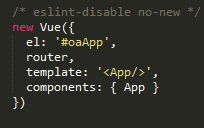
1. vuejs整个项目是单页面项目，所以index.html做为了项目入口。

Index.html中的id为oaApp的div充当着主要容器。

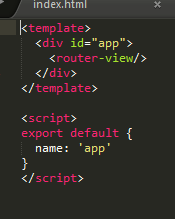
1. index.html的直接调用是main.js



这里用vue组件绑定了 #oaApp，然后用 <App /> 组件填充这个div内容。

所以，项目中所运转的所有内容，又都在 <App />这个组件容器中。所以，vuejs整个项目相当于一个大组件。

3、

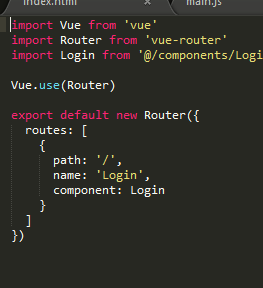


App.vue 中，定义了组件，然后其中的 <router-view />就是路由配置，意思就是这个单页面项目，将通过路由器，在单页面中展示不同的逻辑内容。

在App.vue中， export 导出

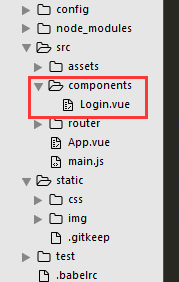
在main.js 中 import导入，这样main.js就使用上了App.vue。同理，index.html使用着main.js，这样形成了一个链调用的过程。

1. 在router文件夹下的index.js中配置着每个逻辑组件的路由，意思就是怎么跳转，对应的url跳转到要显示的哪个逻辑组件



这个是添加了的登录组件的路由，需要首页，列表页等新的需要展示的组件的时候，继续在这里面添加，上面import也要对应导入。

1. 新定义的组件模块都在components文件夹下，



这个是我们自己定义的登录模块组件。

1. **index.html中并没有引入main.js，那么他们是怎么产生关系的？或者说index.html怎么就引用上main.js了呢？**

如果你是用vue.js官网提供的脚手架工具并沿用默认配置的话，你执行npm run dev的时候会出来页面，是因为你根目录下的package.json文件里script配置了"dev": "node build/dev-server.js"，也就是其实执行的是dev-server.js这个文件，里面有定义var webpackConfig = require('./webpack.dev.conf');因为我们这个脚手架工具里是用webpack来打包项目文件的，依赖的webpack.dev.conf文件里又定义了var baseWebpackConfig = require('./webpack.base.conf');在这个依赖webpack.base.conf文件里面entry入口文件就配置了app: './src/main.js'，所以当你运行npm run dev的时候就从main.js这个入口文件开始执行了

所以，如果我们这里做了更改，是可以调用其他js文件的。

1. **main.js中的**

**Vue.config.productionTip = false**

**是什么意思？**



**具体什么意思暂时没有研究出来**