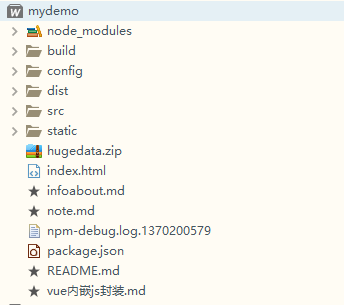
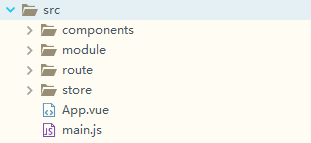
# Vue使用手册

1. 目录结构

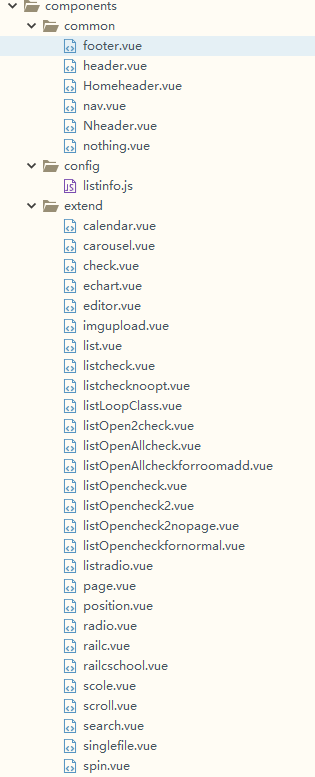


1. mydemo 为全项目目录
2. node\_modules 为环境依赖安装目录
3. build： webpack打包配置目录
4. config： webpack 配置文件目录
5. dist：打包编译文件目录即nigix代理目录
6. src：项目模块以及路由，vuex等
7. static： 静态文件配置目录
8. hugedata.zip： 大数据静态网页包
9. index.html入口文件
10. infoabout.md: js语法要求以及一些api说明
11. note.md: vue的一些用法
12. npm-dbug.log： npm打包日志
13. package.json 环境依赖配置文件
14. 其他说明文档以及用法
15. 具体使用说明
16. 1-5 8-14略过
17. Src使用说明
18. Src



* 1. Components为基础组件目录

1. 目录结构



1. Common目录
2. 存放公用头尾信息以及导航信息组件详见footer,header…
3. Config目录
4. 存放components公用配置信息目录，详见listinfo.js
5. Extend目录
6. 存放公用组件及模板目录如

* Calendar 日历组件
* Carousel.vue 幻灯组件，未使用
* Check.vue 复选组件
* Echart.vue 百度echart第三方组件
* Editor.vue wangeditor第三方富文本组件
* Imgupload.vue 图片上传组件 未使用
* List.vue 模板组件 只包含列表
* Listchecknoopt.vue 模板组件 不包含操作项
* Listloopclass.vue 模板组件 后台在线巡课
* Listopen2check.vue 模板组件 两层展开
* Listopenallcheck.vue 模板组件 全部展开（复选）
* Listopenallcheckforroomadd.vue 模板组件 后台添加教室
* Listopencheck.vue 模板组件 默认关闭
* Listopencheck2.vue 模板组件 未使用
* Listopencheck2nopage.vue 模板组件 无分页
* Listopencheckfornormal.vue 模板组件 后台常态管理
* Listradio.vue 单选组件 多行未使用
* Page.vue 分页组件
* Position.vue 面包屑导航组件
* Radio.vue 单选组件 一组
* Railc.vue 火车票查询组件，应用课表
* Railcschool.vue 火车票查询组件，应用校园电视台
* Scole.vue 评分组件
* Scroll.vue 模拟滚动条组件 未使用
* Search.vue 查询组件
* Singlefile.vue 单文件上传组件
* Spin.vue 加载中组件

1. 日历组件

* Prop参数

props: ['valshow', 'min', 'iconstyle', 'clear', 'iconclass', 'part', 'inputstyle', 'hmsshow', 'ids', 'ymdhms', 'dateval'],

1. Valshow: true/false 是否直接显示日期，如果为false,选中日历时间时需要手动赋值时间给input value
2. Min: 最小日期 与ymdhms参数保持日期格式一致
3. Iconstyle 日历图标样式style
4. Clear：true/false 是否显示清空
5. Iconclass: 日历小图标样式表名称
6. Part必填项，日历节点生成存放位置，默认document.body
7. Inputsytle 日历文本框style样式
8. Hmsshow:true/false 是否只显示时分秒，年月日正常赋值
9. Ids 日历id
10. Ymdhms: ymdhMs, ymd为必须项，可以自由组合，hMsymd…
11. dateval 默认日期，与ymdhms数据格式保持一致，没有保留为空
12. 复选组件

* Prop参数

props: ['info', 'selectarr', 'styles']

1. Info 复选数组信息[{value: ‘’, txt: ‘’}]
2. Selectarr 已勾选数组索引[]
3. Styles check复选sytle
4. Echart

* Prop参数

props: ['styles', 'echartid', 'echartdata', 'title']

1. Styles 图表style
2. Echartid 图表id
3. Echartdata echart option 内容
4. Title echart 标题
5. wangeditor第三方富文本组件

* Prop参数

props: ['styles']

1. Styles 编辑器style
2. List.vue 模板组件 只包含列表(listopen,listcheck开头的模板组件参考此例)

* Prop参数

props: ['bingos', 'totalnum', 'pageinfo', 'pageval', 'pagesize', 'pagestyles', 'pathinfo', 'jumpinfo', 'info', 'percentinfo', 'colums', 'head', 'optevent'],

1. Bingos 分页选中数组[‘selected’, ‘’]
2. Totalnum 数据总条数
3. Pageinfo 总页码
4. Pageval 当前页码
5. Pasesize 每页条数
6. Pagestyles 分页style样式
7. Pathinfo 路由公用信息{'prex': 'AJportal', 'optname': '门户'}
8. Jumpinfo 路由公用信息{'prex': 'AJportal', 'optname': '门户'}
9. 'info' 数据数组
10. Percentinfo 数据项所占百分比['wid10', 'wid80', 'wid10']
11. Colums 数据项接口配置项['check', 'name', 'code', 'des', 'opt']
12. Head 数据项头部信息['check', '名称', '代码', '描述', '操作']
13. Optevent 数据操作项数据['editlist', 'dellist']，该项可以在config目录下的具体文件里面配置，也可以根据权限在接口里面配置，区别在于接口文件数据项里面是否包含optevent
14. Page.vue 分页组件

* Prop参数

props: ['pagestyles', 'bingo', 'pagesize', 'pageindex', 'totalpage', 'totalnum']

1. Pagesytles 分页数据
2. Bingo 当前选中分页样式
3. Pagesize 每页多少条数据
4. Pageindex 当前页码
5. Totalpage 总页数
6. Totalnum 总条数
7. 页码规则： 小于十页，全部展示，大于十页显示前三页，后三页，位于首页时，首页和上一页置灰，不可用，同理，最后一页的时候，下一页和末页置灰不可用
8. Position.vue 面包屑导航组件

* 从路由记录集获取当前路由信息，匹配屑导航

1. Radio.vue 单选组件 一组

* Prop参数

props: ['radioinfo', 'genderval', 'type']

1. Radioinfo 单选信息（详情api查看js api组件）

radioinfocome = {

classname: 'radiofather come',

val: [0, 1],

title: ['否', '是'],

opt: {

father: '.come span',

init: {

val: true,

index: 0

},

inputflag: true

}

}

1. 'genderval' 默认勾选项索引值
2. Type add/edit 添加编辑选项
3. Railc.vue 火车票查询组件，应用课表

* Prop参数

props: ['railid', 'titleinfo', 'dateval', 'resetflag', 'days', 'lostday', 'show']

1. Railid id
2. Titleinfo 课表左上角的展示信息
3. Dateval ymd 年月日信息
4. Resetflag --
5. Days: 展示几天的信息，默认7
6. Lostday 是否从周一开始
7. Show: true/false 是否显示titleinfo,上周，下周按钮
8. Railcschool.vue 火车票查询组件

* Prop参数

props: ['railid', 'dateval', 'railpart', 'resetflag', 'days', 'topclass', 'lostday', 'show']

1. Railid id
2. railpart 日历组件样式表名称
3. Dateval ymd 年月日信息
4. Resetflag --
5. Topclass：父组件样式表名称
6. Days: 展示几天的信息，默认7
7. Lostday 是否从周一开始
8. Show: true/false 是否显示titleinfo,上周，下周按钮
9. Scole.vue 评分组件（具体参考js api文档）

* Props 参数

props: ['items', 'starid', 'disabled', 'reset', 'resetupdate', 'scoletype', 'type', 'ind', 'bool', 'totalwid', 'starnum', 'inputstyle', 'wid', 'backgroundsize', 'val', 'totalscole'],

1. Items
2. Starid id
3. Disabled 是否禁用input输入
4. Reset –
5. Resetupdate true/false 数据更新，重新实例化组件
6. Scoletype ‘init’ 回调取整分值
7. Type 评分类型支持半星，整星，自由星
8. Ind 索引号
9. Bool 是否禁用评分
10. Totalwid 总长度
11. Sternum 几颗星
12. Inputstyle input style
13. Wid 初始宽度
14. Backgroundsize 背景
15. Val 初始分值
16. Totalscole 总分值
17. Search.vue 查询组件

* Props 参数

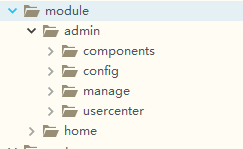
props: ['Gsearchstyle', 'styles', 'searchbtnstyles'],

1. Gsearchstyle 父组件样式style
2. Styles input style
3. Searchbtnstyles 查询按钮样式
4. Spin.vue 加载中组件

* Props 参数

props: ['spinshow']

1. Spinshow true/false 是否显示
   1. Module为前后台模块目录
2. 模块目录结构

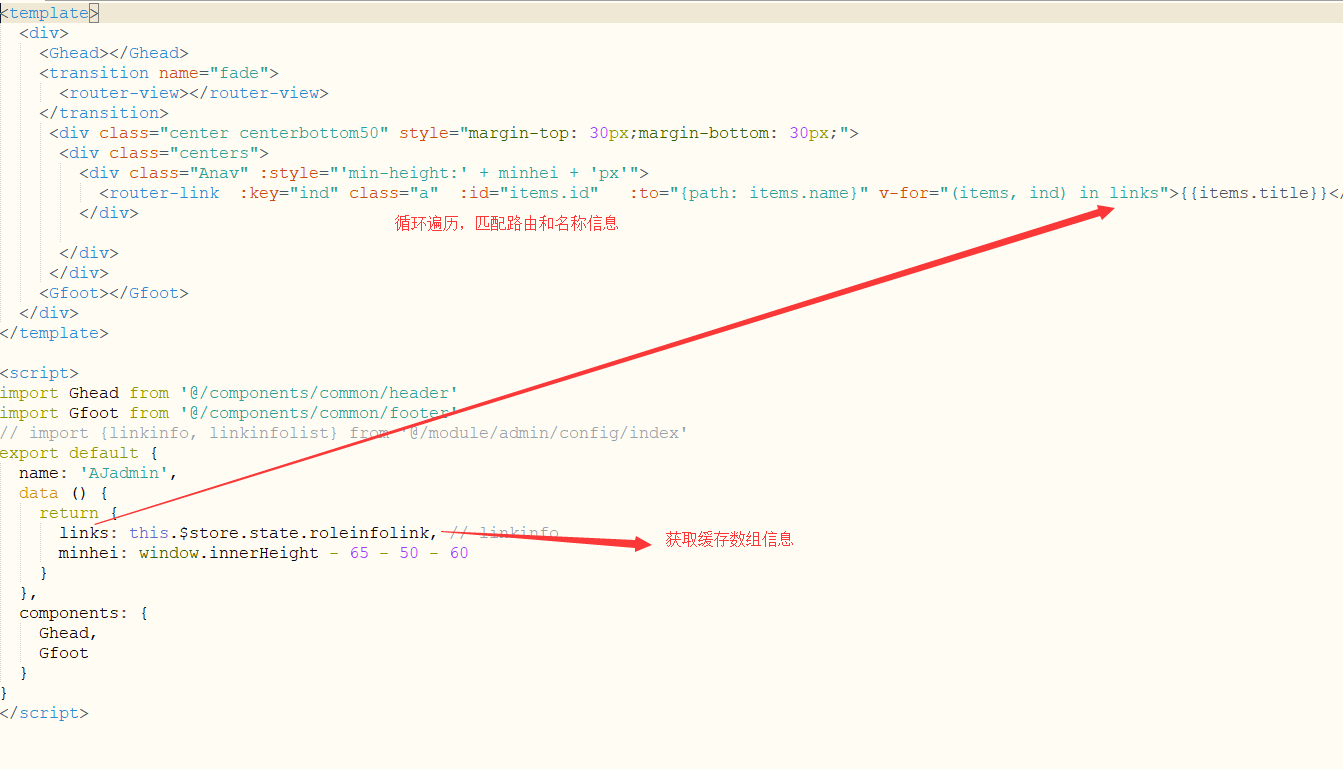


1. Admin为后台目录
2. Components为后台所有单文件目录

* 文件目录

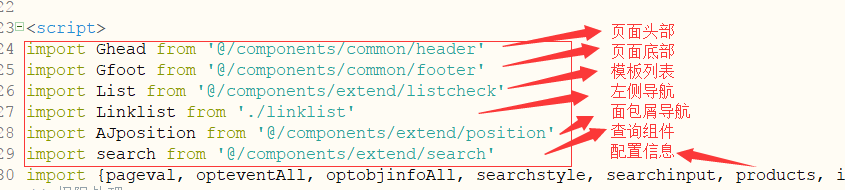
1. Add结尾的vue文件为添加
2. Edit结尾的vue文件为修改
3. Del结尾的vue文件为删除
4. 正常结尾的vue文件一版为列表文件

* Link.vue 登录后台后，默认进入界面，从vuex缓存获取所有模块，展示所有有权限的模块



* Term.vue 学期列表案例

1. Script部分说明



配置信息引入的路径为config目录下对应的js文件eg:

import {pageval, opteventAll, optobjinfoAll, searchstyle, searchinput, products, infoval, OeN, OdN, OPelist, OPdlist, bingos, colums, percentinfo, pagesize, listurl, listhead, pathinfo} from '@/module/admin/config/terminfo'

注释： @符号代表src目录，详见webpack.base.conf.js配置文件

1. **Vue单文件写法**(如不了解，具体请参考vue官方文档，了解vue生命周期，组件，事件绑定，传值， 路由， vuex等vue相关知识)

export default {

name: 'AJterm',

data () {

return {

products: products

}

}

}

1. **data 解释**

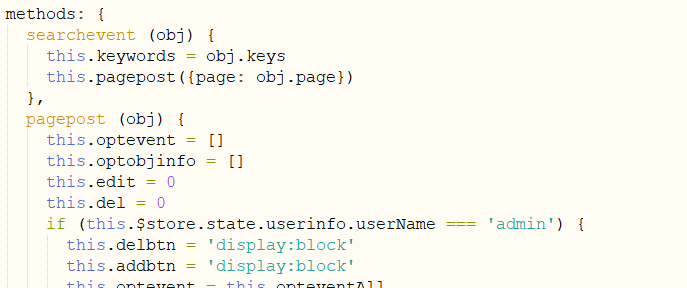


1. **created** 数据准备齐全后，执行接口调用或者其他操作



eg: 调用学期数据方法

1. **methods 方法集合，该方法定义界面所有使用的方法，包含点击事件，post请求，弹窗等方法，如上图pagepost**

****

1. **pagepost列表获取及权限分配**
2. 权限分配



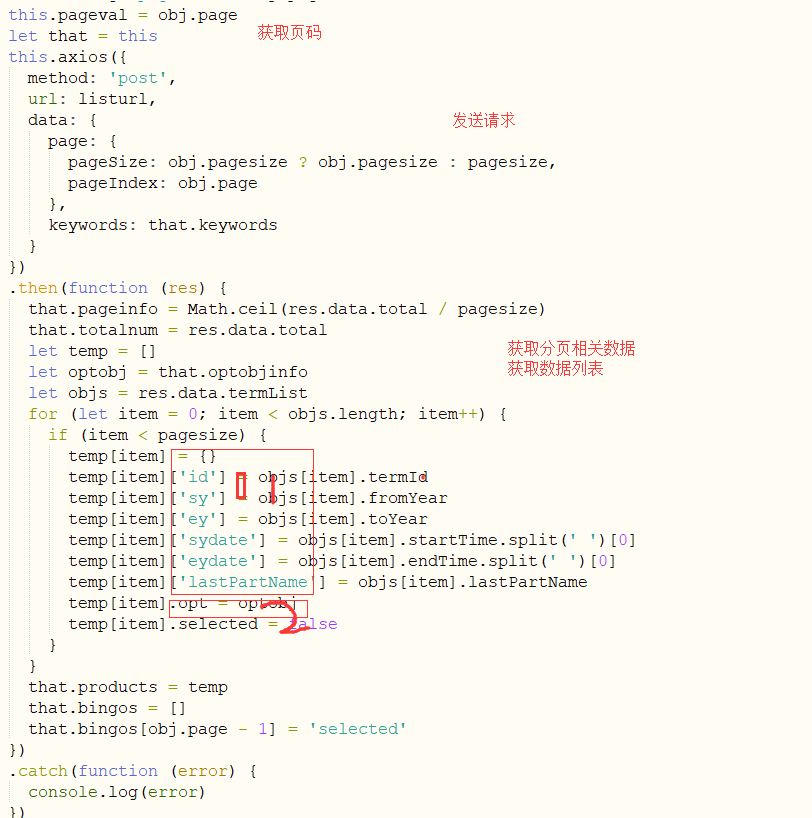
* 权限区分角色，超管和普通用户
* 如果是超管，可以定制角色权限数组，如果不是，从vuex缓存获取权限配置数组，进行权限分配
* 前四行为每次请求数据，初始化权限分配数组或者计数器；

第五行区分超管和普通用户

第6-9行为超管权限配置

第11行以后为普通用户权限分配的过程，optevent数组为权限按钮的事件匹配名称，optobjinfo数组为权限按钮的名称

* 数据列表获取



1标识的部分为配置信息(terminfo.js)里面的colums信息

export const colums = ['check', 'sy', 'ey', 'lastPartName', 'sydate', 'eydate', 'opt']

2标识的部分为配置信息(terminfo.js)里面的opteven

export const opteventAll = ['editlist', 'dellist']

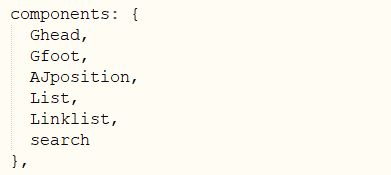
bingo为分页选中当前页的数据

totalnum为分页总条数

pageinfo为总页码

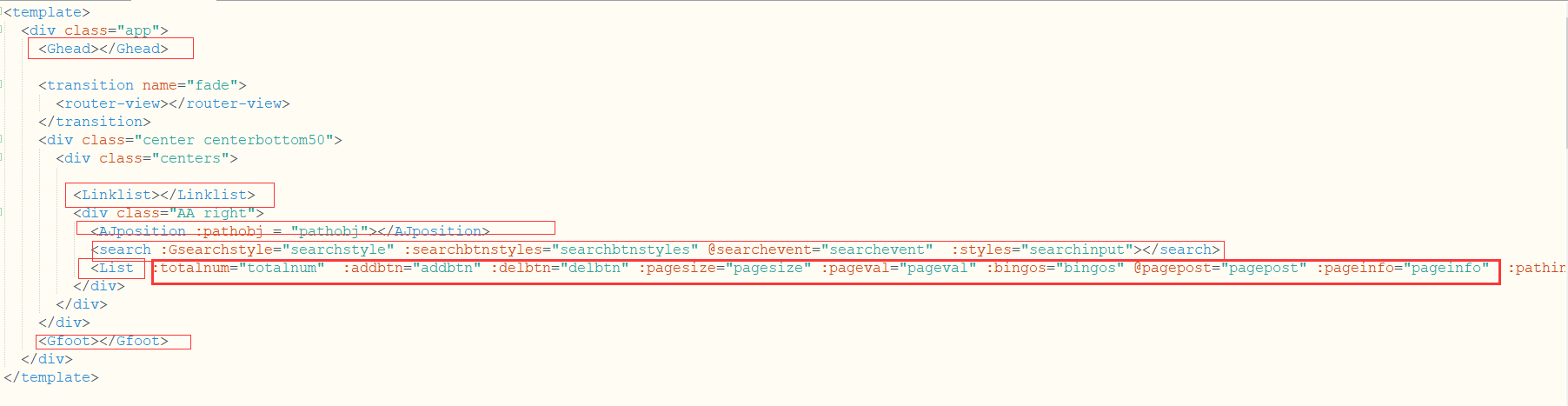
pageval为当前分页

1. **components部分**

****

**通过组件注册，将引入的组件注入到模板当中**

1. **template部分**

****

* 上图中细线的部分，为7中注册的组件
* List组件为就是上面我面讲到的模板，通过：xx=””的形式，将分页信息，数据信息通过组件传递到list模板里面
* 其中@pagepost为自定义时间，用于点击list模板的分页后，向父组件（term.vue）传递页码参数，进行数据的重新获取

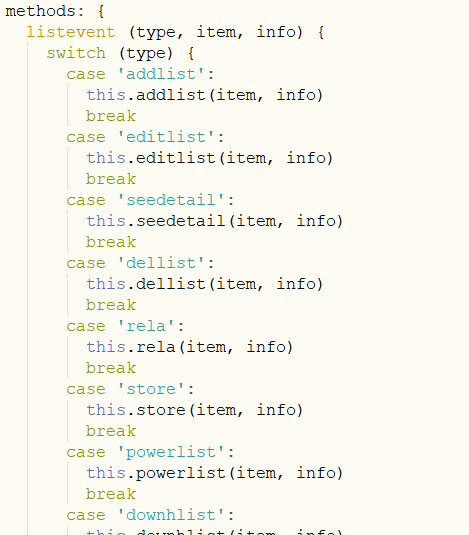
1. **List模板部分**

* **上面我们讲解了模板**List.vue 模板组件的props的各项含义，8中通过：xx=””的形式传递的参数，其实就是listcheck.vue模板相中的props参数，这样，我们就可以在listcheck.vue模板中获取对应的数据，进行操作

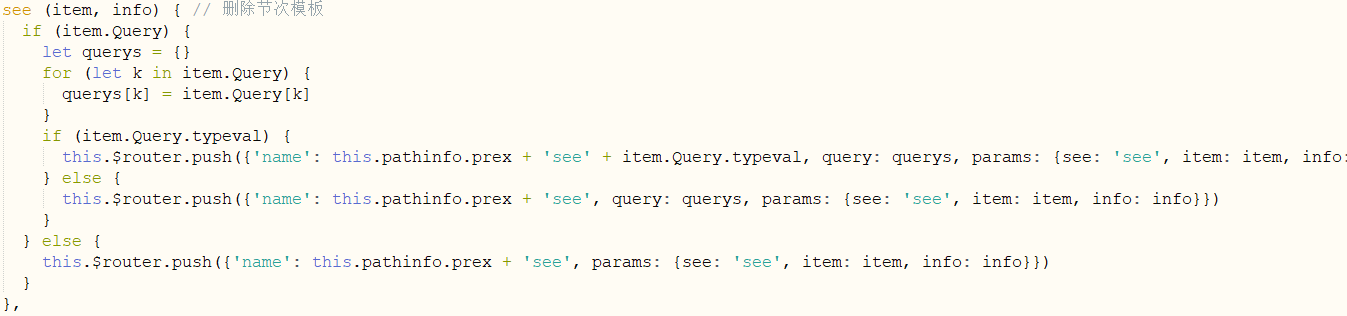
1. **Listcheck模板**

****

* 上图1标识的部分，通过props传递的delbtn属性，控制该按钮是否显示
* 上图2标识的部分为接口数据数组，进行遍历循环，添加列表数据
* 上图3标识的部分为通过判断接口是否包含opt字段与props传递的colums属性做比对，区分是否是操作项，通过判断check并比对，判断是否是复选项以及其他选项
* 上图3部分@Listevent就是通过传递接口optevent数组（也就是terminfo.js里面的['editlist', 'dellist']权限区分项，区分不同的事件，并响应不同的事件）我们来看listcheck的methods事件集合
* Methods集合



* 如上图，通过listevent传参，匹配不同的的方法调用，拿查看see为例，我们调用see方法



如上图所示，如果接口文档包含Query参数，我们会获取query参数的值，进行路由的变种匹配，并传递参数

如上图ifitem.query.typeval作为判断，需要在路由跳转的时候，匹配路由的名称，这样我们就可以进行名称的叠加匹配，传递query参数（即url传参？aa=xx这样，可以传递多个参数，query数据结构为

{aa: xx, bb: cc} == url?aa=xx&bb=cc）

如果没有传递参数，即没有Query字段，我们将自然匹配预先设定好的路由名称，这就是为什么在js的配置信息文件中配置pathinfo的原因

export const pathinfo = {'prex': 'AJterm', 'optname': '学期'} 详见terminfo.js

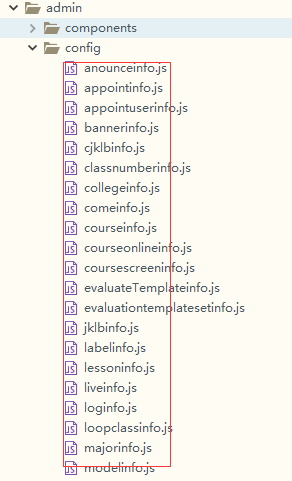
匹配出来的结果就是

this.pathinfo.prex + see/see+item.Query.typeval== AJtermsee/AJtermseelist等等，

该名称在路由匹配表里面是定义好的，所以这里的匹配跟路由表保持一致，就可以与模板进行关联，路由部分会说明此部分

1. Config为后台所有单文件配置信息目录，基本与components目录一一对应

* 目录



* 以term为例进行讲解

1. 常量说明（如果不明白export const， import{} from这种声明方式，请参考es6官方文档，了解es6相关的模块化相关<http://es6.ruanyifeng.com/#docs/>关于module的部分知识）
2. 常量说明（注意：建议（非必须）不要在模板文件保留文字信息，后期维护只需要维护js配置文件即可）



1. 上文已经降到term.vue单文件script部分，

import { OaN} from '@/module/admin/config/terminfo', 其中OaN就是这里的常量信息。Terminfo无需写js文件，webpack配置文件里面已经定义好了文件遍历的文件类型，详见webpack.base.conf.js

resolve: {

extensions: ['.js', '.vue', '.json'],

alias: {

'vue$': 'vue/dist/vue.esm.js',

'@': resolve('src')

}

}

@符号代表src目录是从这里来的

1. 其他信息，具体参考具体的配置文件
2. Manage目录为后台课程管理目录
3. 目录结构

* Config目录配置信息文件 js
* Vue单文件， 包含课程管理所有

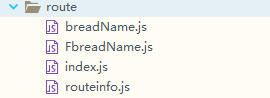
1. Usercenter为后台个人中心目录
2. 目录结构

* Config目录 配置信息文件 js
* Vue单文件，包含个人中心所有

1. Home为前台目录
2. 目录结构

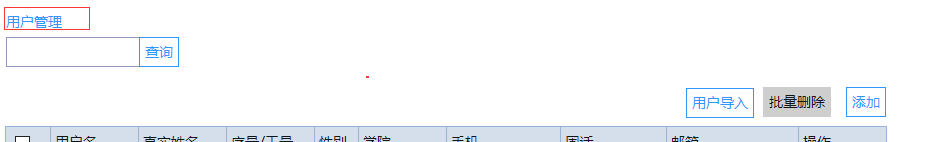
* Config目录 配置信息文件 js
* Vue单文件，包含首页，校园电视台，直播，课程
  1. Route路由配置目录

1. 路由目录



1. breadName.js 后台面包屑导航名称常量（如下图）

FbreadName.js 前台面包屑导航名称常量（前台暂无，预留数据）



1. index.js 后台路由模块的对应匹配表
2. linkinfo对象数组，对应跳转后台接口获取的模块名称，成员名必须唯一

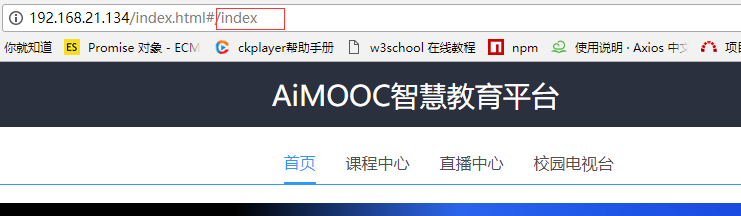
{

'name': 'user', // path 路径

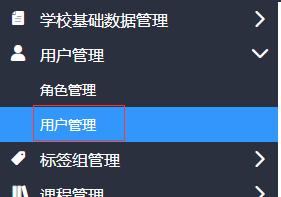
'title': '用户管理' // 路由名称

}

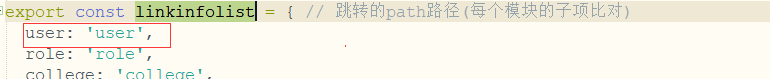
* name匹配的是每一个路由的名称即path参数（如下图）



* title为后台首页或者左侧路由列表看到的名称（如下图）

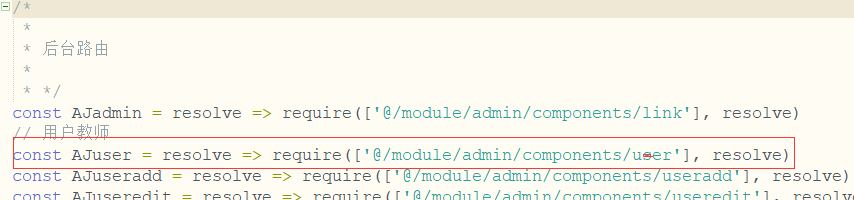


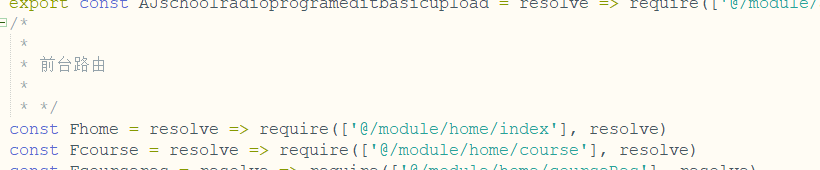
1. Linkinfolist对象为接口获取模块名称与linkinfo对象数组的对应跳转关系表



* 上图所示，user为接口获取数据， ‘user’为本地路由配置的path信息，再次进行模块与路由的重新匹配，建议保持一致

1. routeinfo.js 前后台所有路由对应关系表
2. 目录



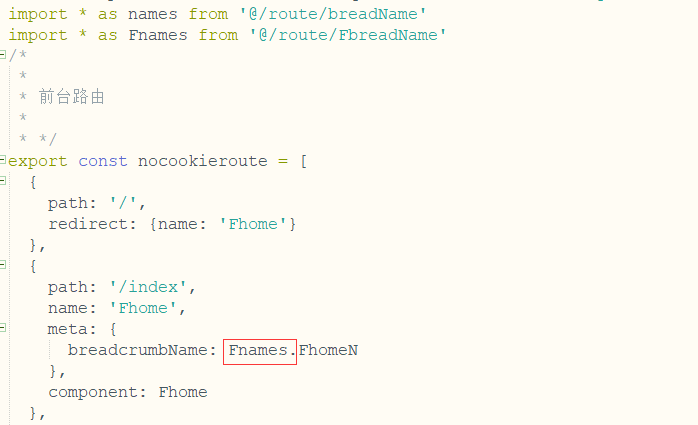


1. 这里分为前台路由表和后台路由表，后台以AJ开头，前台以F开头，以示区分。

这里我们定义了一个AJuser变量，AJuser实际代表user.vue文件，同理，AJuseradd/AJuseredit/AJuserdel各自代表不同的文件

Require部分导入vue单文件，这样就可以将路由与实际的文件模板做关联了

1. 路由配置信息



* 这里我们通过names、Fnames别名，引入面包屑导航的整个配置信息，在路由信息的配置文件里面使用
* Nocookieroute为前台路由配置信息，因为前台路由无需登录，即可查看

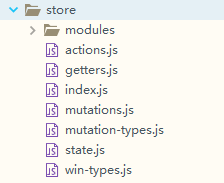


需要注意前台路由最后一个配置项，如果所有的路由匹配不到，我们就会设置自动跳转的首页，后者404界面等

* 后台路由配置



1. Children为子路由配置，配置规则与父路由一致
2. 需要注意，子路由的path不需要/开头
   1. Store vuex缓存变量目录
3. 目录



1. Modules为多模块时缓存变量的目录（暂未使用）
2. Actions.js 为全局变量触发传递文件
3. Getters.js 为Vuex 允许我们在 store 中定义“getter”（可以认为是 store 的计算属性）。就像计算属性一样，getter 的返回值会根据它的依赖被缓存起来，且只有当它的依赖值发生了改变才会被重新计算
4. Mutations.js action时间触发后变量的接受着，用于赋值变量给store
5. Mutation-typs.js 统一变量定义文件
6. State.js store全局变量定义文件
7. Win-types.js 弹窗配置信息文件
8. Index.js 组件注册文件以及vuex实例化导出文件

* 目录



* Vuex如果不了解，请查看vuex官方文档了解（<https://vuex.vuejs.org/zh-cn/intro.html>）
* 自定义组件及用法请查看每个组件里面的readme.md文件，里面有详细说明

自定义组件为es6编码，babelcli转码，uglifyjs压缩编译 –m-c-o,去除注释，变量替换.

如不了解babel,babelcli请参考官方文档

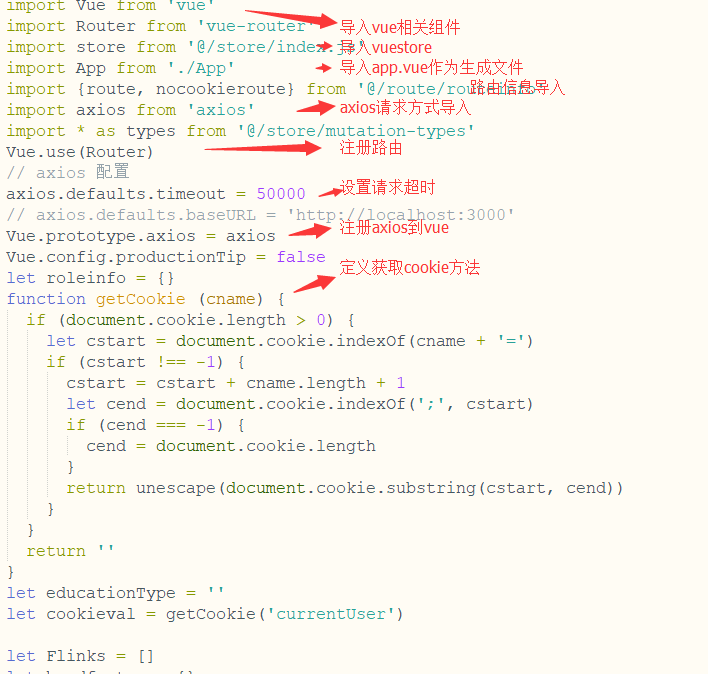
<http://www.ruanyifeng.com/blog/2016/01/babel.html>

如不了解uglifyjs,请参考官方文档

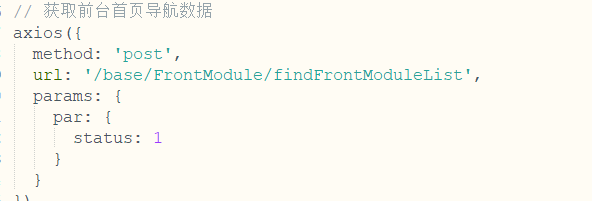
<http://blog.fens.me/nodejs-uglifyjs2-js/>

* 1. App.vue 入口文件
  2. Main.js 入口文件 详见build/webpack.base.conf.js

1. 目录



1. 获取前台导航信息，因为不登录也可以看前台的页面，所以需要在vue为实例化之前，获取前台所有导航



1. 未登录情况



1. 登陆后





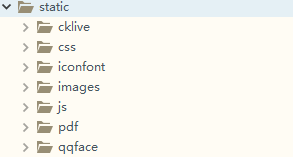
上图部分，屏蔽了部分模块的功能，也可以修改数据库进行操作

获取，模块信息后，触发store缓存方法，保存为全局变量

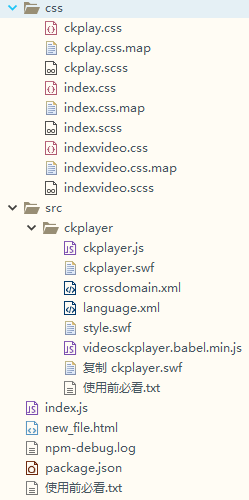
最后，重新实例化vue

2．Static目录

1. 目录



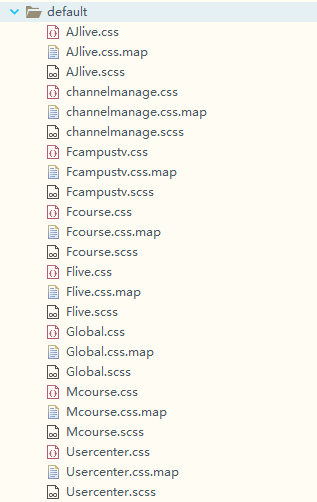
1. Cklive存放ckplay相关文件目录
2. Css样式表目录
3. Iconfont 字体目录
4. Images 图片目录
5. Js js文件目录
6. Pdf pdf文件预览目录
7. Qqface 表情包目录
8. Cklive



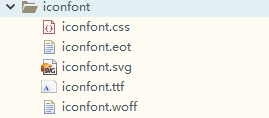
1. Css 存放本地h5播放器以及直播播放器等样式表目录
2. Css采用sass编译，如不了解，请参考sass官方文档

<https://www.sass.hk/guide/>

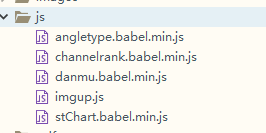
1. Src目录ckplayer目录
2. Css目录



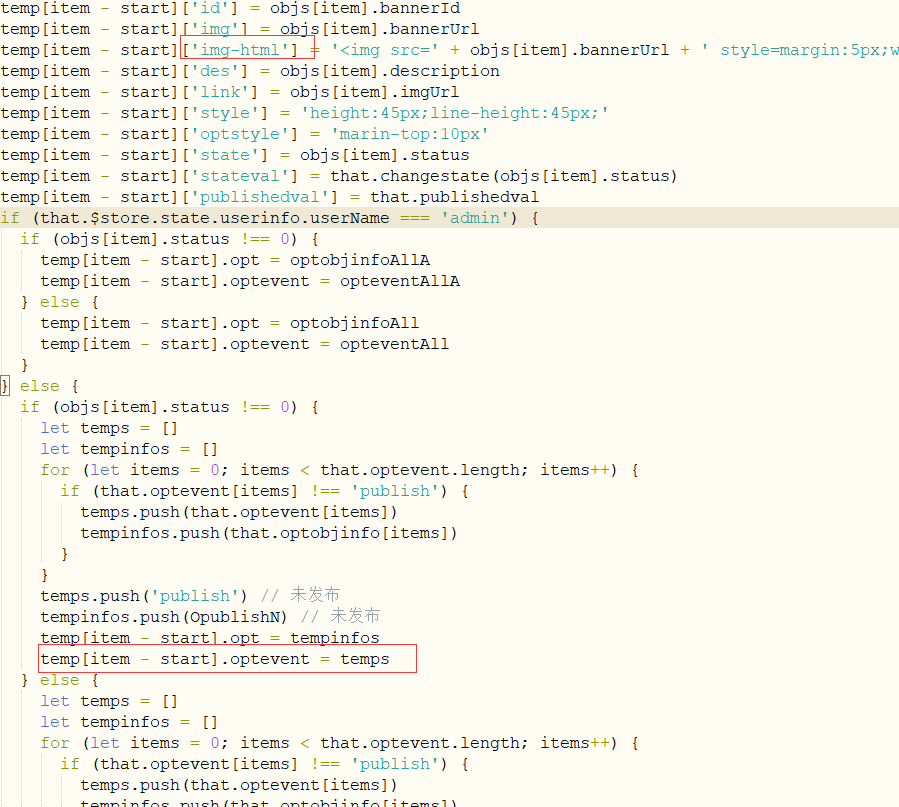
1. Global.scss 全局sass文档
2. Mcourse.scss 课程管理sass文档
3. Usercenter.scss 个人中心sass文档
4. Flive.scss 前台直播中心sass文档
5. Fcampustv.scss 前台校园电视台sass文档
6. AJlive.scss 后台直播中心sass文档
7. Fcourse.scss 前台课程中心sass文档
8. Channelmanage.scss 后台频道管理sass文档
9. Iconfont



1. Js



1. Angletype.babel.min.js 课程详情页教学分析
2. Channelrank.babel.min.js 后台模块管理排序
3. Danmu.babel.min.js 直播详情页弹幕
4. Imgup.js 图片上传
5. Stchart.babel.min.js st分析图表
6. 权限分配其他方式



1. 如上图，权限分配也可以在接口里面进行配置optevent参数，如果没有配置，则直接获取常量optevent,如果配置了optevent,将直接获取接口内容的操作项，不在获取模板绑定的操作项
2. 上图img-html为插入dom节点标识，如果有该标识，则会将该条内容插入为html
3. 注意事项
4. 全局环境依赖于node，npm,webpack,babel,sass,es6,axios,echart
5. 相关文档连接

Node <https://nodejs.org/zh-cn/>

Npm <https://www.npmjs.com/>

Webpack <http://webpack.github.io/docs/>

Axios <https://www.npmjs.com/package/axios>

echart <http://echarts.baidu.com/demo.html#line-step>