Getting Started > Authentication > OAuth

OAuth

Auth.js comes with over 80 providers preconfigured. We constantly test ~20 of the most popular ones, by having them enabled and actively used in our example application. You can choose a provider below to get a walk-through, or find your provider of choice in the sidebar for further details.

Google

1 Register OAuth App in Google's dashboard

First you have to setup an OAuth application on the Google developers dashboard.

> If you haven’t used OAuth before, you can read the beginners step-by-step guide on how to setup "Sign in with GitHub" with Auth.js.

When registering an OAuth application on Google, they will all ask you to enter your application’s callback URL. See below for the callback URL you must insert based on your framework.

Callback URL

Next.js Qwik SvelteKit Express

```
[origin]/api/auth/callback/google
```

> Many providers only allow you to register one callback URL at a time. Therefore, if you want to have an active OAuth configuration for development and production environments, you'll need to register a second OAuth app in the Google dashboard for the other environment(s).

2 Setup Environment Variables

Once registered, you should receive a **Client ID** and **Client Secret**. Add those in your application environment file:

Next.js Qwik SvelteKit Express

.env.local

```
AUTH_GOOGLE_ID={CLIENT_ID}
AUTH_GOOGLE_SECRET={CLIENT_SECRET}
```

Auth.js will automatically pick up these if formatted like the example above. You can also use a different name for the environment variables if needed, but then you'll need to pass them to the provider manually.

3 Setup Provider

Let's enable Google as a sign in option in our Auth.js configuration. You'll have to import the `Google` provider from the package and pass it to the `providers` array we setup earlier in the Auth.js config file:

Next.js Qwik SvelteKit Express

In Next.js we recommend setting up your configuration in a file in the root of your repository, like at `auth.ts`.

./auth.ts

```
import NextAuth from "next-auth"
import Google from "next-auth/providers/google"

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

Add the `handlers` which `NextAuth` returns to your `api/auth/[...nextauth]/route.ts` file so that Auth.js can run on any incoming request.

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

```
import { handlers } from "@/auth"
export const { GET, POST } = handlers
```

4 Add Signin Button

Next, we can add a signin button somewhere in your application like the Navbar. It will trigger Auth.js sign in when clicked.

Next.js Next.js (Client) Qwik SvelteKit Express

./components/sign-in.tsx

```
import { signIn } from "@/auth"

export default function SignIn() {
  return (
    <form
      action={async () => {
        "use server"
        await signIn("google")
      }}
    >
      <button type="submit">Signin with Google</button>
    </form>
  )
}
```

5 Ship it!

Click the “Sign in with Google" button and if all went well, you should be redirected to Google and once authenticated, redirected back to the app!

> 💡 You can build your own Signin, Signout, etc. pages to match the style of your application, check out session management for more details.

For more information on this provider check out the detailed Google provider docs page.

Last updated on January 9, 2025

Auth.js © Balázs Orbán and Team - 2025