

# AntDesign组件库

王红元 coderwhy



实力IT教育

■ vue中添加class是一件非常简单的事情：

■ 你可以通过传入一个对象：

```
<div class="static"
  v-bind:class="{ active: isActive, 'text-danger': hasError }">
</div>
```

■ 你也可以传入一个数组：

```
<div v-bind:class="[activeClass, errorClass]"></div>
```

■ 甚至是对象和数组混合使用：

```
<div v-bind:class="[{ active: isActive }, errorClass]"></div>
```

- React在JSX给了我们开发者足够多的灵活性，你可以像编写JavaScript代码一样，通过一些逻辑来决定是否添加某些class：

```
<div>
  <h2 className={"title" + (isActive ? "active": "")}>我是标题</h2>
  <h2 className={["title", (isActive ? "active": "")].join(" ")}>我是标题</h2>
</div>
```

- 这个时候我们可以借助于一个第三方的库：classnames
  - 很明显，这是一个用于动态添加classnames的一个库。

```
classnames('foo', 'bar'); // => 'foo bar'
classnames('foo', { bar: true }); // => 'foo bar'
classnames({ 'foo-bar': true }); // => 'foo-bar'
classnames({ 'foo-bar': false }); // => ''
classnames({ foo: true }, { bar: true }); // => 'foo bar'
classnames({ foo: true, bar: true }); // => 'foo bar'
classnames('foo', { bar: true, duck: false }, 'baz', { quux: true }); // => 'foo bar baz quux'
classnames(null, false, 'bar', undefined, 0, 1, { baz: null }, ''); // => 'bar 1'
```

# AntDesign的介绍

■ AntDesign，简称 antd 是基于 Ant Design 设计体系的 React UI 组件库，主要用于研发企业级中后台产品。

□ 中后台的产品 属于**工具性产品**，很多优秀的设计团队通过自身的探索和积累，形成了自己的设计体系。

■ AntDesign的特点：

□ 🌈 提炼自企业级中后台产品的交互语言和视觉风格。

□ 📦 开箱即用的高质量 React 组件。

□ 🛡️ 使用 TypeScript 开发，提供完整的类型定义文件。

□ ⚙️ 全链路开发和设计工具体系。

□ 🌐 数十个国际化语言支持。

□ 🎨 深入每个细节的主题定制能力。

■ 全链路开发和设计指的是什么？







□ 全链路这个词我记得是16年左右阿里提出的；

□ 从**业务战略—用户场景—设计目标—交互体验—用户流程—预期效率**全方面进行分析和考虑；

□ 这个主要是产品经理会考虑的一个点；

## ■ AntDesign的兼容性：

- 现代浏览器和 IE11 ( 需要 [polyfills](#) ) 。
- 支持服务端渲染。
- Electron

 IE / Edge	 Firefox	 Chrome	 Safari	 Opera	 Electron
IE11, Edge	last 2 versions	last 2 versions	last 2 versions	last 2 versions	last 2 versions

■ antd@2.0 之后不再支持 IE8 , antd@4.0 之后不再支持 IE9/10。

■ 目前稳定的版本：v4.4.0

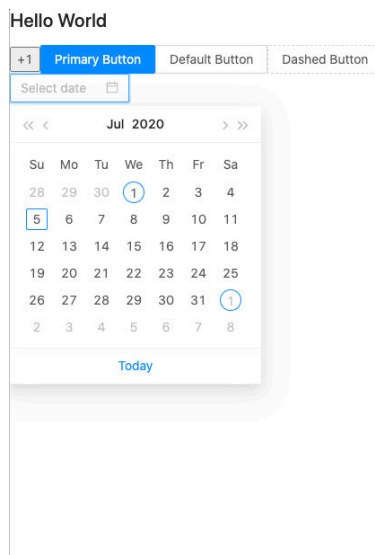
## ■ 使用 npm 或 yarn 安装

`npm install antd -save`  
或  
`yarn add antd`

## ■ 我们需要在index.js中引入全局的Antd样式：

```
import "antd/dist/antd.css";
```

## ■ 在App.js中就可以使用一些组件了：



```
<Button type="primary">Primary Button</Button>
<Button>Default Button</Button>
<Button type="dashed">Dashed Button</Button>
<br />
<DatePicker onChange={this.onChange}
  allowClear={false}
  dateRender={current => {
    const style = {};
    if (current.date() === 1) {
      style.border = '1px solid #1890ff';
      style.borderRadius = '50%';
    }
    return (
      <div className="ant-picker-cell-inner" style={style}>
        {current.date()}
      </div>
    );
  }} />
```

## ■ 考虑一个问题：Antd是否会将一些没有用的代码（组件或者逻辑代码）引入，造成包很大呢？

- antd 官网有提到：antd 的 JS 代码默认支持基于 ES modules 的 tree shaking，对于 js 部分，直接引入 `import { Button } from 'antd'` 就会有按需加载的效果。

- 上面的使用过程是无法对主题进行配置的，好像对主题等相关的高级特性进行配置，需要修改create-react-app 的默认配置。
- 如何修改create-react-app 的默认配置呢？
  - 前面我们讲过，可以通过yarn run eject来暴露出来对应的配置信息进行修改；
  - 但是对于webpack并不熟悉的人来说，直接修改 CRA 的配置是否会给你的项目带来负担，甚至会增加项目的隐患和不稳定性呢？
  - 所以，在项目开发中是不建议大家直接去修改 CRA 的配置信息的；
- 那么如何进行修改默认配置呢？社区目前有两个比较常见的方案：
  - react-app-rewired + customize-cra；（这个是antd早期推荐的方案）
  - craco；（目前antd推荐的方案）

## ■ 第一步：安装craco：

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

## ■ 第二步：修改package.json文件

□ 原本启动时，我们是通过react-scripts来管理的；

□ 现在启动时，我们通过craco来管理；

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

## ■ 第三步：在根目录下创建craco.config.js文件用于修改默认配置

```
module.exports = {  
  // 配置文件  
}
```



■ 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；

■ 安装 craco-less：

```
yarn add craco-less
```

■ 修改craco.config.js中的plugins：

□ 使用modifyVars可以在运行时修改LESS变量；

■ 引入antd的样式时，引入antd.less文件：

```
import 'antd/dist/antd.less';
```

■ 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。

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

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

- 在项目开发中，某些组件或者文件的层级会较深，
  - 如果我们通过上层目录去引入就会出现这样的情况：`../../../../components/button`；
  - 如果我们可以配置别名，就可以直接从根目录下面开始查找文件：`@/components/button`，甚至是`:components/button`；
- 配置别名也需要修改webpack的配置，当然我们也可以借助于 `craco` 来完成：

```
...
const path = require("path");
const resolve = dir => path.resolve(__dirname, dir);

module.exports = {
  ...,
  webpack: {
    alias: {
      '@': resolve("src"),
      'components': resolve("src/components"),
    }
  }
}
```

- 在导入时就可以按照下面的方式使用了：

```
import HYCommentInput from '@/components/comment-input';
import HYCommentItem from 'components/comment-item';
```

## ■ 我们通过AntDesign来编写一个案例：



coderwhy 2 minutes ago

你好啊，李银河

🗑 删除



coderwhy a few seconds ago

当我跨过沉沦的一切，向着永恒开战的时候，你是我的军旗。

🗑 删除



coderwhy a few seconds ago

你生了气就哭，我一看见你哭就目瞪口呆，就像一个小孩子做了坏事在未受责备之前目瞪口呆一样，所以什么事你先别哭，先来责备我，好吗？

🗑 删除

添加评论