gatsby-transformer-react-docgen

Parses inline component-documentation using react-docgen.

Install

```
npm install gatsby-transformer-react-docgen
```

Usage

Add a plugin-entry to your `gatsby-config.js`

```
module.exports = {
  plugins: [`gatsby-transformer-react-docgen`],
}
```

For custom resolvers or handlers, all config options are passed directly to react-docgen. In addition any custom handlers are passed the component file `Node` object as their last argument for more Gatsby specific handler behavior.

```
module.exports = {
  plugins: [
    {
      resolve: "gatsby-transformer-react-docgen",
      options: {
        resolver: require("./custom-resolver"),
      },
    },
  ],
}
```

File parsing and babel configs

By default, your local `.babelrc` will be used to determine how to parse source files. Don't worry if you don't have a local babel config and are using Gatsby's default settings! If there isn't any config react-docgen will use it's own, permissive parsing options.

In the case of more complex sites with local custom configs, such as in a monorepo, you may have to tell babel (via react-docgen), how to properly resolve your local babel config. See the react-docgen documentation for more details.

```
module.exports = {
  plugins: [
    {
      resolve: "gatsby-transformer-react-docgen",
      options: {
        babelrcRoots: ["../packages/*"],
      },
    },
  ],
}
```

You'll also need to include a source-plugin, such as gatsby-source-filesystem, so that the transformer has access to source data.

> *Note: that at least one of your React Components must have PropTypes defined.*

How to query

An example *graphql* query to get nodes:

```
{
  allComponentMetadata {
    edges {
      node {
        displayName
        description
        props {
          name
          type
          required
        }
      }
    }
  }
}
```