A statically generated blog example using Next.js and Ghost

This example showcases Next.js's Static Generation feature using Ghost as the data source.

> *This boilerplate demonstrates simple usage and best practices. If you are looking for a more feature rich Next.js generator for Ghost including the Casper theme, check out next-cms-ghost.*

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
- Prismic
- Contentful
- Strapi
- Agility CMS
- ButterCMS
- Storyblok
- GraphCMS
- Kontent
- 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-ghost cms-ghost-app
# or
yarn create next-app --example cms-ghost cms-ghost-app
# or
pnpm create next-app -- --example cms-ghost cms-ghost-app
```

Setp 1. Run Next.js in development mode

To get started, no configuration is needed as this example sources the content from a demo Ghost CMS.

```
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 2. Set up environment variables

If you already have a Ghost CMS running, you should create new Content API keys in the Ghost Admin panel under Integrations -> Add custom integration. Once your keys are generated, copy them into the environment variables as follows. Your `.env.local` file should look like this:

```
GHOST_API_URL=...
GHOST_API_KEY=...
```

Make sure to use the Content API Key.

Step 3. Set up Headless Ghost CMS

If you do not have access to a Ghost CMS, you need to create one for your own content. The demo Ghost CMS is running on Hetzner Cloud, which is described in Ghost CMS on Hetzner Cloud. Note that a Ghost install on localhost is not sufficient for public deploys, as the images on your local computer are not accessible from outside.

Step 4. 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.