1. 初始化项目

vue init simulatedgreg/electron-vue my-pro

cd my-pro

cnpm i

1. 启动开发环境
2. npm run dev

RROR in Template execution failed: ReferenceError: process is not defined

1. Index.ejs 问题

Node 安装最新版本

Src/Main.js

mainWindow = new BrowserWindow({

    height: 563,

    useContentSize: true,

    width: 1000,

    webPreferences: {

      nodeIntegration: true

    }

  })

Webpack.web.config.js

Webpack.renderer.comfig.js

templateParameters(compilation, assets, options) {

        return {

          compilation: compilation,

          webpack: compilation.getStats().toJson(),

          webpackConfig: compilation. options,

          htmlWebpackPlugin: {

            files: assets,

            options: options,

          },

          process,

        };

      },

1. Index.ejs 改成index.html
2. CopyWebpackPlugin Copy共有资源

Mkdir resource

Mkdir renders

new CopyWebpackPlugin([

      {

        from: path.join(\_\_dirname, '../src/resource'), // 渲染进程共有的资源

        to: path.join(\_\_dirname, '../dist/electron/resource')

      },

      {

        from: path.join(\_\_dirname, '../src/renders'), // 除了vue渲染进程外其他的渲染页面

        to: path.join(\_\_dirname, '../dist/electron/renders')

      }

    ]),

1. Uncaught Error: Cannot find module 'axios'

externals: [

    // ...Object.keys(dependencies || {}).filter(

    //   (d) => !whiteListedModules.includes(d)

    // ),

  ],

1. 启动生产环境
2. 如果已有electron-build 模板打包工具是electron-builder, 否则是electron-packager

<https://github.com/SimulatedGREG/electron-vue/blob/master/template/package.json>

npm 镜像设置

registry=https://registry.npm.taobao.org

sass\_binary\_site=https://npm.taobao.org/mirrors/node-sass/

phantomjs\_cdnurl=http://npm.taobao.org/mirrors/phantomjs

ELECTRON\_MIRROR=http://npm.taobao.org/mirrors/electron/

ELECTRON\_BUILDER\_BINARIES\_MIRROR=http://npm.taobao.org/mirrors/electron-builder-binaries/

1. Electron、electron-builder包下载

Get "https://github-production-release-asset-2e65be.s3.amazonaws.com/9384267/b487f180-9921-11ea-830a-154affe9bdc2?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20200523%2Fus-east-1%2Fs3%2Faws4\_request&X-Amz-Date=20200523T121857Z&X-Amz-Expires=300&X-Amz-Signature=f7336e9cc8bf2fe598191c020cd6d47dcb50aa4313b0f5194b31f50e0f83de32&X-Amz-SignedHeaders=host&actor\_id=0&repo\_id=9384267&response-content-disposition=attachment%3B%20filename%3Delectron-v9.0.0-win32-x64.zip&response-content-type=application%2Foctet-stream":Electron、electron-builder包下载

添加electron和electron-builder包C:\Users\Administrator\AppData\Local

1. Package.json配置

DevDependencies: {

 "electron": "^9.0.0",

    "electron-builder": "^22.6.0",

    "electron-debug": "^1.5.0",

    "electron-devtools-installer": "^2.2.4",

    "electron-rebuild": "^1.8.1",

}

"nsis": {

      "oneClick": false,

      "allowElevation": true,

      "allowToChangeInstallationDirectory": true,

      "createDesktopShortcut": true,

      "createStartMenuShortcut": true,

      "shortcutName": "chat"

    },

    "win": {

      "icon": "build/icons/icon.ico",

      "target": [

        {

          "target": "nsis",

          "arch": [

            "ia32",

            "x64"

          ]

        }

      ]

    },

1. 项目结构

**### structure**

> ├─ .electron 打包和开发环境构建脚本和配置文件

> ├─ build 打包路径

> ├─ dist 源码打包路径

> ├─ static 整个项目静态资源

> ├─ src

> ├──  resource index.html 渲染页面共有资源文件

> ├──  main 开发和生产环境 入口文件

> ├──  renders 渲染页面资源

> ├──  renderer vue页面根路径

> ├────  assets 资源

> ├────  comnon 渲染进程交互主进程

> ├────  components

> ├────  plugins

> ├────  request

> ├────  router

> ├────  store

> ├────  views

> ├─ test 测试

> ├─ package.json

> ├─ config-sys.js 系统配置

> ├─ eslint.js eslint配置

1. 开发

2020/5/25 – 2020/5/28

聊天弹窗 组件通信

打开窗口视频，背景黑色

右侧可拉伸区域

左上角菜单隐藏，滚动条

Q:

1、进程通信统一进行管理✓

包含渲染进程之间，主进程和渲染进程之间，统一抽取公用类

Src/main/ipcmain/index.js 主进程事件

Static/resource/js/ipcrender.js 渲染进程事件

2、渲染窗口引入html的方式

双向绑定的框架例如vue.js web页面引入使用模板方式

Vue使用webpack打包过之后再loadfile

纯静态的页面 jquery html

3、渲染进程之间文件复用 🗶

统一放在static文件夹下，主渲染窗口可以调用其他的static下的渲染页面文件，但是static下不能复用src/renderer下的文件，如何解决

4、存储问题

使用node数据库，还是利用h5的存储

2020/5/29

1、渲染延迟问题，✓ 右侧边框扩展问题 🗶

2、关闭到缩略图中✓

3、版本信息展示，下载包，（更新）

Q:

1. 进程文件之间的组织管理

借用webpack 别名配置和node path引用resource

1. 共有的资源之间的引用

同上

3、项目文件结构

一个进程对应一个入口文件

.electron-vue

|-- config.js渲染进程对应文件 key: value

Src/main -> 主进程

|-- Index.js 主进程管理操作系统相关内容 用global变量挂载

Src/Renderer -> 主渲染进程

Src/pages/talk -> 渲染进程talk src/talk/index.html

Src/pages/users -> 渲染进程 user src/talk/index.html

Src/resource -> 渲染进程共有资源，共有样式，icon图标，js文件

Src/static -> 静态资源

主renderer配置不动，另外起一个webpack进程，生成多页面入口文件，一个入口文件对应一个渲染窗口

窗口的文件全部在src/pages文件夹下

1. 静态资源引用问题

模板文html件static资源引用，配置webpack模板变量

1. new HtmlWebpackPlugin({
2. filename: config.name,
3. template: config.entry,
4. base: process.env.NODE\_ENV !== 'production' ? '/' : '../',
5. chunks: ['manifest', 'vendor', config.key], // 指定生成的模板文件引入的入口文件
6. minify: {
7. collapseWhitespace: true,
8. removeAttributeQuotes: true,
9. removeComments: true
10. },
11. nodeModules: process.env.NODE\_ENV !== 'production'
12. ? path.resolve(\_\_dirname, '../node\_modules')
13. : false
14. }),

主进程 static资源引用，设置node全局变量

渲染进程可用 require(‘electron’).remote.getGlobal()

/\*\*

 \* Set `\_\_static` path to static files in production

 \* https://simulatedgreg.gitbooks.io/electron-vue/content/en/using-static-assets.html

 \*/

if (process.env.NODE\_ENV !== 'development') {

  global.\_\_static = require('path').join(\_\_dirname, '/static').replace(/\\/g, '\\\\')

}

渲染进程开发环境、生产环境 \_\_static设置，相当于设置了window.\_\_static全局变量

<script>

  if (process.env.NODE\_ENV !== 'development') window.\_\_static = require('path').join(\_\_dirname, '../static').replace(

    /\\/g, '\\\\')

</script>

if (process.env.NODE\_ENV !== 'production') {

  rendersConfig.plugins.push(

    new webpack.DefinePlugin({

      '\_\_static': `"${path.join(\_\_dirname, '../static').replace(/\\/g, '\\\\')}"`,

    })

  )

}