Next.js 13 is a game changer

Code For Real
@codeforreal

Code For Real
@codeforreal

Give a follow

Next.js 13 has a lot of new features. Swipe to see it

Dont' forget to save it

Save

NEXT.JS 13

Vercel recently announced **Next.js 13,** an incremental and breaking update of their most popular React framework. The framework seems to have been revamped overall to support multiple features.

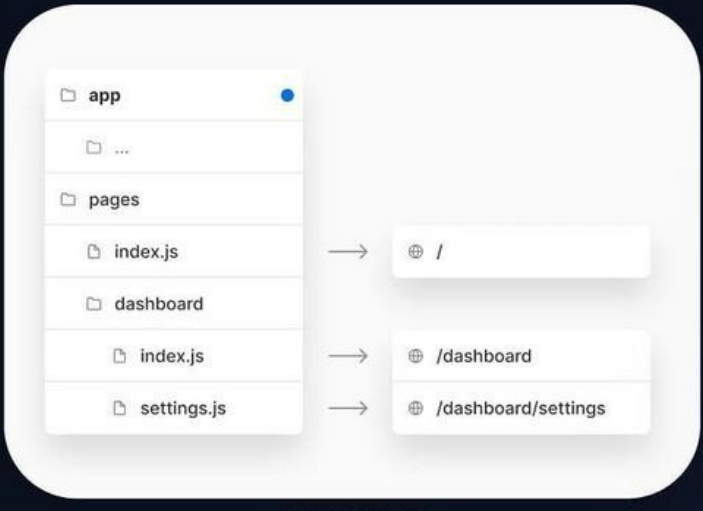
And they are claiming some impressive improvements in JavaScript ecosystem that has never happened before.

Code For Real
@codeforreal

Save

New Routing & Layout System

Vercel is improving their file based routing system in Next.js 13. It introduces a new optional **/app** directory. The idea is to replace the **/pages** directory with **/app** as a single source of all the pages, layouts, context, styling, etc. You can still use the **/pages** directory and the **/app** directory can co-exist.



Credit: Vercel

Code For Real
@codeforreal

Save

The new routing system now detects a page if there is a **page.js file** inside a directory inside this /app.

On top of this, Next.js 13 uses a lot of reserved names like layout.js, loading.js, error.js, etc. layout.js is a new file that contains the component UI which can be inherited by the child routes. Similarly, loading.js UI is shown when the content is loading and error.js UI is shown when the content threw error.

While navigating to different a page, layouts preserve state, remain interactive, and do not re-render. Now, this is really cool. Next.js 13 gives fine grained control over the UI nested inside the routes and makes it possible to manage it separately which was very difficult previously with the introduction of the layout system.

Code For Real
@codeforreal

Save

Data Fetching

Next.js 13 now makes it possible to not use getServerSideProps() and getStaticProps() method to pass props back and forth. With the introduction of new **fetch** implementation, you can directly call any API and await for it and it will be server rendered automatically.
Next.js 13 provides one flexible way to fetch, cache, and revalidate data at the component level. This means all the benefits of Static Site Generation (SSG), Server-Side Rendering (SSR), and Incremental Static Regeneration (ISR) are now available through one API.

```
// This request should be cached until manually invalidated.
// Similar to `getStaticProps`.
// `force-cache` is the default and can be omitted.
fetch(URL, { cache: 'force-cache' });

// This request should be refetched on every request.
// Similar to `getServerSideProps`.
fetch(URL, { cache: 'no-store' });

// This request should be cached with a lifetime of 10 seconds.
// Similar to `getStaticProps` with the `revalidate` option.
fetch(URL, { next: { revalidate: 10 } });
```

Code For Real
@codeforreal

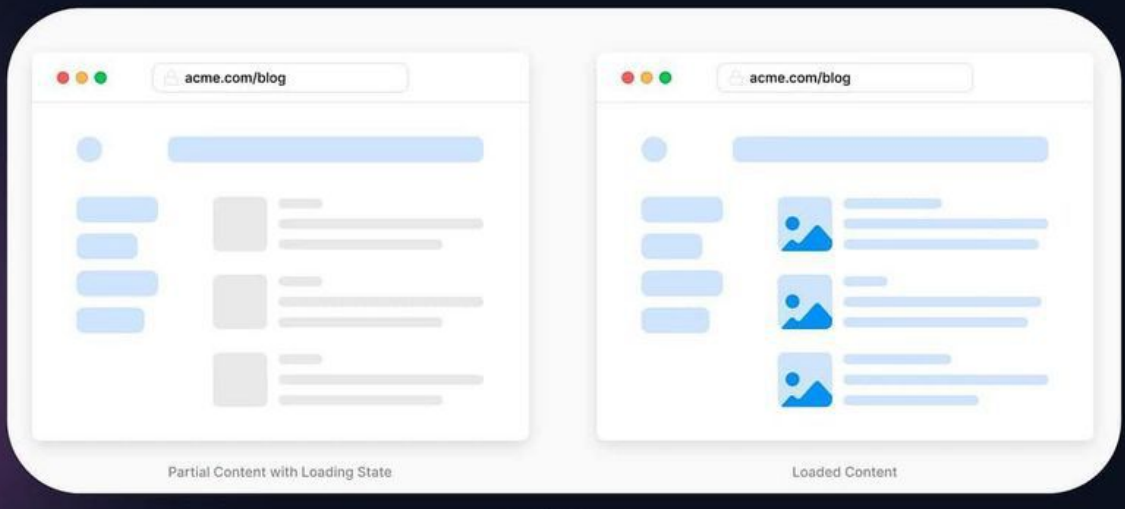
Credit: Vercel

Save

Streaming

The /app directory introduces the ability to progressively render and incrementally stream rendered units of the UI to the client.

With Server Components and nested layouts in Next.js, you're able instantly render parts of the page that do not specifically require data, and show a loading state for parts of the page that are fetching data. With this approach, the user does not have to wait for the entire page to load before they can start interacting with it.



Credit: Vercel

Code For Real
@codeforreal

Save

Server Components

The /app directory introduces support for React's new Server Components architecture. Server and Client Components use the server and the client each for what they're best at - allowing you to build fast, highly-interactive apps with a single programming model that provides a great developer experience.

With Server Components, Next.js allows building complex interfaces while reducing the amount of JavaScript sent to the client, enabling faster initial page

Code For Real
@codeforreal

Save

And last but not the least, the most exciting announcements was the brand build tool called "**Turbopack**". Turbopack is supposed to be a **Webpack** killer and is built on top of **Rust,** so expect it to be blazingly fast. Just to start with some benchmarks, Turbopack is said to be 700x faster than Webpack.

I have covered all about Turbopack in a separate reel, so be sure to check that out for more info.

Code For Real
@codeforreal

Save