

React 安装

React 可以直接下载使用，下载包中也提供了很多学习的实例。

本教程使用了 React 的版本为 16.4.0，你可以在官网 <https://reactjs.org/> 下载最新版。

你也可以直接使用 Staticfile CDN 的 React CDN 库，地址如下：

```
<script src="https://cdn.staticfile.org/react/16.4.0/umd/react.development.js"></script>
<script src="https://cdn.staticfile.org/react-dom/16.4.0/umd/react-dom.development.js"></script>
<!-- 生产环境中不建议使用 -->
<script src="https://cdn.staticfile.org/babel-standalone/6.26.0/babel.min.js"></script>
```

官方提供的 CDN 地址：

```
<script src="https://unpkg.com/react@16/umd/react.development.js"></script>
<script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
<!-- 生产环境中不建议使用 -->
<script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
```

注意：在浏览器中使用 Babel 来编译 JSX 效率是非常低的。

使用实例

以下实例输出了 Hello, world!

React 实例

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8" />
<title>Hello React!</title>
<script src="https://cdn.staticfile.org/react/16.4.0/umd/react.development.js"></script>
<script src="https://cdn.staticfile.org/react-dom/16.4.0/umd/react-dom.development.js"></script>
<script src="https://cdn.staticfile.org/babel-standalone/6.26.0/babel.min.js"></script>
</head>
<body>
<div id="example"></div>
<script type="text/babel">
ReactDOM.render(
<h1>Hello, world!</h1>,
document.getElementById('example')
);
</script>
</body>
</html>
```

[尝试一下 »](#)

实例解析：

实例中我们引入了三个库：react.min.js、react-dom.min.js 和 babel.min.js：

- **react.min.js** - React 的核心库
- **react-dom.min.js** - 提供与 DOM 相关的功能
- **babel.min.js** - Babel 可以将 ES6 代码转为 ES5 代码，这样我们就在在目前不支持 ES6 浏览器上执行 React 代码。Babel 内嵌了对 JSX 的支持。通过将 Babel 和 babel-sublime 包（package）一同使用可以让源码的语法渲染上升到一个全新的水平。

```
ReactDOM.render(  
  <h1>Hello, world!</h1>,  
  document.getElementById('example')  
)
```

以上代码将一个 h1 标题，插入 id="example" 节点中。

注意：

如果我们需要使用 JSX，则 `<script>` 标签的 `type` 属性需要设置为 `text/babel`。

通过 npm 使用 React

如果你的系统还不支持 Node.js 及 NPM 可以参考我们的 [Node.js 教程](#)。

我们建议在 React 中使用 CommonJS 模块系统，比如 browserify 或 webpack，本教程使用 webpack。

国内使用 npm 速度很慢，你可以使用淘宝定制的 cnpm (gzip 压缩支持) 命令行工具代替默认的 npm：

```
$ npm install -g cnpm --registry=https://registry.npm.taobao.org  
$ npm config set registry https://registry.npm.taobao.org
```

这样就可以使用 cnpm 命令来安装模块了：

```
$ cnpm install [name]
```

更多信息可以查阅：<http://npm.taobao.org/>。

使用 create-react-app 快速构建 React 开发环境

create-react-app 是来自于 Facebook，通过该命令我们无需配置就能快速构建 React 开发环境。

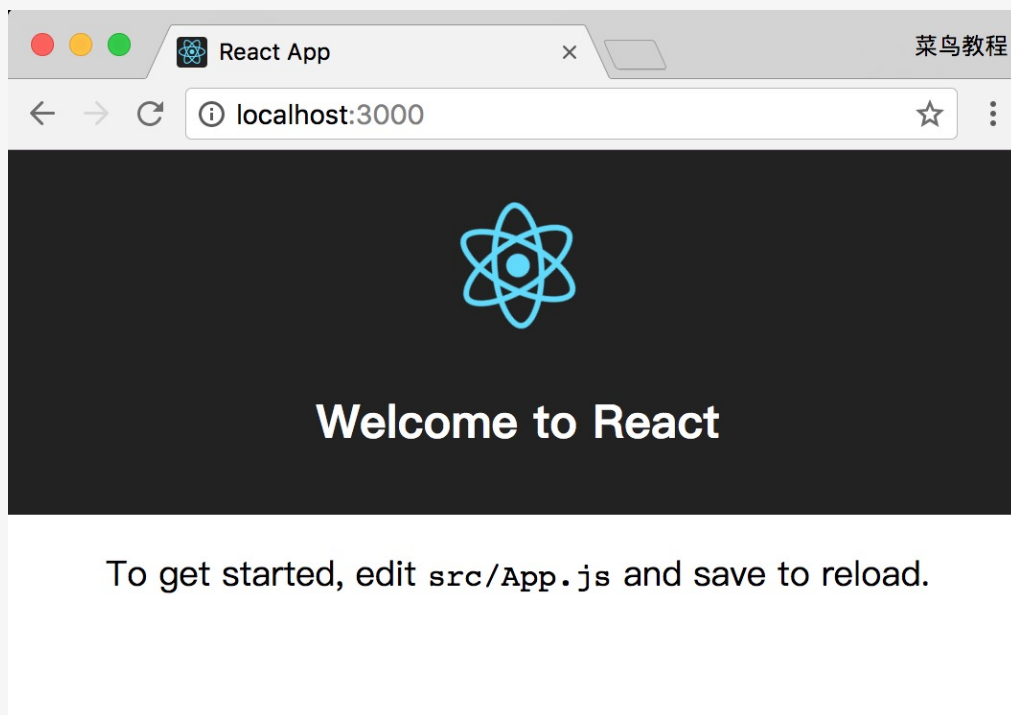
create-react-app 自动创建的项目是基于 Webpack + ES6。

执行以下命令创建项目：

```
$ cnpm install -g create-react-app  
$ create-react-app my-app
```

```
$ cd my-app/  
$ npm start
```

在浏览器中打开 <http://localhost:3000/>，结果如下图所示：



项目的目录结构如下：

```
my-app/  
  README.md  
  node_modules/  
  package.json  
  .gitignore  
  public/  
    favicon.ico  
    index.html  
    manifest.json  
  src/  
    App.css  
    App.js  
    App.test.js  
    index.css  
    index.js  
    logo.svg
```

manifest.json 指定了开始页面 index.html，一切的开始都从这里开始，所以这个是代码执行的源头。

尝试修改 src/App.js 文件代码：

src/App.js

```
import React, { Component } from 'react';
import logo from './logo.svg';
import './App.css';
class App extends Component {
  render() {
    return (
      <div className="App">
        <div className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <h2>欢迎来到菜鸟教程</h2>
        </div>
        <p className="App-intro">
          你可以在 src/App.js 文件中修改。
        </p>
      </div>
    );
  }
}
export default App;
```

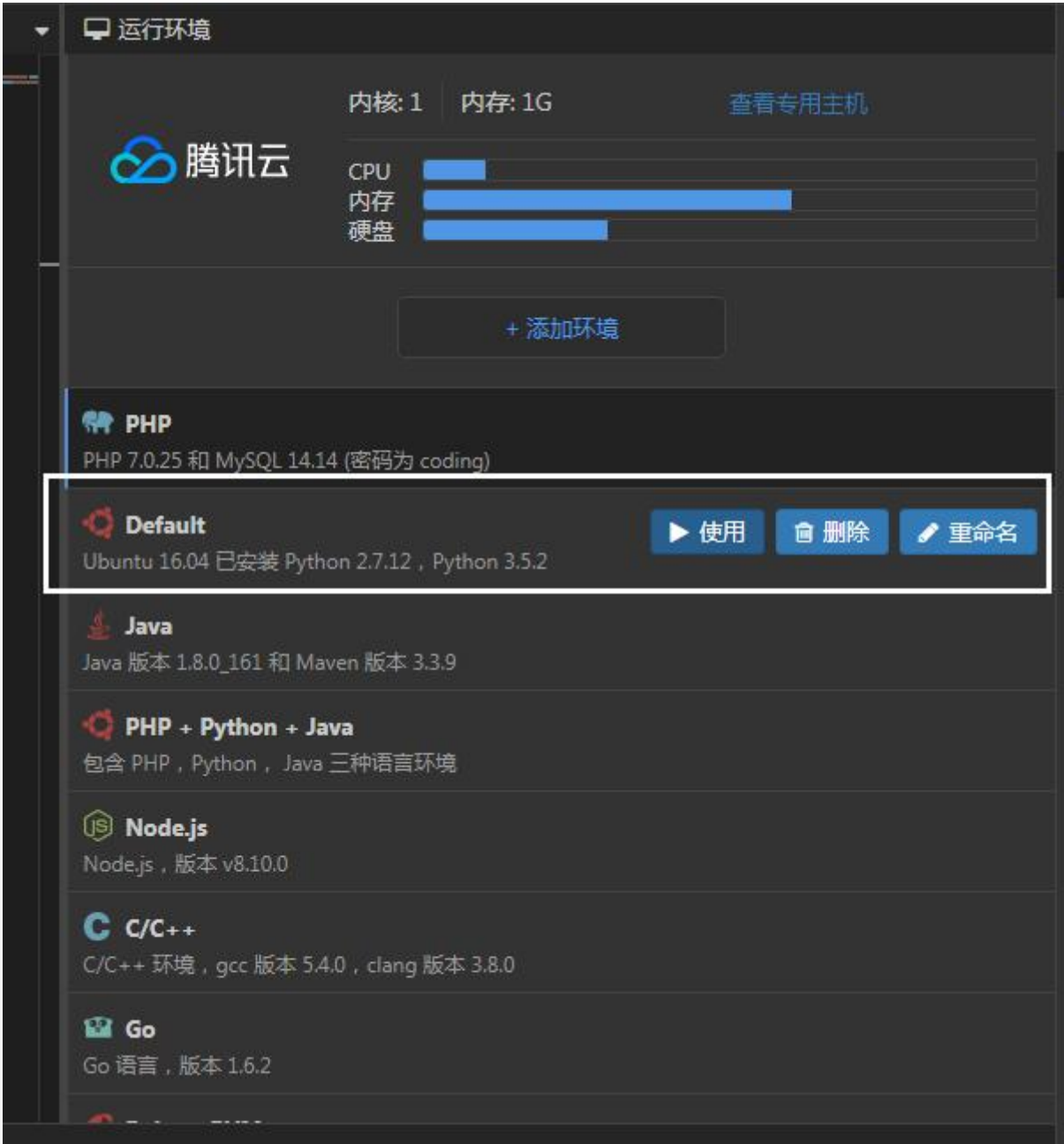
修改后，打开 <http://localhost:3000/>（一般自动刷新），输出结果如下：



在 Cloud Studio 中运行 React

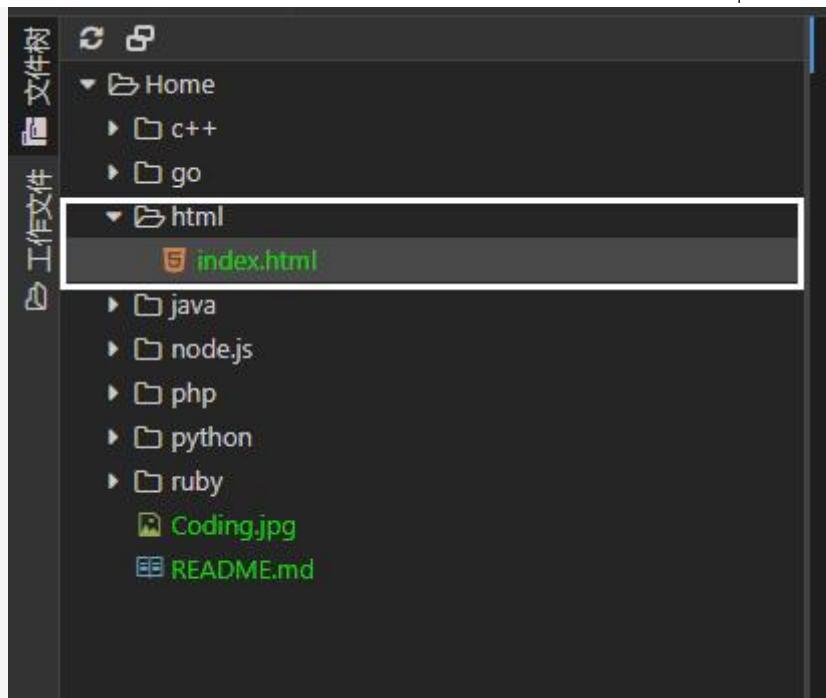
下面我们介绍如何在 Cloud Studio 中安装、使用 React：

- step1：访问[腾讯云开发者平台](#)，注册/登录账户。
- step2：在右侧的运行环境菜单选择 "ubuntu"：



alt=" />

- step3 : 在左侧代码目录中新建 html 目录，编写你的HTML代码，例如 index.html



- step4: 你可以在官网 <https://reactjs.org/> 下载最新版。你也可以直接使用 Staticfile CDN 的 React CDN 库，地址如下：

```
<script src="https://cdn.staticfile.org/react/16.4.0/umd/react.development.js"></script>
<script src="https://cdn.staticfile.org/react-dom/16.4.0/umd/react-dom.development.js"></script>
<!-- 生产环境中不建议使用 -->
<script src="https://cdn.staticfile.org/babel-standalone/6.26.0/babel.min.js"></script>
```

官方提供的 CDN 地址：

```
<script src="https://unpkg.com/react@16/umd/react.development.js"></script>
<script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
<!-- 生产环境中不建议使用 -->
<script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
```

- step5：在终端中输入命令 `sudo vim /etc/nginx/site-enable/default`。将配置文件红框部分修改为如下图所示，然后输入命令：`sudo service nginx restart` 重启 nginx 服务（nginx 安装完成并启动后默认会监听80端口。我们需要将 nginx 的站点目录以及监听的端口号改为我们需要的）

```
server {  
    listen 8080 default_server;  
    listen [::]:8080 default_server;  
  
    # SSL configuration  
    #  
    # listen 443 ssl default_server;  
    # listen [::]:443 ssl default_server;  
    #  
    # Note: You should disable gzip for SSL traffic.  
    # See: https://bugs.debian.org/773332  
    #  
    # Read up on ssl_ciphers to ensure a secure configuration.  
    # See: https://bugs.debian.org/765782  
    #  
    # Self signed certs generated by the ssl-cert package  
    # Don't use them in a production server!  
    #  
    # include snippets/snakeoil.conf;  
  
    root /home/coding/workspace/html;  
  
    # Add index.php to the list if you are using PHP  
    index index.html index.htm index.nginx-debian.html;  
  
    server_name _;  
  
    location / {  
        # First attempt to serve request as file, then  
        # as directory, then fall back to displaying a 404.  
        try_files $uri $uri/ =404;  
    }  
}
```

- step6: 点击最右侧的【访问链接】选项卡，在访问链接面板中填写端口号为：8080（和刚才 nginx 配置文件中的端口号一致），点击创建链接，即可点击生成的链接访问我们刚刚编写的代码，查看 React 效果。



现在 CODING 正在举办一场基于 Cloud Studio 工作空间的【我最喜爱的 Cloud Studio 插件评选大赛】。进入活动官网：<https://studio.qcloud.coding.net/campaign/favorite-plugins/index>，了解更多活动信息。

[← React 教程](#)[React JSX →](#)**2 篇笔记****写笔记**



create-react-app 执行慢的解决方法：

在使用 **create-react-app my-app** 来创建一个新的React应用，在拉取各种资源时,往往会非常慢,一直卡在那：

```
fetchMetadata: sill mapToRegistry uri http://registry.npmjs.org/whatwg-fetch
```

可以看到资源还是使用了 **npmjs.org**，解决方法是换成淘宝的资源：

```
$ npm config set registry https://registry.npm.taobao.org
-- 配置后可通过下面方式来验证是否成功
$ npm config get registry
-- 或 npm info express
```

tianqixin 2年前 (2017-02-14)



React 代码的书写格式和以前的 JS 有很大的不同，下面通过对这段代码进行分析了解一下他。

以前使用JS 定义一个变量使用 **var** 现在用 **const**：

```
const div = document.createElement('div');
```

ReactDOM.render(...) 是渲染方法，所有的 js,html 都可通过它进行渲染绘制，他又两个参数，内容和渲染目标 js 对象。

内容就是要在渲染目标中显示的东西，可以是一个React 部件，也可以是一段HTML或TEXT文本。渲染目标JS对象，就是一个DIV或TABLE,或TD 等HTML的节点对象。

```
ReactDOM.render(<App />, div);
```

ReactDOM.unmountComponentAtNode() 这个方法是解除渲染挂载，作用和 **render** 刚好相反，也就清空一个渲染目标中的 React 部件或 html 内容。

```
ReactDOM.unmountComponentAtNode(div);
```

xjjuser 7个月前 (08-09)