« 返回 | 编辑 | 删除 | 发送给好友

https://blog

创建时间：2020年9月23日(星期三) 晚上10:50 | 分类：未分类 | 字数：9879 | 另存为... | 打印 | 添加到日历

https://blog.csdn.net/dwjpeng2/article/details/82994321

wget http://nodejs.org/dist/latest/node-v10.11.0-linux-x64.tar.gz

tar xf node-v5.10.1-linux-x64.tar.gz -C /usr/local/

cd /usr/local/

mv node-v5.10.1-linux-x64/ nodejs

ln -s /usr/local/nodejs/bin/node /usr/local/bin

ln -s /usr/local/nodejs/bin/npm /usr/local/bin

下载最新的版本（8.1.3），解压 tar -xvf node-v8.1.3-linux-x64.tar.xz

安装最新版本的npm： npm install -g npm

安装cnpm：npm install -g cnpm --registry=http://registry.npm.taobao.org

C:\Users\admini\AppData\Roaming\npm\cnpm

安装electron： cnpm install -g electron 再做这一步之前可能需要将cnpm加入到path环境变量

二.克隆github上的electron入门项目到本地

安装git： sudo apt-get install git

命令： git clone https://github.com/electron/electron-quick-start

注意：环境突然搞坏了，运行不了程序，恢复命令步骤:

安装cnpm：npm install -g cnpm --registry=http://registry.npm.taobao.org

将cnpm加入到path环境变量

安装electron： cnpm install -g electron

再运行程序

export NODE_HOME=/usr/local/nodejs/bin
export PATH=$NODE_HOME:$PATH

install use root

三.执行

切换到项目路径下执行：npm start

或者直接执行 electron .（注意后面有个点）

四.安装打包工具

```
npm install --save-dev electron-packager
```

1. 得到原本的镜像地址

```
npm get registry
```

https://registry.npmjs.org/

2. 设成淘宝的

```
npm config set registry http://registry.npm.taobao.org/

yarn config set registry http://registry.npm.taobao.org/
```

3. 换成原来的

```
npm config set registry https://registry.npmjs.org/
```

打开项目路径

1使用命令 npm install --save-dev electron-packager将electron-package安装到项目的路径下面

注:完成以上两步骤会在 package.json 生成文件

```
"devDependencies": {
    "electron-packager": "^8.5.1"
}
```

3在项目根目录下面的 package.json 里添加类似于如下代码

```
"scripts": {

os系统:"packageDarwin": "electron-packager . 'Hosts' --platform=darwin --arch=x64 --icon=hosts.icns --out=./dist --asar --app-version=2.0.1 --ignore=\"
(dist|src|docs|.gitignore|LICENSE|README.md|webpack.config*|node_modules)\"",

os系统:"packageDarwin": "electron-packager . 'Hosts' --platform=darwin --arch=x64 --icon=hosts.icns --out=./dist --asar --app-version=2.0.1",


windows系统:"packageWin": "electron-packager . 'Hosts' --platform=win32 --arch=x64 --icon=hosts.ico --out=./dist --asar --app-version=2.0.1 --ignore=\"
(dist|src|docs|.gitignore|LICENSE|README.md|webpack.config.js|node_modules)\"",

windows系统:"packageWin": "electron-packager . 'Hosts' --platform=win32 --arch=x64 --icon=hosts.ico --out=./dist --asar --app-version=2.0.1",


linux系统:"packageLinux": "electron-packager . 'Hosts' --platform=linux --arch=x64 --out=./dist --asar --app-version=2.0.1 --ignore=\"
(dist|src|docs|.gitignore|LICENSE|README.md|webpack.config.js|node_modules)\""

linux系统:"packageLinux": "electron-packager . 'Hosts' --platform=linux --arch=x64 --out=./dist --asar --app-version=2.0.1"

}
```

命令说明:
* *location of project*:项目所在路径
* *name of project*:打包的项目名字
* *platform*:确定了你要构建哪个平台的应用(*Windows*、*Mac* 还是 *Linux*)
* *architecture*:决定了使用 *x86* 还是 *x64* 还是两个架构都用
* *electron version*:*electron-prebuilt* 的版本
* *optional options*:可选选项

PS:这里要注意,字段里的 项目名字,*version*,*icon*路径要改成自己的; 例如:*"packager": "electron-packager ~/Desktop/myFirstElectronApp(*项目位置*) Hello(*项目名称*) --linux --out ./OutApp(*项目导出位置*) --version 1.4.13 --overwrite"*

*4*然后,使用命令 *npm run-script package*---即可打包

https://github.com/electron/electron/issues/26827

小雨子1993
(二)使用electron-packager打包成应用文件

前提已经实现了(一)里面的效果连接:https://www.cnblogs.com/yuNotes/p/12884930.html

1.下载electron-packager:运行命令 npm install electron-packager --save-dev

```
D:\huaxin\elelctron\electron-quick-start>npm install electron-packager --save-dev
+ electron-packager@14.2.1
added 78 packages from 60 contributors in 9.645s
```

2.在packageage.json中加入"packager": "electron-packager ./ --platform=[linux/win32/darwin/all] --arch=x64 --overwrite"([]中只能选择一个分别代表linux/windows/mac中使用的包,all代表全部)

```
{
  "name": "electron-quick-start",
  "version": "1.0.0",
  "description": "A minimal Electron application",
  "main": "main.js",
  "scripts": {
    "start": "electron .",
    "packager": "electron-packager ./ --platform=linux --arch=x64 --overwrite"
  },
  "repository": "https://github.com/electron/electron-quick-start",
  "keywords": [
    "Electron",
    "quick",
    "start",
    "tutorial",
    "demo"
  ],
  "author": "GitHub",
  "license": "CC0-1.0",
  "devDependencies": {
    "electron": "^9.0.0",
    "electron-packager": "^14.2.1"
  }
}
```

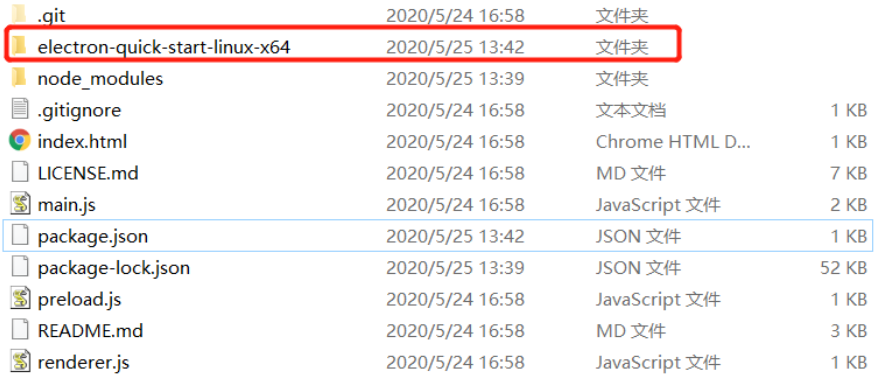
3.运行命令:npm run-script packager(我在windows环境打包all情况下只打包除了liunx环境和windows环境,后续有成功会在跟进)

```
D:\huaxin\elelctron\electron-quick-start>npm run-script packager

> electron-quick-start@1.0.0 packager D:\huaxin\elelctron\electron-quick-start
> electron-packager ./ --platform=all --arch=x64 --overwrite

Packaging app for platform win32 x64 using electron v9.0.0
Packaging app for platform linux x64 using electron v9.0.0
Downloading electron-v9.0.0-mas-x64.zip: [========================================================] 100% ETA: 0.0 seconds
Downloading electron-v9.0.0-darwin-x64.zip: [==================-----------------------------] 38% ETA: 51.9 seconds C
annot create symlinks (on Windows hosts, it requires admin privileges); skipping mas platform
Downloading electron-v9.0.0-darwin-x64.zip: [========================================================] 100% ETA: 0.0 seconds
Cannot create symlinks (on Windows hosts, it requires admin privileges); skipping darwin platform
Wrote new apps to:
D:\huaxin\elelctron\electron-quick-start\electron-quick-start-linux-x64
D:\huaxin\elelctron\electron-quick-start\electron-quick-start-win32-x64
```

4.在打包完成后在你项目的根目录下会出现下面的文件包里面就有你需要的exe文件但是liunx中我还没有细研究后续会更新



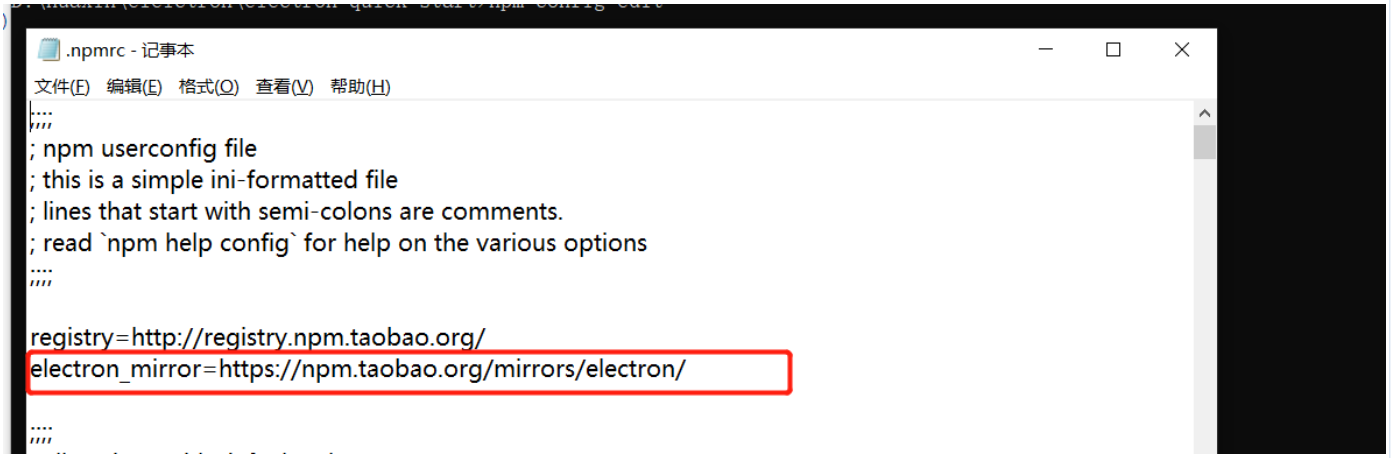
5.到这里你已经的成功了!!!

添加项:在第二步中命令可以看这篇文章: https://www.cnblogs.com/zwhbk/p/6396500.html

在第三步中可能会遇到命令卡在

electron-quick-start@1.0.0 packager D:\huaxin\elelctron\electron-quick-start
> electron-packager ./ --platform=win32 --arch=x64 --overwrite这里可以按下面的步骤进行

(1)运行命令npm config edit,在弹出的文件中加入electron_mirror="https://npm.taobao.org/mirrors/electron/";这句话



(2)删除node_modules文件删除重新进行npm install,然后进行npm run-script packager命令应该就可以了

错误：electron>npm ERR! electron-quick-start@1.0.0 start: `electron .`

沧桑岁月歌 关注

2020.01.20 00:23:55 字数 28阅读 1,218

```
?  electron-quick-start git:(master) npm start
> electron-quick-start@1.0.0 start /Users/baidu/Downloads/bishe/electron-quick-start
> electron .
fs.js:584
  return binding.open(pathModule._makeLong(path), stringToFlags(flags), mode);
                 ^

Error: ENOENT: no such file or directory, open '/Users/baidu/Downloads/bishe/electron-quick-start/node_modules/electron/path.txt'
    at Error (native)
    at Object.fs.openSync (fs.js:584:18)
    at Object.fs.readFileSync (fs.js:431:33)
    at Object.<anonymous> (/Users/baidu/Downloads/bishe/electron-quick-start/node_modules/electron/index.js:4:42)
    at Module._compile (module.js:413:34)
    at Object.Module._extensions..js (module.js:422:10)
    at Module.load (module.js:357:32)
    at Function.Module._load (module.js:314:12)
    at Module.require (module.js:367:17)
    at require (internal/module.js:16:19)
npm ERR! Darwin 15.5.0
npm ERR! argv "/Users/baidu/.nvm/versions/node/v5.5.0/bin/node" "/Users/baidu/.nvm/versions/node/v5.5.0/bin/npm" "start"
npm ERR! node v5.5.0
npm ERR! npm  v3.3.12
npm ERR! code ELIFECYCLE
npm ERR! electron-quick-start@1.0.0 start: `electron .`
npm ERR! Exit status 1
npm ERR!
npm ERR! Failed at the electron-quick-start@1.0.0 start script 'electron .'.
npm ERR! Make sure you have the latest version of node.js and npm installed.
npm ERR! If you do, this is most likely a problem with the electron-quick-start package,
npm ERR! not with npm itself.
npm ERR! Tell the author that this fails on your system:
npm ERR!     electron .
npm ERR! You can get their info via:
npm ERR!     npm owner ls electron-quick-start
npm ERR! There is likely additional logging output above.
npm ERR! Please include the following file with any support request:
npm ERR!     /Users/baidu/Downloads/bishe/electron-quick-start/npm-debug.log
```

解决：执行 electron 的时候是全局的，然后 cnpm install electron --save-dev 局部执行就好了

electron打包整理

最近在折腾把项目打包成桌面应用程序，发现一个工具electron，可以讲项目打包成一个跨平台的应用程序，很方便，来学习一下。

1、先安装electron、electron-packager，安装方法可以使用package.json文件配置，然后npm install
也可以使用cnpm安装，速度会快点，具体如下：

npm install -g cnpm --registry=https://registry.npm.taobao.org
cnpm i

package.json如下：

```
{
  "name": "electron_demo",
  "version": "1.0.0",
  "main": "main.js",
  "scripts": {
  "start": "electron .",
  "packager": "electron-packager ./ stockschool --platform=win32 --arch=x64 --icon=./pazq.ico --out=./ElectronApp"
  },
  "devDependencies": {
  "electron": "^1.7.8",
  "electron-packager": "^10.1.2"
  },
  "dependencies": {}
  }
```

2、安装完成后，准备好要打包的项目，并增加一个main.js,用来声明一个类似webview的东西，来加载页面。

```
const electron = require('electron')
  // Module to control application life.
  const app = electron.app
  // Module to create native browser window.
  const BrowserWindow = electron.BrowserWindow

  const path = require('path')
  const url = require('url')

  // Keep a global reference of the window object, if you don't, the window will
  // be closed automatically when the JavaScript object is garbage collected.
  let mainWindow

  function createWindow() {
      // Create the browser window.
      mainWindow = new BrowserWindow({
          width: 1366,
          height: 727,
          minWidth: 1366, // Integer (可选) - 窗口的最小宽度, 默认值为 .
          minHeight: 727, // Integer (可选) - 窗口的最小高度. 默认值为 .
          maxWidth: 1366, // Integer (可选) - 窗口的最大宽度, 默认无限制.
          maxHeight: 727, // Integer (可选) - 窗口的最大高度, 默认无限制.
          minimizable: true, // Boolean (可选) - 窗口是否可以最小化. 在 Linux 中无效. 默认值为 true.
          maximizable: false, // Boolean (可选) - 窗口是否可以最大化动. 在 Linux 中无效. 默认值为 true.
          useContentSize: false, // width 和 height 将使用 web 页面的尺寸
          center: true,  // Boolean (可选) - 窗口在屏幕居中.
      })
      mainWindow.setMenu(null)

      // and load the index.html of the app.
      mainWindow.loadURL(url.format({
          pathname: path.join(__dirname, './stockschool/index.html'),
          protocol: 'file:',
          slashes: true
      }))
       // 加载应用的 index.html
       // mainWindow.loadURL('file://' + __dirname + '/index.html');

      // Open the DevTools. // 打开开发工具 mainWindow.openDevTools();
      // mainWindow.webContents.openDevTools()

      // Emitted when the window is closed.
      mainWindow.on('closed', function () {
          // Dereference the window object, usually you would store windows
          // in an array if your app supports multi windows, this is the time
          // when you should delete the corresponding element.
          mainWindow = null
      })
  }

  // This method will be called when Electron has finished
  // initialization and is ready to create browser windows.
  // Some APIs can only be used after this event occurs.
  app.on('ready', createWindow)
```

```
// Quit when all windows are closed.
app.on('window-all-closed', function () {
    // On OS X it is common for applications and their menu bar
    // to stay active until the user quits explicitly with Cmd + Q
    if (process.platform !== 'darwin') {
        app.quit()
    }
})

app.on('activate', function () {
    // On OS X it's common to re-create a window in the app when the
    // dock icon is clicked and there are no other windows open.
    if (mainWindow === null) {
        createWindow()
    }
})

// In this file you can include the rest of your app's specific main process
// code. You can also put them in separate files and require them here.
```

3、启动项目：
npm start

4、打包，根据平台打包成一个.exe文件。（导出目录见 package.json中的out配置项）
打包方法：cnpm run packager

出现

```
> electron-quick-start@1.0.0 packager C:\Users\admini\Desktop\electron\electron-quick-start-master
> electron-packager ./ --platform=win32 --arch=x64 --overwrite

Downloading electron-v11.1.0-win32-x64.zip: [------------------------------------] 0% ETA: 0.0 seconds
PS C:\Users\admini\Desktop\electron\electron-quick-start-master> npm config edit
PS C:\Users\admini\Desktop\electron\electron-quick-start-master> cnpm run packager

> electron-quick-start@1.0.0 packager C:\Users\admini\Desktop\electron\electron-quick-start-master
> electron-packager ./ --platform=win32 --arch=x64 --overwrite

Downloading electron-v11.1.0-win32-x64.zip: [------------------------------------] 0% ETA: 0.0 seconds
PS C:\Users\admini\Desktop\electron\electron-quick-start-master> cnpm config edit
PS C:\Users\admini\Desktop\electron\electron-quick-start-master> cnpm run packager

> electron-quick-start@1.0.0 packager C:\Users\admini\Desktop\electron\electron-quick-start-master
> electron-packager ./ --platform=win32 --arch=x64 --overwrite

Packaging app for platform win32 x64 using electron v11.1.0
Wrote new app to C:\Users\admini\Desktop\electron\electron-quick-start-master\electron-quick-start-win32-
x64
PS C:\Users\admini\Desktop\electron\electron-quick-start-master>
```

electron_mirror=https://npm.taobao.org/mirrors/electron/

package.json中的打包配置

electron-packager <应用目录> <应用名称> <打包平台> --out <输出目录> <架构> <应用版本>

electron-packager . HelloWorld --platform=win32 --arch=x64 --icon=./pazq.ico --out=./ElectronApp --version=0.0.1

[上次修改时间：2021年2月17日(星期三) 晚上10:23]