

# With Cookie Auth and Fauna

In this example, we authenticate users and store a token in a secure (non-JS) cookie. The example only shows how the user session works, keeping a user logged in between pages.

This example uses [Fauna](#) as the auth service and DB.

The repo includes a minimal auth backend built with [API Routes](#) and [Fauna for Auth](#). The backend allows the user to create an account (a User document), login, and see their user id (User ref id).

Session is synchronized across tabs. If you logout your session gets removed on all the windows as well. We use the HOC `withAuthSync` for this.

The helper function `auth` helps to retrieve the token across pages and redirects the user if not token was found.

## How to use

Execute `create-next-app` with [npm](#) or [Yarn](#) to bootstrap the example:

```
npx create-next-app --example with-cookie-auth-fauna with-cookie-auth-fauna-app
# or
yarn create next-app --example with-cookie-auth-fauna with-cookie-auth-fauna-app
# or
npm create next-app -- --example with-cookie-auth-fauna with-cookie-auth-fauna-app
```

## Run locally

First, you'll need to create an account on [Fauna](#), then follow these steps:

1. In the [FaunaDB Console](#), click "New Database". Name it whatever you like and click "Save".
2. Click "New Collection", name it `User`, leave "Create collection index" checked, and click "Save".
3. Now go to "Indexes" in the left sidebar, and click "New Index". Select the `User` collection, call it `users_by_email`, and in the "terms" field type `data.email`. Select the "Unique" checkbox and click "Save". This will create an index that allows looking up users by their email, which we will use to log a user in.
4. Next, go to "Security" in the sidebar, then click "New Key". Create a new key with the `Server` role, call it `server-key`, and click "Save". Your key's secret will be displayed, copy that value.
5. Create the `.env.local` file (which will be ignored by Git) based on the `.env.local.example` file in this directory:

```
cp .env.local.example .env.local
```

6. Paste the secret you copied as the value for `FAUNA_SERVER_KEY` in the `.env.local` file. Keep this key safely as it has privileged access to your database.

For more information, read the [User Authentication Tutorial in Fauna](#).

Now, install it and run:

```
npm install
npm run dev
```

```
# or  
yarn  
yarn dev
```

## Deploy on Vercel

You can deploy this app to the cloud with [Vercel](#) ([Documentation](#)).

### Deploy Your Local Project

To deploy your local project to Vercel, push it to GitHub/GitLab/Bitbucket and [import to Vercel](#).

**Important:** When you import your project on Vercel, make sure to click on **Environment Variables** and set them to match your `.env.local` file.

### Deploy from Our Template

Alternatively, you can deploy using our template by clicking on the Deploy button below.

