Connor Elsea Follow
Aspiring Computer Science Student. LSU Engineering/Honors College 2020. Founder of Elsea Labs. Juni...
Aug 31, 2016 · 3 min read

Adding Sass or Scss to Create-React-App

Recently, the React team and a number of contributors released a fantastic tool for creating configuration-less React applications based on a set of minimal ideas to help beginners dive into building React applications without having to worry about tooling a beginner may find daunting.

There are two ways you can add Sass or Scss to a create-react-app project.

Install With: NPM Scripts

This method is simpler and can work without ejecting your project. If you prefer not to eject, use this method. Click here to view the project's official guide on how to install Sass or Scss using NPM Scripts.

Install With: Webpack

This method is not as simple, but more extensible and automated. Unfortunately, this method requires ejection. Ejecting a project, although useful in many advanced cases, means that futures updates to `create-react-app` will not be easy for you to get.

Before starting, install and save the necessary tools from your packager of choice by running one of the following commands in your project directory.

Yarn: `yarn add sass-loader node-sass --dev`

NPM: `npm install sass-loader node-sass --save-dev`

Adding sass or scss to your create-react-app project will first require you to eject, which can be done by executing the following command in your project's base directory.

```
npm run eject
```

Once ejection has completed, locate the config folder which holds, among other things, two webpack configuration files. Each of these configuration files corresponds to a different environment—development or production. Enter the development configuration file `webpack.config.dev.js` and locate the `loaders` block within the `modules` block.

In webpack, loaders allow the developer to "pre-process" files as they are required/loaded. Create react app uses multiple loaders to handle various build tasks such as transpiling with babel, prefixing with PostCSS, or allowing the importing of assets. To add Sass or Scss to this project, you will be adding a loader that can process Sass and Scss files.

The beginning of the `loaders` block, before any modification, should look something like the following code (Exact code may change in future versions of CRA).

```
loaders: [
  // Process JS with Babel.
  {
    test: /\.js$/,
    include: paths.appSrc,
    loader: 'babel',
    query: require('./babel.dev')
  },
  ...
```

After the beginning of the loaders array, you can add your own loader for Sass and Scss. The "sass" loader featured in the loaders array in the code below is able to process both Sass and Scss files.

If you prefer **Sass**, you can add:

```
{
  test: /\.sass$/,
  include: paths.appSrc,
  loaders: ["style", "css", "sass"]
},
```

If you prefer **Scss**, you can add:

```
{
  test: /\.scss$/,
  include: paths.appSrc,
  loaders: ["style", "css", "sass"]
},
```

Important: In newer versions of create-react-app, if you see an `exclude` array in your webpack config, you need to add the following lines to

```
/\.sass$/,
/\.scss$/,
```

Now that the Sass and Scss loader is in place, you can begin using Sass/Scss files. For example, in the default *App.js* component that comes with create-react-app, you could now write

```
import './App.sass'; // or `.scss` if you chose scss
```

Note, you'd also have to refactor the "App.css" file to Sass and change it's file name to "App.sass".

. . .

You can read more about the sass loader here.

You can read more about webpack loaders here.

You can read more about create-react-app here.

Looking for the Github changes? You can view the file diff on Github here.