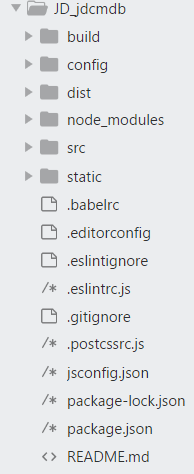
**VUE前端框架和CMS系统的开发流程规范说明文档**

**项目目录：**



**文件夹说明：**

Build: 编译测试开发等脚本文件，基本为webpack默认配置。

Config: 测试开发 && 上线环境配置文件。

Src：项目主要代码文件。

Static: 静态文件临时打包存放目录。

**开发运行：**

安装项目依赖

npm install

开启本地服务，默认为localhost: 8099

npm run dev

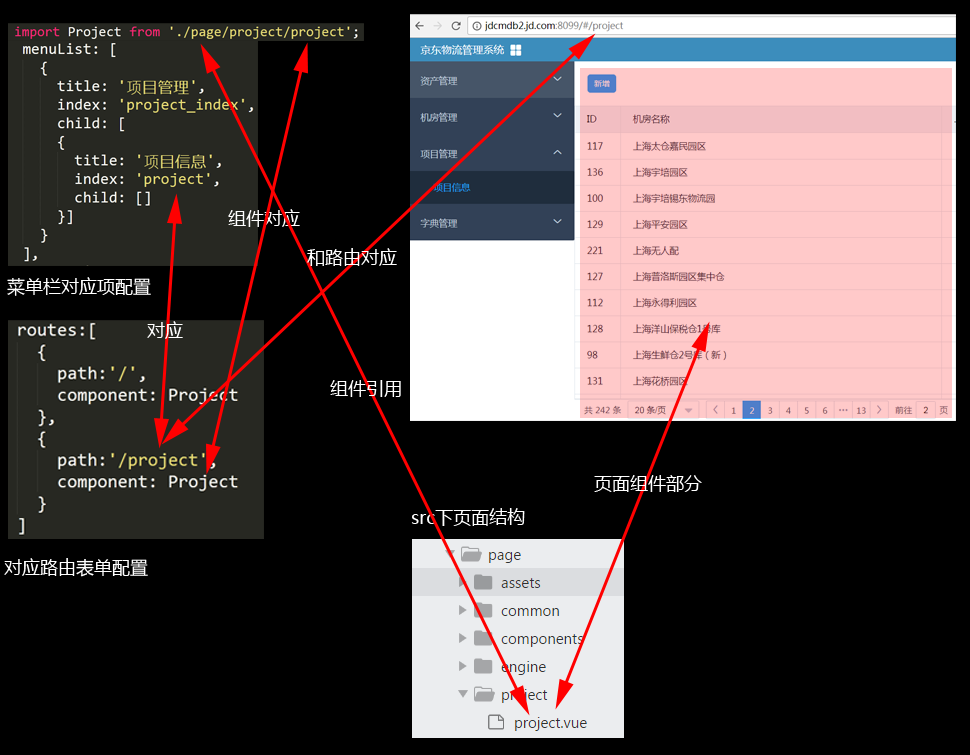
项目打包，构建生产环境

npm run build

**通用项目开发流程说明：**

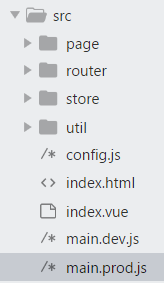
**一：路由配置：配置src/config.js文件，包括系统名称，左边菜单栏配置，路由配置等。**

菜单配置和路由配置关联示意图：



**二：单页面开发规范**

Src文件夹构成：



Page中放置所有页面级组件（在路由配置中对应关系），其中src/page/common文件夹放置页面级通用基础组件，src/page/components文件夹放置可复用性质组件。

Router实现路由配置（VUE-router），可自动生成不需要特别关注。Store是统一VUEX全局代码配置，后面会进一步介绍，util放置一些工具JS文件。

Index.vue、index.html是总体页面布局组件，main.dev.js和main.prod.js是VUE入口执行文件。其中index组件控制页面整体框架下图可以说明：



Header组件包含了用户登录逻辑，在src/config.js文件中的isTestlogin配置项可配置是否进行登录验证，如启用Header组件右上会包含用户登录信息以及相关操作。

LeftMemu组件封装了左侧菜单栏配置策略已和路由打通（src/config.js中配置），之前介绍了配置生成策略。

按照VUE开发规范，全部组件声明为单文件组件，关于这样做的优点见：

<https://cn.vuejs.org/v2/guide/single-file-components.html>

共包括三部分：



Template放置组件模板，srcipt放置组件相关VUE策略代码，style为当前组件样式，打包配置做了作用域隔离，所以可以保证单文件组件良好的封装性。方便引用和复用。

VUE 组件开发api需要熟悉见：

<https://cn.vuejs.org/v2/api/#%E9%80%89%E9%A1%B9-%E6%95%B0%E6%8D%AE>

es6语法：

<http://es6.ruanyifeng.com/>

页面中基础组件，建议使用element UI组件库，见官网：

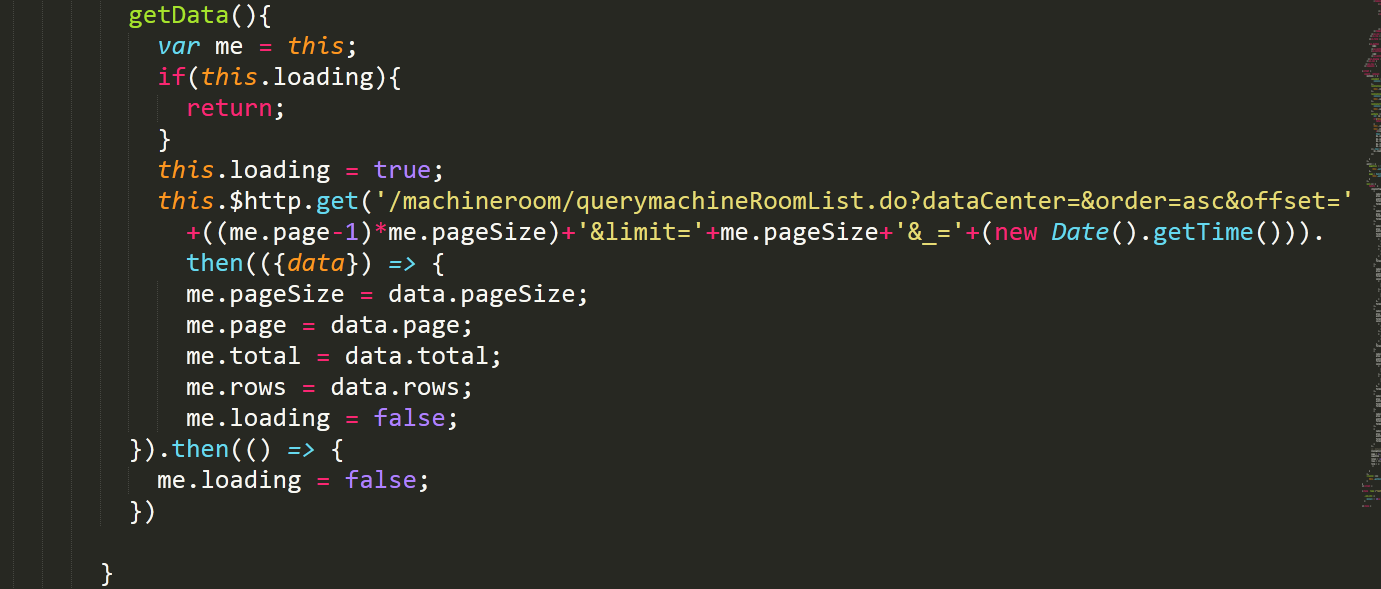
<http://element.eleme.io/#/zh-CN>

该组件库已在框架入口层面引入，全局可使用，如需使用第三方组件可自行引用安装。

**三：table子组件的进一步封装和代码策略建议**

表格类控件在报表性质页面中经常用到，这里专门针对表格开发了一个示例页面src/page/project/project.vue，其中有几点需要注意，建议采用：

1. 页面统一刷新数据方法：



pageSize为单页面信息条数，page为当前页面，total页码总数，rows表格数据，此些字段与后端数据接口对应即可，getData方法可放置在需要通过控件切换的页面字段的watch函数中。

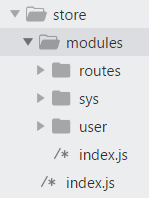
1. 页面data数据配置了showConfigElment数组，可配置和后端对应（本例为rows字段）需要展示的数据，这里和针对表格数据的添加和编辑功能做了绑定，编辑和新建弹出窗（el-dialog组件）的字段和showConfigElment数组配置字段是一一对应的。
2. 获取数据的方式在全局注册了$http对象，使用方法见：<https://segmentfault.com/a/1190000008470355>

**四：VUEX在全局的配置和使用**

关于VUEX相关配置文档：

<https://vuex.vuejs.org/zh-cn/>

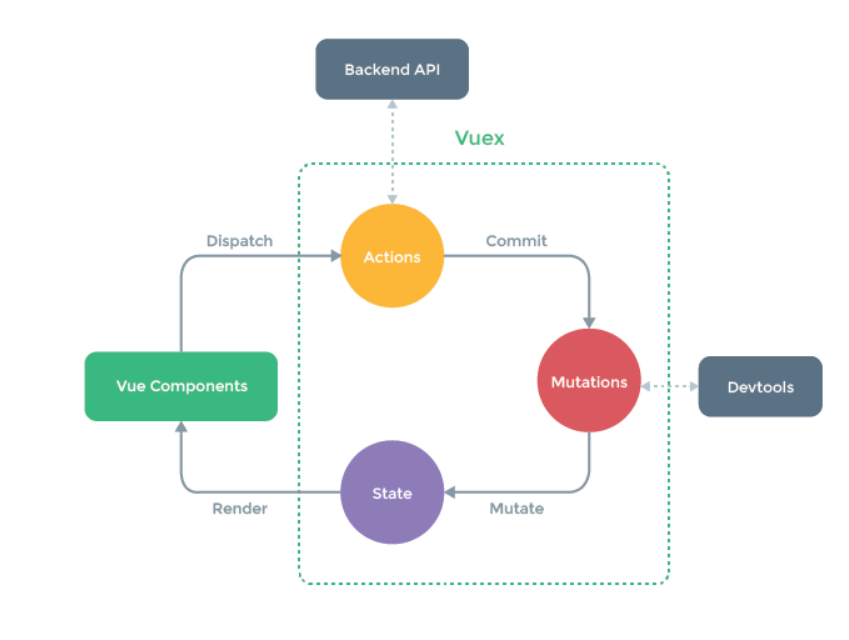
VUEX模块在src/store文件夹下，目录结构如下：



index.js注册了全局的store实例（全局唯一），modules下封装了各子系统模块，以user子模块为例管理了全局用户登录系统。可创建新子系统配置和管理系统层面的模块、业务、页面等代码间的通信。

在使用上尽量保证数据的单一流向，不要违背了VUEX设计初衷。

如user子系统，用户名信息在Header模块中绑定，用户登录Header模块会调用store/user模块中user/logout的action做统一操作，进而执行mutations，进而更新state，之后又重新绑定到Header组件状态。而用户登入操作是在router中实现（数据流策略类似），二者的通信通过store进行。



**四：开发和调试**

1. 可统一安装vue-devtools工具方便开发阶段查看vue-dom及对应的数据变化情况 ：<https://github.com/vuejs/vue-devtools>
2. 开发阶段的跨域请求配置可使用web-pack框架内建的proxyTable插件做跨域代理处理，配置在根目录下config/index.js中。本例的配置策略见下图：

