# Gatsby Redux Lazy Loading

## Goal

The goal is to show a pattern of *not* just loading a slice of a Redux store when a component mounts, but *when the user interacts with the site.* In this case, there is an expensive reducer that:

- has a large 3rd party dependency
- imports a large JSON file and sets it in the store

This is very useful for parts of the store that are:

- not present on every page
- not present on a page without a certain user action

For less expensive slices of the store, this pattern may be overkill, but for expensive parts it is worth it.

## Why

The reason we have to be especially "careful" with Redux with Gatsby is that Gatsby's default code-splitting will be looking for code that is unique to a certain page and bundle it up, but if it sees code across more than one page, it will end up in the `app` or `commons` bundle and be loaded on every page, whether it is used or not. In this example the large 3rd party dependency and large array of dummy data would otherwise be loaded on every page, over 400kb! See this article by Ben Robertson on Gatsby code splitting.

## Pattern

The app works based on the implementation of the Redux Modules or Reducer Registry pattern found here by Nicholas Gallagher.

## Running Locally

```
cd examples/using-redux-w-interaction-code-splitting
npm install
npm run start
```

## Live demo:

gatsbyreduxcodesplittingexampl.gtsb.io