next-intl example

This example uses next-intl, a minimal, but complete solution for managing internationalization in Next.js apps.

Features

- 🌟 I18n is an essential part of the user experience, therefore this library doesn't compromise on flexibility and never leaves you behind when you need to fine tune a translation. Messages use the proven ICU syntax which covers interpolation, numbers, dates, times, plurals, ordinal pluralization, label selection based on enums and rich text.
- ⚔️ Based on battle-tested building blocks from Format.JS (used by `react-intl`), this library is a thin wrapper around high-quality, lower-level APIs for i18n.
- 💯 Built-in number and date formatting that is integrated with translations, e.g. allowing for the usage of global formats for a consistent look & feel of your app.
- 💡 A hooks-only API ensures that you can use the same API for `children` as well as for attributes which expect strings.
- 🚀 Integrates with both static as well as server side rendering.

Deploy your own

Deploy the example using Vercel or preview live with StackBlitz

▲ Deploy

How to use

Execute `create-next-app` with npm or Yarn to bootstrap the example:

```
npx create-next-app --example with-i18n-next-intl
# or
yarn create next-app --example with-i18n-next-intl
# or
pnpm create next-app -- --example with-i18n-next-intl
```

Deploy it to the cloud with Vercel (Documentation).