<https://clerk.com/docs/references/nextjs/custom-signup-signin-pages>

1. [Next.js](https://clerk.com/docs/references/nextjs/overview)
2. Add custom sign up and sign in pages

**Build your own sign-in and sign-up pages for your Next.js app with Clerk**

This guide shows you how to use the [<SignIn />](https://clerk.com/docs/components/authentication/sign-in) and [<SignUp />](https://clerk.com/docs/components/authentication/sign-up) components with the [Next.js optional catch-all route](https://nextjs.org/docs/pages/building-your-application/routing/dynamic-routes#catch-all-segments) in order to build custom sign-in and sign-up pages for your Next.js app.

If Clerk's prebuilt components don't meet your specific needs or if you require more control over the logic, you can rebuild the existing Clerk flows using the Clerk API. For more information, check out the [custom flows](https://clerk.com/docs/custom-flows/overview) guides.

Note

Watch the video version of this guide on the Clerk YouTube channel → [YouTube (4 minutes)](https://youtu.be/fsuHLafTYyg).

Note

Just getting started with Clerk and Next.js? Check out the [quickstart tutorial](https://clerk.com/docs/quickstarts/nextjs)!

[**Build a sign-up page**](https://clerk.com/docs/references/nextjs/custom-signup-signin-pages#build-a-sign-up-page)

Create a new file that will be used to render the sign-up page. In the file, import the [<SignUp />](https://clerk.com/docs/components/authentication/sign-up) component from @clerk/nextjs and render it.

App Router

Pages Router

app/sign-up/[[...sign-up]]/page.tsx

import { SignUp } from '@clerk/nextjs'

export default function Page() {

return <SignUp />

}

[**Build a sign-in page**](https://clerk.com/docs/references/nextjs/custom-signup-signin-pages#build-a-sign-in-page)

Create a new file that will be used to render the sign-in page. In the file, import the [<SignIn />](https://clerk.com/docs/components/authentication/sign-in) component from @clerk/nextjs and render it.

App Router

Pages Router

app/sign-in/[[...sign-in]]/page.tsx

import { SignIn } from '@clerk/nextjs'

export default function Page() {

return <SignIn />

}

[**Make the sign-up and sign-in routes public**](https://clerk.com/docs/references/nextjs/custom-signup-signin-pages#make-the-sign-up-and-sign-in-routes-public)

By default, clerkMiddleware() makes all routes public. This step is specifically for applications that have configured clerkMiddleware() to make [all routes protected](https://clerk.com/docs/references/nextjs/clerk-middleware#protect-all-routes). If you have not configured clerkMiddleware() to protect all routes, you can skip this step.

To make the sign-up and sign-in routes public:

* Navigate to your middleware.ts file.
* Create a new [route matcher](https://clerk.com/docs/references/nextjs/clerk-middleware#create-route-matcher) that matches the sign-up and sign-in routes, or you can add them to an existing route matcher that is making routes public.
* Create a check to see if the user's current route is a public route. If it is not a public route, use [auth().protect()](https://clerk.com/docs/references/nextjs/auth#protect) to protect the route.

middleware.ts

import { clerkMiddleware, createRouteMatcher } from '@clerk/nextjs/server'

const isPublicRoute = createRouteMatcher(['/sign-in(.\*)', '/sign-up(.\*)'])

export default clerkMiddleware((auth, request) => {

if (!isPublicRoute(request)) {

auth().protect()

}

})

[**Update your environment variables**](https://clerk.com/docs/references/nextjs/custom-signup-signin-pages#update-your-environment-variables)

In the previous steps, a path prop is passed to the <SignIn /> and <SignUp /> components. This is because the components need to know which route they are originally mounted on.

In Next.js applications, you can either pass the path prop, *or* you can define the NEXT\_PUBLIC\_CLERK\_SIGN\_IN\_URL and NEXT\_PUBLIC\_CLERK\_SIGN\_UP\_URL environment variables, like so:

.env.local

NEXT\_PUBLIC\_CLERK\_SIGN\_IN\_URL=/sign-in

NEXT\_PUBLIC\_CLERK\_SIGN\_UP\_URL=/sign-up

[**Visit your new pages**](https://clerk.com/docs/references/nextjs/custom-signup-signin-pages#visit-your-new-pages)

Run your project with the following terminal command from the root directory of your project:

npm

yarn

pnpm

terminal

npm run dev

Visit your new custom pages locally at [localhost:3000/sign-in](http://localhost:3000/sign-in) and [localhost:3000/sign-up](http://localhost:3000/sign-up).