

Dokumentace

- Data fetching
 - Static generation
 - Proběhne v build fázi, implementace metod `getStaticProps` a `getStaticPaths`.
 - `getStaticPaths` má hodnotu `fallback`:
 - `fallback: false` (default) během buildu.
 - `fallback: true` - rebuild bude probíhat na pozadí.
 - `fallback: blocking` - build bude probíhat před prvním requestem.
 - `getStaticProps` proběhne vždy během buildu
 - Metoda `revalidate()` umožňuje provést rebuild na pozadí.
 - Metoda `unstable_revalidate()` umožňuje provést rebuild on demand.
 - Server Side rendering
 - Implementuje se jen na serveru pomocí `getServerSideProps`.
 - Pages using `getServerSideProps` will be server side rendered at request time and only be cached if cache-control headers are configured.
 - Nastavení hlaviček (cachova na 59s):
 - `Cache-Control: public, s-maxage=10, stale-while-revalidate=59`
 - Jaký kód se posílá na klienta: <https://next-code-elimination.vercel.app>.
 - Incremental Static Regeneration (průběžné buildování stránek):
 - Kombinace `revalidate: XX + fallback: blocking`.
 - On Demand revalidace:
 - Zavolá se externě `res.unstable_revalidate()` a ta vymaže cache.
- Static File Serving
 - Jen ve složce `public`.
- Fast Refresh
 - Přidat komentář `// @refresh reset` pokud chci, aby došlo při editaci ke kompletnímu refreshi stránky.
 - Hooky `useEffect`, `useMemo`, and `useCallback` se vždy updatují při refreshi stránky.
- Typescript
 - Používat return typy `GetStaticProps`, `GetStaticPaths`, `GetServerSideProps`.
- Environment Variables
 - Variables, které chci v public repu deklarovat s předponou `NEXT_PUBLIC`.
 - Možno používat uvnitř proměnné jako `$VAR`.
- Supported Browsers and Features
 - Polyfill `fetch()` je dostupný všude.
- Handling Scripts
 - Vlastní script tag, několik strategií, jak je načítat.
- Routing
 - Parametr `href` je možné předávat jako objekt.
 - Catch all routes can be made optional by including the parameter in double brackets (`[...slug]`).
 - Import typů přes `import type { NextApiResponse, NextApiRequest } from 'next'`.
 - Možno doplnit dvojité závorky `[...slug]` k parametru, pak se na routu přesměruje i index.
 - Parametr `shallow` umožňuje jen změnit URL bez opětovného načtení dat přes API.
- API routes
 - Doplnit CORS, standardně na stejné doméně, já používám `api.xxx`.

- Z rout se exportuje `config`, kde je možné nastavit pár middleware.
- Production
 - Databáze a server musí být ve stejné lokaci.
 - Projít checklist před releasem.
- Deployment
 - Příklad na implementaci s Dockerem.
 - Next.js will automatically load the latest version of your application in the background when routing.
- Dynamic Import
 - Podpora pro dynamický import komponent: `dynamic(() => import('./components/a'))`.
- Automatic Static Optimization
 - Stránka je statická, pokud nemá implementovány vlastnosti `getServerSideProps` a `getInitialProps`.
 - Pokud možno, nepoužívat `getInitialProps`, vypne se optimalizace.
- Static HTML Export
 - Pokud je aplikace statická a nepotřebuje server, vyexportovat jn html pomocí `next export` místo buildu. Úpravy v konfiguraci `next.js exportPathMap`. Některé důležité vlastnosti chybí.
 - `getInitialProps` je asi sttarší, nepoužívat.
- Using MDX
 - Vlastní MDX parser od Next.js.
- Errors
 - Možno vytvořit stránku `pages/_error.js` pro zpracování chyb, které jdou pak do stránky `404.js` a `500.js`. Nebo využít `import Error from 'next/error'`.
- Measuring performance
 - Definice `reportWebVitals()` pro sbírání dat o výkonnosti Web Vitals pro analytiku.
- Middleware
 - V souboru `pages/_middleware.tsx` a podadresářích, spouští se na edge platformách.
- Debugging
 - Nastavení debugování podle návodu.
- Error Handling
 - Implementace `ErrorBoundary` uvnitř App.
- Codemods
 - Utilita pro převod starých věcí na nové featury.
- Internationalized Routing
 - Next.js doesn't know about variants of a page so it's up to you to add the `hreflang` meta tags using `next/head`.
 - For pages using `getStaticProps` with Dynamic Routes, all locale variants of the page desired to be prerendered need to be returned from `getStaticPaths`.
- Security Headers
 - Užitečné jsou zejména `X-Content-Type-Options` či `Content-Security-Policy`.
- CLI
 - Port je možné specifikovat jako `next dev -p 4000`
- Router
 - `router.push()` a `nextjs/link` a parametry `scroll` (bool).
 - Router events pro různé události
- Next.config.js Custom Page Extensions
 - Možno povolit třeba jen stránky s koncovkou `page.react.js`, tím se člověk vyhne tomu, že je nutné používat podtržítka pro privátní stránky.

Další zdroje

- Projit [nextjs-awesome](#) aplikace
- Projit příklady v repu nextjs