

Next.js

Alen Murtić



01

Next.js introduction

02

Next.js features

03

Server-side rendering with Next.js



01

Next.js introduction

Motivation

- React by default fetches all JS to the client and then starts rendering
 - User gets empty page (or app shell)
 - Slower time to interaction because all user can initially see are placeholders
 - Users with slower devices and/or internet speed can have double the amount of slowness
- Server-side rendering content, especially metadata, is optimal for SEO (search engine optimization, i.e. higher ranking on Google)
- Entered: Next.js

Next.js introduction

Next.js background

- Started as a server-side rendering library for React
 - Implemented core SSR features and allowed extending them via custom Express.js server logic (e.g. translations logic)
- Became full-fledged React SSR framework
 - Added static-site generation (before that, [Gatsby](#) was the best SSG framework)
 - Removing customization ability in favor of its own solutions

Next 14 and React 18

- Next 13 introduces many new **beta** features and changes, they are stable in Next 14
 - [Blogpost](#) and [more docs](#)
 - In alignment with server components introduced in React 18
 - Mostly has to do with **app folder** and **app routing**
- Given how Sofascore uses Next.js, those changes won't give us new features, but could introduce potential issues
 - We still use Next 12, but hope to upgrade sometime soon
- Because of that, at this year's academy, please don't use **app** folder for your project
- There will be a separate lesson for app folder features

02



Next.js features

Next.js features

Getting started

- Instead of using create react-app, use the following command:
 - `yarn create next-app [app name] --typescript` - default template
 - You'll get a (hopefully) better template for your project
- Project structure is different than classic React
 - All generated files are in `pages` directory or inside `src/pages`
- Server-side and client-side routing and rendering out of the box
- Important: **disable strict mode in `next.config.js` file**
 - Among other things, it runs `useEffect` twice on render

Next.js features

Pages

- Two types of pre-rendering:
- 1. **Static generation** - HTML is generated **at build time**
 - Definite (final) number of pages
- 2. **Server-side rendering** - HTML is generated **on each request**
 - Indefinite (unknown) number of pages based on parameters
- Pages folder offers routing out of the box
- **_app.tsx** file - all components are wrapped in its default export
 - The place to add SWRConfig or similar things
- **index.tsx** file - default page

Next.js features

Static pages

- Without data - generates single HTML file
 - Example: `pages/about.tsx`
- With data - HTML + JSON props - two options:
 - Fixed path, content depends on external data - `getStaticProps`
 - Example - `pages/privacy.tsx`
 - Variable path (dynamic routing) - `getStaticPaths` + `getStaticProps`
 - Example - `pages/country/[alpha2].tsx`
 - The idea is to have a limited number of pages, e.g. Sofascore motorsport categories
- Still can use client-side rendering



Next.js example - next.md



03

Server-side rendering with Next.js



Server-side rendering with Next.js

SSR Basics

- HTML is not created at build time, but at each request
- Must implement method `getServerSideProps`
 - Fetch from dynamic routes
- Default export is your page component

Server-side rendering with Next.js

Client vs server

- `useEffect` is run on client
- `console.log(...)` on server is written in the terminal running the app
 - Important for debugging
- Most of other code is run on server
- After being rendered on server, React is rehydrated on client
 - Rehydration is something like a partial re-rendering (in layman's terms)
 - To force re-rendering on client, use `useIsServer` hook

Server-side rendering with Next.js

Routing

- Next gives you routing by default
 - Link components instead of a
 - useRouter() hook
 - `const router = useRouter()` and `router.push(path)`
 - Translated routes are also an option
- [More on routing](#)



Next.js SSR example - ssr.md



**Thank you for your
attention!**

