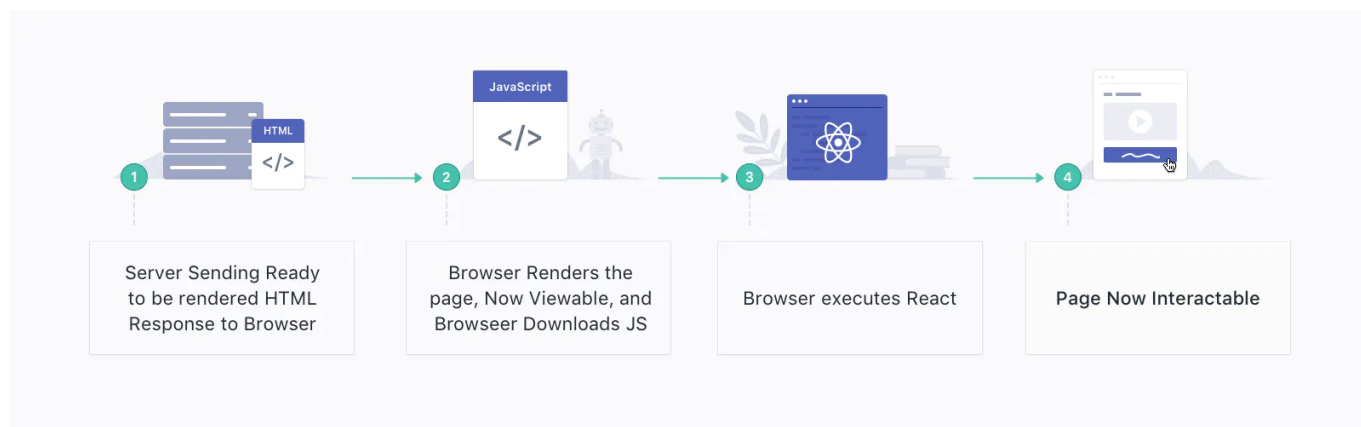


Nội dung chương 2 - NextJS concepts



Source: <https://prismic.io/blog/seo-with-react-and-nextjs>

AGENDA

1. Chương 2 này mình tập trung vào điều gì?
2. Faster build with Rust-based SWC
3. NextJS sử dụng cho FE hay BE?
4. Code sẽ có phần chạy trên server, có phần chạy trên client

1. Chương 2 này mình tập trung vào điều gì?

- Tìm hiểu các **CONCEPTS** của **NextJS** qua các ví dụ đơn giản.
- Chưa làm feature / project nào hoàn chỉnh (để dành cho 3 chương tiếp theo)
- Tham khảo từ:
 - NextJS learn document: <https://nextjs.org/learn/basics/create-nextjs-app>
 - MashingMagazine: <https://www.smashingmagazine.com/category/next.js>
 - Và nhiều nguồn tài liệu khác online. Mình sẽ ghi kèm nguồn mỗi khi sử dụng.

| # | Name |
|---|---------------------------------------|
| 1 | File-system routings |
| 2 | Navigation |
| 3 | Layout |
| 4 | Data Fetching (try with useSWR) |
| 5 | CSR, SSR, SSG |
| 6 | ISR (Incremental Static Regeneration) |
| 7 | Protected route |
| 8 | SEO |
| 9 | i18n |

2. Faster build with Rust-based SWC

Adopting Rust-based SWC

We're working on integrating [SWC](#), a super-fast JavaScript and TypeScript compiler written in Rust, that will replace two toolchains used in Next.js: Babel for individual files and Terser for minifying of output bundles.

As part of replacing Babel with SWC, we are porting all custom code transformations Next.js uses to SWC transforms written in Rust to maximize performance. For example, tree shaking unused code inside ``getStaticProps``, ``getStaticPaths``, and ``getServerSideProps``.

As part of replacing Terser, we are working on ensuring the SWC minifier has similar output to Terser while massively improving performance and parallelization of minification.

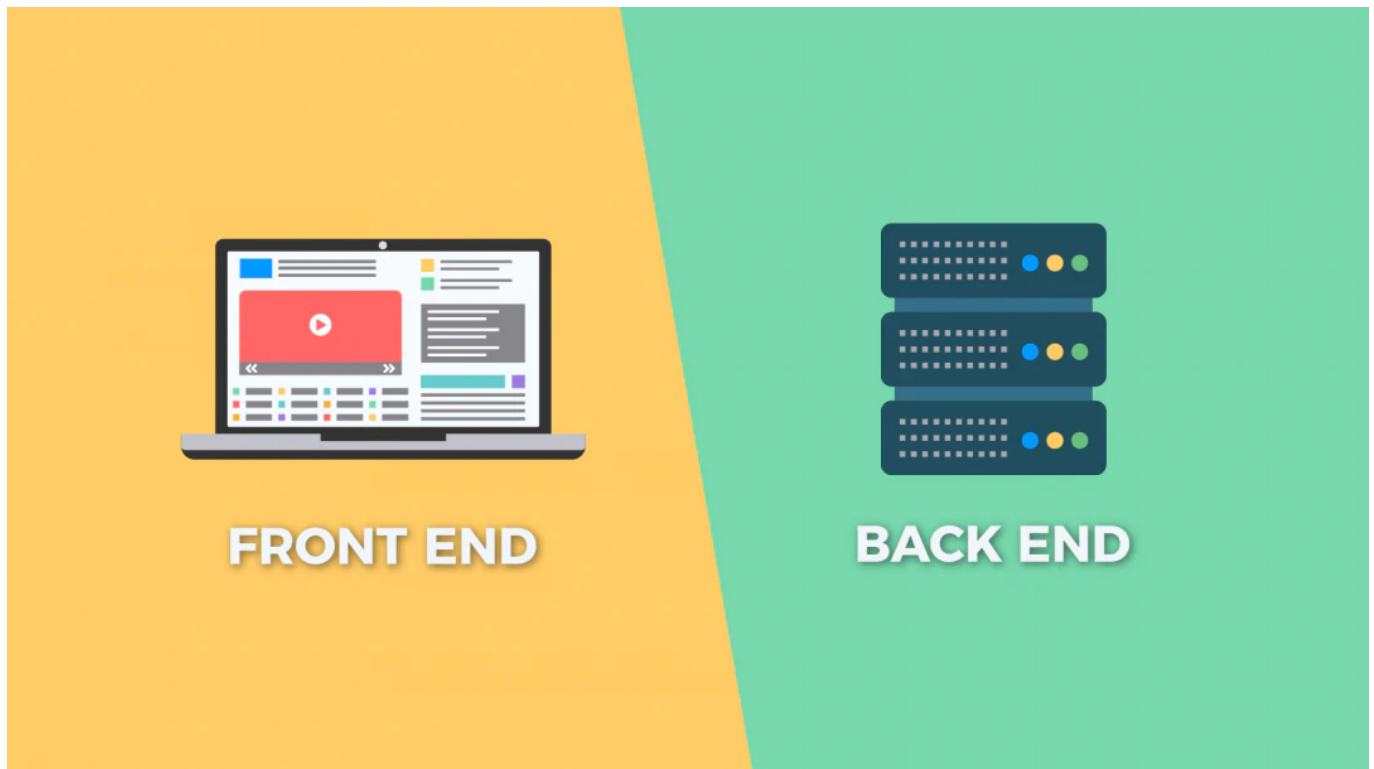
In early tests, previous code transformations using Babel dropped from **~500ms to ~10ms** and code minification from Terser dropped from **~250ms to ~30ms** using SWC. Overall, this resulted in **twice as fast builds**.

We're excited to announce [DongYoon Kang](#), the creator of [SWC](#), and [Maia Teegarden](#), contributor to [Parcel](#), have joined the Next.js team at Vercel to work on improving both ``next dev`` and ``next build`` performance. We will be sharing more results from our SWC adoption in the next release when it's made stable.

Source: [Next.js 11.1 release / Blog](#)

3. NextJS sử dụng cho FE hay BE?

- NextJS cơ bản được host trên NodeJS server --> **làm Backend được** --> nhưng **KHÔNG NÊN**
- Nên tách biệt Frontend và Backend.
 - Coding styles / convention khác nhau
 - Big codebase / complex work flow for git / github
 - Not optimized for horizontal scale
 - Bonus, real world cases, most of the time, we keep FE and BE repo separated.



Source: [Frontend and Backend | ironhack.com](https://ironhack.com)

4. Code sẽ có phần chạy trên server, có phần chạy trên client

Mình có thể thấy những đoạn code tương tự thế này để check client / server:

```
if (typeof window !== 'undefined') {  
  // do something on client side  
}
```

Series - NextJS + Typescript 🍌

- Tác giả: **Hậu Nguyễn**
- Được phát hành trên kênh youtube **Easy Frontend**.
- Tài liệu pdf và videos đều có bản quyền thuộc về Easy Frontend.
- Videos được phát hành cho fan cứng trước, public sau.
- [Đăng ký fan cứng](#) để xem series này đầy đủ và sớm nhất nhé.

 Kết nối với mình

- ☒ Follow Facebook: <https://www.facebook.com/nvhauesmn/>
- ☒ Like Fanpage: <https://www.facebook.com/learn.easyfrontend>
- ☒ Youtube Channel: <https://www.youtube.com/easyfrontend>