

[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

在开始之前，你可能需要安装 [yarn](#)。

```
$ yarn create react-app antd-demo

# or

$ npx 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

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 from 'react';
import { Button } from 'antd';
```

```
import './App.css';

const App = () => (
  <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';
```

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](#)。

我们现在已经把 antd 组件成功运行起来了，开始开发你的应用吧！

高级配置

这个例子在实际开发中还有一些优化的空间，比如无法进行主题配置。

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 [craco](#)（一个对 create-react-app 进行自定义配置的社区解决方案）。

现在我们安装 craco 并修改 `package.json` 里的 `scripts` 属性。

```
$ yarn add @craco/craco
```

```
/* package.json */
"scripts": {
  - "start": "react-scripts start",
  - "build": "react-scripts build",
  - "test": "react-scripts test",
  + "start": "craco start",
  + "build": "craco build",
  + "test": "craco test",
}
```

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

```
/* craco.config.js */
module.exports = {
  // ...
};
```

自定义主题

按照 [配置主题](#) 的要求，自定义主题需要用到类似 [less-loader](#) 提供的 less 变量覆盖功能。我们可以引入 [craco-less](#) 来帮助加载 less 样式和修改变量。

首先把 `src/App.css` 文件修改为 `src/App.less`，然后修改样式引用为 less 文件。

```
/* src/App.js */  
- import './App.css';  
+ import './App.less';
```

```
/* src/App.less */  
- @import '~antd/dist/antd.css';  
+ @import '~antd/dist/antd.less';
```

然后安装 `craco-less` 并修改 `craco.config.js` 文件如下。

```
$ yarn add craco-less
```

```
const CracoLessPlugin = require('craco-less');  
  
module.exports = {  
  plugins: [  
    {  
      plugin: CracoLessPlugin,  
      options: {  
        lessLoaderOptions: {  
          lessOptions: {  
            modifyVars: { '@primary-color': '#1DA57A' },  
            javascriptEnabled: true,  
          },  
        },  
      },  
    ],  
  ],  
};
```

这里利用了 [less-loader](#) 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](#) 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。

antd 内建了深色主题和紧凑主题，你可以参照 [使用暗色主题和紧凑主题](#) 进行接入。

同样，你可以使用 [react-app-rewired](#) 和 [customize-cra](#) 来自定义 create-react-app 的 webpack 配置。

eject

你也可以使用 create-react-app 提供的 `yarn 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。

小结

以上是在 create-react-app 中使用 antd 的相关实践，你也可以借鉴此文的做法在自己的 webpack 工作流中使用 antd。

上述教程的脚手架源码我们放在 [create-react-app-antd](#) 中，你可以直接下载使用。

接下来我们会介绍如何在 [TypeScript](#) 和 [Umi](#) 中使用 antd，欢迎继续阅读。