A statically generated blog example using Next.js and Strapi

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

Demo

https://next-blog-strapi.vercel.app/

Deploy your own

Once you have access to the environment variables you'll need, deploy the example using Vercel:

▲ Deploy

Related examples

- WordPress
- DatoCMS
- Sanity
- TakeShape
- Prismic
- Contentful
- Agility CMS
- Cosmic
- ButterCMS
- Storyblok
- GraphCMS
- 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-strapi cms-strapi-app
# or
yarn create next-app --example cms-strapi cms-strapi-app
# or
pnpm create next-app -- --example cms-strapi cms-strapi-app
```

Configuration

Step 1. Set up Strapi locally

Use the provided Strapi template Next example to run a pre-configured Strapi project locally. See the Strapi template docs for more information

```
npx create-strapi-app@3 my-project --template next-example --quickstart
# or: yarn create strapi-app@3 my-project --template next-example --quickstart
npm run develop # or: yarn develop
```

This will open http://localhost:1337/ and prompt you to create an admin user.

After you sign in there should already be data for **Authors** and **Posts**. If you want to add more entries, just do the following:

Select **Author** and click **Add New Author**.

- Use dummy data for the name.
- For the image, you can download one from Unsplash.

Next, select **Posts** and click **Add New Post**.

- 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.
- Set the **status** field to be **published**.

Step 2. Set up environment variables

While the Strapi server is running, open a new terminal and `cd` into the Next.js app directory you created earlier.

```
cd cms-strapi-app
```

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`:

- `STRAPI_PREVIEW_SECRET` can be any random string (but avoid spaces), like `MY_SECRET` - this is used for Preview Mode.
- `NEXT_PUBLIC_STRAPI_API_URL` should be set as `http://localhost:1337` (no trailing slash).

Step 3. Run Next.js in development mode

Make sure that the local Strapi server is still running at http://localhost:1337. Inside the Next.js app directory, run:

```
npm install
npm run dev

# or

yarn install
yarn dev
```

Your blog should be up and running on http://localhost:3000!

The best place to debug is inside the `fetchAPI` function in `lib/api.js`. If you need help, you can post on GitHub discussions.

Step 4. Try preview mode

If you go to the `/posts/draft` page on localhost, you won't see this post because it's not published. 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=draft
```

- `<secret>` should be the string you entered for `STRAPI_PREVIEW_SECRET`.
- `<slug>` should be the post's `slug` attribute.

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

To add more preview pages, create a post and set the **status** as `draft`.

Step 5. Deploy Strapi

To deploy to production, you must first deploy your Strapi app. The Strapi app for our demo at https://next-blog-strapi.vercel.app/ is deployed to Heroku (here's the documentation) and uses Cloudinary for image hosting (see this file).

Step 6. 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.

▲ Deploy