

1 Installing Auth.js

Start by installing the appropriate package for your framework.

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

[npm](#) [pnpm](#) [yarn](#) [bun](#)

```
npm install next-auth@beta
```

i Installing `@auth/core` is not necessary, as a user you should never have to interact with `@auth/core`.

2 Setup Environment

The only environment variable that is mandatory is the `AUTH_SECRET`. This is a random value used by the library to encrypt tokens and email verification hashes. (See [Deployment](#) to learn more). You can generate one via the official [Auth.js CLI](#) running:

```
npx auth secret
```

This will also add it to your `.env` file, respecting the framework conventions (eg.: Next.js' `.env.local`).

3 Configure

Next, create the Auth.js config file and object. This is where you can control the behaviour of the library and specify custom authentication logic, adapters, etc. We recommend all frameworks to create an `auth.ts` file in the project. In this file we'll pass in all the options

to the framework specific initialization function and then export the route handler(s), signin and signout methods, and more.

i You can name this file whatever you want and place it wherever you like, these are just conventions we've come up with.

Next.js Qwik SvelteKit Express

1. Start by creating a new `auth.ts` file at the root of your app with the following content.

TS `./auth.ts`

```
import NextAuth from "next-auth"

export const { handlers, signIn, signOut, auth } = NextAuth({
  providers: [],
})
```

2. Add a Route Handler under `/app/api/auth/[...nextauth]/route.ts`.

💡 This file must be an App Router Route Handler, however, the rest of your app can stay under `page/` if you'd like.

TS `./app/api/auth/[...nextauth]/route.ts`

```
import { handlers } from "@/auth" // Referring to the auth.ts we just created
export const { GET, POST } = handlers
```

3. Add optional Middleware to keep the session alive, this will update the session expiry every time its called.

TS `./middleware.ts`

```
export { auth as middleware } from "@/auth"
```

4

Setup Authentication Methods

With that, the basic setup is complete! Next we'll setup the first authentication methods and fill out that `providers` array.

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