

# 开始使用MontageJS

MontageJS是一个用于创建富单页面应用（SPAs）的HTML5框架，框架中包括以下特性：针对手持设备特殊优化的组件，纯页面模板，可重用组件，组件与对象之间的双向数据绑定，隐式事件委托，可控的页面绘制周期，同时框架也提供了丰富的命令行工具来帮助开发者创建和编译工程项目。

## 划时代的前端开发框架

---

现在如果我们要使用一个框架，大体过程是这样的，下载整个框架包然后导入到我们的项目中，其实大多数的时候我们只会用到框架中的很小的一部分功能，但是却不得不把整个框架引入到项目中。这样做会让我们的项目变的臃肿，运行缓慢，特别是在有网络带宽限制，甚至设备性能限制的的网络应用中。

MontageJS用一种全新的方式完美地解决了上述问题。我们不用从一个URL去下载框架本身，因为MontageJS使用[CommonJS](#)的模块方式发布，[CommonJS](#)属于[npm](#)包管理系统的一部分，所以开发者可以非常容易地把框架引入到项目中。在开发的时候我们只需要在项目中用关键字 `require` 引用相关的包就可以了。最后在项目发布阶段使用Montage的优化工具[mop](#)对项目进行编译，它会分析你的项目，只把你用到的模块，组件编辑到最终结果中，所以你的最终产品是一个高度优化，体积很小的代码包。

## 配置开发环境

---

通过下面的两个步骤我们可以安装MontageJS的开发环境和命令行工具：

- [Node.js 和 npm](#)
- [Minit](#), MontageJS的命令行工具

当然我们也需要在开发的机器上安装一个常用的浏览器。比如Chrome, Safari, Firefox等等。

## 准备工作

[下载](#)安装Node.js，如果你开发的机器还没用安装。因为MontageJS需要Node.js提供的版本和包依赖管理功能，同时我们也需要用Node.js的npm命令行工具来帮助我们安装MontageJS本身。

### 步骤 1：安装mininit

mininit是一个命令行工具，我们可以使用它来很方便的创建一个MontageJS应用的种子项目，创建好的工程项目中已经包括了我们在之后开发的过程中所有需要的东西。通过下面的命令行我们就可以在不同的操作系统上面安装mininit:

- **Mac OS X / Linux:** 打开终端输入下面的命令:

```
$ sudo npm install -gq mininit@latest
```

- **Windows:** 运行命令行工具输入下面得命令

```
npm install -gq mininit@latest
```

### 步骤 2：创建一个新项目

通过下面的命令我们可以创建一个MontageJS的种子项目（参数“app-nam”是工程项目的名称，使用的时候请设置为当前的项目名称）

```
$ minit create:app -n app-name
```

运行上面的命令之后会在当前目录中新建一个跟我们项目名称一样的文件夹，比如上面的命令行就会新建一个“app-name”的文件夹，在这个文件夹中已经包含所有MontageJS项目需要的东西了。

**备注:** 可以打开"app-name"文件夹中的readme文本文件查看新建的MontageJS工程项目的目录结构

### 步骤 3：验证项目是否初始化成功

通过以下的步骤验证

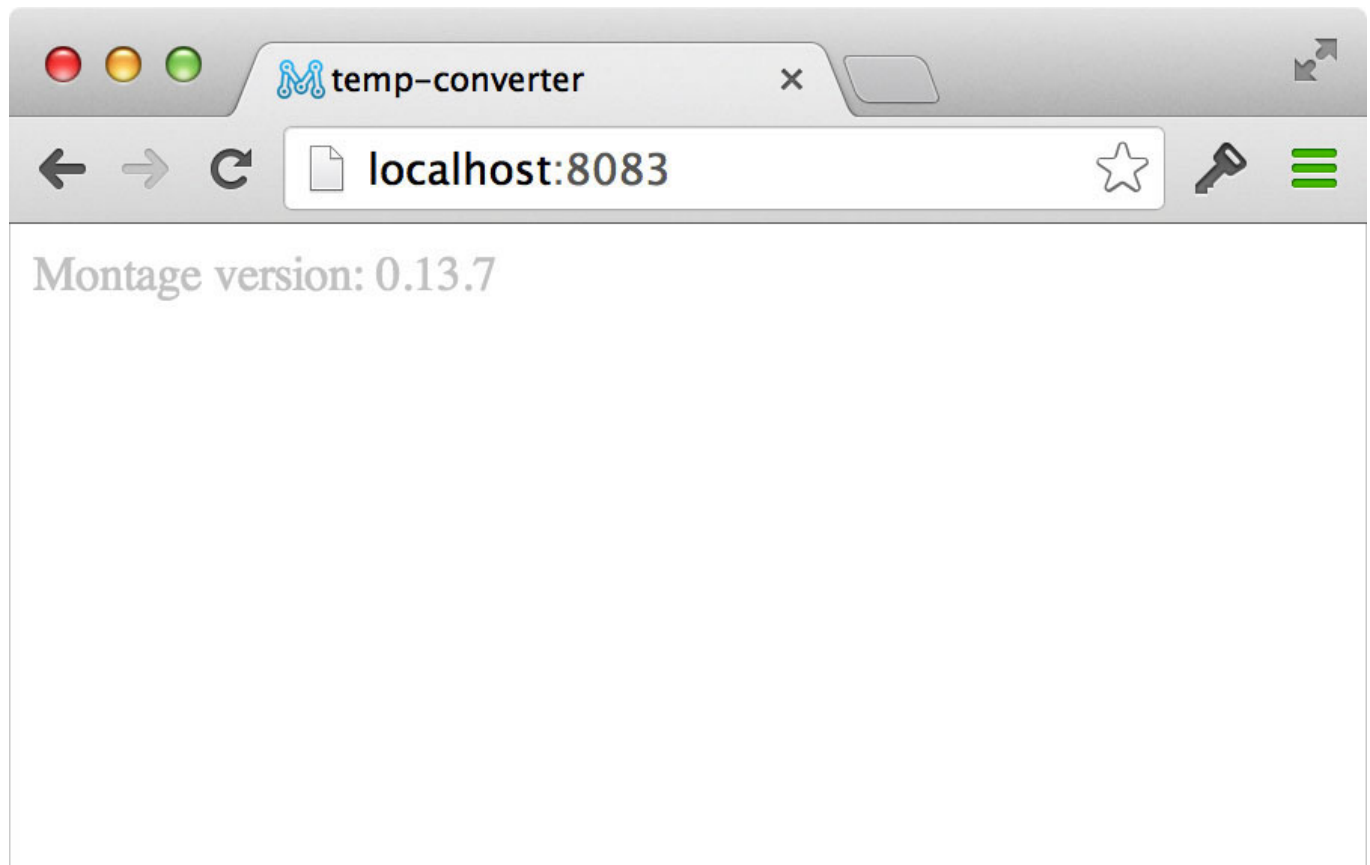
1. 在命令行工具中切换到当前项目的文件夹路径然后运行以下的命令：


```
$ cd app-name  
$ minit serve &
```

备注：MontageJS运行的时候需要通过XMLHttpRequest (XHR)机制从远端服务器加载相关组件和模块（所以我们需要一个服务器才能使我们的程序运行起来）；命令`minit serve &`就会启动一个这样的本地服务器。命令末尾的`&`表示启动的本地服务器在后台运行，这样我们就可以在启动服务之后把命令行窗口关闭。

2. 打开浏览器输入网址<http://localhost:8083/>。

如果一切顺利你将看到一个空白的网页，在网页的左上角有版本信息，（如下面的屏幕截图）。





图片 1. MontageJS开发环境配置成功

现在为止，所有的配置已经完成，可以开始编码了。

## 下一阶段

---

- 如何用MontageJS编写[Hello MontageJS](#)
- 通过左边的菜单查看MontageJS的文档，教程和示例代码