

Next.js 기본 2

1 Next.js

Pre-rendering

(<https://nextjs.org/learn/basics/data-fetching>)

hydration: 페이지가 브라우저에 로드되고 자바스크립트 코드가 실행되면서 페이지가 인터랙티브하게 동작할 상태가 되는 과정

Next.js는 기본적으로 모든 페이지를 pre-rendering 한다.

disable javascript 해보기

반면, create-react-app 으로 만든 순수한 react-app에게
javascript disable은 치명적이다.

Two Forms for Pre-rendering

(<https://nextjs.org/learn/basics/data-fetching/two-forms>)

SSG vs SSR

(build time vs each request)

단, npm run dev 로 킨 경우는 무조건 each request 마다 빌드한다.

Next.js의 장점

각각의 페이지마다 원하는
pre-rendering 로직을 적용할 수 있다

모든 페이지가 외부 Data fetching을 필요로 하는 것은 아니다.

필요없는 페이지는 build time에 statically generated 된다.

모든 페이지가 외부 Data fetching을 필요로 하는 것은 아니다.

데이터가 필요한 페이지에는 `getStaticProps`라는 `async` 함수를 구현하면 된다. (그럼 빌드 타임에 해당 함수를 실행해서 데이터를 페칭해온다.)

단, development 모드에서는 each request 마다 `getStaticProps`가 실행된다.

(<https://nextjs.org/learn/basics/data-fetching/with-data>)

Blog Data 추가

/posts 에 md 파일 추가

(<https://nextjs.org/learn/basics/data-fetching/blog-data>)

title / date 는 YAML Front Matter

gray-matter 라는 라이브러리로 해석 가능함

(<https://nextjs.org/learn/basics/data-fetching/implement-getstaticprops>)

외부 데이터 가져오기

(<https://nextjs.org/learn/basics/data-fetching/getstaticprops-details>)

fetch() 함수는 next.js 가 client-side / server-side 모두에 polyfills 해둠

외부 api 접근 / 직접 DB 접근도 가능함

사용자의 요청마다 외부 데이터 가져오기

(<https://nextjs.org/learn/basics/data-fetching/request-time>)

getServerSideProps 쓰면

TTFB(Time to first byte)는 getStaticProps 보다 느림

Client-side Rendering을 고려한다면 SWR(캐싱 / 패칭) 추천

정리

Next.js 기본 2

hydration

js 실행으로 interactive 할 준비 과정

SSG

Build time 에 서버에서만 동작

SSR

Request time에 동작 / TTFB slow

CSR

Request time 이후 동작 SWR