

A statically generated blog example using Next.js and Prismic

This example showcases Next.js's [Static Generation](#) feature using [Prismic](#) as the data source.

Demo

<https://next-blog-prismic.vercel.app/>

Deploy your own

Once you have access to [the environment variables you'll need](#), deploy the example using [Vercel](#):



Related examples

- [WordPress](#)
- [DatoCMS](#)
- [Sanity](#)
- [TakeShape](#)
- [Contentful](#)
- [Strapi](#)
- [Agility CMS](#)
- [Cosmic](#)
- [ButterCMS](#)
- [Storyblok](#)
- [Kontent](#)
- [Ghost](#)
- [Umbraco Heartcore](#)
- [Blog Starter](#)
- [Builder.io](#)

How to use

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

```
npx create-next-app --example cms-prismic cms-prismic-app
# or
yarn create next-app --example cms-prismic cms-prismic-app
# or
pnpm create next-app -- --example cms-prismic cms-prismic-app
```

Configuration

Step 1. Create an account and a repository on Prismic

First, [create an account on Prismic](#).

After creating an account, create a **repository** from the [dashboard](#) and assign to it any name of your liking.

Step 2. Create an `author` type

From the repository page, create a new **custom type**:

- The name should be `author`.

Next, add these fields (you don't have to modify the settings):

- `name` - **Key Text** field
- `picture` - **Image** field

Alternatively, you can copy the JSON in [types/author.json](#), then click on **JSON editor** and paste it there.

Save the type and continue.

Step 3. Create a `post` type

From the repository page, create a new **custom type**:

- The name should be `post`.

Next, add these fields (you don't have to modify the settings unless specified):

- `title` - **Title** field
- `content` - **Rich Text** field
- `excerpt` - **Key Text** field
- `coverimage` - **Image** field
- `date` - **Date** field
- `author` - **Content relationship** field, you may also add `author` to the **Constraint to custom type** option to only accept documents from the `author` type.
- `slug` - **UID** field.

Alternatively, you can copy the JSON in [types/post.json](#), then click on **JSON editor** and paste it there.

Save the type and continue.

Step 4. Populate Content

Go to the **Content** page, it's in the menu at the top left, then click on **Create new** and select the **author** type:

- You just need **1 author document**.
- Use dummy data for the text.
- For the image, you can download one from [Unsplash](#).

Next, select **Post** and create a new document.

- We recommend creating at least **2 Post documents**.
- Use dummy data for the text.
- You can write markdown for the **content** field.
- For images, you can download them from [Unsplash](#).
- Pick the **author** you created earlier.

Important: For each document, you need to click **Publish** after saving. If not, the document will be in the draft state.

Step 5. Set up environment variables

Follow the instructions in [this post](#) to generate a new access token.

Next, copy the `.env.local.example` file in this directory to `.env.local` (which will be ignored by Git):

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

Then set each variable on `.env.local` :

- `PRISMIC_API_TOKEN` should be the **Permanent access token** you just created
- `PRISMIC_REPOSITORY_NAME` is the name of your repository (the one in the URL)
- `PRISMIC_REPOSITORY_LOCALE` is the locale of your repository. Defaults to `en-us`

Your `.env.local` file should look like this:

```
PRISMIC_API_TOKEN=...
PRISMIC_REPOSITORY_NAME=...
PRISMIC_REPOSITORY_LOCALE=...
```

Make sure the locale matches your settings in the Prismic dashboard.

Step 6. Run Next.js in development mode

```
npm install
npm run dev

# or

yarn install
yarn dev
```

Your blog should be up and running on <http://localhost:3000>! If it doesn't work, post on [GitHub discussions](#).

Step 7. Try preview mode

On your repository page, go to **Settings**, click on **Previews** and then **Create a New Preview** for development, fill the form like so:

- **Site Name:** may be anything, like `development`
- **Domain of Your Application:** `http://localhost:3000`
- **Link Resolver:** `/api/preview`

Once saved, go to one of the posts you've created and:

- **Update the title.** For example, you can add `[Draft]` in front of the title.
- Click **Save**, but **DO NOT** click **Publish**. By doing this, the post will be in draft state.
- Right next to the **Publish** button you should see the **Preview** button, displayed with an eye icon. Click on it!

You should now be able to see the updated title. To exit preview mode, you can click on **Click here to exit preview mode** at the top of the page.

Step 8. 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.

