NextJS (приложение my-nextjs-app)

Виды рендеринга

CSR- приложение рендерится на клиенте. Пример обычное реакт приложение - компоненты помеченные ‘use client’

SSR – приложение рендерится на сервере и отдается клиенту. При этом возможна гидратация - по умолчанию

SSG- рендерится на сервере в процессе билда (то есть может отдавать страницу по частям – по аналогии как чат GPT дает ответ) Без гидратации - Указываем generateStaticParams

ISG –смесь предыдущих двух подходов - не совсем понял

1. Установка

npx create-next-app@latestЗапуск

npm run dev

2. Серверный и клиентский рендеринг

По умолчанию либо если в верху фаила указано 'use server' - компонента либо страница будет рендериться на сервере. Если же указано 'use client' - отрисовка на стороне клиента.

Рекомендуется все базовые компоненты (из которых строится приложение) - рендерить на сервере, а все что содержит хуки, обращения к дому, онклики и так далее (кнопки инпуты и т д) - отрисовывать на клиенте

3. Реализация SSG, SSR, ISG

В 14 нексте это все реализовывается через доп параметр в fetch

- кэширование данных (по дефолту)

fetch('https://...', { cache: 'force-cache' })

- без кэширования

fetch('https://...', { cache: 'no-store' })

- ревалидация (проверять и подтягивать новые данные через количество секунд)

fetch('https://...', { next: { revalidate: 3600 } })

4. App Route

Пути в url будут формироваться взависимости от структуры папок в src/app . За название в пути отвечает название папки, сам фаил называется всегда page.tsx. Если название папки в виде (folderName) - она не участвует в создании пути. Если же название в виде [someId] - то путь динамический и из url можно будет вытянуть someId

В корне src/app фаил not-found.tsx - будет отрисовываться если 'нет такого пути'

Лайауты - можно на любом уровне создавать фаил layout.tsx и все последующие страницы будут в него обернуты.

5. Маршрутизация и навигация

<Link href="/page4">Link to page 4</Link>

-------

const pathname = usePathname();

// /page2

-------

const params = useParams<{ tag: string; item: string }>()

// Route -> /shop/[tag]/[item]

// URL -> /shop/shoes/nike-air-max-97

// `params` -> { tag: 'shoes', item: 'nike-air-max-97' }

-------

const router = useRouter()

<button type="button" onClick={() => router.push('/dashboard')}>

-------

const searchParams = useSearchParams()

const search = searchParams.get('search')

// `search` -> 'my-project'

6. Meta-данные

Что б указать title приложения, описание, иконку и другие данные - метаданные подключаются на любую страницу в виде

export const metadata: Metadata = {

title: "Title for first layout",

description: "Generated by create next app",

};

7. Turbopack

При его подключении в режиме разработки он будет собирать приложение вместо вебпака - и он быстрее в несколько раз.

В package.json изменить script "dev": "next dev --turbo",

Что такое гидротация ? При открытии приложения сначала формируется серверная часть компонент - и этого достаточно для критического рендеринга. Но так же после этого часть функционала формируется на клиенте (вешаются обработчики, выполняются все действия связанные с DOM и т д ) . Процесс исполнения скрипта на клиенте с целью изменения функционала страницы, полученной с сервера - это и есть гидратация.