文件结构：

---- api : puer-mock，模拟后端接口。使用方法见https://github.com/ufologist/puer-mock

启动命令：puer -a api/\_mockserver.js -p 8082 (开发环境使用的是8082端口，如果需要修改，需要在webpack.dev.config.js里相应修改代理地址)；也可以使用自己熟悉的mock-server，但要注意相应修改开发环境的代理地址。

---- doc : 项目文档

---- src : 项目源文件

|------- assets : 自定义资源文件夹，css,fonts,img.script 放置区域

|-------exui : iview-theme 的less文件，可在这里直接通过less修改原有iview的主样式。

样式生成命令：iview-theme build -o dist/

然后在入口main.js里修改生成的iview.css路径。

iview-theme需要额外安装，可参考iview官网说明，因为可以引用其他UI库，所以这里就不详细写了。

|-------libs : 各类公共函数及路由、存储过程配置。

|-------template : 模板文件夹

|-------views : 页面组件文件夹,按照模块进行分类和命名，例如common/login.vue

　 |-------components：组件库

|-------modal : 弹出层组件库

|------- input-modal.vue : 表单输入组件，搭配table-grid使用

|-------search : 组合搜索组件库

|------- adv-search.vue : 报表高级搜索框，搭配table-grid使用，可用json配置输出界面

|-------tree : 树形控件组件库

|------- tree-region.vue : 地区树

　　 |------- tree-company.vue : 公司树

|-------app.vue : 入口组件，一般不要动它

|-------main.js : 入口JS，一般不要动它

|-------router.js : 路由文件，meta{requireAuth: true} 代表需要token认证

|-------vendors.js : 入口依赖库文件，最后会整合成一个文件，最后打包时尽量不要超过1M

---- index.html : 开发环境页面，不要修改! 不要修改! 不要修改!

---- index\_prod.html : 生产环境主页,自动生成，不需要修改。

---- package.json : 依赖库配置文件，一般不用修改。

---- \_mockserver.json 　: 后台接口模拟文件，可参考http://mockjs.com/examples.html编写模拟的接口数据

---- webpack.base.config.js : webpack基本设置,一般不要动它

---- webpack.dev.config.js : webpack 开发环境配置,一般不要动它;启动命令:npm run dev

---- webpack.prod.config.js : webpack 生产环境配置,一般不要动它;启动命令:npm run build

依赖库安装：

根目录下命令:

npm install -g

npm install puer -g

npm install puer-mock --save -dev

npm install --save babel-polyfill 解决ie9报错问题

需要掌握的技术：

Vue.js --------- 数据驱动的新一代JS库

Vuex --------- vue的数据状态存储插件

Axios --------- 更加强大的AJAX

Vue-router -------- vue的路由设置，可以在这里做很多钩子进行简化操作

Webpack ------- 前端打包整合工具

IviewUI(或者可以用elementUIi进行替换) ---------UI组件库

Mockjs ------- 后端接口数据模拟