

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

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

## Demo

<https://next-blog-storyblok.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](#)
- [Prismic](#)
- [Contentful](#)
- [Agility CMS](#)
- [Cosmic](#)
- [Strapi](#)
- [ButterCMS](#)
- [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-storyblok cms-storyblok-app
# or
yarn create next-app --example cms-storyblok cms-storyblok-app
# or
pnpm create next-app -- --example cms-storyblok cms-storyblok-app
```

## Configuration

### Step 1. Create an account on Storyblok

[Create an account on Storyblok.](#)

When signing up, select **Create a new space**. Its name can be anything.

## Step 2. Create the `Authors` folder

From the dashboard, create a new folder called `Authors`.

- For **Default content type**, select **Add new**.
- Name of content type should be `author`.
- Choose **Blank** as the content type blueprint.

## Step 3. Create an `author` entry

In the `Authors` folder, create a new entry.

- **Name** can be anything, such as `Test Author`.

After creating the entry, click **Define schema**.

- Add a key called `picture`. Then click `picture` and set the **Type** as **Asset** and select **Images**.

Then upload an image to `picture`. You can use an [image from Unsplash](#).

Finally, after uploading, click **Publish**.

## Step 4. Create the `Posts` folder

After publishing the author, go back to the dashboard by clicking **Content** on the sidebar.

This time, create a new folder called `Posts`.

- For **Default content type**, select **Add new**.
- Name of content type should be `post`.
- Choose **Post** as the content type blueprint.

## Step 5. Create a `post` entry

In the `Posts` folder, create a new entry.

- **Name** can be anything.

Now, populate each field.

- **Title** can be any text.
- **Image** can be an [image from Unsplash](#).
- **Intro** can be any text.
- **Long Text** can be any text.
- **Author** should be the author created earlier.

Finally, click **Publish**.

You can create more posts under the `Posts` folder. Make sure to publish each one.

## Step 6. Set up environment variables

Go to the **Settings** menu for your space, and click **API-Keys**.

Then copy the **preview** token on the page.

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` :

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

## Step 7. 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, you can post on [GitHub discussions](#).

## Step 8. Try preview mode

To try preview mode, create another post like before (you can try duplicating), but **do not publish it - just save it**:

Now, if you go to the post 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=<slug>
```

- `<secret>` should be the string you entered for `STORYBLOK_PREVIEW_SECRET` .
- `<slug>` should be the post's `slug` (which can be seen on the **Config** section).

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.

## Step 9. 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.

