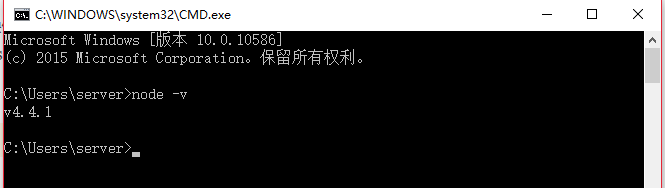
[从零开始使用Electron + jQuery开发桌面应用 （一） HelloWorld](https://segmentfault.com/a/1190000004843033)

写在前面：本人技术一般，此文章为本人边学边写而成，难免出现差错与技术方面不完美的情况，仅供参考。如有错误，欢迎指正。  
能翻墙就尽量翻墙，国内网络你懂的  
本文以Windows开发平台为基础写的，MAC用户仅供参考  
（[electron中文API下载](http://download.csdn.net/detail/lvyuanpeng/9487115)）

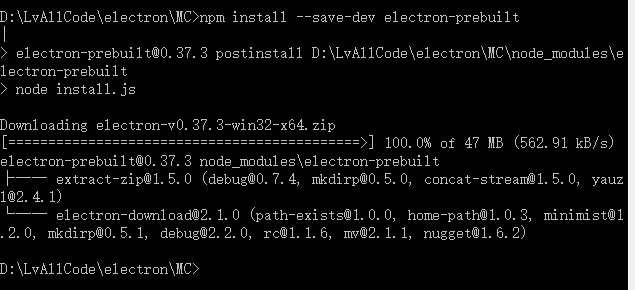
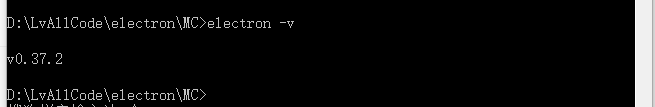
1.首先安装[node](https://nodejs.org/en/" \t "_blank)

安装完成以后在cmd里运行 node -v检测安装是否成功。  


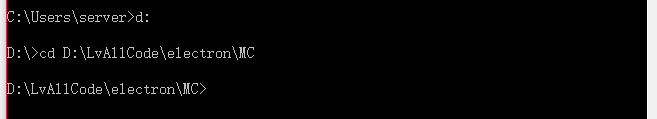
2.安装[Electron](http://electron.atom.io/" \t "_blank)

使用npm 安装 Electron:

npm **install** –g --save-dev electron-prebuilt

  
安装完成后运行 electron -v 检查是否安装成功  


3.一个简单的Electron程序

首先，切换到应用开发根目录，我的在D:\LvAllCode\electron\MC  
  
然后创建Electron简单文件结构  
在根目录下使用npm init命令创建package.json文件，根据提示填写就行，  
执行一次npm install --save-dev electron-prebuilt  
最后差不多是这个样子

{

"name": "my-app",

"version": "1.0.0",

"description": "my-study-simple-proj",

"main": "app/main.js",

"dependencies": {

"electron-prebuilt": "^0.37.3"

},

"devDependencies": {

"electron-prebuilt": "^0.37.3"

},

"scripts": {

"test": "hello",

"start": "electron ."

},

"author": "CodingWorker",

"license": "MIT"

}

然后在根目录下创建app文件夹，在app/下创建main.js（me：注意，electron的版本不同main.js的代码可能不同）,内容如下：

// 应用控制模块app,第三方模块

//var app = require("app");

// 创建窗口模块，类似浏览器的窗口

//var BrowserWindow = require('browser-window');

// 主窗口

//var mainWindow = {};

//var electron = require('electron');

//const app = electron.app;

//const BrowserWindow = electron.BrowserWindow;

var { app, BrowserWindow } = require('electron')

// 初始化并准备创建主窗口

app.on('ready', function () {

// 创建一个宽800px 高700px的窗口

mainWindow = new BrowserWindow({

width: 1000,

height: 800,

});

// 载入应用的inde.html

mainWindow.loadURL('file://' + \_\_dirname + '/html/index.html');

// 打开开发工具界面

mainWindow.openDevTools();

// 窗口关闭时触发

mainWindow.on('closed', function(){

// 想要取消窗口对象的引用， 如果你的应用支持多窗口，你需要将所有的窗口对象存储到一个数组中，然后在这里删除相对应的元素

});

});

然后在app/html下创建index.html文件，内容如下：

<!doctype html>

<html>

<head>

<meta charset="utf-8" />

<title>我的世界</title>

</head>

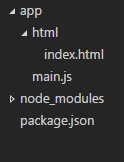
<body>

<span style="color:#fff;">Hello World</span>

</body>

</html>

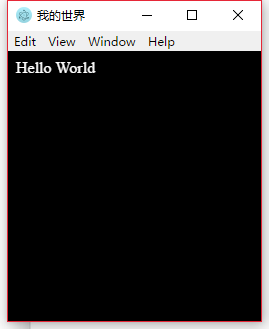
最后，项目整体文件目录大概是这个样子的：



4.启动你的项目

在根目录下，命令行输入 npm start(需要在package.json中配置

"scripts": {"start": "electron ."}

)或者输入 electron .(注意有个点，并且electron装的是全局的)   


5.配置[VS Code](https://www.visualstudio.com/en-us/products/code-vs.aspx" \t "_blank)开发环境

先装全局glup：npm install gulp -g  
再装本地：npm install --save-dev gulp  
至于为啥[看这](https://www.zhihu.com/question/36023122/answer/65611814" \t "_blank)  
https://segmentfault.com/img/bVuubf

根目录下创建gulpfile.js配置文件,内容如下：

// 获取依赖

var gulp = require('gulp'),

childProcess = require('child\_process'),

electron = require('electron-prebuilt');

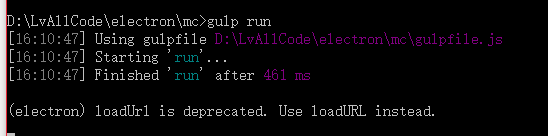
// 创建 gulp 任务

gulp.task('run', function () {

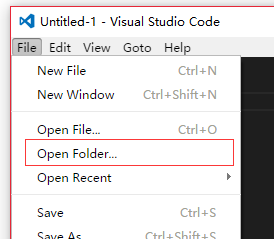
childProcess.spawn(electron, ['.'], {stdio:'inherit'});

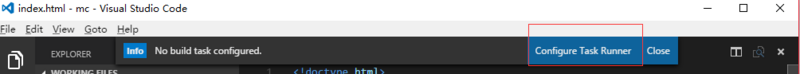
});

运行gulp任务：gulp run

  
又看到了熟悉的界面了！！

在VS Code里打开你的工程：

  
摁下快捷键ctrl+shift+b,第一次启动的时候会提示你配置task文件，点击Configure Task Runner

  
之后VS Code会自动在./.vscode目录下创建一个tasks.json配置文件，我们来编辑一下这个文件：

{

"version": "0.1.0",

"command": "gulp",

"isShellCommand": true,

"args": [

"--no-color"

],

"tasks": [ {

"taskName": "run",

"args": [],

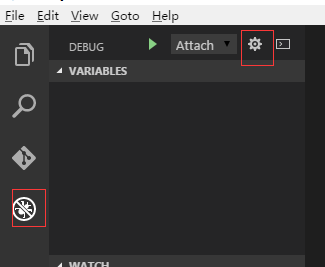
"isBuildCommand": true

}]

}

保存，之后再摁下ctrl+shift+b就可以直接运行我们的程序了。

6.使用VS Code调试

（此部分仅供参考，因为我自己没跑成功o(>﹏<)o，如有知道为什么的还请大神告知）  
点击DEBUG界面的设置按钮  
  
VS Code自动在./.vscode目录下创建了launch.json   
我们不需要启动 main.js 的配置 所以移除它，现在launch.json 应该如下所示：

{

"version": "0.2.0",

"configurations": [

{

"name": "Attach",

"type": "node",

"request": "attach",

"port": 5858,

"address": "localhost",

"restart": false,

"sourceMaps": false,

"outDir": null,

"localRoot": "${workspaceRoot}",

"remoteRoot": null

}

]

}

更改之前创建的 gulpfile.js 文件

gulp.task('run', function () {

childProcess.spawn(electron, ['--debug=5858','.'], {stdio:'inherit'});

});

在调试面板中选择Attach 然后点击RUN按钮即可开始调试。

**至此我们的第一个Electron程序就开发完成了，**[**下一节**](https://segmentfault.com/a/1190000004863646)**我们看如何打包成执行程序**