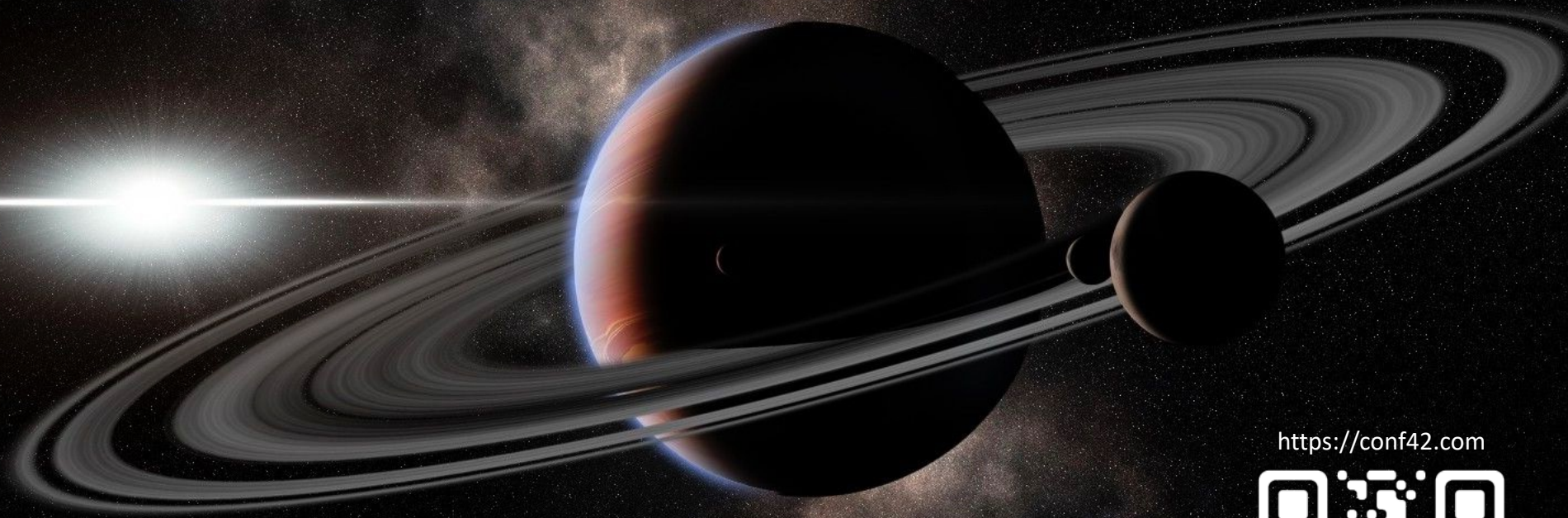


THE REMIX PHILOSOPHY IS NOT JUST FOR REMIX



Ken Snyder, Conf42

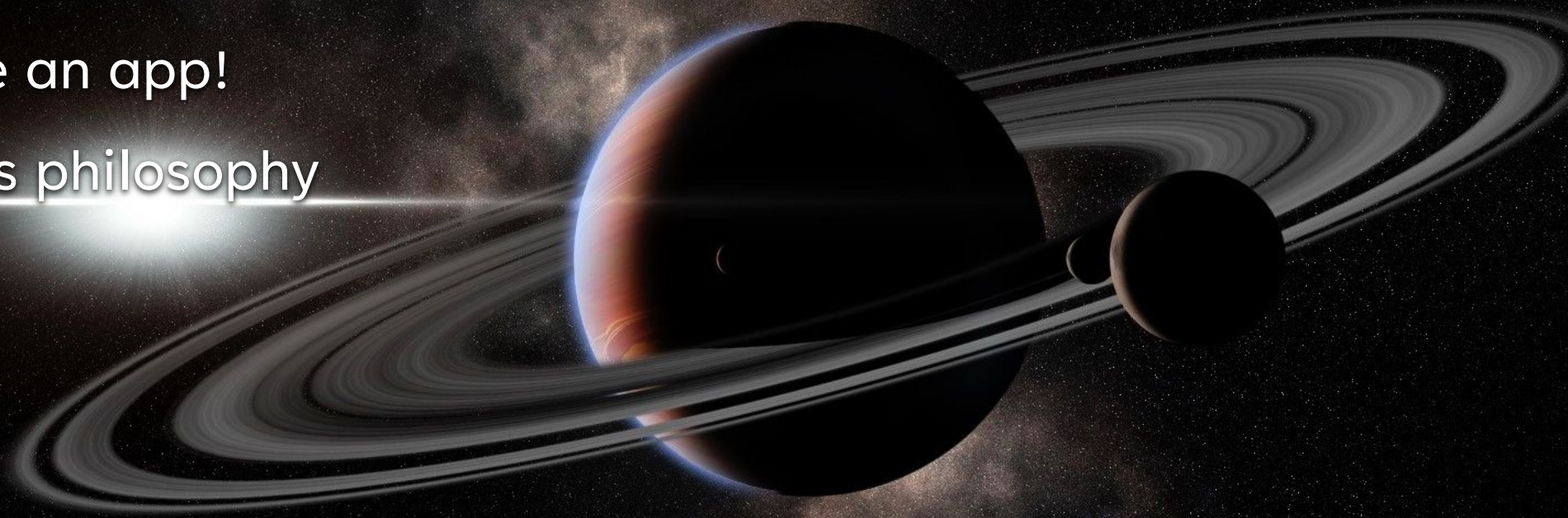
November 2023

<https://conf42.com>



What we'll cover

1. Why should you care
2. Let's create an app!
3. Applying its philosophy



A little background

- Shoreline **streamlines patient education** management by centralizing resources and making them easy to share via text, email, and EHR.
- UtahJS is a **JavaScript developer group** with monthly meetups and yearly conferences.
- Remix is a **meta-framework** for web applications running on Node* and React

**Or any other JavaScript or TypeScript runtime*

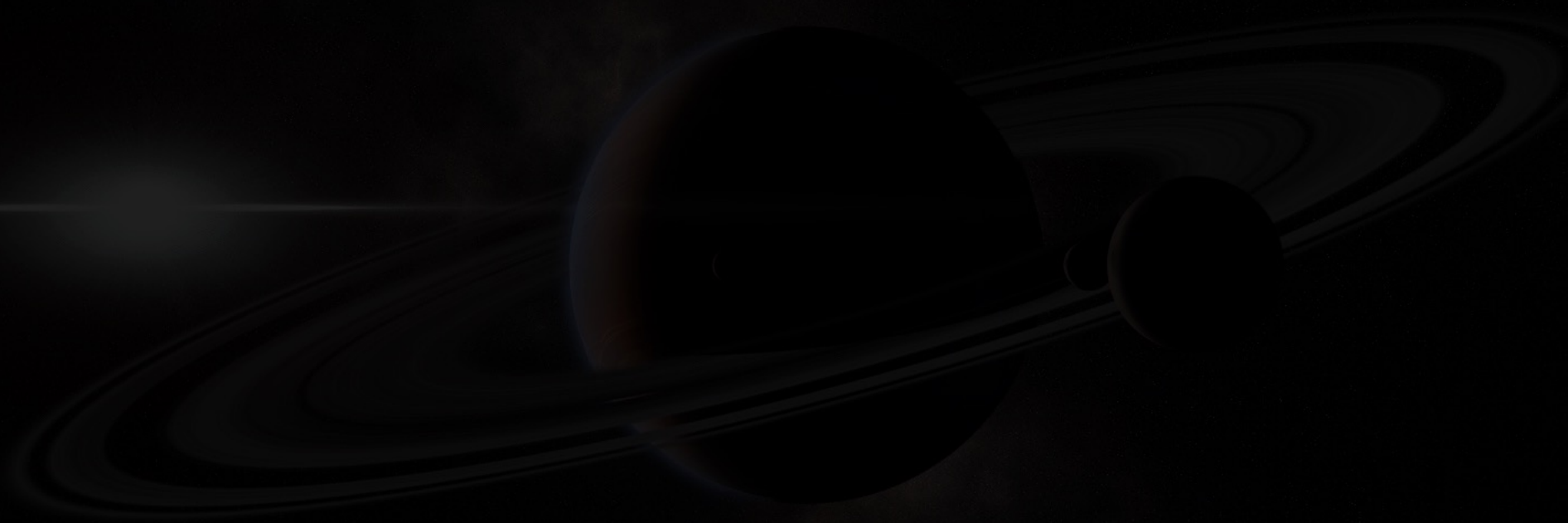


Why should you care?

- First-render performance
- Immediate interactivity
- React Router
- Part of Shopify

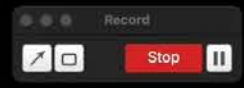


Let's create a real Remix App!



Kens-MBP:Sites ksnyder\$

Record

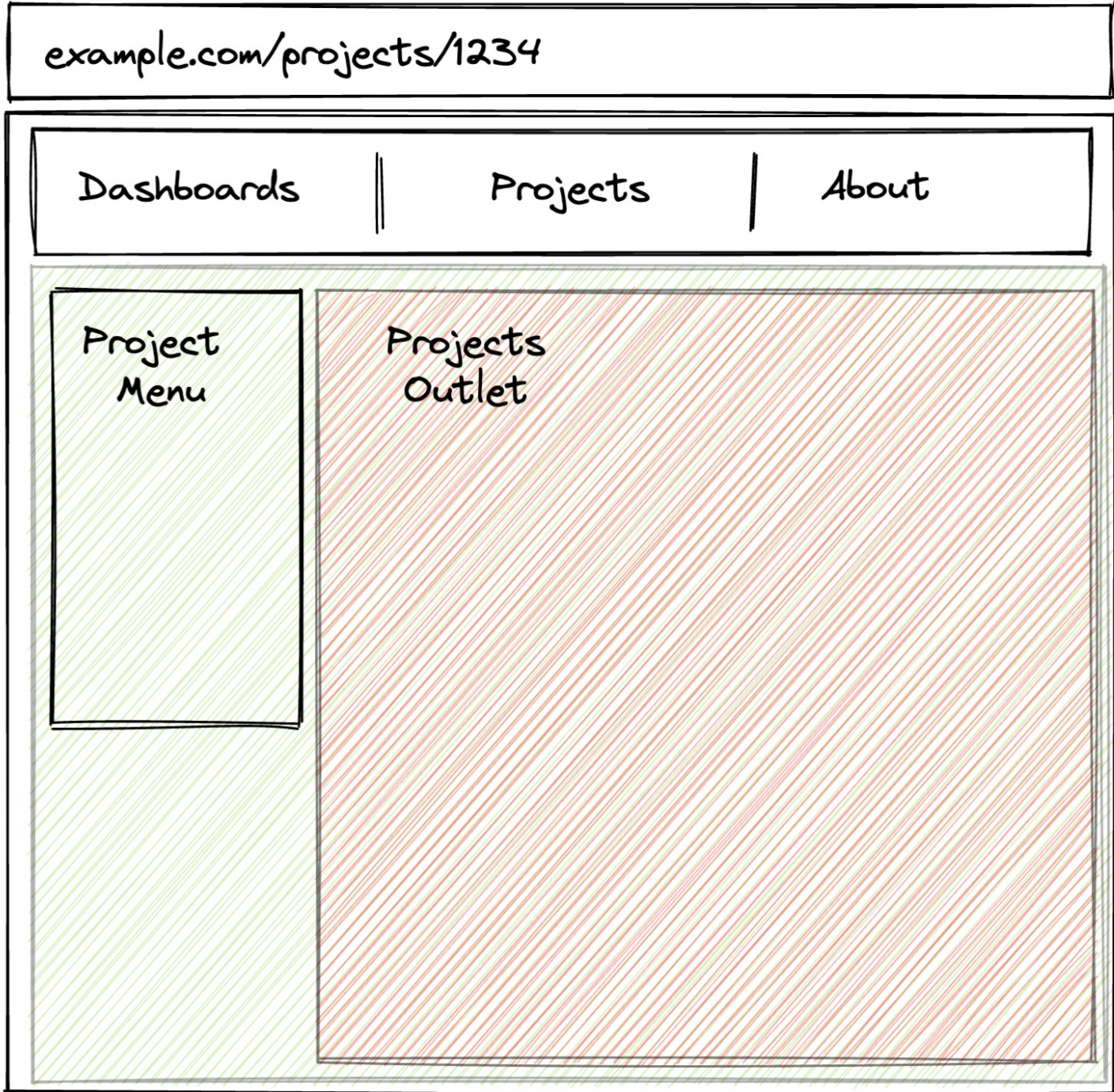


Try it on GitHub



<https://github.com/kensnyder/remix-demo-shopping-list>

<Outlet />



More complex forms

Shoreline
Beta

Manage Users ADD USER

Search FILTER

<input type="checkbox"/>	First name	Last name	Email address	Organization	Last login	Actions
<input type="checkbox"/>	AB				10/24/2023 9:12 AM	
<input type="checkbox"/>	AB				10/24/2023 8:23 AM	
<input type="checkbox"/>	BM				N/A	
<input type="checkbox"/>	BH				10/27/2023 11:19 AM	
<input type="checkbox"/>	BD				N/A	

Items per page: 10 < 1 2 3 4 5 ... 8 >

More complex forms

Shoreline
Beta

KS

Manage Users

+ ADD USER

Search FILTER

<input type="checkbox"/>	First name	Last name	Email address	Organization	Last login	Actions
<input type="checkbox"/>	AB				10/24/2023 9:12 AM	
<input type="checkbox"/>	AB				10/24/2023 8:23 AM	
<input type="checkbox"/>	BM				N/A	
<input type="checkbox"/>	BH				10/27/2023 11:19 AM	
<input type="checkbox"/>	BD				N/A	

Items per page: 10 < 1 2 3 4 5 ... 8 >

More complex forms

Shoreline
Beta

KS

Manage Users

+ ADD USER

Search **FILTER**

<input type="checkbox"/>	First name	Last name	Email address	Organization	Last login	Actions
<input type="checkbox"/>	AB				10/24/2023 9:12 AM	
<input type="checkbox"/>	AB				10/24/2023 8:23 AM	
<input type="checkbox"/>	BM				N/A	
<input type="checkbox"/>	BH				10/27/2023 11:19 AM	
<input type="checkbox"/>	BD				N/A	

Items per page: 10 1 ...

More complex forms

Shoreline
Beta

KS

Manage Users

+ ADD USER

Search **FILTER** `<button type="submit">Filter</button>`

<input type="checkbox"/>	First name	Last name	Email address	Organization	Last login	Actions
<input type="checkbox"/>	AB				10/24/2023 9:12 AM	
<input type="checkbox"/>	AB				10/24/2023 8:23 AM	
<input type="checkbox"/>	BM				N/A	
<input type="checkbox"/>	BH				10/27/2023 11:19 AM	
<input type="checkbox"/>	BD				N/A	

Items per page: 10 < 1 2 3 4 5 ... 8 >

More complex forms

Shoreline
Beta

KS

Manage Users

+ ADD USER

Search FILTER

<button type="submit" name="sort" value="email"><UpIcon /></button>
<button type="submit" name="sort" value="-email"><DownIcon /></button>

<input type="checkbox"/>	First name	Last name	Email address	Organization	Last login	Actions
<input type="checkbox"/>	[redacted]	[redacted]	[redacted]		10/24/2023 9:12 AM	
<input type="checkbox"/>	[redacted]	[redacted]	[redacted]		10/24/2023 8:23 AM	
<input type="checkbox"/>	[redacted]	[redacted]	[redacted]		N/A	
<input type="checkbox"/>	[redacted]	[redacted]	[redacted]		10/27/2023 11:19 AM	
<input type="checkbox"/>	[redacted]	[redacted]	[redacted]		N/A	

Items per page: 10

< 1 2 3 4 5 ... 8 >

More complex forms

Shoreline
Beta

KS

Manage Users

+ ADD USER

Search FILTER

<input type="checkbox"/>	First name	Last name	Email address	Organization	Last login	Actions
<input type="checkbox"/>	AB				10/24/2023 9:12 AM	
<input type="checkbox"/>	AB				10/24/2023 8:23 AM	
<input type="checkbox"/>	BM				N/A	
<input type="checkbox"/>	BH				10/27/2023 11:19 AM	
<input type="checkbox"/>	BD				N/A	

Items per page: 10

< 1 **2** 3 4 5 ... 8 >

`<button type="submit" name="page" value="2">2</button>`

Philosophy

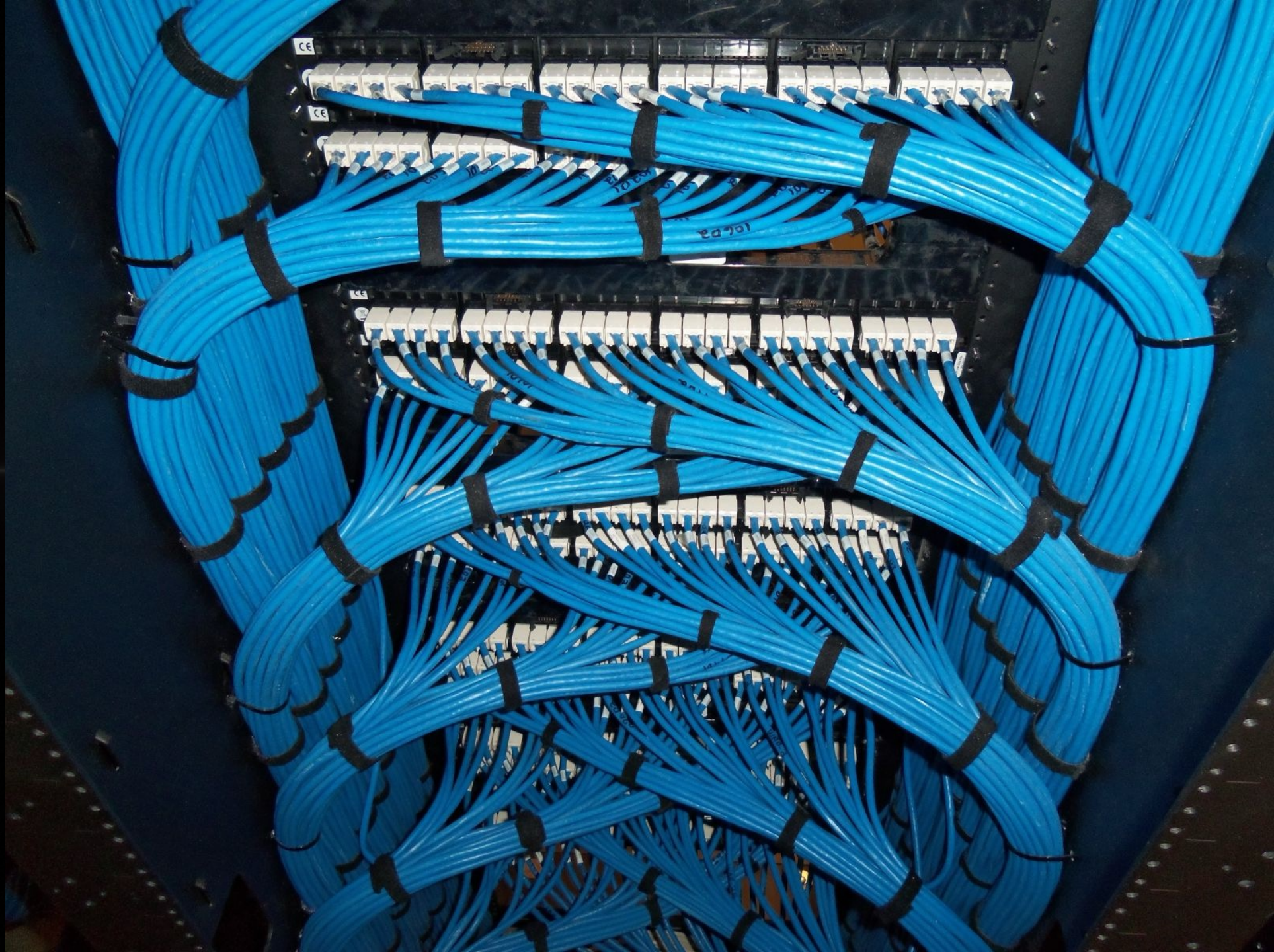
1. Embrace server/client model
2. Work with web foundations
3. Work without JavaScript
4. Roadmap & future flags



Philosophy

1. Embrace server/client model
 2. Work with web foundations
 3. Work without JavaScript
 4. Roadmap & future flags
- 

Bundles



Bundles

Server Bundle

```
4 > export const items = [...  
10 ];  
11  
12 export async function loader({ request }: LoaderFunctionArgs) {  
13   return json({ items });  
14 }  
15  
16 export default function Items() {  
17   const { items } = useLoaderData<typeof loader>();  
18   return (  
19     <main>  
20       <ul>  
21 > {items.map(item => (...  
42   </ul>  
43     <Form method="POST" key={items.length}>  
44       <input name="name" autoFocus />  
45       <button>Add item</button>  
46     </Form>  
47   </main>  
48 );  
49 }  
50 }  
51  
52 export async function action({ request }: ActionFunctionArgs) {  
53   const formData = await request.formData();  
54   const name = formData.get('name') as string;  
55   items.push({  
56     id: +new Date(),  
57     name,  
58     inCart: false,  
59   });  
60   return null;  
61 }
```

Browser Bundle

Single page app



```
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head> ...  
7 </head>  
8 <body>  
9   <noscript>Please enable JavaScript</noscript>  
10  <script src="bundle.js"></script>  
11 </body>  
12 </html>
```

```
!function(e,t,n){var r=document.getElementsByTagName("script");function o(a,b){function c(){var d=document.createElement("script");d.src=a;d.async=true;d.onload=function(){o(a,b)};r[0].parentNode.appendChild(d)}c()}o(n,"/bundle.js")}(window,document,"/bundle.js")
```

fetch('/todos')

```
{  
  "name": "Example B&E",  
  "date": "2017/16 17:47:19",  
  "uptime": "08  
  00:34:19",  
  "scale": "0",  
  "maxscale": "100",  
  "cpu": "10.15%",  
  "mem": "13.46%",  
  "disk": "100%",  
  "network": "100%",  
  "format": "00",  
  "display": "00",  
  "daylight_saving": "00",  
  "timezone": "00",  
  "format": "00",  
  "display": "00",  
  "daylight_saving": "00",  
  "timezone": "00",  
  "label": "Internal  
  sensor",  
  "temp": "69.55",  
  "tempo": "20.75",  
  "light": "68.65",  
  "hight": "20.51",  
  "low": "68.77",  
  "flow": "20.43",  
  "alarm": "0",  
  "type": "16",  
  "enabled": "1",  
  "label": "Ext Sensor",  
  "temp": "52.00",  
  "tempo": "20.00",  
  "light": "53.00",  
  "hight": "20.00",  
  "low": "53.00",  
  "flow": "20.00",  
  "alarm": "0",  
  "type": "10",  
  "enabled": "10",  
  "switch": "00",  
  "label": "1",  
  "enabled": "1",  
  "alarm": "1",  
  "status": "00"}  
}
```

```
15 <main>  
16 > <ul> ...  
18 </ul>  
19 <form>  
20   <input ... />  
21   <button ... />  
22 </form>  
23 </main>
```

Remix app



```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head> ...
7 </head>
8 <body>
9 <main>
10 <ul> ...
12 </ul>
13 <form>
14   <input ... />
15   <button ... />
16 </form>
17 </main>
18 <script src="hydrator.js" type="module"></script>
19 </body>
20 </html>

```

```

14000 0... (function...
14001 0... (function...
14002 0... (function...
14003 0... (function...
14004 0... (function...
14005 0... (function...
14006 0... (function...
14007 0... (function...
14008 0... (function...
14009 0... (function...
14010 0... (function...
14011 0... (function...
14012 0... (function...
14013 0... (function...
14014 0... (function...
14015 0... (function...
14016 0... (function...
14017 0... (function...
14018 0... (function...
14019 0... (function...
14020 0... (function...
14021 0... (function...
14022 0... (function...
14023 0... (function...
14024 0... (function...
14025 0... (function...
14026 0... (function...
14027 0... (function...
14028 0... (function...
14029 0... (function...
14030 0... (function...
14031 0... (function...
14032 0... (function...
14033 0... (function...
14034 0... (function...
14035 0... (function...
14036 0... (function...
14037 0... (function...
14038 0... (function...
14039 0... (function...
14040 0... (function...
14041 0... (function...
14042 0... (function...
14043 0... (function...
14044 0... (function...
14045 0... (function...
14046 0... (function...
14047 0... (function...
14048 0... (function...
14049 0... (function...
14050 0... (function...
14051 0... (function...
14052 0... (function...
14053 0... (function...
14054 0... (function...
14055 0... (function...
14056 0... (function...
14057 0... (function...
14058 0... (function...
14059 0... (function...
14060 0... (function...
14061 0... (function...
14062 0... (function...
14063 0... (function...
14064 0... (function...
14065 0... (function...
14066 0... (function...
14067 0... (function...
14068 0... (function...
14069 0... (function...
14070 0... (function...
14071 0... (function...
14072 0... (function...
14073 0... (function...
14074 0... (function...
14075 0... (function...
14076 0... (function...
14077 0... (function...
14078 0... (function...
14079 0... (function...
14080 0... (function...
14081 0... (function...
14082 0... (function...
14083 0... (function...
14084 0... (function...
14085 0... (function...
14086 0... (function...
14087 0... (function...
14088 0... (function...
14089 0... (function...
14090 0... (function...
14091 0... (function...
14092 0... (function...
14093 0... (function...
14094 0... (function...
14095 0... (function...
14096 0... (function...
14097 0... (function...
14098 0... (function...
14099 0... (function...
14100 0... (function...

```

addInteractivity()

Remix

REACT NOTES

Search NEW

Meeting Notes
12/30/20

A note with a v...
12/5/21

I wrote this not...
12/5/21

Click a note on the left to view something! 😊

Network

Filter: Invert Hide data URLs

Na... V S P T. S. T P Waterfall

Fully painted

Fully interactive

REACT NOTES

Search NEW

Meeting Notes
12/30/20

A note with a v...
12/30/20

I wrote this not...
12/3/21

Click a note on the left to view something! 😊

Network

Filter: Invert Hide data URLs

Na... V S P T. S. T P Waterfall

Fully painted and interactive

Server & runtime adapters

- Cloudflare Workers
- Deno
- Bun
- Netlify
- Vercel
- Azure Functions
- AWS Lambda
- Google Cloud Functions
- Express
- Fastify
- Many more



Server & runtime adapters

- Cloudflare Workers
- Deno
- Bun
- Netlify
- Vercel
- Azure Functions
- AWS Lambda
- Google Cloud Functions
- Express
- Fastify
- Many more



Server & runtime adapters

- Cloudflare Workers
- Deno
- Bun
- Netlify
- Vercel
- Azure Functions
- AWS Lambda
- Google Cloud Functions
- Express
- Fastify
- Many more



Official list

remix-stack

☆ Star

Here are 116 public repositories matching this topic...

Language: All ▾

Sort: Most stars ▾



📄 epicweb-dev / epic-stack

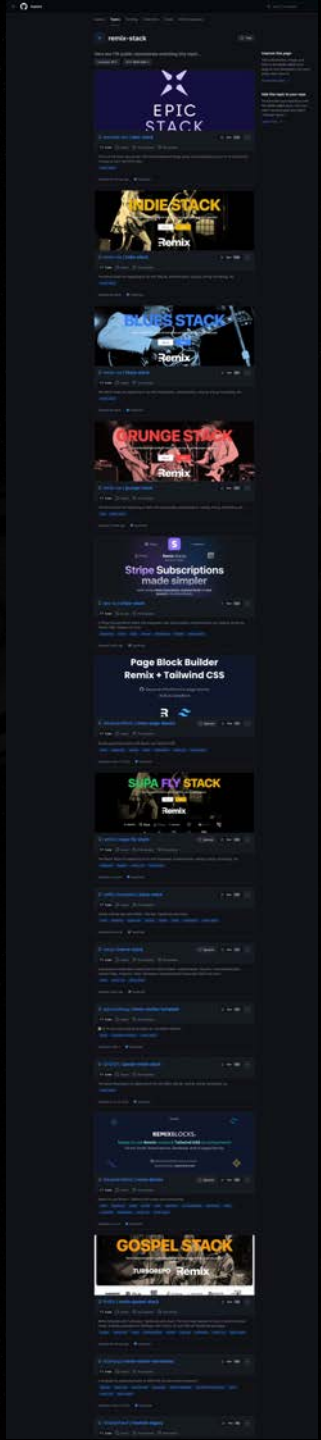
☆ Star 3.1k ▾

<> Code ⓘ Issues 🔄 Pull requests 💬 Discussions

This is a Full Stack app starter with the foundational things setup and configured for you to hit the ground running on your next EPIC idea.

remix-stack

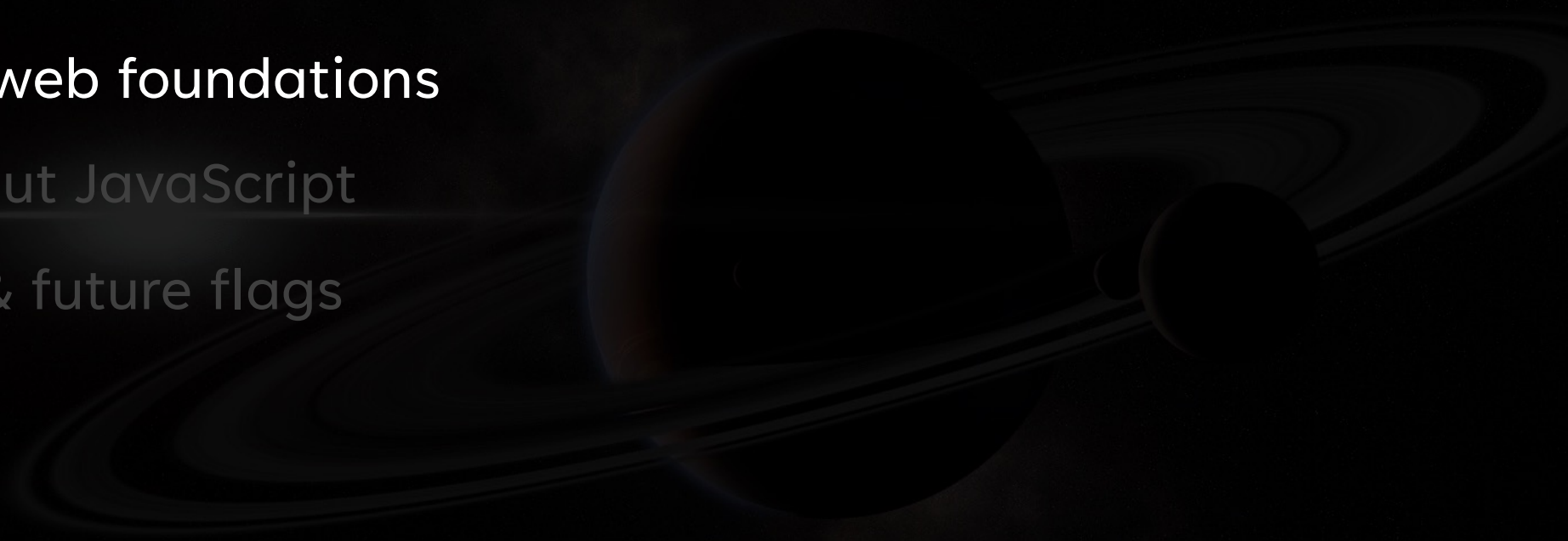
Updated 25 minutes ago ● TypeScript





Philosophy

1. Embrace server/client model
2. **Work with web foundations**
3. Work without JavaScript
4. Roadmap & future flags



This Jen, is the internet

The Elders of the Internet



Standards

- Request
- Response
- fetch
- FormData
- Headers



Standards

- URL
- URLSearchParams
- HTTP Prefetching
- HTTP Caching
- HTML Forms



Ladies and gentlemen, I'd like to present to you - the Internet!

Request, Response

```
4 // Bun
5 Bun.serve({
6   port: 80,
7   fetch: async (request: Request) => {
8     return new Response('Hello World');
9   },
10 });
11
12 // Deno
13 Deno.serve(
14   {
15     port: 80,
16   },
17   async (request: Request) => {
18     return new Response('Hello World');
19   }
20 );
21
22 // Cloudflare Workers
23 export default {
24   fetch: async (request: Request) => {
25     return new Response('Hello World');
26   },
27 };
28
```

fetch loves Request & Response

```
75 // Same signatures
76 await fetch(url, init);
77 new Request(url, init);
78
79 // Or you can directly use a Request object:
80 const request = new Request(url, init);
81 await fetch(request);
```

```
85 // And note that fetch returns a Response promise
86 > function fetch(request: Request) : Promise<Response> {...
89 }
```


Request, Response, fetch

```
75 // Same signatures
76 await fetch(url, init);
77 new Request(url, init);
78
79 // Or you can directly use a Request object:
80 const request = new Request(url, init);
81 await fetch(request);
```

```
85 // And note that fetch returns a Response promise
86 > function fetch(request: Request) : Promise<Response> {...
89 }
```

Request, Response, fetch

```
75 // Same signatures
76 await fetch(url, init);
77 new Request(url, init);
78
79 // Or you can directly use a Request object:
80 const request = new Request(url, init);
81 await fetch(request);
```

```
85 // And note that fetch returns a Response promise
86 > function fetch(request: Request) : Promise<Response> {...
89 }
```

Headers (Request & Response)

```
60 request.headers instanceof Headers; // true
61
62 request.headers.get('Cookie')
```

```
29 new Response('Hello World', {
30   headers: new Headers({
31     'Content-type': 'text/plain',
32   }),
33 });
34
35 new Response('Hello World', {
36   headers: new Headers([
37     ['Content-type', 'text/plain'],
38     ['Set-Cookie', '< cookie 1 >'],
39     ['Set-Cookie', '< cookie 2 >'],
40   ]),
41 });
42
```

URLSearchParams

```
43 new URLSearchParams('filter=Hello&sort=name');
44
45 new URLSearchParams({
46   filter: 'Hello',
47   sort: 'name',
48 });
49
50 new URLSearchParams([
51   ['filter', 'Hello'],
52   ['sort', 'name'],
53   ['tags[]', 'JavaScript'],
54   ['tags[]', 'TypeScript'],
55 ]);
56
```

Web standards in plain React code



Traditional React component

```
84 function AddUserPageJson() {
85   const addUser = useCallback(async (evt) => {
86     evt.preventDefault();
87     const formData = new FormData(evt.target);
88     const { first, last, email } = Object.fromEntries(formData);
89     await fetch('/users', {
90       method: 'POST',
91       headers: { 'Content-type': 'application/json' },
92       body: JSON.stringify({ first, last, email }),
93     });
94     navigate('/users')
95   }, []);
96   return (
97     <form onSubmit={addUser}>
98       <input type="text" name="first" placeholder="First name" />
99       <input type="text" name="last" placeholder="Last name" />
100      <input type="email" name="email" placeholder="Email" />
101      <button>Submit</button>
102    </form>
103  );
104 }
```

Traditional React component

```
84 function AddUserPageJson() {
85   const addUser = useCallback(async (evt) => {
86     evt.preventDefault();
87     const formData = new FormData(evt.target);
88     const { first, last, email } = Object.fromEntries(formData);
89     await fetch('/users', {
90       method: 'POST',
91       headers: { 'Content-type': 'application/json' },
92       body: JSON.stringify({ first, last, email }),
93     });
94     navigate('/users')
95   }, []);
96   return (
97     <form onSubmit={addUser}>
98       <input type="text" name="first" placeholder="First name" />
99       <input type="text" name="last" placeholder="Last name" />
100      <input type="email" name="email" placeholder="Email" />
101      <button>Submit</button>
102    </form>
103  );
104 }
```

Traditional React component

```
84 function AddUserPageJson() { 107
85   const addUser = useCallback(async (evt) => { 108
86     evt.preventDefault(); 109
87     const formData = new FormData(evt.target); 110
88     const { first, last, email } = Object.fromEntries( 111
89       await fetch('/users', { 112
90         method: 'POST', 113
91         headers: { 'Content-type': 'application/json' }, 114
92         body: JSON.stringify({ first, last, email }) 115
93       }); 116
94     navigate('/users') 117
95   }, []); 118
96   return ( 119
97     <form onSubmit={addUser}> 120
98       <input type="text" name="first" placeholder="First name" /> 121
99       <input type="text" name="last" placeholder="Last name" /> 122
100       <input type="email" name="email" placeholder="Email" /> 123
101       <button>Submit</button> 124
102     </form> 125
103   ); 126
104 }
```


More traditional React page

```
84 function AddUserPageJson() {
85   const addUser = useCallback(async (evt) => {
86     evt.preventDefault();
87     const formData = new FormData(evt.target);
88     const { first, last, email } = Object.fromEntries(formData.entries());
89     await fetch('/users', {
90       method: 'POST',
91       headers: { 'Content-type': 'application/json' },
92       body: JSON.stringify({ first, last, email });
93     });
94     navigate('/users');
95   }, []);
96   return (
97     <form onSubmit={addUser}>
98       <input type="text" name="first" placeholder="First name" />
99       <input type="text" name="last" placeholder="Last name" />
100      <input type="email" name="email" placeholder="Email" />
101      <button>Submit</button>
102    </form>
103  );
104 }

107 function AddUserPageFormData() {
108   const addUser = useCallback(async (evt) => {
109     evt.preventDefault();
110     const formData = new FormData(evt.target);
111     await fetch('/users', {
112       method: 'POST',
113       headers: { 'Content-type': 'application/x-www-form-urlencoded' },
114       body: formData,
115     });
116     navigate('/users');
117   }, []);
118   return (
119     <form onSubmit={addUser}>
120       <input type="text" name="first" placeholder="First name" />
121       <input type="text" name="last" placeholder="Last name" />
122       <input type="email" name="email" placeholder="Email" />
123       <button>Submit</button>
124     </form>
125   );
126 }
```

...Or just use Remix

```
128 import { Form } from '@remix-run/react';
129
130 function AddUserPageRemix() {
131   return (
132     <Form action="/users">
133       <input type="text" name="first" placeholder="First name" />
134       <input type="text" name="last" placeholder="Last name" />
135       <input type="email" name="email" placeholder="Email" />
136       <button>Submit</button>
137     </Form>
138   );
139 }
```

Philosophy

1. Embrace server/client model
2. Work with web foundations
3. **Work without JavaScript**
4. Roadmap & future flags



Work without JavaScript

- Forms still work before hydration
- Support for IE11



shopper

EXPLORER

- SHOPPER
 - .cache
 - app
 - routes
 - TS _index.tsx
 - TS shop.items.\$id.tsx
 - TS shop.items.tsx
 - TS shop.tsx
 - TS entry.client.tsx
 - TS entry.server.tsx
 - root.tsx M
 - build
 - node_modules
 - public
 - .eslintrc.cjs
 - .gitignore
 - .prettierrc.cjs
 - package-lock.json
 - package.json
 - README.md
 - remix.config.js
 - remix.env.d.ts
 - tsconfig.json

```
15
16 export default function App() {
17   return (
18     <html lang="en">
19       <head>
20         <meta charSet="utf-8" />
21         <meta name="viewport" content="width=device-width, initial-scale=1" />
22         <Meta />
23         <Links />
24       </head>
25       <body>
26         <Outlet />
27         <ScrollRestoration />
28         <Scripts />
29         <LiveReload />
30       </body>
31     </html>
32   );
33 }
34
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

remix dev

```
info building...
info built (301ms)
[remix-serve] http://localhost:3000 (http://192.168.1.140:3000)

GET /shop/items/200 - - 33.465 ms
```

Ln 34, Col 1 Spaces: 2 UTF-8 LF TypeScript JSX Prettier

localhost

Shop

- Bread [Delete](#) [Load in cart](#) [Unload from cart](#)

Elements Console Sources Network Timelines Storage

Filter Full URL XHR/Fetch

Name	Domain	Type	Initiator	Tr...	Time
No Filter Results					

Clear Filters

0 0 0B 0B 0

Philosophy

1. Embrace server/client model
2. Work with web foundations
3. Work without JavaScript
4. Roadmap & future flags



Roadmap & RFCs are published on GitHub


Roadmap

Board | List | Gantt

Filter by keyword or by field Discard

- Backlog 1**
 - Draft: Warn when remix packages are out of sync
- Planned 7**
 - remix #7639: Remix SPA Target
 - remix #7641: Single Fetch
 - remix #7643: Middleware
 - remix #7645: Server Context
 - remix #7646: Migrate Integration tests to @remix-run/testing
 - Draft: Make build and watch from remix-dev public API for hydrogen
 - remix #7824: Remix Package
- In Progress 2**
 - remix #7635: Client Data
 - remix #7633: Vite
- Merged 3**
 - remix #7710: Fetcher Tweaks
 - Draft: Build error popup
 - Draft: View Transitions
- Released 0**


Roadmap Highlights

- Vite build system
 - Loader/action middleware
 - Optimistic update tools
- 

Future flags

```
1  /**
2   * @type {import('@remix-run/dev').AppConfig}
3   */
4  module.exports = {
5    future: {
6      v2_dev: true,
7      v2_headers: true,
8      v2_errorBoundary: true,
9      v2_meta: true,
10     v2_normalizeFormMethod: true,
11     v2_routeConvention: true,
12   },
13   /* other configuration options here */
14 };
```

Applying the philosophy outside Remix

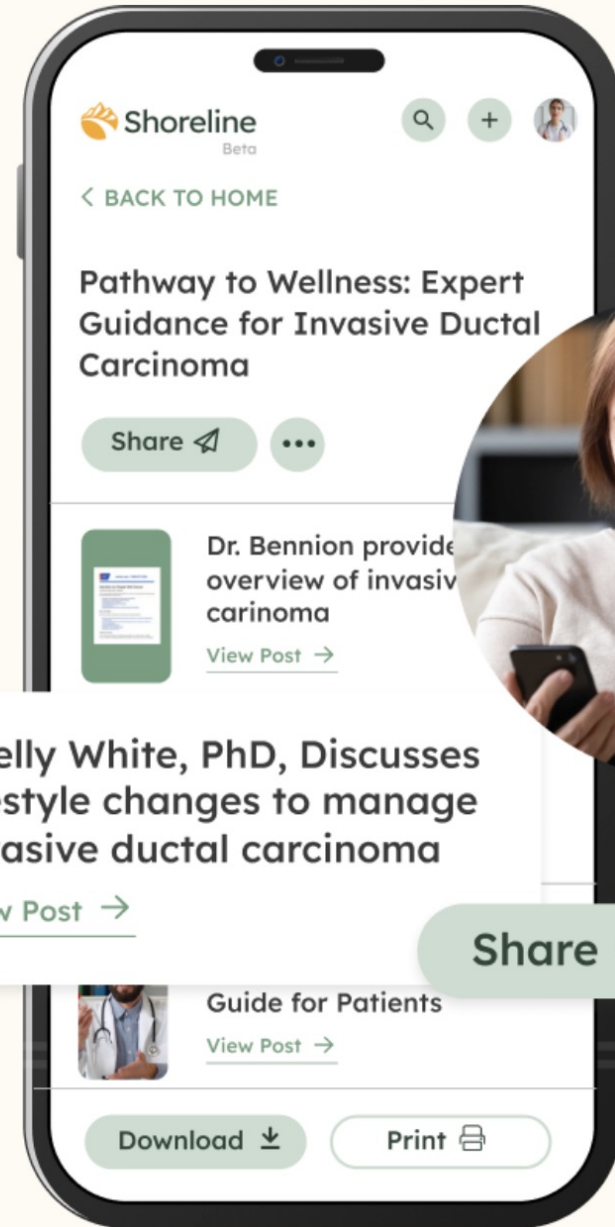
- Directly use fetch, FormData, URL
 - Learn web standards
 - Learn runtime standards
 - Progressive enhancement
- 

We chose a new mindset

- I don't worry about:
 - Where my code runs
 - Routing, submission and rendering target
- I am liberated from:
 - Server-side rendering duplication
 - Worry about first-render performance

Hello Shoreline

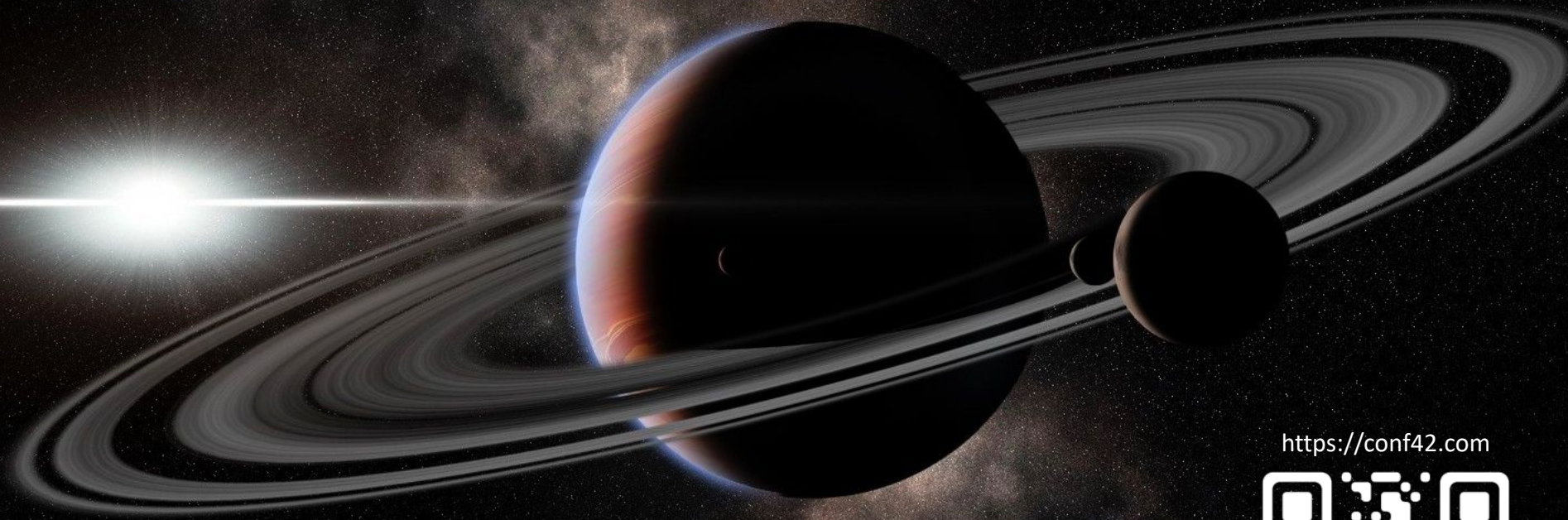
<https://shoreline.health>



Share ↗

THANK YOU

Remix



Ken Snyder, Conf42

November 2023

<https://conf42.com>

