



Server-side events in Next.js

Gabriel Răducu

NOVEMBER 2023

Contents

01 EPAM & me

02 Industry focus

03 The problem

04 The “React” way

05 Alternatives

06 Q&A

About EPAM

Global

55K+ total employees

50+ countries

Romania

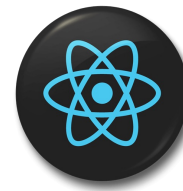
Since 2020

450+ Employees

50+ colleagues in Digital Engagement
Practice, working remotely

Lead Web Engineer

1.3 years in EPAM
a decade of Web development

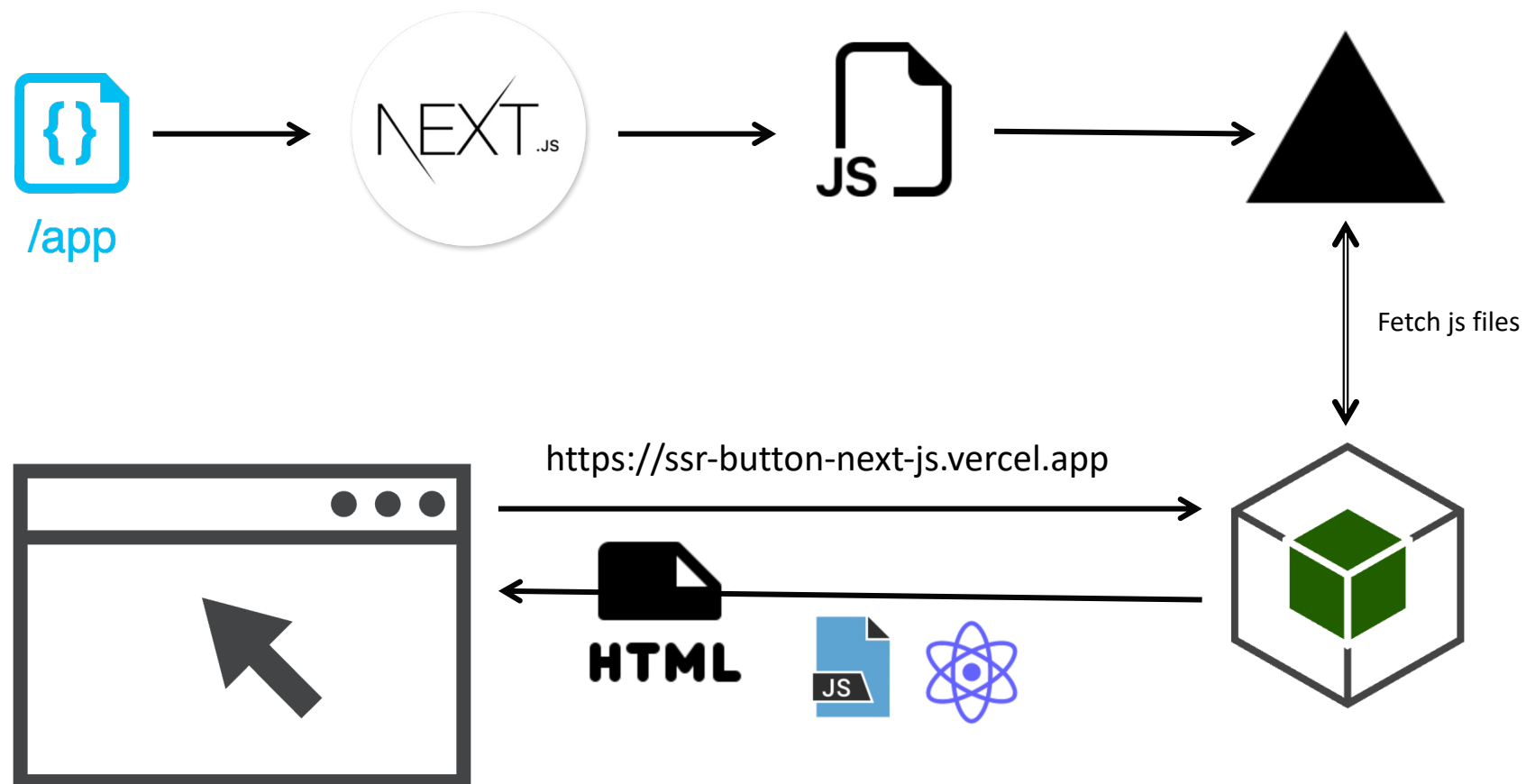


More HTML

Less javascript

No hydration

COMMON SSR FLOW



1. How do we do SSR and Static pages in nextjs “pages” routing?
2. How do we do SSR and Static in nextjs “app” routing?
3. How does ‘use client’ impact the SSR output?

THE PROBLEM



λ (Server) server-side renders at runtime (uses `getInitialProps` or `getServerSideProps`)

○ (Static) automatically rendered as static HTML (uses no initial props)

⚠ Entire page / deopted into client-side rendering. <https://nextjs.org/docs/messages/deopted-into-client-rendering> /



SSR

Static

CSR

Write a button that fetches the latest data from the server using a SSR behavior on click event



page.tsx ×

```
1  import { contentfulClient } from '../../env/contentful/contentful.client';
2  import { ButtonTitleEntryId } from '../../env/contentful/contentful.constants';
3  import { fetchData } from '../../utils/fetch-name';
4  import { inlinePre } from './constants';
5
6  export default async function PageDefault() {
7    const data = await fetchData();
8    const title = await contentfulClient.getEntry(ButtonTitleEntryId);
9    const state = true;
10   return (
11     <>
12       <button
13         title={title.fields.ctaLabel as string}
14         className={`bg-${state ? 'blue' : 'red'}-500`}
15       >
16         no state, static data - {data}
17       </button>
18       <pre>{inlinePre}</pre>
19     </>
20   );
21 }
```



ADDING STATE



<epam>

```
page.tsx  x
6
7  export default async function TheReactWay() {
8    const data = await fetchData();
9    const title = await contentfulClient.getEntry(ButtonTitleEntryId);
10   return (
11     <
12       <ClientButton data={data} title={title.fields.ctaLabel as string} />
13       <pre>{inlinePre}</pre>
14     >
15   );
16 }

button.client.tsx  x
1  "use client";
2  import { FC, useState } from "react";
3  import { fetchData } from "../utils/fetch-name";
4
5  > type ClientButtonProps = { ...
6  }
7
8  export const ClientButton: FC<ClientButtonProps> = ({ data, title }) => {
9    const [state, setState] = useState(true);
10    const [sData, setData] = useState(data);
11    const handleClick = () => {
12      setState((old) => !old);
13      fetchData().then(setData);
14    }
15    return (
16      <button
17        onClick={handleClick}
18        title={title}
19        className={`bg-${state ? "blue" : "red"}-500`}
20      >
21        with onClick - {sData}
22      </button>
23    );
24  };
25 }
```

INITIAL IDEA



```
page.tsx x
6 export default async function InitialIdea() {
7   const data = await fetchData();
8   const title = await contentfulClient.getEntry(ButtonTitleEntryId);
9   return (
10     <
11       <button
12         id="click-btn"
13         title={title.fields.ctaLabel as string}
14         value="blue"
15         className="bg-blue-500"
16       >
17         with JS event listener - {data}
18     </button>
19   );
}
```

```
layout.tsx x
1 import Script from "next/script";
2
3 > export default function VanillaLayout({ ...
4 > }): { ...
5 > } {
6 >   {
7 >     {
8 >       return (
9 >         <
10 >           <Script id="btn-loader" strategy="afterInteractive">
11 >             {
12 >               const el = document.querySelector('#click-btn');
13 >               el?.addEventListener('click', (e) => {
14 >                 const state = e.currentTarget.value === "blue" ? "red" : "blue";
15 >                 e.currentTarget.classList = "bg-" + state + "-500";
16 >                 e.currentTarget.value = state;
17 >                 fetch('/api').then((res) => res.text()).then((val) => {
18 >                   el.innerText = "with JS event listener - " + val;
19 >                 }).catch((er) => {
20 >                   console.log('failed', er);
21 >                 })
22 >               })
23 >             }
24 >           </Script>
25 >           {children}
26 >         </>
27 >       );
28 >     }
29 >   }
30 > }
31 > }
```

SOLUTION 1



<epam>

```
page.tsx ×
6   export default async function Solution1() {
7     const data = await fetchData();
8     const title = await contentfulClient.getEntry(ButtonTitleEntryId)
9
10    return (
11      <>
12        <button
13          className="bg-blue-500"
14          name="btn-color"
15          value="blue"
16          title={title.fields.ctaLabel as string}
17          hx-post="/api"
18          hx-swap="outerHTML"
19        >
20          with htmx - {data}
21        </button>
22      </>
23    );
24  }

layout.tsx ×
1   import Script from "next/script";
2
3   export default function HTMXLayout({
4     children,
5   }: {
6     children: React.ReactNode;
7   }) {
8     return (
9       <>
10        <Script src="https://unpkg.com/htmx.org@1.9.4/dist/htmx.min.js" />
11        {children}
12      </>
13    );
14  }
```

SOLUTION 2



<epam>

```
page.tsx  X
7
8  export default async function Solution2() {
9    const title = await contentfulClient.getEntry(ButtonTitleEntryId);
10   const data = await fetchData();
11   const color = cookies().get('x-color')?.value ?? 'blue';
12   return (
13     <>
14       <form action={SeverFunction}>
15         <input type="hidden" value={color} name="color" />
16         <button
17           title={title.fields.ctaLabel as string}
18           type="submit"
19           className={`bg-${color}-500`}
20         >
21           with RS action - {data}
22         </button>
23       </form>
24     </>
25   );
26 }
```

```
sign-in.action.ts  X
1  "use server";
2
3  import { revalidatePath } from "next/cache";
4  import { cookies } from 'next/headers';
5
6  export async function SeverFunction(e: FormData) {
7    const formColor = e.get("color") === "blue" ? "red" : "blue";
8
9    cookies().set('x-color', formColor);
10   revalidatePath('/server-actions-way');
11   return {message: formColor};
12 }
```

Thank you!

Let the Q&A begin!

Gabriel Răducu
Lead Web Engineer @ EPAM



[gabrielraducu](#)



[ebitzu](#)



<https://gr-cv.vercel.app/>