mukul rajpoot

DIFFERENT DATA FETCHING TECHNIQUE IN NEXT.JS

munkul rajpoot

WHAT IS NEXT.JS?

NEXT.JS - The React Framework for Production

Next.js gives you the best developer experience with all the features you need for production: hybrid static & server rendering, TypeScript support, smart bundling, route pre-fetching, and more. Zero config needed.

It provides a common structure that allows you to easily build a frontend React application, and transparently handles server-side rendering for you.

mukul rajpoot

NEXT.JS FEATURES

1. Hot Code Reloading - Next.js reloads the page when it detects any change saved to disk.
2. File-based Routing - Any URL is mapped to the filesystem, to files put in the pages folder, and you don't need any configuration (you have customization options of course).
3. Server Rendering - You can render React components on the server side, before sending the HTML to the client.
4. Ecosystem Compatibility - Next.js plays well with the rest of the JavaScript, Node, and React ecosystem.

mukul rajpoot

NEXT.JS FEATURES

1. Automatic Code Splitting - Pages are rendered with just the libraries and JavaScript that they need, no more. Instead of generating one single JavaScript file containing all the app code, the app is broken up automatically by Next.js in several different resources.Loading a page only loads the JavaScript necessary for that particular page
2. Prefetching - The Link component, used to link together different pages, supports a prefetch prop which automatically prefetches page resources in the background.

mukul rajpoot

SETUP A NEXT.JS PROJECT

To install Next.js, you need to have Node.js installed.

If you're familiar with create-react-app, create-next-app is the same thing - except it creates a Next app instead of a React app, as the name implies.

```
npx create-next-app <project-name>
```

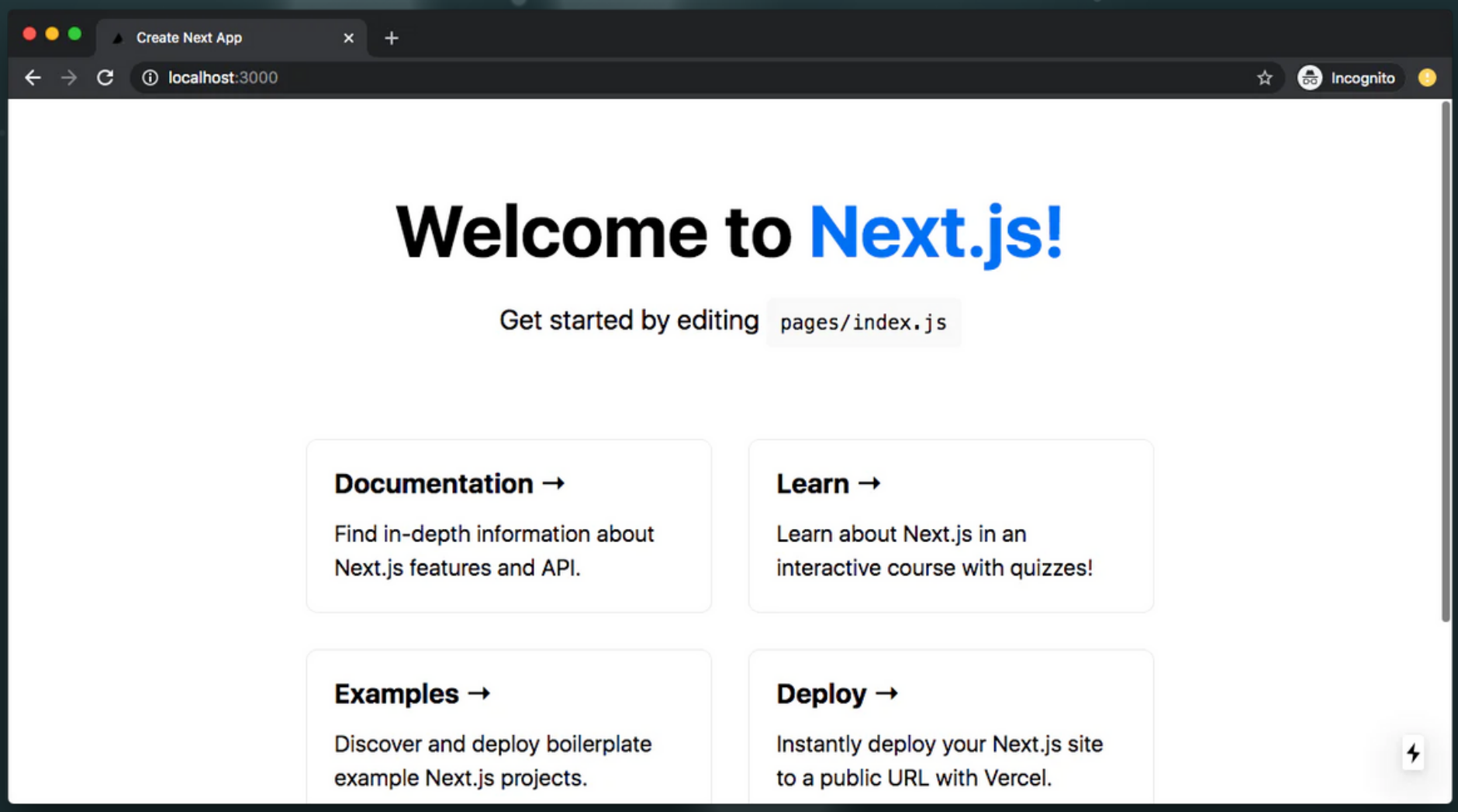
After completion, you can spin up the development server by

```
npm run dev
```

mukul rajpoot

SETUP A NEXT.JS PROJECT

And your next.js project running on http://localhost:3000



mukul rajpoot

Data Fetching in Next.js

Data fetching in Next.js allows you to render your content in different ways, depending on your application's use case. These include pre-rendering with Server-side Rendering or Static Generation, and updating or creating content at runtime with Incremental Static Regeneration.

- Server-Side Rendering (SSR)
- Static-Site Generation (SSG)
- Client-Side Rendering (CSR)
- Incremental Static Generation (ISR)

mukul rajpoot

SERVER-SIDE RENDERING IN NEXT.JS

Server-side rendering (SSR), is the ability of an application to contribute by displaying the web-page on the server instead of rendering it in the browser.Server-side sends a fully rendered page to the client;

If you export a function called getServerSideProps from a page, Next.js will pre-render this page on each request using the data returned by getServerSideProps.

getServerSideProps only runs on server-side and never runs on the browser

mukul rajpoot

STATIC-SITE GENERATION IN NEXT.JS

Static Generation describes the process of compiling and rendering a website or app at build time. The output is a bunch of static files, including the HTML file itself and assets like JavaScript and CSS.

If you export a function called `getStaticProps` from a page, Next.js will pre-render this page at build time using the props returned by `getStaticProps`.

`getStaticProps` runs only on the server-side, it will never run on the client-side.

mukul rajpoot

CLIENT-SIDE RENDERING IN NEXT.JS

Client-side rendering (CSR) means rendering pages directly in the browser using JavaScript. All logic, data fetching, templating and routing are handled on the client rather than the server.

You can fetch data on the client side using the useEffect hook as you are doing previously in react applications

It's important to note that using client-side data fetching can affect the performance of your application and the load speed of your pages.

mukul rajpoot

CLIENT-SIDE RENDERING IN NEXT.JS

The team behind Next.js has created a React hook library for data fetching called SWR. It is highly recommended if you are fetching data on the client-side. It handles caching, revalidation, focus tracking, refetching on intervals, and more.

SWR will automatically cache the data for us and will revalidate the data if it becomes stale. Also, increases the performance of the application.

mukul rajpoot

INCREMENTAL-SITE REGENERATION IN NEXT.JS

Next.js allows you to create or update static pages after you've built your site.

Incremental Static Regeneration (ISR) enables you to use static-generation on a per-page basis, without needing to rebuild the entire site. With ISR, you can retain the benefits of static while scaling to millions of pages.

To use ISR add the revalidate prop to getStaticProps

mukul rajpoot

INCREMENTAL-SITE REGENERATION IN NEXT.JS

When a request is made to a page that was pre-rendered at build time, it will initially show the cached page. Let's say we pass 10 seconds to revalidate props in getstaticProps:

- Any requests to the page after the initial request and before 10 seconds are also cached and instantaneous.
- After the 10-second window, the next request will still show the cached (stale) page
- Next.js triggers a regeneration of the page in the background.
- Once the page has been successfully generated, Next.js will invalidate the cache and show the updated page. If the background regeneration fails, the old page would still be unaltered.

mukul rajpoot

NEXT.JS IS A TRUE BEAST
HAPPY LEARNING