后台管理系统框架的搭建，以及开发脚手架

1、后台管理系统框架技术栈介绍： 本项目选用Vue2.0 +Vue-cli+webpack搭建；

2、插件： Vue-router2.0、Vuex、Axios、Element-UI

3、项目架构大致为： vue-cli搭建项目类似于spa单页面应用的组件化开发各个视图，

vue-router管理路由，vuex管理全局数据变量，axios于后台交互，webpack管理项目代 码

项目构建：

build：webpack相关配置

config：配置开发环境和生产环境

node\_modules： 项目依赖

src：项目主要结构文件

--api： 与后台数据交互

--assets： 静态文件，以及打包生成问价主目录

--components： 公共组件文件

--data： mock数据管理

--router： 项目路由

--view： 项目主视图区vue单文件

--vuex： 应用层数据管理

--app.vue： 项目入口文件

--main.js： 入口配置js

--static： 全局css，image，iconfont等

index.html: 项目HTML挂载

.eslintrc.js: 项目引入eslint规范

package.json： 项目依赖包管理

其余为： 相关配置文件，

路由设置：

参考文档：vue-router2.0 ——https://router.vuejs.org

主要设置为：路由嵌套，设置url重定向(redirect)等

功能文档

1、安装node项目依赖，http://nodejs.cn/download/，解压安装，配置环境变量

2、单页面项目构建：

# 全局安装vue ----> npm install --global vue

# 全局安装vue-cli ----> npm install --global vue-cli

# 创建一个基于webpack模板的新项目 ---->vue init webpack my-project (my-project为项目名称)

# 安装依赖 ----> cd my-project / npm install

# 运行项目 ----> npm run dev

说明： 以上步骤，可快速搭建并启动一个带热重载、保存时静态检查以及用于生产环境的 基于vue+vue-cli的单页面应用

3、目录结构调整：修改app.vue入口文件，在src目录下新增view文件夹，其功能为各个 主页面视图入口vue单文件

4、引入插件vue-router ----> npm install vue-router --save

src目录下新增router/index.js 编辑路由

注释： 项目入口文件app.vue

<div id="app">

<router-view></router-view>

</div>

即为路由视图入口，即路由渲染区域。

router/index.js结构：

# 引入vue、vuex、各个视图组件（view文件夹的.vue单文件）

# Vue.use(Router)

# 实例化路由，主要结构为：一级路由，二级路由...其中一级路由为控制页 面主内容区域、登录、注册、忘记密码、404页面等，二级路由均为主内 容子路由，控住导航栏对应内容显示区域，其余路由依次类推，结构如代 码所示。

注意实例化路时属性的配置：

mode：利用history.pushState API来完成URL的跳转而无需重新加载页面

linkActiveClass： 激活路由时的类名，可以为其添加相应的样式

meta： 其属性requireAuth 添加该字段，表示进入这个路由需要登录验证

Mode属性在上线时需要服务器端做相关配置

相关文档地址：<https://router.vuejs.org/zh-cn/essentials/history-mode.html>

Ngix 配置为：location / { try\_files $uri $uri/ /index.html; }

5、引入插件vuex（vue.js应用程序开发的状态管理模式），本项目主要依赖该插件管理全局应用级数据，大致为全局变量，多组件依赖变量等数据的管理；（具体例子参考当前项目的登录即可,以及官方中文文档）

结构： 在src目录下增加vuex目录，主要文件包括：

# state.js：应用层级数据的唯一数据源；简单说就是定义全局变量，变量初始化。

使用：在组件内通过计算属性获取 例： 获取count变量

count () {

return this.$store.state.count

}

修改得通过在mutations中定义操作方法，通过commit（mutation）处理

# getters.js: 获取state中的数据，做过滤处理； 例：

getters: {

doneTodos: state => {

return state.todos.filter(todo => todo.done)

}

}

# mutations.js：更改vuex的store中的状态的唯一方法是提交mutation，mutation类似于事件： 每个mutation都有一个字符串的事件类型（type）和回调函数（handler），该回调函数就是实际进行状态更改的地方，冰倩只会接受state作为第一个参数 ；可以通过types.js定于types，也可以在mutations.js里面定义事件类型，最好用大写定义。

# actions.js 类似于mutation，不同在于mutation执行同步操作，action执行异步；action提交的是mutation，而不是直接变更状态；

#modules文件夹，主要功能为当项目庞大是，对各个模块做区分，降低action，mutation的耦合度。 每个module可以是单独的state，action，mutation，state， 可以是将其中某些部分耦合到上级目录的对应的js文件内做处理。

6、引入element-ui

npm install element-ui --save

使用主要是设计文档内各个组件的使用，以及查阅组件相关属性以及绑定的方法，数据的传递等。参考文档：http://element.eleme.io/#/zh-CN/

#借助babel-plugin-component,我们可以只引入需要的组件，以达到减小项目体积的目的。

npm install babel-plugin-component --save -dev

修改.babellrc

{

"presets": [

["es2015", { "modules": false }]

],

"plugins": [["component", [

{

"libraryName": "element-ui",

"styleLibraryName": "theme-default"

}

]]]

}

参考文档：

Vue2.0：http://cn.vuejs.org/v2/guide/

Vuex：http://vuex.vuejs.org/zh-cn/intro.html

Vue-router：http://router.vuejs.org/zh-cn/index.html

Axios：https://www.oschina.net/p/axios

Element-UI: http://element.eleme.io/#/zh-CN

注意事项：

1、登录状态： # 初始化页面， 获取后台token， 判断是否处于登录状态，状态为true则 进入主页， false则跳转登录页。

# 登录： 发送用户及密码，分发异步action，action触发mutation提交commit 变化，改变state状态，同时本地存储用户信息以及登录状态标识，设置 时间控制清除本地存储信息。

# 请求拦截， 后台设置session expires实现， 前端每次请求需要发送

Cookie作为唯一标识，通过后台判断返回前端， 前端通过不同的code 值做出相应的判断拦截；

1. axios插件配置：axios作为前端model中心，处理前后端数据交互，即接口请求。主要 设置为--引入相关模块，格式化数据， 全局提示， router做拦截路由跳转等。

# 生成两个实例化方法，其一为普通GET， POST请求后台接口， headers 设为：'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8'， 其二为上传组件接口请求， headers设为'Content-Type': 'multipart/form-data; boundary=---1232'；

# 全局设置headers、timeout、请求拦截、相应拦截； 其中在请求拦截处 理请求头设置，错误信息提示灯， 相应拦截为处理与后台协议商定的请 求返回code值的拦截处理。

# 跨域处理： 通过CORS（跨域资源共享），后台设置请求头，解决问题。

由于axios插件的withCredentials默认是false，意思就是不携带cookie 信息，那就让它为true，我是全局性配置的，即在main.js设置 axios.defaults.withCredentials = true

3、 路由切换： 切换路由之前会在钩子函数render页面之前判断routers下的meta属性的 requireAuth属性，判断是否需要验证登录，获取登录状态，控制页面 路由；状态值为true 自动进入next(), 状态为false是，跳转登录。

vue-router 2.0地址：https://router.vuejs.org/zh-cn/

4、 ElementUI的使用

table组件： 使用template定义列内容scope.row.detail为继承该列的字段，通过template 遍历json数据内包括的数据

<template scope="scope">

<div v-for="item in scope.row.detail" class="detail\_list">

</div>

</template>

pagenation组件： 所传参数均使用字符串，因为element做了一次封装，所以不能使 用模板字符串双大括号语法

5、elementui upload组件：

注： 上传图片，文件等需要操作的文件，与一般的post方法的请求头content-type不一样，也就是请求数据的格式不一样，需作区别处理。

上传数据格式form-data，需使用表单提交，或者原生html5上传，需要创建formdata对象作为发送给后台的数据。上传与其他表单与后台交互好、分开操作，挂载在七牛云、腾讯云等等，通过上传成功返回数据，关联其他表单。