

Getting Started > Session Management > Signin and Signout

Handling Signin and Signout

To signin your users, make sure you have at least one [authentication method](#) setup. You then need to build a button which will call the sign in function from your Auth.js framework package.

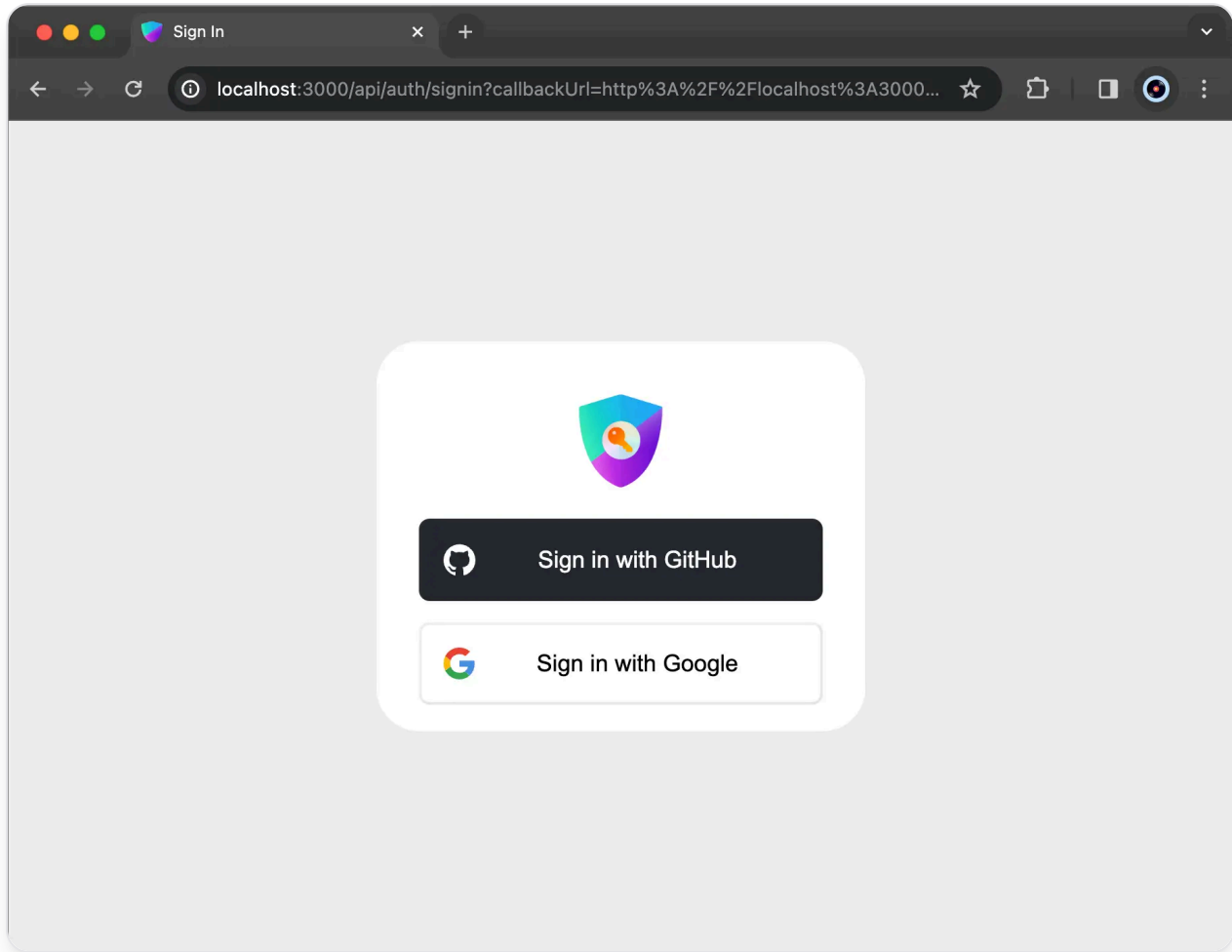
[Next.js](#) [Next.js \(Client\)](#) [Qwik](#) [SvelteKit](#) [Express](#)

```
TS ./components/auth/signin-button.tsx

import { signIn } from "@auth"

export function SignIn() {
  return (
    <form
      action={async () => {
        "use server"
        await signIn()
      }}
    >
      <button type="submit">Sign in</button>
    </form>
  )
}
```

You can also pass a provider to the `signIn` function which will attempt to login directly with that provider. Otherwise, when clicking this button in your application, the user will be redirected to the configured sign in page. If you did not setup a [custom sign in page](#), the user will be redirected to the default signin page at `/[basePath]/signin`.



Once authenticated, the user will be redirected back to the page they started the signin from. If you want the user to be redirected somewhere else after sign in (i.e `/dashboard`), you can do so by passing the target URL as `redirectTo` in the sign-in options.

[Next.js](#) [Next.js \(Client\)](#) [Qwik](#) [SvelteKit](#) [Express](#)

TS app/components/signin-button.tsx

```
import { signIn } from "@auth.ts"

export function SignIn() {
  return (
    <form
      action={async () => {
        "use server"
        await signIn("github", { redirectTo: "/dashboard" })
      }}
    >
      <button type="submit">Sign in</button>
    </form>
  )
}
```

```
    </form>
  )
}
```

Signout

Signing out can be done similarly to signing in. Most frameworks offer both a client-side and server-side method for signing out as well.

[Next.js](#) [Next.js \(Client\)](#) [Qwik](#) [SvelteKit](#) [Express](#)

To sign out users with a form action, you can build a button that calls the exported signout function from your Auth.js config.

```
TS app/components/signout-button.tsx

import { signOut } from "@/auth.ts"

export function SignOut() {
  return (
    <form
      action={async () => {
        "use server"
        await signOut()
      }}
    >
      <button type="submit">Sign Out</button>
    </form>
  )
}
```



Note that when signing out of an OAuth provider like GitHub in an Auth.js application, the user will not be signed out of GitHub elsewhere.

Last updated on January 9, 2025

About Auth.js

[Introduction](#)

[Security](#)

[Discord Community](#)

Download

[GitHub](#)

[NPM](#)

Acknowledgements

[Contributors](#)

[Sponsors](#)

Auth.js © Balázs Orbán and Team - 2025