infrastructure
- 3 Semantics, structure, and APIs of HTML documents
- 4 The elements of HTML
- 5 Microdata
- 6 User interaction
- 7 Loading web pages
- 8 Web application APIs
- 9 Communication
- 10 Web workers
- 11 Worklets
- 12 Web storage
- 13 The HTML syntax
- 14 The XML syntax
- 15 Rendering
- 16 Obsolete features
- 17 IANA considerations
- Index
- References
- Acknowledgments
- Intellectual property rights

Full table of contents

- 1 Introduction
 - 1.1 Where does this specification fit?
 - 1.2 Is this HTML5?
 - 1.3 Background

Standards

Provide a common understanding.

Assist in migrating between languages and libraries.

Provide additional functionality for free.

`<a />`

HTML Tag

How do you link to another page?

`<table>...</table>`

HTML Tag

How do you render tabular data?

2xx or 3xx

Status Code

How do you know if an HTTP request succeeded?

4xx or 5xx

Status Code

How do you know if an HTTP request failed?

Accept: mime/format

HTTP Header

How do you ask for a response in a specific format?

Content-Type: mime/format

HTTP Header

How do you know what format a response is in?

GET

HTTP Method

How do you signify you want to view a url?

(hint: it's an HTTP method)

POST

HTTP Method

How do you signify you want to send data to the server to process?

Well Actually

<dialog />

HTML Tag

How do you create a modal?

<div popover />

HTML Attribute

How do you create a tooltip?

Shared Knowledge

Familiar environment (ex: HTTP).

Lots of documentation.

Utilize existing standards.

Try to keep up to date.

2. Seek Simplicity

HTML

What's simple?

Progressive Enhancement

A software design philosophy that aims to provide as much essential content and functionality to as many users as possible, while delivering a better experience to users with more advanced web browsers.

HATEOAS (Hypertext As The Engine Of Application State)

```
<div>
  • <h1>Your Account Balance</h1>
  • <h2>$100.00</h2>
  • <a href="/account/12345/deposit">Deposit</a>
  • <a href="/account/12345/withdraw">Withdraw</a>
  • <a href="/account/12345/transfer">Transfer</a>
  • <form action="/account/12345" method="delete">
  • | • <input type="submit" value="Delete" />
  • </form>
</div>
```

HATEOAS (Hypertext As The Engine Of Application State)

```
<div>
  · · <h1>Your Account Balance</h1>
  · · <h2>-$10.00</h2>
  · · <a href="/account/12345/deposit">Deposit</a>
</div>
```

Server Rendering

What's simple?

Multi-Page Application (MPA)

GET `/widgets` -> listing of items

GET `/widgets/12412` -> show specific item

GET `/widgets/12412/edit` -> edit form for a specific item

POST `/widgets` -> create an item

PATCH/PUT `/widgets/12412` -> update an item

DELETE `/widgets/12412` -> delete an item

DataTables

10 ▾ entries per page		Search: <input type="text"/>		
Name	Position	Office	Age	Start date
▶ Prescott Bartlett	Technical Author	London	27	5/6/2011
▶ Gavin Cortez	Team Leader	San Francisco	22	10/25/2008
▶ Lael Greer	Systems Administrator	London	21	2/26/2009
▶ Gloria Little	Systems Administrator	New York	59	4/9/2009
▶ Tiger Nixon	System Architect	Edinburgh	61	4/24/2011
▶ Quinn Flynn	Support Lead	Edinburgh	22	3/2/2013
▶ Finn Camacho	Support Engineer	San Francisco	47	7/6/2009
▶ Sakura Yamamoto	Support Engineer	Tokyo	37	8/18/2009
▶ Olivia Liang	Support Engineer	Singapore	64	2/2/2011
▶ Zorita Serrano	Software Engineer	San Francisco	56	5/31/2012
Name	Position	Office	Age	Start date
Showing 1 to 10 of 57 entries		◀ ◁ 1 2 3 4 5 6 ▷ ▶		

Single Page Application (SPA)

Built for more interactive applications.

Meant to reduce complexity.

Increases scalability.

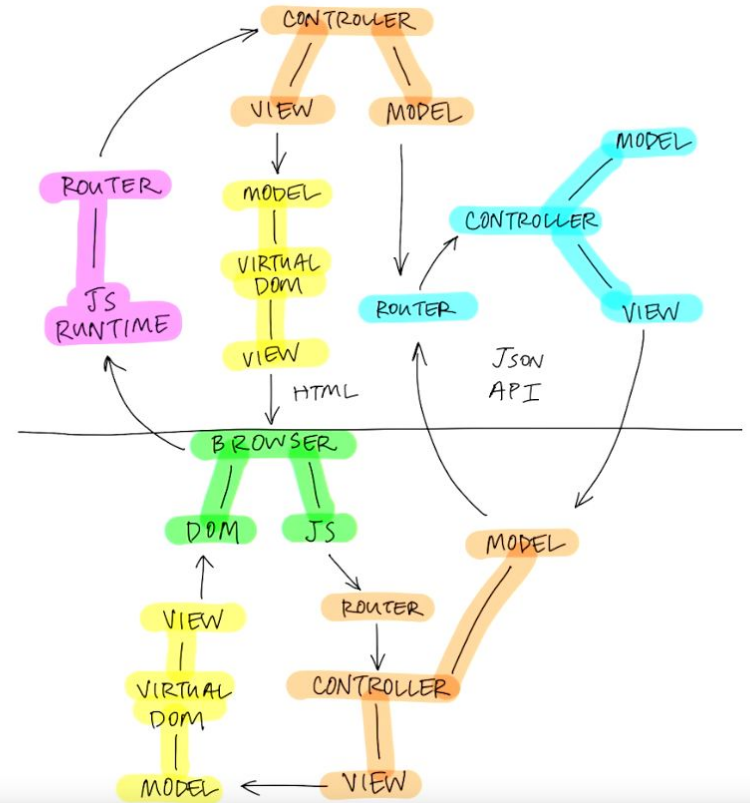
Safe Space

This space intentionally left blank.

wat

I Can't Believe It's Not Native!

by Sam Stephenson



Rendering HTML on the server?

Or server-side... rendering?



Who routes the router?



Model View Controller++



I can't ctrl+click? FFS!!!



Single Page Application (SPA)

Duplicated business logic.

Introduced complex toolchains.

Customized implementations.

Seek Simplicity

Starts with you.

You are not FaceBook & Google.

Always start simple. You will *always* want to rewrite it.

3. Supportive Technology





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.



Turbo + Stimulus + Strada

Hotwire is an alternative approach to building modern web applications without using much JavaScript by sending *HTML* instead of *JSON* over the wire.

2013

Party like it's Rails 4

Controller

app/controllers/users_controller.js

```
@user = User.find_by(params[:id])

respond_to do |format|
  format.html
  format.js
end
```

View

app/views/users/create.js.erb

```
$("<%= escape_javascript(render @user)  
%>").appendTo("#users");
```

2020

Party like it's Rails 7 and HTMX 1.0

HTML Over the Wire

Background page fetching.

Partial page updates.

Lazy loading content.

RESTful requests with multiple response types.

Web socket support.

Background Page Loads

Works on links and form submissions.

Makes requests with AJAX instead of browser default.

Swaps `<body>` content; may update `<head>`; may morph dom instead of swap.

Immediate back navigation using cached html.

Hotwire

How to Use

```
<script type="module">  
  import hotwiredTurbo from  
    'https://cdn.skypack.dev/@hotwired/turbo';  
</script>
```

HTMX

How to Use

```
<script  
src="https://unpkg.com/htmx.org@1.9.12"  
integrity="sha384-..."  
crossorigin="anonymous"></script>
```

```
<body hx-boost="true">
```

Using DataTables

Examples

Basic initialisation

Advanced initialisation

Styling

Data sources

API

Ajax

Server-side

Plug-ins

Manual

Reference

Extensions

Plug-ins

More Help

Blog

Forums

Support

FAQs

Get DataTables

Download

Purchase

Zero configuration

DataTables has most features enabled by default, so all you need to do to use it with your own tables is to call the construction function: `new DataTable('#myTable')` for plain Javascript or `$('#myTable').DataTable();` when using jQuery.

Searching, ordering and paging goodness will be immediately added to the table, as shown in this example.

10 ▾ entries per page

Search:

Name	Position	Office	Age	Start date	Salary
Airi Satou	Accountant	Tokyo	33	2008-11-28	\$162,700
Angelica Ramos	Chief Executive Officer (CEO)	London	47	2009-10-09	\$1,200,000
Ashton Cox	Junior Technical Author	San Francisco	66	2009-01-12	\$86,000
Bradley Greer	Software Engineer	London	41	2012-10-13	\$132,000
Brenden Wagner	Software Engineer	San Francisco	28	2011-06-07	\$206,850
Brielle Williamson	Integration Specialist	New York	61	2012-12-02	\$372,000
Bruno Nash	Software Engineer	London	38	2011-05-03	\$163,500
Caesar Vance	Pre-Sales Support	New York	21	2011-12-12	\$106,450
Cara Stevens	Sales Assistant	New York	46	2011-12-06	\$145,600
Cedric Kelly	Senior Javascript Developer	Edinburgh	22	2012-03-29	\$433,060

Name	Position	Office	Age	Start date	Salary
------	----------	--------	-----	------------	--------

Showing 1 to 10 of 57 entries

« ‹ 1 2 3 4 5 6 › »

Javascript

HTML

CSS

Comments (0)

The Javascript shown below is used to initialise the table shown in this example:

```
1 new DataTable('#example', {
2   layout: {
3     bottomEnd: {
4       paging: {
5         boundaryNumbers: false
6       }
8     }
9   }
10 }
```

Javascript

Partial HTML Updates

Requests to existing HTML pages can have parts of response swapped in.

Special headers and responses can be used to trim size of response.

Hotwire

How to Use

```
<turbo-frame id="widgets">  
  <a href="/widgets/new" >New Widget</a>  
  <a href="/some/link"  
data-turbo="false">Contact Us</a>  
</turbo-frame>
```

Hotwire

How to Use

```
<turbo-stream action="append" target="messages">
  <template>
    <div id="message_10">
      I show up after #message_10
    </div>
  </template>
</turbo-stream>
```

```
<turbo-stream action="prepend" target="messages">
  <template>
    <div id="message_1">
      I show up before #message_1
    </div>
  </template>
</turbo-stream>
```


HTMX

How to Use

```
<input type="text" name="q"  
  hx-get="/trigger_delay"  
  hx-select="#search-results"  
  hx-trigger="keyup delay:500ms changed"  
  hx-target="#search-results"  
  placeholder="Search..."  
>  
  
<div id="search-results"></div>
```

Using DataTables

Examples

Basic initialisation

Advanced initialisation

Styling

Data sources

API

Ajax

Server-side

Plug-ins

Manual

Reference

Extensions

Plug-ins

More Help

Blog

Forums

Support

FAQs

Get DataTables

Download

Purchase

Zero configuration

DataTables has most features enabled by default, so all you need to do to use it with your own tables is to call the construction function: `new DataTable('#myTable')` for plain Javascript or `$('#myTable').DataTable()` when using jQuery.

Searching, ordering and paging goodness will be immediately added to the table, as shown in this example.

10 entries per page

Search:

Name	Position	Office	Age	Start date	Salary
Airi Satou	Accountant	Tokyo	33	2008-11-28	\$162,700
Angelica Ramos	Chief Executive Officer (CEO)	London	47	2009-10-09	\$1,200,000
Ashton Cox	Junior Technical Author	San Francisco	66	2009-01-12	\$86,000
Bradley Greer	Software Engineer	London	41	2012-10-13	\$132,000
Brenden Wagner	Software Engineer	San Francisco	28	2011-06-07	\$206,850
Brielle Williamson	Integration Specialist	New York	61	2012-12-02	\$372,000
Bruno Nash	Software Engineer	London	38	2011-05-03	\$163,500
Caesar Vance	Pre-Sales Support	New York	21	2011-12-12	\$106,450
Cara Stevens	Sales Assistant	New York	46	2011-12-06	\$145,600
Cedric Kelly	Senior Javascript Developer	Edinburgh	22	2012-03-29	\$433,060

Showing 1 to 10 of 57 entries

« ‹ 1 2 3 4 5 6 › »

Javascript

HTML

CSS

Comments (0)

The Javascript shown below is used to initialise the table shown in this example:

```
1 new DataTable('#example', {
2   layout: {
3     bottomEnd: {
4       paging: {
5         boundaryNumbers: false
6       }
8     }
9   }
10 }
```

Javascript

2024

Party like it's Rails v8

Return to Simple

Supports DOM morphing to simplify responses.

Web sockets trigger page loads instead of sending HTML.

DOM Morphing

Virtual DOM diffing and merging of new content.

Hotwire

How to Use

model

```
class Conversation < ApplicationRecord  
  
  broadcasts_refreshes  
  
end
```

view

```
<%= turbo_stream_from @conversation %>
```

layout

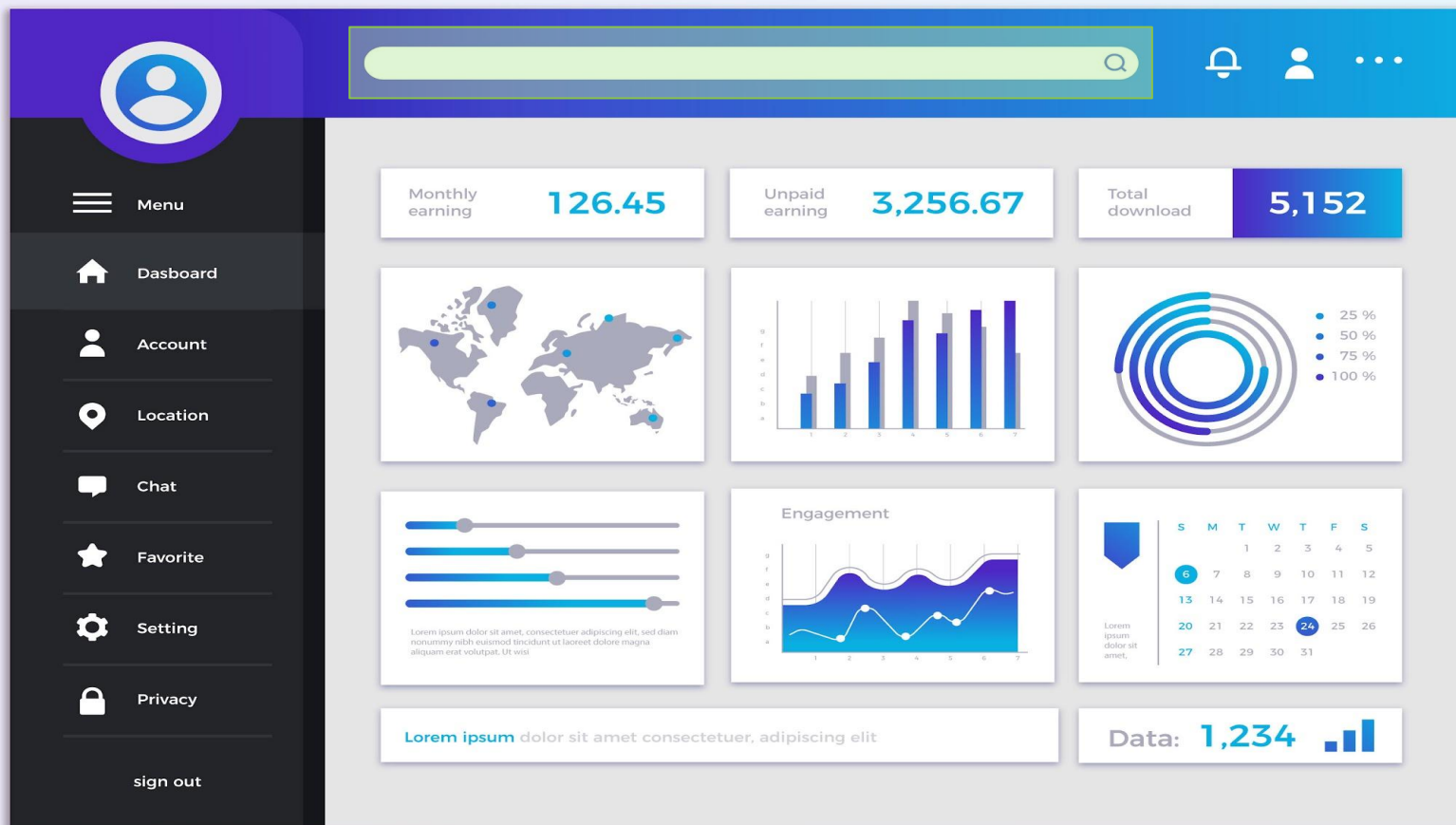
```
<%= turbo_refresh_method_tag :morph %>
```

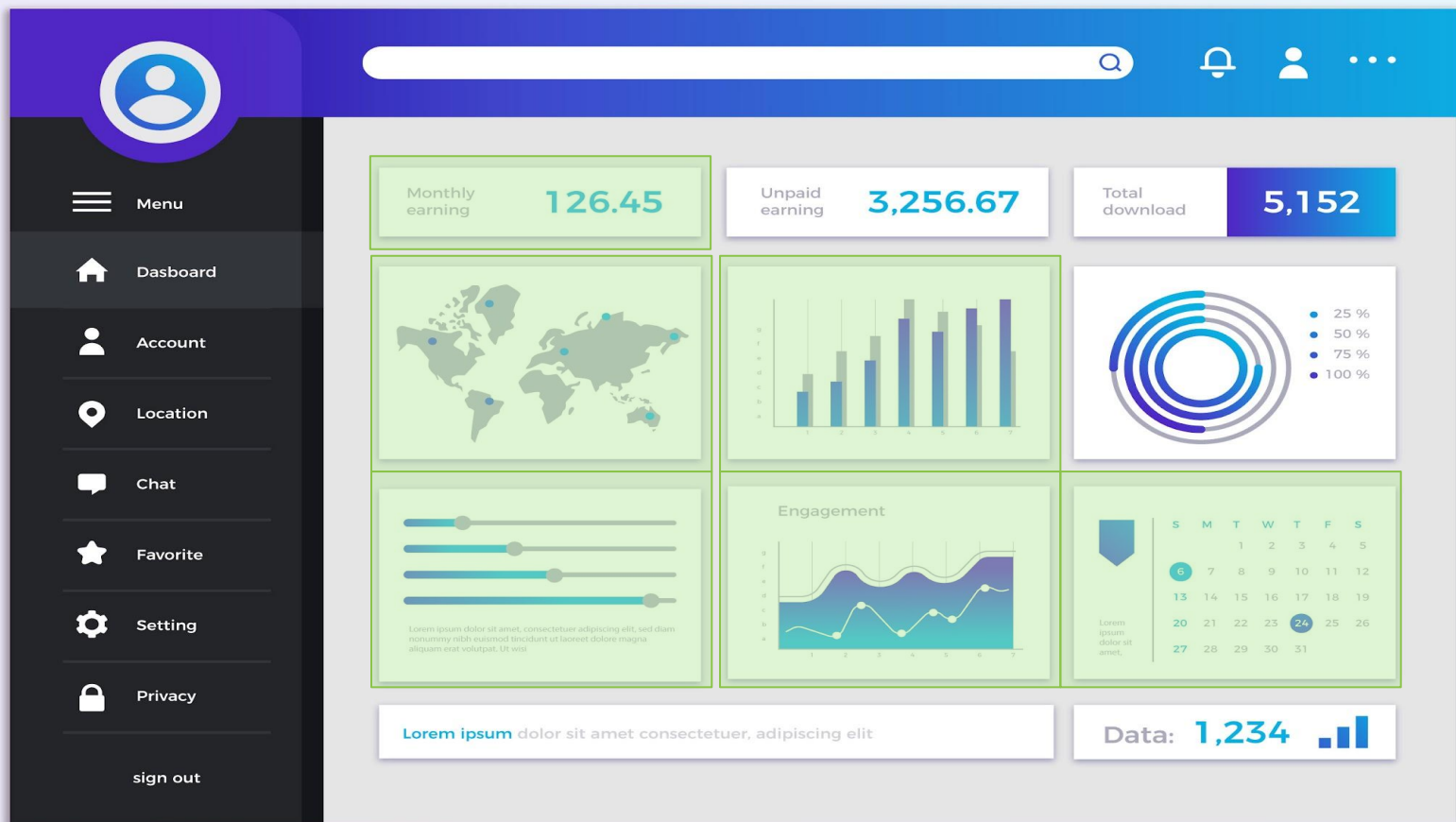
```
<%= turbo_refresh_scroll_tag :preserve %>
```

HTMX

How to Use

```
<script
src="https://unpkg.com/htmx/dist/htmx.min.js"></script>
<div hx-ext="morph">
  <button hx-get="/example"
hx-swap="morph:innerHTML">
    Morph My Inner HTML
  </button>
  <button hx-get="/example"
hx-swap="morph:outerHTML">
    Morph My Outer HTML
  </button>
  <button hx-get="/example" hx-swap="morph">
    Morph My Outer HTML
  </button>
</div>
```





How do we do better?

1. Support shared knowledge.
2. Start simple.
3. Use technologies that support our goals.
4. Question complexity when you see it.
5. Be an advocate for boring ol' HTML.

Bonus Complexities!

Replace sass with css.

Replace webpack/esbuild/bun with JS modules, [unpkg](#), and/or import maps.

Don't choose a UI library that requires a specific JS framework.

Thanks!

Jonathan Knapp
jon@coffeeandcode.com



Check out [Cleveland GiveCamp](#) for an amazing volunteering opportunity to help local nonprofits.

July 26th-28th at Hyland in Westlake, Ohio.

Resources

<https://html.spec.whatwg.org>

<https://http.cat>

<https://hypermedia.systems>

<https://restcookbook.com>

<https://v1.htmx.org/migration-guide-hotwire-turbo/>

<https://www.youtube.com/watch?v=SWets0rlezA>

COFFEE
AND
CODE