

# A statically generated blog example using Next.js and Sanity

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

You'll get:

- Sanity Studio running on localhost
- Sub-second as-you-type previews in Next.js
- [On-demand revalidation of pages](#) with [GROQ powered webhooks](#)

## Demo

<https://next-blog-sanity.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](#)
- [TakeShape](#)
- [Prismic](#)
- [Contentful](#)
- [Strapi](#)
- [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-sanity cms-sanity-app
# or
yarn create next-app --example cms-sanity cms-sanity-app
# or
pnpm create next-app -- --example cms-sanity cms-sanity-app
```

## Configuration

### Step 1. Create an account and a project on Sanity

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

After creating an account, install the Sanity cli from npm `npm i -g @sanity/cli`.

### Step 2. Create a new Sanity project

In a separate folder run `sanity init` to initialize a new studio project.

This will be where we manage our data.

When going through the init phase make sure to select **Yes** to the **Use the default dataset configuration** step and select **Clean project with no predefined schemas** for the **Select project template** step.

### Step 3. Generate an API token

Log into <https://manage.sanity.io/> and choose the project you just created. Then from **Settings**, select **API**, then click **Add New Token** and create a token with the **Read** permission.

### Step 4. Set up environment variables

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`:

- `NEXT_PUBLIC_SANITY_PROJECT_ID` should be the `projectId` value from the `sanity.json` file created in step 2.
- `NEXT_PUBLIC_SANITY_DATASET` should be the `dataset` value from the `sanity.json` file created in step 2 - defaults to `production` if not set.
- `SANITY_API_TOKEN` should be the API token generated in the previous step.
- `SANITY_PREVIEW_SECRET` can be any random string (but avoid spaces), like `MY_SECRET` - this is used for [Preview Mode](#).
- `SANITY_STUDIO_REVALIDATE_SECRET` should be setup the same way as `SANITY_PREVIEW_SECRET` - this is used for [on-demand revalidation](#) with [webhooks](#).

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

```
NEXT_PUBLIC_SANITY_PROJECT_ID=...
NEXT_PUBLIC_SANITY_DATASET=...
SANITY_API_TOKEN=...
SANITY_PREVIEW_SECRET=...
SANITY_STUDIO_REVALIDATE_SECRET=...
```

### Step 5. Prepare the project for previewing

5.1. Install the `@sanity/production-preview` plugin with `sanity install @sanity/production-preview`.

5.2. Create a file called `resolveProductionUrl.js` (we'll get back to that file in a bit).

5.3. Open your studio's `sanity.json`, and add the following entry to the `parts`-array:

```
{
  "plugins": [
    "@sanity/production-preview"
  ],
  "parts": [
    //...
+   {
+     "implements": "part:@sanity/production-preview/resolve-production-url",
+     "path": "./resolveProductionUrl.js"
+   }
  ]
}
```

Now, go back to `resolveProductionUrl.js` and add a function that will receive the full document that was selected for previewing:

```
const previewSecret = 'MY_SECRET' // Copy the string you used for
SANITY_PREVIEW_SECRET
const projectUrl = 'http://localhost:3000'

export default function resolveProductionUrl(document) {
  return `${projectUrl}/api/preview?
secret=${previewSecret}&slug=${document.slug.current}`
}
```

For more information on live previewing check the [full guide](#).

## Step 6. Copy the schema file

After initializing your Sanity studio project there should be a `schemas` folder.

Replace the contents of `schema.js` in the Sanity studio project directory with [./schemas/schema.js](#) in this example directory. This will set up the schema we'll use this for this example.

## Step 7. Populate Content

To add some content go to your Sanity studio project directory and run `sanity start`.

After the project has started and you have navigated to the URL given in the terminal, 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 8. 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 9. Try preview mode

On Sanity, go to one of the posts you've created and:

- **Update the title.** For example, you can add `[Draft]` in front of the title.
- As you edit the document it will be saved as a draft, but **DO NOT** click **Publish**. By doing this, the post will be in the draft state.

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 view the preview, go to the post edit page on Sanity, click the three dots above the document and select **Open preview** ([see the instruction here](#))

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.

## Step 10. 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.



## Step 11. Setup Revalidation Webhook

- Open your Sanity manager, go to **API**, and **Create new webhook**.
- Set the **URL** to use the Vercel app url from [Step 10](#) and append `/api/revalidate` , for example:  
`https://cms-sanity.vercel.app/api/revalidate`
- Set the **Trigger on** field to ☒ Create ☒ Update ☒ Delete
- Set the **Filter** to `_type == "post" || _type == "author"`
- Set the **Secret** to the same value you gave `SANITY_STUDIO_REVALIDATE_SECRET` earlier.
- Hit **Save!**

### Testing the Webhook

- Open the Deployment function log. (**Vercel Dashboard** > **Deployment** > **Functions** and filter by `api/revalidate` )
- Edit a Post in your Sanity Studio and publish.
- The log should start showing calls.
- And the published changes show up on the site after you reload.

### Next steps

- Mount your preview inside the Sanity Studio for comfortable side-by-side editing
- [Join the Sanity community](#)