Sentry

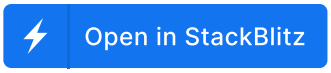
Note: Currently, the minimum Next.js supported version of `@sentry/nextjs` *is 10.0.8.*

This is an example showing how to use Sentry to catch and report errors and monitor the performance of both the front and back ends, using the official Sentry SDK for Next.js. This example contains the following:

- `sentry.server.config.js` and `sentry.client.config.js` are used to configure and initialize Sentry
- `next.config.js` automatically injects Sentry into your app using `withSentryConfig`
- `_error.js` (which is rendered by Next.js when handling certain types of exceptions) is overridden so those exceptions can be passed along to Sentry
- Each API route is handled with `withSentry`

Preview

Preview the example live on StackBlitz:



Deploy your own

It only takes a few steps to create and deploy your own version of this example app. Before you begin, make sure you have linked your Vercel account to GitHub, and set up a project in Sentry.

Option 1: Deploy directly to Vercel

You can deploy a copy of this project directly to Vercel.

This will clone this example to your GitHub org, create a linked project in Vercel, and prompt you to install the Vercel Sentry Integration. (You can read more about the integration on Vercel and in the Sentry docs.)

Option 2: Create locally before deploying

Alternatively, you can create a copy of this example app locally so you can configure and customize it before you deploy it.

Create and configure your app

To begin, execute `create-next-app` with npx or Yarn, to create the app and install dependencies:

```
npx create-next-app --example with-sentry nextjs-sentry-example
# or
yarn create next-app --example with-sentry nextjs-sentry-example
# or
pnpm create next-app -- --example with-sentry nextjs-sentry-example
```

Next, run `sentry-wizard`, which will create and populate the settings files needed by `@sentry/nextjs` to initialize the SDK and upload source maps to Sentry:

```
npx @sentry/wizard -i nextjs
```

Beware that above command creates files extended with `.wizardcopy.js` as the example already contains the config files.

Once the files are created, you can further configure your app by adding SDK settings to `sentry.server.config.js` and `sentry.client.config.js`, and `SentryWebpackPlugin` settings to `next.config.js`.

(If you'd rather do the SDK set-up manually, you can do that, too.)

You should now be able to build and run your app locally, upload source maps, and send errors and performance data to Sentry. For more details, check out the Sentry Next.js SDK docs.

Deploy your app to Vercel

Vercel reads you code from GitHub, so you first need to create an empty GitHub repo for your project and then add it to your local repo as a remote:

```
git remote add origin https://github.com/<org>/<repo>.git
```

Next, create a project in Vercel and link it to your GitHub repo.

In order for Vercel to upload source maps to Sentry when building your app, it needs a Sentry auth token. The wizard automatically sets up your personal token locally; to use that token on Vercel, add an environment variable to your Vercel project with the key `SENTRY_AUTH_TOKEN` and the value you'll find in `.sentryclirc` at the root level of your project. To use an org-wide token instead, set up the Vercel Sentry Integration. (You can read more about the integration on Vercel and in the Sentry docs.)

Finally, commit your app and push it to GitHub:

```
git add .
git commit -m "Initial commit"
git push
```

This will trigger a deployment in Vercel. Head over to your Vercel dashboard, click on your project, and then click "Visit" to see the results!