

---

# Table of Contents

Introduction	1.1
开始	1.2
前提	1.2.1
安装	1.2.2
使用	1.2.3
服务端渲染	1.2.4
例子	1.2.5
自定义	1.3
主题	1.3.1
样式	1.3.2
颜色	1.3.3
组件	1.4
导航栏	1.4.1
自动填充	1.4.2
头像	1.4.3
徽章	1.4.4
按钮	1.4.5
扁平按钮	1.4.5.1
凸起按钮	1.4.5.2
悬浮按钮	1.4.5.3
图标按钮	1.4.5.4
卡片	1.4.6
日期选择器	1.4.7
提示框	1.4.8
分割器	1.4.9
抽屉	1.4.10
网格	1.4.11
图标	1.4.12
Font 图标	1.4.12.1
SVG 图标	1.4.12.2

列表	1.4.13
菜单	1.4.14
普通菜单	1.4.14.1
图标菜单	1.4.14.2
下拉菜单	1.4.14.3
纸片	1.4.15
列表	1.4.16
进度条	1.4.17
环形进度条	1.4.17.1
线形进度条	1.4.17.2
加载进度条	1.4.17.3
弹出框	1.4.18
选择框	1.4.19
滑块	1.4.20
切换控件	1.4.21
多选按钮	1.4.21.1
单选按钮	1.4.21.2
开关控件	1.4.21.3
消息提示窗	1.4.22
步进	1.4.23
副标题	1.4.24
表格	1.4.25
选项卡	1.4.26
输入框	1.4.27
时间选择器	1.4.28
工具条	1.4.29
更多介绍	1.5
联系我们	1.5.1
加入我们	1.5.2
案例展示	1.5.3
相关项目	1.5.4
资源	1.6
<a href="#">GitHub</a>	1.6.1
<a href="#">React</a>	1.6.2



# Material-UI(中文)

Material-UI是基于Material Design设计的一套丰富的React组件。这是一个实现的非常漂亮的类库包括CSS样式和动画。有两套独立的风格可以选择-黑色和亮色。

这是关于Material-UI的中文翻译版本，供学习使用。



## 前提

学习Material-UI，你需要先去了解[React](#),Material-UI是一套React组件，所以了解React进行web开发是非常重要的。

如果你已经熟悉单页面应用(SPAs)和Node,你可以直接跳过前提和安装章节。

否则，下面我们将快速的介绍什么是SPAs和Node。这是非常有用的，特别是你刚好有有限的WEB开发经验，或者你的经验仅仅只是使用HTML、CSS、JS做过传统的多页面网站。

## 单页面应用

很久以前，网站仅仅使用HTML、CSS做静态的页面，JavaScript支持少量的用户交互和动画效果。对于大多数的用户交互，特别是数据的处理，完整的从客户端发送到服务器，然后服务器响应到客户端，这种方式一般是阻塞的，用户在数据发送的过程中，是处于等待状态，用户体验非常差。

后面随着Ajax异步处理的出现，客户端在发送数据的同时，用户可以做其它事情而不受影响。然而，大多数客户的交互仍然需要服务器的响应，而网站依然没有桌面程序响应那么快和流畅，所以SPAs就可以解决这个问题，它可以把网站像桌面程序的速度一样流畅，提升用户的体验度。

SPAs是一个站点，但里面只包含了一个页面，也就是所有的

- [Angular](#)
- [Ember](#)
- [Backbone](#)
- [React](#)

所有的代码包含在一个页面中，这对于项目的架构是一个大的挑战。幸运的是，我们可以使用工具把我们的代码分解成小的模块组织在一起（类似于面向对象编程里面的类和接口），这就Node的用武之地。

## 安装

Material-UI使用[npm](#)进行安装。

### react-tap-event-plugin

我们使用[react-tap-event-plugin](#)去监听touch / tap / click 事件，使之得到快速的响应，这个插件是临时的，一旦React的官方发布相应的插件，我们将马上去掉。但是，在那之前，你一定要在你的应用启动时注入这个插件。

```
import injectTapEventPlugin from 'react-tap-event-plugin';

// Needed for onTouchTap
// Check this repo:
// https://github.com/zilverline/react-tap-event-plugin
injectTapEventPlugin();
```

**react-tap-event-plugin** 对React所有的组件都提供了一个 **onTouchTap()** 方法，去处理手机端的点击事件，快速响应我们的应用程序。这对于按钮非常有用。

### Roboto 字体

Material-UI采用谷歌的[Roboto](#)。所以你需要包含Roboto在你的项目中。这里提供了一些[说明](#)，希望对你有用。

## 使用

Material-UI组件非常容易使用。最快的开始方式是使用 `MuiThemeProvider` 来插入到你的程序的主题包里。接着你可以使用任何在我们文档中的组件。

这里是一个非常简单的快速开始的例子:

### **./App.js**

```
import React from 'react';
import ReactDOM from 'react-dom';
import getMuiTheme from 'material-ui/styles/getMuiTheme';
import MuiThemeProvider from 'material-ui/styles/MuiThemeProvider';
import MyAwesomeReactComponent from './MyAwesomeReactComponent';

const App = () => (
  <MuiThemeProvider muiTheme={getMuiTheme()}>
    <MyAwesomeReactComponent />
  </MuiThemeProvider>
);

ReactDOM.render(
  <App />,
  document.getElementById('app')
);
```

### **./MyAwesomeReactComponent.js**

```
import React from 'react';
import RaisedButton from 'material-ui/RaisedButton';

const MyAwesomeReactComponent = () => (
  <RaisedButton label="Default" />
);

export default MyAwesomeReactComponent;
```

请查阅其他的文档来看他们是如何被导入的。

注意在上面的例子，我们用了：

```
import RaisedButton from 'material-ui/RaisedButton';
```

替换成



```
import {RaisedButton} from 'material-ui';
```

这将让你的网站搭建过程更快，并且你的输出更小。为了完整的把所有Material-UI的组件 `import`，请看 `/index.js` 里的Material-UI npm package的根目录。

## 自定义

我们实现了一个默认的主题来生成所有Material-UI的组件。美化组件到你喜欢的是非常容易和简单的事情。这个可以通过以下两种方法来完成：

- (通过主题)`theme`，你可以使用自定义的主题来改变这些组件的风格。
- (通过线内的风格)`style`，你可以覆盖掉之前的风格设置。

## 服务端渲染

当在服务端使用Material-UI，我们必须使用相同的环境。

## 示例

我们在[例子](#)中提供了两个示例项目，你可以学习如何去使用 material-ui。

- 第一使用[browserify](#)去自动化构建我们的项目。
- 第二使用[webpack](#)。

当前文档的源代码的文档也包含在仓库中。这是一个比较复杂的项目，我们使用了webpack和material-ui的组件，你可以在[这里](#)去查看详细内容。























