在 create-react-app 中使用

[create-react-app](https://github.com/facebookincubator/create-react-app) 是业界最优秀的 React 应用开发工具之一，本文会尝试在 create-react-app 创建的工程中使用 antd 组件，并自定义 webpack 的配置以满足各类工程化需求。

* [安装和初始化](https://ant.design/docs/react/use-with-create-react-app-cn#%E5%AE%89%E8%A3%85%E5%92%8C%E5%88%9D%E5%A7%8B%E5%8C%96)
* [引入 antd](https://ant.design/docs/react/use-with-create-react-app-cn#%E5%BC%95%E5%85%A5-antd)
* [高级配置](https://ant.design/docs/react/use-with-create-react-app-cn#%E9%AB%98%E7%BA%A7%E9%85%8D%E7%BD%AE)
* [eject](https://ant.design/docs/react/use-with-create-react-app-cn#eject)
* [源码和其他脚手架](https://ant.design/docs/react/use-with-create-react-app-cn#%E6%BA%90%E7%A0%81%E5%92%8C%E5%85%B6%E4%BB%96%E8%84%9A%E6%89%8B%E6%9E%B6)

安装和初始化[#](https://ant.design/docs/react/use-with-create-react-app-cn#%E5%AE%89%E8%A3%85%E5%92%8C%E5%88%9D%E5%A7%8B%E5%8C%96)

在开始之前，你可能需要安装 [yarn](https://github.com/yarnpkg/yarn/)。

$ yarn create react-app antd-demo

工具会自动初始化一个脚手架并安装 React 项目的各种必要依赖，如果在过程中出现网络问题，请尝试配置代理或使用其他 npm registry。

然后我们进入项目并启动。

$ cd antd-demo

$ yarn start

此时浏览器会访问 <http://localhost:3000/> ，看到 Welcome to React 的界面就算成功了。

引入 antd[#](https://ant.design/docs/react/use-with-create-react-app-cn#%E5%BC%95%E5%85%A5-antd)

这是 create-react-app 生成的默认目录结构。

├── README.md

├── package.json

├── public

│ ├── favicon.ico

│ └── index.html

├── src

│ ├── App.css

│ ├── App.js

│ ├── App.test.js

│ ├── index.css

│ ├── index.js

│ └── logo.svg

└── yarn.lock

现在从 yarn 或 npm 安装并引入 antd。

$ yarn add antd

修改 src/App.js，引入 antd 的按钮组件。

import React, { Component } from 'react';

import Button from 'antd/es/button';

import './App.css';

class App extends Component {

render() {

return (

<div className="App">

<Button type="primary">Button</Button>

</div>

);

}

}

export default App;

修改 src/App.css，在文件顶部引入 antd/dist/antd.css。

@import '~antd/dist/antd.css';

.App {

text-align: center;

}

...

好了，现在你应该能看到页面上已经有了 antd 的蓝色按钮组件，接下来就可以继续选用其他组件开发应用了。其他开发流程你可以参考 create-react-app 的[官方文档](https://github.com/facebookincubator/create-react-app/blob/master/packages/react-scripts/template/README.md)。

高级配置[#](https://ant.design/docs/react/use-with-create-react-app-cn#%E9%AB%98%E7%BA%A7%E9%85%8D%E7%BD%AE)

我们现在已经把组件成功运行起来了，但是在实际开发过程中还有很多问题，例如上面的例子实际上加载了全部的 antd 组件的样式（gzipped 后一共大约 60kb）。

此时我们需要对 create-react-app 的默认配置进行自定义，这里我们使用 [react-app-rewired](https://github.com/timarney/react-app-rewired) （一个对 create-react-app 进行自定义配置的社区解决方案）。

引入 react-app-rewired 并修改 package.json 里的启动配置。由于新的 [react-app-rewired@2.x](https://github.com/timarney/react-app-rewired#alternatives) 版本的关系，你还需要安装 [customize-cra](https://github.com/arackaf/customize-cra)。

$ yarn add react-app-rewired customize-cra

/\* package.json \*/

"scripts": {

- "start": "react-scripts start",

+ "start": "react-app-rewired start",

- "build": "react-scripts build",

+ "build": "react-app-rewired build",

- "test": "react-scripts test",

+ "test": "react-app-rewired test",

}

然后在项目根目录创建一个 config-overrides.js 用于修改默认配置。

module.exports = function override(config, env) {

// do stuff with the webpack config...

return config;

};

使用 babel-plugin-import[#](https://ant.design/docs/react/use-with-create-react-app-cn#%E4%BD%BF%E7%94%A8-babel-plugin-import)

注意：antd 默认支持基于 ES module 的 tree shaking，js 代码部分不使用这个插件也会有按需加载的效果。

[babel-plugin-import](https://github.com/ant-design/babel-plugin-import) 是一个用于按需加载组件代码和样式的 babel 插件（[原理](https://ant.design/docs/react/getting-started-cn#%E6%8C%89%E9%9C%80%E5%8A%A0%E8%BD%BD)），现在我们尝试安装它并修改 config-overrides.js文件。

$ yarn add babel-plugin-import

+ const { override, fixBabelImports } = require('customize-cra');

- module.exports = function override(config, env) {

- // do stuff with the webpack config...

- return config;

- };

+ module.exports = override(

+ fixBabelImports('import', {

+ libraryName: 'antd',

+ libraryDirectory: 'es',

+ style: 'css',

+ }),

+ );

然后移除前面在 src/App.css 里全量添加的 @import '~antd/dist/antd.css'; 样式代码，并且按下面的格式引入模块。

// src/App.js

import React, { Component } from 'react';

- import Button from 'antd/es/button';

+ import { Button } from 'antd';

import './App.css';

class App extends Component {

render() {

return (

<div className="App">

<Button type="primary">Button</Button>

</div>

);

}

}

export default App;

最后重启 yarn start 访问页面，antd 组件的 js 和 css 代码都会按需加载，你在控制台也不会看到这样的[警告信息](https://zos.alipayobjects.com/rmsportal/vgcHJRVZFmPjAawwVoXK.png)。关于按需加载的原理和其他方式可以阅读[这里](https://ant.design/docs/react/getting-started-cn#%E6%8C%89%E9%9C%80%E5%8A%A0%E8%BD%BD)。

自定义主题[#](https://ant.design/docs/react/use-with-create-react-app-cn#%E8%87%AA%E5%AE%9A%E4%B9%89%E4%B8%BB%E9%A2%98)

按照 [配置主题](https://ant.design/docs/react/customize-theme-cn) 的要求，自定义主题需要用到 less 变量覆盖功能。我们可以引入 customize-cra 中提供的 less 相关的函数 [addLessLoader](https://github.com/arackaf/customize-cra#addlessloaderloaderoptions) 来帮助加载 less 样式，同时修改 config-overrides.js 文件如下。

$ yarn add less less-loader

- const { override, fixBabelImports } = require('customize-cra');

+ const { override, fixBabelImports, addLessLoader } = require('customize-cra');

module.exports = override(

fixBabelImports('import', {

libraryName: 'antd',

libraryDirectory: 'es',

- style: 'css',

+ style: true,

}),

+ addLessLoader({

+ javascriptEnabled: true,

+ modifyVars: { '@primary-color': '#1DA57A' },

+ }),

);

这里利用了 [less-loader](https://github.com/webpack/less-loader#less-options) 的 modifyVars 来进行主题配置，变量和其他配置方式可以参考 [配置主题](https://ant.design/docs/react/customize-theme-cn) 文档。

修改后重启 yarn start，如果看到一个绿色的按钮就说明配置成功了。

你也可以使用 [craco](https://github.com/sharegate/craco) 和 [craco-antd](https://github.com/FormAPI/craco-antd) 来实现和 customize-cra 一样的修改 create-react-app 配置的功能。

eject[#](https://ant.design/docs/react/use-with-create-react-app-cn#eject)

你也可以使用 create-react-app 提供的 [yarn run eject](https://facebook.github.io/create-react-app/docs/available-scripts#npm-run-eject) 命令将所有内建的配置暴露出来。不过这种配置方式需要你自行探索，不在本文讨论范围内。