ESLint is an open source JavaScript linting utility. Code linting is a type of static analysis that is frequently used to find problematic patterns. There are code linters for most programming languages, and compilers sometimes incorporate linting into the compilation process.

JavaScript, being a dynamic and loosely-typed language, is especially prone to developer error. Without the benefit of a compilation process, JavaScript code is typically executed in order to find syntax or other errors. Linting tools like ESLint allow developers to discover problems with their JavaScript code without executing it.

How to use ESLint

Gatsby ships with a built-in ESLint setup. For *most* users, our built-in ESLint setup is all you need. If you know however that you'd like to customize your ESLint config e.g. your company has their own custom ESLint setup, this shows how this can be done.

You'll replicate (mostly) the ESLint config Gatsby ships with so you can then add additional presets, plugins, and rules.

```
# First install the necessary ESLint dependencies
npm install --save-dev eslint-config-react-app
```

Now that your packages have been installed, create a new file at the root of the site named `.eslintrc.js` using the command below.

```
# Create a config file for ESLint
touch .eslintrc.js
```

Configuring ESLint

Copy the snippet below to the newly created `.eslintrc.js` file. Then add additional presets, plugins, and rules as desired.

```
module.exports = {
  globals: {
    __PATH_PREFIX__: true,
  },
  extends: `react-app`,
}
```

Note: When there is no ESLint file Gatsby implicitly adds a barebones ESLint loader. This loader pipes ESLint feedback into the terminal window where you are running or building Gatsby and also to the console in your browser developer tools. This gives you consolidated, immediate feedback on newly-saved files. When you include a custom `.eslintrc` file, Gatsby gives you control over the ESLint configuration. This means that it will override the built-in `eslint-loader` only leaving the required rules activated (`no-anonymous-exports-page-templates` & `limited-exports-page-templates`). For the non-required config you'll need to activate each rule on your own. One way to do this is to use the Community plugin `gatsby-plugin-eslint` . This also means that the default ESLint config Gatsby ships with (the `eslintConfig` export) will be entirely overwritten. If you would still like to take advantage of those rules, you'll need to copy them to your local file.

Disabling ESLint

Creating an empty `.eslintrc` file at the root of your project will mostly disable ESLint for your site. The empty file will disable the built-in `eslint-loader` because Gatsby assumes once you have an ESLint file you are in charge of linting. However, the required ESLint rules for Fast Refresh (`no-anonymous-exports-page-templates` & `limited-exports-page-templates`) will still be activated and shown in the terminal output.