

teach权限及业务线配置梳理

一、导读

不知可曾有过一碰权限深似海，配置权限分不清的感觉？

本篇会梳理一下权限和配置在工程中的含义，及具体的用法，还有一些自己的碎碎念。

关于权限、配置的历史，我只简单说一下，还请组里的前辈们来补充。

二、pre define

管理中心页面：对应的ftl `front-k12-teach/src/teachold/views/org/index.ftl`

期次页面：对应的ftl, `front-k12-teach/src/teachold/views/org/course/termEdit.ftl`

红色 强调

紫色 待补充或待确认

咖啡色 疑问

三、含义

权限本质上是跟角色挂钩，比如说「知识库导航是否展示」的权限，需要去判断这个账号是否是运营、管理员等角色，是的话，就赋予该账号这个权限字段。我们前端拿到的只是一个权限字段，判断的工作都给后端做掉了。另一种做法是前端拿到角色之后自己去判断。

关于业务线配置，之所以有这个东西存在，是因为曾经100分、2b、2c各条产品线共用teach工程，后来拆分出来之后，100分又与geek共用teach工程。举2个栗子：极客不展示试卷库导航，100分展示；极客展示一键结业功能，100分不展示。

四、配置

配置的引入

资源路径：`front-k12-teach/src/teachold/javascript/org/page/page.js`

teachold下page.js在init阶段引入了k12-cache-config中的业务区分配置：


```
ing.js  index.ftl  config.ftl  contact.ftl  release_geek.conf  release_dev.conf x  JS page.js .../common  JS page.js .../org/...  JS org.js .../page  JS answerSheet.js

# 输入文件通过路径筛选规则
# - 正则表达式, 忽略大小写
# - 确保可以通过new RegExp转换成正则
# - 规则用来检验要处理的文件路径, 默认处理所有文件, 忽略以","起始的目录和文件
#FILE_FILTER = \.(\.ftl|html)$
# 输入文件通过路径排除规则
# - 正则表达式, 忽略大小写
# - 确保可以通过new RegExp转换成正则
# - 规则用来检验不要处理的文件路径, 默认不排除文件
#FILE_EXCLUDE = \.(js|mc|css)$
# 输入输出文件编码, 默认为utf-8, 这项目必须保证所有文件的编码一致, 如css/js/ftl等文件的编码一致
#FILE_CHARSET = gbk
##### 别名配置 #####
##### 别名配置列表 #####
# 页面引入的脚本或者样式可以使用服务器端模板标记, 如
<script src="{config_lib_root}define.js${nej_config}"></script>
# 别名匹配规则, 可以通过new RegExp转换成正则表达式
#ALIAS_MATCH = \${{(.*)}}
# 别名配置列表
ALIAS_DICTIONARY = {"csRoot":"/src/teachold/css","jsRoot":"/src/teachold/javascript/","nejRoot":"/lib/nej/src/define.js?pro=/src/teach/javascript/&old=/src/teachold/javascript/&
edu=/src/edu/javascript/&mode=web"}
##### 别名配置列表 #####
```

当然如果你自己想引入额外的配置, 可以使用setting.js中的一些方法, 这里就不多介绍了。

配置的使用

本质上是根据配置来判断是否做某件事情。

举例: TERM_PRACTICE_WITH_EXAM用来配置是否支持考试tab

资源路径: front-k12-teach/src/teachold/javascript/org/module/term/practice.js

```
var _tabData = {
  data: {
    tabs: [
      {data:{title:"习题",id: _umi.getUMI('TERM_PRACTICE_TEST')}},
      {data:{title:"考试",id: _umi.getUMI('TERM_PRACTICE_EXAM'),disabled: !_setting.get(_termKey)['TERM_PRACTICE_WITH_EXAM']}}
    ]
  }
}
```

另一种解

本质是通过打包来区分:

```
config.ftl .../config x  site.ftl .../config  contact.ftl  release_geek.conf  release_dev.conf  JS p

You, 3 days ago | 3 authors (hzzlinannan and others)
1 <!-- Root Config 开发时请自行更换config路径 -->
2 <!-- assign csRoot = "/src/teach/css/"
3     jsRoot = "/src/teach/javascript/"
4     nejRoot = "/lib/nej/src/define.js?pro=/src/teach/javascript/&old=/src/teachold/javascript/&util=
5     rui=../../node_modules/@study/edu-front-regularUI/src/js/&pool=/lib/&config=/lib/k12-cache-co
6
7 <!-- @VERSION -->
8 <script>
9     location.config = {
10         root: '/src/teach/html/'
11     };
12 </script>
13 hzzlinannan, a year ago * @init project
14 <!-- @IGNORE {mode: 'geek'} -->
15 <!-- include "../config/k12/site.ftl" -->
16 <!-- /@IGNORE -->
17 <!-- @IGNORE {mode: 'k12'} -->
18 <!-- include "../config/geek/site.ftl" -->
19 <!-- /@IGNORE -->
```

```
hzwangxinyu, 4 months ago | 3 authors (hzlinannan and others)
<#assign siteName="网易100分">
<#assign siteLogo="http://tkc.stu.126.net/res/images/common/logo/k12logo2.png">
<#assign siteLogo3="http://tkc.stu.126.net/res/images/common/logo/k12logo2.png">
<#assign siteFavicon="http://tkc.stu.126.net/res/favicon/k12.ico">
<#assign host="100.163.com">
<#assign initUploadUrl="/api/courseDesign/TermManager/getEduUploaderToken.json">
<#assign adminUploadUrl="/api/courseDesign/TermManager/getEduUploaderToken.json">
<#assign logoutUrl="http://100.163.com/path/member/logout">
<#assign gaAccount="UA-85176345-5">
<#assign gaProduct="k12">

<!-- @NOPARSE -->
<script>
  (function(g){
    g.eduProduct = {
      "siteName": "${siteName?js_string}",
      "siteLogo": "${siteLogo?js_string}",
      "host": "${host}",
      "initUploadUrl": "${initUploadUrl}",
      "adminUploadUrl": "${adminUploadUrl}",
      "logoutUrl": "${logoutUrl}",
      "gaAccount": "${gaAccount}",
      "gaProduct": "${gaProduct}",
      "platType": "-12"
    }
  })(window);
</script>
<!-- /@NOPARSE -->
```

然后通过window.eduProduct.platType或者gaProduct来判断是哪个产品线。

在front-k12-teach/src/teachold/javascript/common/ui/base.js 有添加isGeek字段，子类可以使用：

```
REG.define([
  'old/common/filter/base',
  'old/common/directive/base',
  'pool/edu-front-util/src/util',
  'old/util/constant',
  'pool/cache-base/src/setting',
  'old/common/util/privilegeUtil',
  'old/org/constant'
], function (_filter, _directive, util, Validation, setting, privilegeUtil, orgConstant) {

  var _baseUI = Regular.extend({
    name: 'baseUI',
    urlPrefix: window.urlPrefix,
    webUser: window.webUser,
    $defaultConfig: window.defaultConfig,
    validator: Validation.validate,
    config: function () {
      util.extend(this.data, {
        privilegeConstant: orgConstant.PROVIDER_PRIVILEGE,
        isRealTime: util.getValueByVisitor(window, 'eduProduct.gaProduct') === 'geek'
      });
      this.supr();
    },
    //根据rules规则校验UI
    validate: function (isRealTime) {
      if (isRealTime !== false) { //传递来的配置高于组件属性配置
        isRealTime = this.data.isRealTime; // 实时提示还可以输入多少字
      }
      var value = this.data.value;
      var rules = this.data.rules;
      if (!rules || !rules.length) { //没rules不存在则不进行校验
        return;
      }
      var result = Validation.validate(value, rules, isRealTime);
      this.data.state = result.success ? 'success' : 'error';
      this.data.tip = result.message;
      this.$update();
    },
    return result;
```

由于这种区分在打包阶段才会生效，因此在本地或者非打包环境测试的时候需要做点手脚，比如在调试100分的时候，注释掉config.ftl里面的geek/site.ftl引入。

这种做法写起来很方便，不需要写配置。

但是有个问题，如果teach后台以后每要拿给新产品用，就得新加一个字段来区分，并且在各处添加判断，所以不推荐，不过工程里已经有一些这种用法了，之后有相关的需求可以统一改。

五、权限

权限列表的引入

权限列表是后端塞到model里面的，管理中心和期次页面都可以通过 window.providerPrivilege 取到，不过精度有所不同，管理中心页面是精确到机构，期次页面是精确到期次。

什么意思呢？“remark_template_import”表示「点评模板导入的权限」，在管理中心页面，你是拿不到这个字段的；在期次页面，你如果是这个期次的讲师，可以拿到，否则不行。

```
index.ftl x JS lesson.js projectEvaluateUI.html JS projectEvaluateUI.js JS base.js config.ftl ../common config.ftl ../config
83 <#if providerWithAuthInfo??>
84     window.providerWithAuthInfo = ${objectToJsonFunction(providerWithAuthInfo)};
85     <#if providerWithAuthInfo.providerAuthorization??>
86         window.expTime = ${providerWithAuthInfo.providerAuthorization.expTime}
87     </#if>
88 </#if>
89
90 <#if providerRoleVos??>
91     window.providerRoles = [
92         <#list providerRoleVos as providerRole>
93             {
94                 roleType : ${providerRole.roleType?js_string},
95                 roleName : "${providerRole.roleName?js_string}"
96             },
97         </#list>
98     ];
99 </#if>
100
101 <#if providerPrivilege??>
102     window.providerPrivilege = ${objectToJsonFunction(providerPrivilege)};
103 </#if>
```

BTW，实际上后端在两个页面返回的值是不同的，虽然前端是以同一个字段来取：

管理中心里面拿到的是providerPrivilege，期次页面拿到的是viewPrivilege。

权限配置的引入

wtf，什么是权限配置？

这个概念是从 `cache-base/src/setting.js` 来的：

```

/**
 * 设置权限并转换成所包含的功能map并保存
 *
 * @example
 *
 * NEJ.define([
 *   'pool/cache-base/src/setting'
 * ],function(setting){
 *   // 单项设置更新, 设置项合并
 *   setting.setPermission(
 *     ['has_a_permission', 'has_b_permission'],
 *     {
 *       "has_a_permission": { a权限
 *         "component-key": { 某组件或模块
 *           "switch-a": true a权限的开关
 *         }
 *       }
 *     }
 *   );
 * });
 *
 * @method module:pool/cache-base/src/setting.setPermission
 * @param {Array} arr - 权限, 字符串数组。 ex: ['has_a_permission', 'has_b_permission']
 * @param {Array} funcMap - 权限和功能的映射
 * @returns {void}
 */
exports.setPermission = function (arr, funcMap) {
  u._$forIn(funcMap, function (map) {
    CTRL_PERMISSION = u._$merge(CTRL_PERMISSION, map);
  });

  u._$forEach(arr, function (permission) {
    // DT_PERMISSION = u._$merge(DT_PERMISSION, funcMap[permission]);
    u._$forIn(funcMap[permission], function(_it, _key){
      DT_PERMISSION[_key] = DT_PERMISSION[_key] || _it;
      DT_PERMISSION[_key] = u._$merge(DT_PERMISSION[_key], _it);
    });
  });
};

```

这个配置描述一下, 意思就是: 含有某个权限的组件或者模块, 有没有对应的权限开关。

没错, 这个说法看起来很懵逼, 但是不要怀疑, 从注释和目前的用法都是这样的。

来看看这段代码, 实际上就是往CTRL_PERMISSION和DT_PERMISSION存了些东西。

存了啥? 有什么区别?

先通过case看看_.\$merge方法:

```

> var a = {x:1,y:2}
  b = {x:2,z:3}
  u._$merge(a,b)
< ▶ {x: 2, y: 2, z: 3}

> var a = {x:{a:1,b:2},y:2}
  b = {x:{c:3},y:3}
  u._$merge(a,b)
< ▼ {x: {...}, y: 3} ⓘ
  ▶ x: {c: 3}
    y: 3
  ▶ __proto__: Object

```

然后大概读一下代码，你会发现funcMap必须是三层结构的，如果是两层会报错：

```

{
  "has_a_permission": {
    "component-key": {
      "switch-a": true
    }
  }
}

```

最终存的东西长这样：

```

> DT_PERMISSION
< ▼ {module-term-sidebar: {...}, module-course-list: {...}, module-course-edit-basic-permission: {...}, module-course-edit-intro: {...}, module-label: {...}} ⓘ
  ▶ module-course-edit-basic-permission: {ALLOW_BASIC_EDIT: true}
  ▶ module-course-edit-intro: {ALLOW_INTRO_EDIT: true}
  ▼ module-course-list:
    ALLOW_DRAG_TERM: true
    ALLOW_SET_DEFAULT_TERM: true
    SHOW_CANCEL_AUDIT: true
    SHOW_CREATE_COURSE: true
    SHOW_CREATE_TERM: true
    SHOW_DEL_COURSE: true
    SHOW_DEL_TERM: true
    SHOW_EDIT_COURSE: true
    SHOW_EDIT_TERM: true
    ▶ __proto__: Object
  ▶ module-label: {KNOWLEDGE_LABEL_EDIT: true, KNOWLEDGE_LABEL_DRAG: true, KNOWLEDGE_LABEL_ADD: true}
  ▶ module-term-sidebar: {TERM-SIDE-STUDENTENROLL: true, TERM-SIDE-PRICE: true, TERM-SIDE-VISIBLE: true, TERM-SIDE-INTRO: true, TERM-SIDE-PROPERTY: true, ...}
  ▶ __proto__: Object

> CTRL_PERMISSION
< ▼ {module-course-list: {...}, module-course-edit-basic-permission: {...}, module-course-edit-intro: {...}, module-label: {...}, module-term-sidebar: {...}} ⓘ
  ▶ module-course-edit-basic-permission: {ALLOW_BASIC_EDIT: false}
  ▶ module-course-edit-intro: {ALLOW_INTRO_EDIT: true}
  ▼ module-course-list:
    SHOW_CANCEL_AUDIT: true
    SHOW_DEL_TERM: true
    SHOW_EDIT_TERM: true
    ▶ __proto__: Object
  ▶ module-label: {KNOWLEDGE_LABEL_DRAG: true}
  ▶ module-term-sidebar: {TERM-SIDE-PRICE: true, TERM-SIDE-VISIBLE: true, TERM-SIDE-INTRO: true, TERM-SIDE-PROPERTY: true, TERM-SIDE-RULES: true, ...}
  ▶ __proto__: Object

```

主要的区别就是，DT_PERMISSION是arr过滤后的，CTRL_PERMISSION是过滤前的。

还有个区别是CTRL_PERMISSION是不充分merge的，意思就是后面merge的第二级结构会直接覆盖掉前面的第二级结构。不知道为什么要这样实现？

那这个arr传什么呢？没错，就是providerPrivilege。所以，你在DT_PERMISSION里面取的数据都是有权限的。

来看看我们工程里是在哪导入权限配置的：

```

index.html x termEdit.html JS lesson.js projectEvaluateUI.html JS projectEvaluateUI.js base.js
// 合并module-term, module-questionnaire, module-question, k12-module-question
pageConfig.rules.alias = _util._$merge(
  pageConfig.rules.alias,
  moduleLabelConfig.rules.alias,
  moduleResourcesConfig.rules.alias,
  moduleQuestionConfig.rules.alias,
  k12ModuleQuestionConfig.rules.alias,
  moduleLiveManagementConfig.rules.alias,
  moduleOrgConfig.rules.alias
);
pageConfig.modules = _util._$merge(
  pageConfig.modules,
  moduleLabelConfig.modules,
  moduleResourcesConfig.modules,
  moduleQuestionConfig.modules,
  k12ModuleQuestionConfig.modules,
  moduleLiveManagementConfig.modules,
  moduleOrgConfig.modules
);
window.pageConfig = pageConfig;
// 判断
if(window.providerWithAuthInfo && (window.providerWithAuthInfo.externalId > 0)){
  pageConfig.rules.rewrite['/courses/buy'] = '/courses';
}else{
  pageConfig.rules.rewrite['/courses/self'] = '/courses';
}
_setting.setPermission(window.providerPrivilege, _permission);
window.isGeekProvider = window.scope.applicationId == "netease-geek";
_setting.setScope({
  applicationId: window.scope.applicationId,
  scopeTypes: 101
});

```

这个_permission来自于front-k12-teach/src/teach/javascript/config/permission/org.js:

```

conf JS page.js .../common JS page.js .../org/... JS org.js .../page JS answerSheet.js JS page.js .../javascript/page JS page.js .../javasc
    },
    'team_term_edit': {
      'module-course-list': {
        'SHOW_EDIT_TERM': true, //显示编辑期次
        'SHOW_DEL_TERM': true, //显示删除期次
        'SHOW_CANCEL_AUDIT': true //显示撤销审核
      },
      'module-course-edit-basic-permission': {
        'ALLOW_BASIC_EDIT': false // 是否编辑基本信息页
      }
    },
    'v_k12_term_student_enroll': {
      'module-term-sidebar': {
        'TERM-SIDE-STUDENTENROLL': true // 加入学生
      }
    },
    'v_k12_term_practice_mark': {
      'module-term-sidebar': {
        'TERM-SIDE-MARK': true // 批改
      }
    },
    'v_k12_term_default': {
      'module-term-sidebar': {
        'TERM-SIDE-PRICE': true, // 价格设置
        'TERM-SIDE-VISIBLE': true, // 可见性
        'TERM-SIDE-INTRO': true, // 旧的课程信息 (介绍页)
        'TERM-SIDE-PROPERTY': true, // 期次属性
        'TERM-SIDE-RULES': true, // 评分规则
        'TERM-SIDE-TEAM': true, // 课程团队设置
        'TERM-SIDE-COURSEWARE': true, // 课件
        'TERM-SIDE-EXAM': false, // 习题及考试
        'TERM-SIDE-EXERCISE-PAPER': true, // 练习卷
        'TERM-SIDE-RESOURCE': true // 视频库
      }
    }
  }
};

```

顺便一提，虽然用可以这么用，但是这样的写法已经违背了设计的初衷，设计的初衷（从注释可得）是一个component-key下只有一个switcher，但这里有多个，语义上就会显得很奇怪。

权限的使用

权限列表的使用

util

资源路径: front-k12-teach/src/teachold/javascript/common/util/privilegeUtil.js

allowed

```
/**
 * 根据权限判断是否允许
 * @param _privileges 允许的权限列表数组
 * @param _selfPrivilege 需要鉴定的权限列表数组
 * 想了半天说白了, 就是判断_privileges 与 _selfPrivilege 是否有相同的属性, _selfPrivilege已经默认取全局的权限了
 */
_p.allowed = function(_privileges, _selfPrivilege) {
    if (!_privileges) {
        return false;
    }

    if (g.notPrivilegeControl){
        return true;
    }

    var allowed = false;

    _selfPrivilege = _selfPrivilege || g.providerPrivilege || g.privilege; // 取全局的用户权限

    if (!_util._isArray(_privileges)) {
        _privileges = [_privileges];
    }

    if (_privileges && _privileges.length > 0) {
        allowed = _arrayUtil._$arrCross(_privileges, _selfPrivilege);
    }

    return allowed;
};
```

一般用法是 `allowed(['priv1', 'priv2'])`, 即判断是否有指定权限。

已经写到front-k12-teach/src/teachold/javascript/common/ui/base.js中, 子类都可以用:

```
},
isShow: function (switcher) {
    return setting.isShow({
        "switcher": switcher,
        "key": this.settingKey
    });
},
allowed: function(privilege, selfPriv) {
    return privilegeUtil.allowed(privilege, selfPriv);
}
})

.filter(_filter)
.directive(_directive);
```

checkPrivilege

```

52
53     /**
54     * 判断是否具有权限
55     *
56     * @public
57     * @param {Object} options 模块options
58     * @param {Object} menuList menuList
59     * @returns {void}
60     */
61     _p.checkPrivilege = function (options, menuList) {
106
107

```

用于判断umi权限，没权限的话跳转到404页面。待补充。

可以看下这篇 权限系统项目总结

directive

资源路径: front-kl2-teach/src/teachold/javascript/common/directive/privilegeDirective.js

r-allowed

```

_directive['r-allowed'] = function(elem, value) {
  this.$watch(value, function(newValue, oldValue){
    if(!_privilegeUtil.allowed(newValue)){ // 权限不允许则隐藏节点
      elem.style.display = 'none';
    }
  })
}

```

跟allowed方法用法差不多:

```

( CAN_DELETE_TERM_ADMIN_PRIVILEGE ))
<a class="f-fcred" r-allowed=[{privilege.TERM_MANAGER, privilege.TERM_EDIT}] on-click={this.delTerm(item.id,course.id)}>删除</a>
{/if}

```

r-allowedEnter

```

// 是否显示入口: 多个权限数组, 满足一个就显示
_directive['r-allowedEnter'] = function(elem, value) {
  // this.$watch(value, function(newValue, oldValue){
  var allowed = true,
  appConfig;

  if(this.$appConfigKey && setting.get(this.$appConfigKey)[value.config] === false){
    allowed = 'allowed';
  }

  if (value.type === 'expression') {
    value = this.$get(value);
  };

  if (allowed && value.privilege && value.privilege.length > 0) {
    for (var i = value.privilege.length - 1; i >= 0; i--) {
      allowed = _privilegeUtil.allowed(value.privilege[i]);

      if (allowed) { // 有一个满足就显示
        break;
      };
    };
  };

  if (!allowed) {
    elem.style.display = 'none';
  };
  // })
}

```

除了appConfig那句的配置，逻辑跟r-allowed是一毛一样的，写复杂了，建议不要使用。

看看工程里是怎么用的 front-kl2-teach/src/teachold/javascript/org/ui/center/common/sideBar.html:

```
<li>
  <dl>
    <dt class="navItem"><i class="icon u-icon-course-manage f-fc6 f-fs16"></i><span class="cate f-fc6 f-fs16">课程管理</span></dt>
    <dl class="navItem" r-class="{('current': this.data.umi == this.$umi.getUMI('COURSE_LIST'))}><a class="link f-fc3 f-fs14" href="#{this.$umi.getUMI('COURSE_LIST')}">课程管理</a></dl>
    <dl class="navItem" r-hide="{(courseLive) r-class="{('current': this.data.umi == this.$umi.getUMI('LIVE_MANAGEMENT_MULTI'))}><a class="link f-fc3 f-fs14" href="#{this.$umi.getUMI('LIVE_MANAGEMENT_MULTI')}">直播管理</a></dl>
  </dl>
  </li>
  <li r-hide="{!showKnowledge}>
    <dl r-allowedEnter="{('privilege': privilege.KNOWLEDGE_IMPORT)}>
      <dt class="navItem"><i class="icon u-icon-course-manage f-fc6 f-fs16"></i><span class="cate f-fc6 f-fs16">知识库</span></dt>
      <dl class="navItem" r-class="{('current': this.data.umi == this.$umi.getUMI('KNOWLEDGE_IMPORT'))} r-allowedEnter="{('privilege': privilege.KNOWLEDGE_IMPORT)}><a class="link f-fc3 f-fs14" href="#{this.$umi.getUMI('KNOWLEDGE_IMPORT')}">导入</a></dl>
      <dl class="navItem" r-class="{('current': this.data.umi == this.$umi.getUMI('LABEL_SPECIAL'))} r-allowedEnter="{('privilege': privilege.KNOWLEDGE_IMPORT)}><a class="link f-fc3 f-fs14" href="#{this.$umi.getUMI('LABEL_SPECIAL')}">知识点标签</a></dl>
      <dl class="navItem" r-class="{('current': this.data.umi == this.$umi.getUMI('LABEL_GENERAL'))} r-allowedEnter="{('privilege': privilege.KNOWLEDGE_IMPORT)}><a class="link f-fc3 f-fs14" href="#{this.$umi.getUMI('LABEL_GENERAL')}">通用能力标签</a></dl>
      <dl class="navItem" r-class="{('current': this.data.umi == this.$umi.getUMI('LABEL_RELATEDVIDEO'))} r-allowedEnter="{('privilege': privilege.KNOWLEDGE_IMPORT)}><a class="link f-fc3 f-fs14" href="#{this.$umi.getUMI('LABEL_RELATEDVIDEO')}">关联视频</a></dl>
    </dl>
  </li>
</li>
```

权限配置的使用

权限配置相关的api只有两个，都写在 front-kl2-teach/lib/cache-base/src/setting.js 里面，一个是isShow，一个是isShow2 ...

isShow

```
/**
 * 根据功能开关和权限查看是否显示
 *
 * @example
 *
 * NEJ.define([
 *   'pool/cache-base/src/setting'
 * ],function(setting){
 *   // 单项设置更新，设置项合并
 *   setting.merge({
 *     "key": "module-backend-admin",
 *     "switcher": "SHOW_MODULE_OK"
 *   });
 * });
 *
 * @method module:pool/cache-base/src/setting.isShow
 * @param {Object} options
 * @param {String} options.key 模块key
 * @param {String} options.switcher 开关
 * @returns {Boolean}
 */
exports.isShow = function (options) {
  // 判断业务线配置开关
  var hasSet = (DT_SETTING[options.key][{}][options.switcher] !== false; // 默认不配置就返回true

  // 判断是否有用户权限控制开关
  if (CTRL_PERMISSION[options.key]){
    hasSet = hasSet && ((DT_PERMISSION[options.key][{}][options.switcher] === true)
  }
  return hasSet;
};
```

可以看出来，这段代码不仅有对权限配置的判断，还有对业务线配置的判断。

大致描述就是：返回结果是true还是false，由业务线配置和权限配置共同决定；如果没有做相应的配置，默认返回true。

看看工程中的用法：

```
if (!setting.isShow({key: _key, switcher: 'SHOW_MODULE_COURSE_CLASS'})) {
  if (_res.list && _res.list instanceof Array) {
    for (var i = _res.list.length - 1; i >= 0; i--) {
      _res.list[i].hidedown = true;
    }
  }
}
```

除此之外，还在下面两个base文件中实现了isShow，子类都可以使用，不过需要提前配置settingKey:

front-kl2-teach/src/teach/javascript/common/ui/base.js

front-k12-teach/src/teachold/javascript/common/ui/base.js

```
    },
    isShow: function(switcher){
        return setting.isShow({
            "switcher": switcher,
            "key": this.settingKey
        });
    }
})
```

用法如:

```
    },
    termScheduleDraft:{}
};
this.settingKey = "module-course-edit-intro-permisson";
this.data.scene = "intro";
```

```
    canEdit: function(data){
        return this.isShow('ALLOW_INTRO_EDIT');
    },
```

isShow2

有时候我们可能想强制判断权限配置，而不是没有配置就绕过，那这种场景就需要用到isShow2了:

```
exports.isShow2 = function (options,privilegeDefaultFalse) {
    if(!privilegeDefaultFalse){
        return exports.isShow(options);
    }

    // 判断业务线配置开关
    var hasSet = (DT_SETTING[options.key]||{})[options.switcher] !== false && !!CTRL_PERMISSION[options.key]; // 默认不配置返回false

    // 判断是否有用户权限控制开关
    if(CTRL_PERMISSION[options.key]){
        hasSet = hasSet && ((DT_PERMISSION[options.key]||{})[options.switcher] === true)
    }
    return hasSet;
};
```

可以看出来代码跟isShow大同小异，只是如果没有权限配置的话，会返回false。不过写法不太友好...我的话会这样写:

```
exports.isShow2 = function (options) {
    var hasSet = u.getValueByVisitor(DT_SETTING,options.key+'.'+options.switcher) !== false;

    hasSet = hasSet && u.getValueByVisitor(DT_PERMISSION,options.key+'.'+options.switcher) === true;

    return hasSet;
};
```

实际上DT_PERMISSION有的key，CTRL_PERMISSION一定有，相应的判断可能是多此一举了。

我们来看看工程中是怎么用的 front-k12-teach/lib/cache-term/src/term/menu.js:

```
// 旧的模块如果还需要则可以自己加配置
{
  "id" : "TERM-SIDE-INTRO",
  "flag" : CONST.TERM_MENU_INTRO,
  "name" : "期次介绍页",
  "href" : "#/intro",
  visible: function (term,config,setting) {
    return !!setting[this.id] && !!s.isShow2({key: SETTING_KEY, switcher:this.id},setting[PRVILIGE_DEFAULT_FALSE]); // 保留旧的配置
  }
},
```

这里是期次页面的sidebar显示隐藏的配置，我debug的时候发现 PRVILIGE_DEFAULT_FALSE 不存在，所以逻辑直接走到了isShow。

然后根据k12-cache-config的业务线配置及front-k12-teach/src/teach/javascript/config/permission/org.js 中的权限配置进行判断。

碎碎念

感觉权限配置这块数据结构略麻烦，最外层knowledge_bank_add_tag实际上是没有用的，感受一下：

```
//标签相关
'knowledge_bank_add_tag': { //知识点树添加权限
  "module-label":{
    "KNOWLEDGE_LABEL_ADD": true // 添加开关
  }
},
```

六、当提到权限的时候你应该怎么写

导航的权限

待补充

讲师、期次负责人权限

跟后端商定好，让他们在viewPrivilege或者providerPrivilege加个字段，然后用下面的资源的api去判断吧，具体的用法可以往上翻有写：

front-k12-teach/src/teachold/javascript/common/directive/privilegeDirective.js

front-k12-teach/src/teachold/javascript/common/util/privilegeUtil.js

front-k12-teach/src/teachold/javascript/common/ui/base.js

记得顺手在 front-k12-teach/src/teachold/javascript/org/constant.js 加上相应的字段，然后用的时候去拿这个不要直接写死，方便在变更的时候统一修改：

```

'WYB_ACCOUNT_MOD': "wangyibao_account_modify",

// 标签相关权限
'KNOWLEDGE_BANK_ADD_TAG': "knowledge_bank_add_tag",
'KNOWLEDGE_BANK_EDIT_TAG': "knowledge_bank_edit_tag",
'KNOWLEDGE_BANK_DRAG_TAG': "knowledge_bank_drag_tag",

//微课库相关全新啊
'MICRO_COURSE_SETTING': 'v_k12_term_micro_course',

'REMARK_TEMPLATE_IMPORT': 'remark_template_import' // 点评模板权限
}

```

不推荐使用权限配置，因为还得到 front-k12-teach/src/teach/javascript/config/permission/org.js 做额外的配置，但是这些配置并没有多大的意义。

如果考虑到某个feature的显示隐藏由业务线配置和权限配置共同决定，那可以用一下isShow和isShow2，不过你也可以不用，自己封装一个isShow3 (这里随手写了一个不要当真)，这样就可以不用写权限配置了：

```

/**
 * @param {Object} options
 * @param {String} options.key 组件的settingKey
 * @param {String} options.switcher 权限及配置开关字段
 */
isShow3 = function(options){
    var hasSet = setting.get(options.key)[options.switcher] !== false;
    hasSet = hasSet && privilegeUtil.allowed(options.switcher);

    return hasSet;
}

```

运营权限

```

JS courses.js src/teachold/javas... 4
JS org.js src/teach/javascript/... 4, M
JS menu.js lib/cache-term/src/... +9
JS privilegeDirective.js (Wo... +9, M
JS root.js raw/teachold/javas... 6, M
JS nav.ftl pub/teach/views/common
JS nav.ftl src/teach/views/common
JS providerinfo.ftl src/teach/Views/c...
FRONT-K12-TEACH
├─ permission
│   └─ JS org.js 4, M
│       ├── requestConfig.js
│       └─ JS videoConfig.js
├─ lib
├─ module
├─ page
├─ ui
├─ scss
├─ views
│   └─ common
│       ├── common.ftl
│       ├── hrefManagement.ftl
│       ├── macro.ftl
│       ├── meta.ftl
│       ├── nav.ftl
│       ├── providerinfo.ftl
│       ├── schools.ftl
│       └─ script.ftl
└─ config

```

```

2
3 <script>
4   <#if provider?>
5     window.provider = ${objectToJsonFunction(provider)};
6   </#if>
7
8   <#if providerWithAuthInfo?>
9     window.providerWithAuthInfo = ${objectToJsonFunction(providerWithAuthInfo)};
10    <#if providerWithAuthInfo.providerAuthorization?>
11      window.expTime = ${providerWithAuthInfo.providerAuthorization.expTime}
12    </#if>
13  </#if>
14
15  <#if isBackAdmin?>
16    window.adminFlag = ${isBackAdmin?string('1','0')};
17  </#if>
18
19  <#if providerRoleVos?>
20    window.providerRoles = [
21      <#list providerRoleVos as providerRole>
22        {
23          roleType : ${providerRole.roleType?s_string},
24          roleName : "${providerRole.roleName?s_string}"
25        },
26      </#list>
27    ];
28  </#if>
29
30  <#if providerPrivilege?>
31    window.providerPrivilege = ${objectToJsonFunction(providerPrivilege)};
32  </#if>
33  </script>
34  <!-- /@OPARSE -->
35
36

```

providerInfo是引入到common.ftl里面的，而common.ftl被所有页面引用，所以所有页面都可以通过window.adminFlag来取到运营权限。

这个adminFlag是否包含超级管理员，**待确认**

七、疑问

不知道vuex这种方案是否适用于管理这类配置？