

# create-react-app当中使用sass

在cra里面是可以使用sass的，并且默认情况下，脚手架已经帮我们配置好了 sass 的webpack

## 安装node-sass

在cra的脚手架里面，虽然内置的webpack已经帮我们进行了配置，但是它缺少运行环境 node-sass，所以我们需要主动的去安装这个包才行

```
$ yarn add node-sass -D
```

安装完成以后就可以直接使用 sass 了

## 使用全局sass文件

在开发的时候，我们有时候也需要使用全局的sass文件，如在 @vue/cli 里面，我们是新建了一个 vue.config.js 里面进行配置，在 react 的脚手架 cra 里面，如果要使用全局sass文件也是一样的要进行配置

vue是通过 vue.config.js 这个文件来实现自定义配置的，而 react 的脚手架也可以，它会通过一个叫 config-overrides.js 这个文件来进行配置

使用全局 sass 文件需要使用于第三方的工具包来完成

1. sass-resources-loader，这个包主要用于加载 sass 的全局资源文件
2. react-app-rewired，这是一个很常见的包，它可以让 react 从第三方入口开始启动【后面会让它从 config-overrides.js 这个文件开始启动】
3. customize-cra 这是脚手架自定义配置所需要的包，它要与 config-overrides.js 这个文件结合使用

**第一步：**现在我们就安装上面的三个包

```
$ yarn add sass-resources-loader react-app-rewired customize-cra -D
```

**第二步：**在项目的根目录创建 config-overrides.js 文件，并输入如下代码

```
const { override, adjustStyleLoaders } = require("customize-cra");
module.exports = override(
  adjustStyleLoaders(rule => {
    if (rule.test.toString().includes("scss")) {
      rule.use.push({
        loader: require.resolve("sass-resources-loader"),
        options: {
          resources: "./src/assets/scss/comm.scss" //这里是你自己放公共
          scss变量的路径
        }
      });
    }
  })
);
```

**第三步：**更改 `package.json` 里面 `scripts` 部分的启动命令

#### 更改前

```
"scripts": {  
  "start": "react-scripts start",  
  "build": "react-scripts build",  
  "test": "react-scripts test",  
  "eject": "react-scripts eject"  
},
```

#### 更改后

```
"scripts": {  
  "start": "react-app-rewired start",  
  "build": "react-app-rewired build",  
  "test": "react-app-rewired test",  
  "eject": "react-app-rewired eject"  
}
```

当所有的配置都完成了以后，我们就可以使用全局的sass文件了