

[create-react-app](#) is one of the best React application development tools. We are going to use `antd` within it and modify the webpack config for some customized needs.

---

## Install and Initialization

Before all start, you may need install [yarn](#).

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

# or

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

The tool will create and initialize environment and dependencies automatically, please try config your proxy setting or use another npm registry if any network errors happen during it.

Then we go inside `antd-demo` and start it.

```
$ cd antd-demo
$ yarn start
```

Open the browser at <http://localhost:3000/>. It renders a header saying "Welcome to React" on the page.

## Import antd

Below is the default directory structure.

```
├─ 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
```

Now we install `antd` from yarn or npm.

```
$ yarn add antd
```

Modify `src/App.js` , import Button component from `antd` .

```
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;
```

Add `antd/dist/antd.css` at the top of `src/App.css`.

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

OK, you should now see a blue primary button displayed on the page. Next you can choose any components of `antd` to develop your application. Visit other workflows of `create-react-app` at its [User Guide](#).

We are successfully running antd components now, go build your own application!

## Test with Jest

`create-react-app` comes with `jest` built in. Jest does not support `esm` modules, and Ant Design uses them. In order to test your Ant Design application with Jest you have to add the following to your `package.json`:

```
"jest": {
  "transformIgnorePatterns": [
    "/node_modules/(?!antd|@ant-design|rc-.+?|@babel/runtime).+(js|jsx)$"
  ]
}
```

## Advanced Guides

In the real world, we usually have to modify default webpack config for custom needs such as themes. We can achieve that by using [craco](#) which is one of `create-react-app`'s custom config solutions.

Install craco and modify the `scripts` field in `package.json`.

```
$ 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",
}
```

Then create a `craco.config.js` at root directory of your project for further overriding.

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

## Customize Theme

According to the [Customize Theme documentation](#), we need to modify less variables via loader like [less-loader](#). We can use [craco-less](#) to achieve that,

First we should modify `src/App.css` to `src/App.less`, then import less file instead.

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

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

Then install `craco-less` and modify `craco.config.js` like below.

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

```
const CracoLessPlugin = require('craco-less');

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

By adding `modifyVars` option of [less-loader](#) here, we should see a green button rendered on the page after rebooting the server now.

We provide built-in dark theme and compact theme in antd, you can reference to [Use dark or compact theme](#).

You could also try [react-app-rewired](#) and [customize-cra](#) to customize create-react-app webpack config like craco did.

## eject

You can also eject your application using [yarn run eject](#) for a custom setup of create-react-app, although you should dig into it by yourself.

## Summary

Finally, we used antd with create-react-app successfully, the source code of this guide was pushed to [create-react-app-antd](#) which you could clone and use it directly.

Next part, We will introduce how to use antd in [TypeScript](#) and [Umi](#), let's keep moving!