Custom `App`

Next.js uses the `App` component to initialize pages. You can override it and control the page initialization. Which allows you to do amazing things like:

- Persisting layout between page changes

- Keeping state when navigating pages

- Custom error handling using `componentDidCatch`

- Inject additional data into pages

- [Add global CSS](#)

To override the default `App`, create the file `./pages/_app.js` as shown below:

```
// import App from 'next/app'

function MyApp({ Component, pageProps }) {
  return <Component {...pageProps} />
}

// Only uncomment this method if you have blocking data requirements for
// every single page in your application. This disables the ability to
// perform automatic static optimization, causing every page in your app to
// be server-side rendered.
//
// MyApp.getInitialProps = async (appContext) => {
//   // calls page's `getInitialProps` and fills `appProps.pageProps`
//   const appProps = await App.getInitialProps(appContext);
//
```

```
//   return { ...appProps }
// }


export default MyApp
```

The `Component` prop is the active `page`, so whenever you navigate between routes, `Component` will change to the new `page`. Therefore, any props you send to `Component` will be received by the `page`.

`pageProps` is an object with the initial props that were preloaded for your page by one of our data fetching methods, otherwise it's an empty object.

Caveats

- If your app is running and you just added a custom `App`, you'll need to restart the development server. Only required if `pages/_app.js` didn't exist before.

- Adding a custom `getInitialProps` in your `App` will disable Automatic Static Optimization in pages without Static Generation.

- `App` currently does not support Next.js Data Fetching methods like `getStaticProps` or `getServerSideProps`.

TypeScript

If you're using TypeScript, take a look at our TypeScript documentation.

Related

For more information on what to do next, we recommend the following sections:

Automatic Static Optimization:

Next.js automatically optimizes your app to be static HTML whenever possible.
Learn how it works here.

Custom Error Page:

Learn more about the built-in Error page.

Was this helpful?

😭   😕   😃   🤩

Edit this page on GitHub