黑客派

vue cli 3 项目与 electron 记录

作者：lizhongyue248
原文链接：https://hacpai.com/article/1547556984481
来源网站：黑客派

<img src="https://static.hacpai.com/images/img-loading.svg" alt="" data-src="https://ig.hacpai.com/bing/20180924.jpg?imageView2/1/w/960/h/520/interlace/1/q/100">
最近在重构我们上学期参加比赛做的一个小软件，help teacher 主要是作业收取的一款小工具，局域网内进行作业传输，简单用ert.x 做了一个登录注册以及数据备份与同步。为了参加比赛写的小软件，时间紧迫代码质量普遍不高并且使用 electron-vue 进行建，也发现了不少问题，而那时候第一次接触 vue，也不理解其组件化思想，也不会组件抽离什么的然后这次放假了，有老师对我们项目感兴趣，我们开始重构我们的软件，不再使用 electron-vue ，方面因为 vue cli 3 生成的项目结构十分让我喜欢，约定大于配置也类似于 Spring boot 的思想，所我不需要太多的配置文件，一个就够，这是我非常喜欢的，但是目前 electron-vue 还不支持，而且 ve cli 3 拥有自己的图形化界面，可以更加方便的管理项目依赖以及插件；另一方面，我们以前的项目用的 vue + element ui 进行开发，也发现很多 element 的组件在 electron-vue 中有着 bug 的存在比如很简单的一个 MessageBox 弹框，使用的时候都会报错，然后造成整个软件其它部分的瘫痪，以后面我们急冲冲的引入了其他的弹框组件。这次重构是对上次的一个功能上的完善，也是一个性能优化。不过自己虽然是一个后端，自己来做前端的时候也发现一些异曲同工之处，而且因为以前使用ert.x ，在接触 node 以后发现了很多地方的相似之处。今天这篇文章，主要是记录一下这个项目开的重构过程，所以不会是一种教程似的画风，大多都是一些问题上的处理方法进行记录。得不说的是，作为一个非专业的前端人员，在没有学过 webpack 、es6 语法、babel 等等一些前端面的知识就直接上手开发，还是遇到很多问题的。
<blockquote>
项目地址：<a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fgithub.co%2FGeneralAndKing%2Fteacher-tools" target="_blank" rel="nofollow ugc">teacher-tools</>
</blockquote>
<h2 id="一些说明">一些说明</h2>

为了统一代码风格，使用 eslint，并且统一使用 vscode 进行开发
element 换为 iview， 同时为了方便修改 iview 主题，使用 less 替换原来的 stylus
从 npm 改为使用 yarn
前端数据库依旧使用 nedb（暂定）
为了实时聊天，学习 socketio（暂定）

<h2 id="问题一--项目整合">问题一： 项目整合</h2>
我们在使用 vue cli 3 初始化项目后，他是一个纯正的 Web 项目，我们需要对他进行 electron 行整合，我们寻找了很久方案，最后终于找到一款插件进行整合 —— vue-cli-plugin-electron-builder , 安装很方便，接图形化界面搜索安装即可
<img src="https://static.hacpai.com/images/img-loading.svg" alt="搜索安装" data-src="https://resources.echocow.cn//file/2019/01/cf3427538b7d40b2a7ebd832cc082c1f_image.png">
或者命令行模式安装
<pre><code class="highlight-chroma">vue add electron-builder
</code></pre>
然后通过
<pre><code class="highlight-chroma">yarn electron:serve

yarn electron:build

</code></pre>

进行使用。不过使用新的东西随之而来的问题就是，遇到什么问题或者其它需求的时候，只能自

摸索，而且没啥基础很难受。
<h2 id="1--默认入口">1. 默认入口</h2>
我们移动了插件生成的文件路径以及修改了文件名，那么如何保证他能够正确的使用并进入呢？们卡在这问题很久，最后在 GitHub 的 issue 上面找到答案，于是修改 vue.config.js 如下：
<pre><code class="highlight-chroma">pluginOptions: {
  electronBuilder: {
    builderOptions: {
      files: [
        {
          'filter': ['**/*']
        }
      ],
      extraFiles: ['./extensions/'],
      asar: false
    },
    mainProcessFile: 'src/main/main.js',
    mainProcessWatch: ['src/main'],
    // [1.0.0-rc.4+] Provide a list of arguments that Electron will be launched with during "elect on:serve",
    // which can be accessed from the main process (src/background.js).
    // Note that it is ignored when --debug flag is used with "electron:serve", as you must laun h Electron yourself
    mainProcessArgs: []
  }
},
</code></pre>
<h2 id="2--打包目录">2. 打包目录</h2>
与以往不同的是，我们有多个页面，除了 electron 的，也有 Web 的，需要更具不同的进行打，所以我们查找资料，修改 pages 参数如下：
<pre><code class="highlight-chroma">...
pages: getPages(),
...
/**
* 根据 MODE 选择对应的 pages 进行打包
*/
function getPages () {
  return process.env.MODE === 'web'
  ? {
    client: {
      // entry for the page
      entry: 'src/client/main.js',
      // the source template
      template: 'public/index.html',
      // output as dist/index.html
      filename: 'index.html'
    }
  }
  : {
    app: {
      entry: 'src/app/main.js',
      template: 'public/index.html',
      filename: 'app.html'

```
    },
    communication: {
      entry: 'src/communication/main.js',
      template: 'public/index.html',
      filename: 'communication.html'
    },
    forms: {
      entry: 'src/forms/main.js',
      template: 'public/index.html',
      filename: 'forms.html'
    }
  }
}
</code></pre>
<h2 id="3--全局引入">3. 全局引入</h2>
<p>参照了官网的例子</p>
<pre><code class="highlight-chroma">...
chainWebpack: config =&gt; {
  config.resolve.alias
  .set('app@', resolve('src/app'))
  .set('_n', resolve('node_modules'))
  .set('common@', resolve('src/common/'))
  .set('communication@', resolve('src/communication/'))
  .set('form@', resolve('src/form/'))
  .set('client@', resolve('src/client/'))
  const types = ['vue-modules', 'vue', 'normal-modules', 'normal']
  types.forEach(type =&gt; addStyleResource(config.module.rule('less').oneOf(type)))
}
...

/**
● 全局 less 引入
● @param {*} rule 传递规则
*/
function addStyleResource (rule) {
rule.use('style-resource')
.loader('style-resources-loader')
.options({
patterns: [
path.resolve(__dirname, './src/common/theme/iview-variables.less')
]
})
}
</code></pre>

<p>不过对于 stylus 还有另外一种方式，不知道为啥 less 不行</p>
<pre><code class="highlight-chroma">css: {
```

```
  loaderOptions: {
    stylus: {
      import: path.resolve(__dirname, './src/styles/global.styl')
    }
  }
}
</code></pre>
<h2 id="问题二-vue">问题二：vue</h2>
<h2 id="1--滚动条问题">1. 滚动条问题</h2>
<p>由于我们自定义了标题，没有使用默认的标题栏，然后就会有一种情况，他的滚动条会在标题右了</p>
<p><img src="https://static.hacpai.com/images/img-loading.svg" alt="滚动条" data-src="https://resources.echocow.cn//file/2019/01/dd4e900c3f174729917af0077e547818_1.jpg"></p>
<p>通过 CSS 修改如下：</p>
<pre><code class="highlight-chroma">@gak-no-visible: rgba(0, 0, 0, 0);
html, body {
  overflow: hidden;
  height: 100%;
}
.gak-bg-no-visible {
  background-color: @gak-no-visible;
}
.gak-scroll {
  height: 100%;
  -webkit-overflow-scrolling: touch;
  overflow-y: auto;
  /*定义滚动条高宽及背景 高宽分别对应横竖滚动条的尺寸*/
  &amp;::-webkit-scrollbar {
    width: 8px;
    height: 4px;
    cursor: pointer;
    .gak-bg-no-visible;
  }
  /*定义滚动条轨道 内阴影+圆角*/
  &amp;::-webkit-scrollbar-track {
    border: none;
    .gak-bg-no-visible;
  }
  /*定义滑块 内阴影+圆角*/
  &amp;::-webkit-scrollbar-thumb{
    border-radius: 10px;
    background-color: rgba(110, 110, 110, 0.2);
  }
}
</code></pre>
<p>修改后如图</p>
<p><img src="https://static.hacpai.com/images/img-loading.svg" alt="滚动条" data-src="https://resources.echocow.cn//file/2019/01/7d17cd6d68644ab6954d9c0d44237935_TIM2019011202528.jpg"></p>
<h2 id="2--soketio">2. soketio</h2>
<p>找了很多组件，原本一开始使用的是 <a href="https://link.hacpai.com/forward?goto=https3A%2F%2Fgithub.com%2FMetinSeylan%2FVue-Socket.io" target="_blank" rel="nofollow ugc">Vue-Socket.io</a>，但是发现对于 vuex 他似乎已经不再支持，所以找了半天，改成使用了 <a hrf="https://link.hacpai.com/forward?goto=https%3A%2F%2Fgithub.com%2Fprobil%2Fvue-soc
```

et.io-extended" target="_blank" rel="nofollow ugc">vue-socket.io-extended，目前用起十分方便。
使用方式采用外部文件扩展的方式，能够在 action 里面调用是及其方便的
<h2 id="3--打包问题">3. 打包问题</h2>
因为我们包含的 vue 页面比较多，并且有一个不是 electron 项目而是 Web 项目，需要指定不的打包路径，最后查阅资料，使用官方文档中的方法如下：
<pre><code class="highlight-chroma">"scripts": {
  "lint": "vue-cli-service lint",
  "electron:build": "vue-cli-service electron:build",
  "electron:serve": "vue-cli-service electron:serve",
  "postinstall": "electron-builder install-app-deps",
  "test:unit": "vue-cli-service test:unit",
  "web:serve": "cross-env MODE=web vue-cli-service serve",
  "web:build": "cross-env MODE=web vue-cli-service build --dest ./extensions/dist"
},
</code></pre>
<blockquote>
 为什么不用 -mode 模式 而是使用 cross-env MODE=web ？因为我们里 -mode 读取不到，可能使用方式不对，所以采用以前的老办法了。
</blockquote>
<h2 id="待解决的问题">待解决的问题</h2>

 <li class="vditor-task"><input disabled type="checkbox"> <code>作业收取</code> 和 <co e>屏幕分享</code>，如果单用 node 的话是完成不了的（或者我们不会），希望用 c++ 来完成，我队友负责
 <li class="vditor-task"><input disabled type="checkbox"> 内存问题，上一次项目出现了一个型情况就是内存占用比预期中的高，并且出现卡顿情况
 <li class="vditor-task"><input disabled type="checkbox"> 多页面情况 vuex 状态不共享
 <li class="vditor-task"><input disabled type="checkbox"> 开发时加载出现 首屏白屏情况，打后不明显
 <li class="vditor-task"><input disabled type="checkbox"> vue devtools 不管用，多次尝试各组件都上了还是不行（要不就是临时的），最后使用本地的，但是只有第一次运行有效，后面都是无
 <li class="vditor-task"><input disabled type="checkbox"> ...很多杂七杂八的小问题

<img src="https://static.hacpai.com/images/img-loading.svg" alt="imagepng" data-src= https://resources.echocow.cn//file/2019/01/70abbb3b474341849b67c77fba3cb906_image.pn ">
C++ 很强
<h2 id="小总结">小总结</h2>
其实是非常喜欢 electron 这种的，因为他的跨平台真的太棒，我队友操作系统是 <code>主 win ows + mint(移动硬盘)</code>，我的是 <code>主 deepin + windows(移动硬盘)</code>，正这种跨平台性能够让我们在不同平台下进行协作开发、然后重构的时候也对目录进行修改，也更好的行组织以及抽取多页面的公共部分，不过自己其实还是喜欢后端哈哈哈哈，自己来做前端也是因为我学校这一届实在找不到人一起来做了，只有我们两个人，做起来时间上很是费力，然后又要学车，大是队友一直在弄，做了很多东西，自己就晚上弄弄页面，希望开学能够看得到成果吧。