

A statically generated blog example using Next.js and DatoCMS

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

Demo

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

Related examples

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

Deploy your own

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



How to use

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

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

Configuration

Step 1. Create an account and a project on DatoCMS

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

After creating an account, create a **new project** from the dashboard. You can select a **Blank Project**.

Step 2. Create an **Author** model

From the project setting page, create a new **Model**.

- The name should be **Author** .

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

- **Name** - **Text** field (**Single-line String**)
- **Picture** - **Media** field (**Single asset**)

Step 3. Create a **Post** model

From the project setting page, create a new **Model**:

- The name should be **Post** .
- **Important:** From the "Additional Settings" tab, turn on **Enable draft/published system**. This lets you preview the content.

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

- **Title** - **Text** field (**Single-line String**)
- **Content** - **Text** field (**Multiple-paragraph Text**)
- **Excerpt** - **Text** field (**Single-line String**)
- **Cover Image** - **Media** field (**Single asset**)
- **Date** - **Date and time** field (**Date**)
- **Author** - **Links** field (**Single link**) , and from the "Validations" tab under "Accept only specified model", select **Author**.
- **Slug** - **SEO** field (**Slug**), and from the "Validations" tab under "Reference field" select **Title**.

Step 4. Populate Content

From the **Content** menu at the top, select **Author** and create a new record.

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

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

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

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

Step 5. Set up environment variables

Go to the **Settings** menu at the top and click **API tokens**.

Then click **Read-only API token** and copy the 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` :

- `DATOCMS_API_TOKEN` should be the API token you just copied.
- `DATOCMS_PREVIEW_SECRET` can be any random string (but avoid spaces), like `MY_SECRET` - this is used for [the Preview Mode](#).

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

```
DATOCMS_API_TOKEN=...  
DATOCMS_PREVIEW_SECRET=...
```

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 DatoCMS, 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 the draft state.

(If it doesn't become draft, you need to go to the model settings for `Post` , go to **Additional Settings**, and turn on **Enable draft/published system**.)

Now, if you go to the post page on localhost, you won't see the updated title. However, if you use the **Preview Mode**, you'll be able to see the change ([Documentation](#)).

To enable the Preview Mode, go to this URL:

```
http://localhost:3000/api/preview?secret=<secret>&slug=<slug>
```

- `<secret>` should be the string you entered for `DATOCMS_PREVIEW_SECRET` .
- `<slug>` should be the post's `slug` attribute (you can check on DatoCMS).

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

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.

