## 快速上手

### 安装

推荐使用 npm 的方式安装，它能更好地和webpack打包工具配合使用。

npm i element-react --save

### 主题

开始前, 你还需要一个主题包, 这里我们推荐使用element-theme-default.

npm install element-theme-default --save

### 使用

import React from 'react';import ReactDOM from 'react-dom';import { Button } from 'element-react';

import 'element-theme-default';

ReactDOM.render(<Button type="primary">Hello</Button>, document.getElementById('app'));

## 国际化

Element 组件内部默认使用中文，若希望使用其他语言，则需要进行多语言设置。以英文为例，在 main.js 中：

import { i18n } from 'element-react'import locale from 'element-react/src/locale/lang/en'

i18n.use(locale);

如果使用其它语言，默认情况下中文语言包依旧是被引入的，可以使用 webpack 的 NormalModuleReplacementPlugin 替换默认语言包。

**webpack.config.js**

{

plugins: [

new webpack.NormalModuleReplacementPlugin(/element-react[\/\\]src[\/\\]locale[\/\\]lang[\/\\]zh-CN/, 'element-react/src/locale/lang/en')

]}

目前 Element 内置了以下语言：

* 简体中文（zh-CN）
* 英语（en）

## 自定义主题

Element 默认提供一套主题，CSS 命名采用 BEM 的风格方便使用者覆盖样式。如果你想完全替换主题色或者部分样式，可以使用下面方法。

### 安装工具

首先安装「主题生成工具」，可以全局安装或者安装在当前项目下，推荐安装在项目里，方便别人 clone 项目时能直接安装依赖并启动，这里以全局安装做演示。

npm i element-theme -g

安装默认主题，可以从 npm 安装或者从 GitHub 拉取最新代码。

# 从 npm

npm i element-theme-default -D

# 从 GitHub

npm i https://github.com/ElementUI/theme-default -D

### 初始化变量文件

主题生成工具安装成功后，如果全局安装可以在命令行里通过 et 调用工具，如果安装在当前目录下，需要通过node\_modules/.bin/et 访问到命令。执行 -i 初始化变量文件。默认输出到 element-variables.css，当然你可以传参数指定文件输出目录。

et -i [可以自定义变量文件]

> ✔ Generator variables file

如果使用默认配置，执行后当前目录会有一个 element-variables.css 文件。内部包含了主题所用到的所有变量，它们使用 CSS4 的风格定义。大致结构如下：

:root {

/\* Colors

-------------------------- \*/

--color-primary: #20a0ff;

--color-success: #13ce66;

--color-warning: #f7ba2a;

--color-danger: #ff4949;

--color-info: #50BFFF;

--color-blue: #2e90fe;

--color-blue-light: #5da9ff;

--color-blue-lighter: rgba(var(--color-blue), 0.12);

--color-white: #fff;

--color-black: #000;

--color-grey: #C0CCDA;

### 修改变量

直接编辑 element-variables.css 文件，例如修改主题色为红色。

--color-primary: red;

### 编译主题

保存文件后，到命令行里执行 et 编译主题，如果你想启用 watch 模式，实时编译主题，增加 -w 参数；如果你在初始化时指定了自定义变量文件，则需要增加 -c 参数，并带上你的变量文件名

et

> ✔ build theme font

> ✔ build element theme

### 引入自定义主题

默认情况下编译的主题目录是放在 ./theme 下，你可以通过 -o 参数指定打包目录。像引入默认主题一样，在代码里直接引用theme/index.css 文件即可。

import '../theme/index.css'import 'element-ui'

### 搭配插件按需引入组件主题

如果是搭配 babel-plugin-component 一起使用，只需要修改 .babelrc 的配置，指定 styleLibraryName 路径为自定义主题相对于.babelrc 的路径，注意要加 ~。

{

"plugins": [["component", [

{

"libraryName": "element-ui",

"styleLibraryName": "~theme"

}

]]]

}