Using multiple zones

With Next.js you can use multiple apps as a single app using its multi-zones feature. This is an example showing how to use it.

- All pages should be unique across zones. For example, the `home` app should not have a `pages/blog/index.js` page.
- The `home` app is the main app and therefore it includes the rewrites that map to the `blog` app in next.config.js
- The `blog` app sets `basePath` to `/blog` so that generated pages, Next.js assets and public assets are within the `/blog` subfolder.

How to use

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

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

With multi zones you have multiple Next.js apps over a single app, therefore every app has its own dependencies and it runs independently.

To start the `/home` run the following commands from the root directory:

```
cd home
npm install && npm run dev
# or
cd home
yarn && yarn dev
```

The `/home` app should be up and running in http://localhost:3000!

Starting the `/blog` app follows a very similar process. In a new terminal, run the following commands from the root directory :

```
cd blog
npm install && npm run dev
# or
cd blog
yarn && yarn dev
```

The `blog` app should be up and running in http://localhost:4000!

Preview

Preview the example live on StackBlitz:

Open in StackBlitz

Deploy on Vercel

You can deploy this app to the cloud with Vercel (Documentation).

Deploy Your Local Project

To deploy the apps to Vercel, we'll use monorepos support to create a new project for each app.

To get started, push the example to GitHub/GitLab/Bitbucket and import your repo to Vercel. We're not interested in the root directory, so make sure to select the `blog` directory (do not start with `home`):

Import flow for blog app

Click continue and finish the import process. After that's done copy the domain URL that was assigned to your project, paste it on `home/.env`, and push the change to your repo:

```
# Replace this URL with the URL of your blog app
BLOG_URL="https://with-zones-blog.vercel.app"
```

Now we'll go over the import flow again using the same repo but this time select the `home` directory instead:

Import flow for home app

With the `home` app deployed you should now be able to see both apps running under the same domain!

Any future commits to the repo will trigger a deployment to the connected Vercel projects. See the blog post about monorepos to learn more.