链滴

怎样构建一个灵活的权限系统

作者：wuhongxu

原文链接：https://ld246.com/article/1474538624216

来源网站：链滴

背景

我所在的公司是一家培训机构，公司给了我一个学生管理系统的项目

思绪

以上是背景，然后公司给我一个要求是权限系统必须要足够的灵活，可单独为任何的角色添加、删除修改的东西，还可以任意添加、删除角色。

好吧，我服，然后开始对权限系统做思考，以下是思路（前方人格分裂？#雾）：

这样的权限系统很简单嘛，我把每个角色可访问的路由，使用json格式存入数据库不就得了。

（你傻啊？存路由？从哪儿取这些路由啊？）。

恩，有道理，我 把网站所有的路由全部存在数据库，然后使用id做唯一标示，然后把角色允许访问的些id使用Json格式存入数据库。

（不允许的链接地址，你虽然会拒绝访问，但是你的按钮还在啊喂！比如【学生管理】这个菜单项，点进去会拒绝访问，但是你的这个菜单项还在菜单里啊）。

恩...好像有道理，那我使用不同的模板，不就行啦。

（我无力吐槽..说好的灵活呢...添加角色的时候你不就只有一脸懵逼？）

那我提前判断哪些显示哪些不能显示呗?

（那不是每次访问的时候，还要把所有的路由遍历一次？）

额。。。好吧。我服我服..不简单啊喂！

清空一下思路，从新思考一下，于是...想法成型了

成型

我们仔细的观察大部分的网站，基本上链接呈现这样一个结构，访问首页->点击导航->到各个页面-再到页面下单独模块->模块内的功能。它就像树一样开支散叶。我们访问的每一个路由到的页面其实是顺着树干（注意是必须顺着树干）到下一个分叉口（我们称之为结点），结点中存储着这里的信息我们就可以取出其中的信息，这棵树就是整体的网站结构，然后我们的权限系统呢？就像是从这棵树去枝，把不能去的地方剃掉，留下能去的地方，然后我们就可以畅通无阻啦。

于是我决定使用这种结构去做网站权限的设计。它有什么好处呢，我总结了一下：

1、直观，网站结构、权限结构一目了然。

2、效率高，树的遍历有一个特点，很符合我们的要求，那就是要访问子节点，必须访问他的父亲，以我们只要限制父亲不能去了，子节点根本不需要访问，直接抛弃。

3、规范性，这是一个隐式的好处，它将我们的所有的链接功能，都做了很好的管理，不再杂乱，而们每次去添加一个路由，都需要添加在树中，很规范，功能的设计更加的模块化了（怎么模块化？后再说）

说了这么多，准备上代码啦~

（不慌啊喂，树到底是啥概念，完全不懂啊？）

请自行百度~~
（喂喂）

树的存储方式，一般有两种，好吧，我的一般写法是这两种，如果有其他写法，请务必告诉我，谢谢。

第一种是在节点中存储子节点组成的链表，大概是这样：

```
public class Node{
    /*节点信息*/
    ...
    /*子节点链表*/
    List<Node> children;
}
```

第二种是使用二叉链表，其实也就是在节点中只存储两个子节点，分别为leftChild,nextSibling。分就是自己的第一个孩子以及右边的兄弟
大概是这样：

```
public class Node{

    /*节点信息*/
    ...
    /*子节点链表*/
    Node leftChild;
    Node nextSibling;
}
```

我是使用的第二种方式，以下是我的代码（小朋友不要轻易模仿）

TreeNode.java 接口

```
/**
 * Created by wuhongxu on 2016/9/4 0004.
 */
public interface TreeNode {
    public void setChildCount(java.lang.Integer childCount);

    public java.lang.Integer getChildCount();

    public void setDegree(java.lang.Integer degree);

    public java.lang.Integer getDegree();

    public TreeNode getParent();

    public void setParent(TreeNode treeNode);

    public void setLeftChild(TreeNode treeNode);

    public TreeNode getLeftChild();

    public void setNextSibling(TreeNode treeNode);
    public TreeNode getNextSibling();
}
```

Mapping.java

```
import com.hudongwx.studentsys.common.BaseMapping;
import com.hudongwx.studentsys.util.TreeNode;

/**
 * create by wuhongxu. */@SuppressWarnings("serial")
public class Mapping extends BaseMapping<Mapping> implements TreeNode {
   public static final Mapping dao = new Mapping();
   private Mapping parent,leftChild,nextSibling;
   public Mapping(){
      init();
   }
   public Mapping(String icon,String title,String url){
      setIcon(icon);
      setTitle(title);
      setUrl(url);
      init();
   }
   public Mapping(String icon,String title,String url,String message){
      this(icon,title,url);
      setMessage(message);
   }
   public void init(){
      /*getParent();
 getLeftChild(); getNextSibling();*/  setChildCount(0);
      setParentId(0);
      setLeftChildId(0);
      setNextSiblingId(0);
   }
   @Override
  public Mapping getParent() {
      return parent;
   }

   @Override
  public void setParent(TreeNode treeNode) {
      parent = (Mapping) treeNode;
   }

   @Override
  public void setLeftChild(TreeNode treeNode) {
      leftChild = (Mapping)treeNode;
   }

   @Override
  public Mapping getLeftChild() {
      return leftChild;
   }
   @Override
  public void setNextSibling(TreeNode treeNode) {
      nextSibling = (Mapping) treeNode;
   }

   @Override
  public Mapping getNextSibling() {
      return nextSibling;
   }

   public boolean equals(Mapping mapping) {
      return mapping.getId().equals(getId()) && mapping.getTitle().equals(getTitle()) && mappi
g.getUrl().equals(getUrl());
   }
}
```

ArrayTree.java

此处更新于2016.12.30

```
package com.hudongwx.crowdSourcing.util;

import java.util.*;

/**
 * 二叉链存储方式实现的树.
 * Date: 2016/12/30 0026 * Time: 14:47 * * @author wuhongxu.
 * @version 1.0.0
 */public class ArrayTreeextends TreeNode> {
    private List tree;
    //暂时缓存上次添加的父节点，方便下次添加到同一父节点下时直接添加不需要传入父节点参数
  private T parent;
    //暂定层数最多一百层
  private int[] index = new int[Common.MAX_DEGREE];
    //这里得到的是复制，不能操作树本身
  public List getList() {
        return new ArrayList<>(tree);
    }
    public ArrayTree setTree(List tree){
        this.tree = tree;

        return this;
    }
    public void addGoodNode(T node){
        if(tree == null)
            tree = new ArrayList<>();
        tree.add(node);
    }
    public void addGoodNode(int index,T node){
        if(tree == null)
            tree = new ArrayList<>();
        tree.add(index,node);
    }
    public ArrayTree() {
        tree = new ArrayList();
    }

    //通过此接口实现额外对node的操作
  public interface onOverListener {
        void onOver(T node, T parent, T nextSibling);
    }

    public interface onCheckListener {
        boolean onCheck(T now);
    }

    public List initTree(T root) {
        clearTree();
        root.setChildCount(0);
        root.setDegree(0);
        tree.add(root);
        return tree;
    }

    public void destroyTree() {
        tree = null;
    }

    public void clearTree() {
        if (!tree.isEmpty())
            tree.clear();
    }

    public boolean isEmpty() {
        return null != tree && tree.isEmpty();
    }

    public T root() {
        if (!tree.isEmpty()) {
            T t = tree.get(0);
            while (t.getParent() != null) {
                t = (T) t.getParent();
            }
            return t;
        }
        return null;
    }

    public T value(T node) {
        if (!tree.isEmpty() && tree.contains(node))
            return node;
        return null;
    }

    public void assign(T node, T newNode) {
        if (!tree.isEmpty() && tree.contains(node))
```

```
          node = newNode;
  }

  public T getParent(T node) {
      if (!tree.isEmpty() && tree.contains(node))
          return (T) node.getParent();
      return null;
  }

  public T getLeftChild(T node) {
      if (tree.isEmpty() && tree.contains(node))
          return (T) node.getLeftChild();
      return null;
  }

  public T getNextSibling(T node) {
      if (tree.isEmpty() && tree.contains(node))
          return (T) node.getNextSibling();
      return null;
  }

  //为了通用，没有为node的做多余操作，所以使用接口来让调用者实现
public boolean insertChild(T node, T parent, T leftSibling, onOverListener opn) {
      if (null == parent)
          return false;
      this.parent = parent;
      if (tree.contains(node) || !tree.contains(parent))
          return false;
      if (leftSibling != null && !tree.contains(leftSibling)) {
          return false;
      }
      if(parent.getDegree() >= Common.MAX_DEGREE)
          return false;
      parent.setChildCount(parent.getChildCount() + 1);

      node.setDegree(parent.getDegree() + 1);
      node.setParent(parent);
      if (null != leftSibling) {
          T tmp = (T) leftSibling.getNextSibling();
          leftSibling.setNextSibling(node);
          node.setNextSibling(tmp);
      } else {
          TreeNode leftChild = parent.getLeftChild();
          parent.setLeftChild(node);
          node.setNextSibling(leftChild);
      }
      if (null != opn)
          opn.onOver(node, parent, leftSibling);
      //添加位置为无论如何添加所有同层的在一起
/*index[node.getDegree()] += index[parent.getDegree()];*/
tree.add(++index[node.getDegree()],node);
      //TODO 思考更快速的位置标记，暂时没想到
for(int i = node.getDegree() + 1; i < Common.MAX_DEGREE; i++)
          index[i]++;
```

```
        return true;
    }

    public boolean insertChild(T node, T parent, int pos, onOverListener ovl) {
        if (null == parent)
            return false;
        T nowChild = (T) parent.getLeftChild();
        T left = null;
        for (int i = 1; i < pos && nowChild != null; i++) {
            left = nowChild;
            nowChild = (T) nowChild.getNextSibling();
        }
        return insertChild(node, parent, left, ovl);
    }

    public boolean insertChild(T node, T parent, onOverListener ovl) {
        return insertChild(node, parent, Integer.MAX_VALUE, ovl);
    }
    public boolean insertChild(T node, T parent){
        return insertChild(node,parent,null);
    }
    public boolean insertChild(T node){
        return insertChild(node,parent,null);
    }
    public boolean insertChild(T node,onOverListener ovl){
        return insertChild(node,parent,ovl);
    }
    public boolean deleteChild(T node) {
        return tree.remove(node);
    }

    public void checkTree(T parent, onCheckListener ocl) {
        if (!tree.contains(parent) || null == ocl)
            return;
        Queue queue = new LinkedList<>();
        queue.add(parent);
        while (!queue.isEmpty()) {
            T now = queue.poll();
            //如果tree不包含此节点，将不对其进行操作遍历(因为其实操作的对象是同一个，所以可能
   会包含在链表中)
   if(!tree.contains(now))
                continue;
            T first = null;
            if(now.getParent() != null)
                first = (T) now.getParent().getLeftChild();
            //如果返回值为false，将放弃子节点遍历，但是会继续遍历本层
   if (!ocl.onCheck(now))
                continue;
            T c = (T) now.getLeftChild();
            while (c != null) {
                queue.offer(c);
                c = (T) c.getNextSibling();
            }
        }
    }

    //bfs完成遍历
  public void checkTree(onCheckListener ocl) {
        checkTree(root(),ocl);
    }
    public void checkTreePreorder(onCheckListener ocl){
        checkTreePreorder(root(),ocl);
    }
    //深搜,顺序还是为从前到后，比正常深搜多一个倒栈操作
  public void checkTreePreorder(T parent,onCheckListener ocl){
        if (!tree.contains(parent) || null == ocl)
            return;
        Stack stack = new Stack<>();
        Stack bufferStack = new Stack();
        stack.push(parent);
        while (!stack.isEmpty()) {
            T now = stack.pop();
            //如果tree不包含此节点，将不对其进行操作遍历(因为其实操作的对象是同一个，所以可能
会包含在链表中)
  if(!tree.contains(now))
                continue;
            T first = null;
            if(now.getParent() != null)
                first = (T) now.getParent().getLeftChild();
            //如果返回值为false，将放弃子节点遍历
  if (!ocl.onCheck(now))
                continue;
            T c = (T) now.getLeftChild();
            while (c != null) {
                //加入缓存栈
  bufferStack.push(c);
                c = (T) c.getNextSibling();
            }
            while(!bufferStack.isEmpty()){
                //倒序入栈
  stack.push(bufferStack.pop());
            }
        }
    }
}
```

使用示例

此处更新于2016.12.30， 读取xml权限信息，并持久化到数据库（为了方便= =）

```
public static void generatorXML(ArrayTree<Mapping> tree) throws IOException {
        Document document = DocumentHelper.createDocument();
        final Map<Mapping,Element> map = new HashMap<>();
        tree.checkTree(now->{
            if(now.getParent() == null){
                Element rt = DocumentHelper.createElement("mapping");
                document.setRootElement(rt);
                Element element = addAttributeToElement(rt, now);
                map.put(now,element);
                return true;
            }
            Element e = map.get(now.getParent()).addElement("mapping");
            map.put(now,addAttributeToElement(e,now));
            return true;
        });
        //输出到控制台
  XMLWriter xmlWriter = new XMLWriter();
        xmlWriter.write(document);

        // 输出到文件
  // 格式
  OutputFormat format = new OutputFormat("    ", true);// 设置缩进为4个空格，并且另起一行为t
ue
  XMLWriter xmlWriter2 = new XMLWriter(
                new FileOutputStream("src\\main\\resources\\permission.xml"), format);
        xmlWriter2.write(document);

        // 另一种输出方式，记得要调用flush()方法,否则输出的文件中显示空白
  /* XMLWriter xmlWriter3 = new XMLWriter(new FileWriter("student2.xml"),
 format); xmlWriter3.write(document2); xmlWriter3.flush();*/  // close()方法也可以

  }
    private static Element addAttributeToElement(Element e, Mapping mapping){
        e.addAttribute(Common.LABEL_ID,""+mapping.getId());
        e.addAttribute(Mapping.LABEL_ICON,mapping.getIcon());
        e.addAttribute(Mapping.LABEL_URL,mapping.getUrl());
        e.addAttribute(Mapping.LABEL_TITLE,mapping.getTitle());
        e.addAttribute(Mapping.LABEL_FUNCTION,""+mapping.getFunction());
        return e.addElement("mappings");
    }
    public static void readXML(final ArrayTree<Mapping> tree,final String path) throws Excepti
n {
        SAXReader saxReader = new SAXReader();
        File file = new File(path);
        if(!file.exists())
            throw new Exception(path+"文件不存在");
        Document document = saxReader.read(file);
        //bfs实现建树
  Queue<Element> queue = new LinkedList<>();
        queue.offer(document.getRootElement());
        while (!queue.isEmpty()){
            Element now = queue.poll();
            if(now.attributeCount() < 4)
                throw new Exception(now.getPath()+"节点的元素个数不足");
            Mapping mapping = new Mapping();
            List<Attribute> attributes = now.attributes();
            for(Attribute attribute : attributes){
                mapping.set(attribute.getName(),attribute.getValue());
            }
             if(now.isRootElement()){
                tree.initTree(mapping);
            }else{
                List<Mapping> list = tree.getList();
                for(Mapping m : list){
                    Element parent = now.getParent().getParent();
                    String s = parent.attributeValue(Common.LABEL_ID);
                    if(Objects.equals(m.get(Common.LABEL_ID), s)){
                        tree.insertChild(mapping,m);
                        break;
                    }
                }
            }
            List<Element> list = now.element("mappings").elements();
            list.forEach(queue::offer);
        }
        //建树完成 注意：因类型关系，此树不能重复使用,必须重新读树
  tree.checkTree(now->{
            log.info(now.toString());
            return true;
        });
        tree.checkTree(now -> {
            if(now.getParent() != null){
                now.set("parentId",now.getParent().get(Common.LABEL_ID));
                /*now.setParentId(now.getParent().getId());*/
  }
            if(now.getLeftChild() != null)
                now.set("leftChildId",now.getLeftChild().get(Common.LABEL_ID));
/*                now.setLeftChildId(now.getLeftChild().getId());*/
  if(now.getNextSibling() != null)
                now.set("nextSiblingId",now.getNextSibling().get(Common.LABEL_ID));
                /*now.setNextSiblingId(now.getNextSibling().getId());*/

  return now.save();
        });
        CacheKit.put(Common.CACHE_60TIME_LABEL,"mappingTree",null);
    }
```

xml文件实例

```
<!--0：功能，1：视图模块，2：菜单，3：二级菜单，以后多级菜单，依次类推-->

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapping>

<mapping id="2326" icon="fa" url="url" title="根目录" function="0">
    <mappings>
        <mapping id="2327" icon="fa fa-desktop" url="/" title="首页" function="2">
            <mappings>
                <mapping id="2337" icon="search" url="/common/search.ftl" title="全局搜索" funct
on="1">
                    <mappings/>
                </mapping>
        <mapping id="1999" icon="dataManager" url="/common/dataManager.ftl" title=
数据管理" function="1">
          <mappings/>
        </mapping>
        <mapping id="2000" icon="routs" url="/common/routs.ftl" title="导航" function="
">
          <mappings/>
        </mapping>
        <mapping id="2338" icon="fast" url="/common/fastOprator.ftl" title="快捷操作" fu
ction="1">
          <mappings/>
        </mapping>
        <mapping id="2339" icon="newest" url="/common/newest.ftl" title="最新报名" fun
tion="1">
          <mappings/>
        </mapping>
        <mapping id="2340" icon="statistics" url="/common/statistics.ftl" title="统计" funct
on="1">
          <mappings/>
        </mapping>
      </mappings>
    </mapping>
    <mapping id="2328" icon="fa fa-users" url="/userManager" title="用户管理" function="
">
      <mappings>
        <mapping id="2341" icon="userList" url="/userManager/userList.ftl" title="用户列
" function="1">
          <mappings>
            <mapping id="2355" icon="addUser" url="addUser" title="添加" function="0"

              <mappings/>
            </mapping>
            <mapping id="2356" icon="updateUser" url="updateUser" title="编辑" functi
n="0">
              <mappings/>
            </mapping>
            <mapping id="2357" icon="deleteUser" url="deleteUser" title="删除" functio
="0">
              <mappings/>
            </mapping>
          </mappings>
        </mapping>
        <mapping id="2342" icon="roleList" url="/userManager/roleList.ftl" title="角色列表"
function="1">
          <mappings>
            <mapping id="2358" icon="addRole" url="addRole" title="添加" function="0"

              <mappings/>
            </mapping>
            <mapping id="2359" icon="updateRole" url="updateRole" title="编辑" functi
n="0">
              <mappings/>
            </mapping>
            <mapping id="2360" icon="deleteRole" url="deleteRole" title="删除" function
```

```
"0">
                            <mappings/>
                        </mapping>
                    </mappings>
                </mapping>
            </mappings>
        </mapping>
        <mapping id="2329" icon="fa fa-calendar" url="/" title="学生管理" function="2">
            <mappings>
                <mapping id="1007" icon="fa fa-flag" url="/classManager" title="班级管理" functio
="3">
                    <mappings>
                        <mapping id="1008" icon="classList" url="/classManager/classList.ftl" title="
级列表" function="1">
                            <mappings>
                                <mapping id="1009" icon="addClass" title="添加" url="addClass" functio
="0">
                                    <mappings/>
                                </mapping>
                                <mapping id="1010" icon="seeClassStudent" title="详情" url="seeClassS
udent" function="0">
                                    <mappings/>
                                </mapping>
                                <mapping id="1011" icon="updateClass" title="编辑" url="updateClass" f
nction="0">
                                    <mappings/>
                                </mapping>
                                <mapping id="1012" icon="deleteClass" title="删除" url="deleteClass" fu
ction="0">
                                    <mappings/>
                                </mapping>
                            </mappings>
                        </mapping>
                        <mapping id="1013" icon="showStudentInfo" url="/studentManager/showSt
dentInfo.ftl" title="班级详情" function="1">
                            <mappings>
                                <mapping id="1014" icon="seeStudent" title="查看" url="seeStudent" fu
ction="0">
                                    <mappings/>
                                </mapping>
                                <mapping id="1015" icon="updateStudent" title="修改" url="updateStud
nt" function="0">
                                    <mappings/>
                                </mapping>
                                <mapping id="1016" icon="deleteStudent" title="删除" url="deleteStude
t" function="0">
                                    <mappings/>
                                </mapping>
                                <mapping id="1017" icon="projectScore" title="项目评分" url="projectSc
re" function="0">
                                    <mappings/>
                                </mapping>
                            </mappings>
                        </mapping>
                    </mappings>
                </mapping>
                <mapping id="6000" icon="fa fa-flag" url="/studentManager/jumpToAddStudent"
itle="信息录入"
                        function="3">
                    <mappings>
                        <mapping id="6001" icon="input" url="/studentManager/inputStudentInfo.ftl"
title="学生信息录入"
                                function="1">
                            <mappings>
                            </mappings>
                        </mapping>
                    </mappings>
                </mapping>
            </mappings>
        </mapping>
        <mapping id="5000" icon="fa fa-usd" url="/subsidy" title="补助管理" function="2">
            <mappings>
                <mapping id="5001" icon="subsidyManager" url="/subsidyManager" title="补助
请" function="3">
                    <mappings>
                        <mapping id="5002" icon="apply" url="/subsidyManager/applyList.ftl" title=
申请列表信息" function="1">
                            <mappings>
                                <mapping id="5003" icon="seeApply" title="查看" url="seeApply" functi
n="0">
                                    <mappings/>
                                </mapping>
                                <mapping id="5004" icon="deleteApply" title="删除" url="deleteApply" f
nction="0">
                                    <mappings/>
                                </mapping>
                                <mapping id="5005" icon="addApplyClass" title="+添加申请班级" url="a
dApplyClass"
                                        function="0">
                                    <mappings/>
                                </mapping>
                            </mappings>
                        </mapping>
                    </mappings>
                </mapping>
                <mapping id="5100" icon="approvalManager" url="/approvalManager" title="补
审批" function="3">
                    <mappings>
                        <mapping id="5101" icon="approval" url="approvalManager/approvalList.ftl"
itle="补助申请列表"
                                function="1">
                            <mappings>
                                <mapping id="5102" icon="seeApproval" title="查看" url="seeApproval"
unction="0">
                                    <mappings/>
                                </mapping>
                                <mapping id="5103" icon="updateApproval" title="同意" url="updateAg
eeApproval" function="0">
```

```
                  <mappings/>
                </mapping>
                <mapping id="5104" icon="deleteApproval" title="驳回" url="deleteDisa
reeApproval" function="0">
                  <mappings/>
                </mapping>
              </mappings>
            </mapping>
          </mappings>
        </mapping>
        <mapping id="5010" icon="subsidyManager" url="/subsidyManager/historyMana
er" title="申请历史" function="3">
          <mappings>
            <mapping id="5011" icon="apply" url="/subsidyManager/applicationHistoryLi
t.ftl" title="申请历史列表" function="1">
              <mappings>
                <mapping id="5012" icon="seeDetails" title="查看" url="seeDetails" funct
on="0">
                  <mappings/>
                </mapping>
              </mappings>
            </mapping>
          </mappings>
        </mapping>
      </mappings>
    </mapping>
    <mapping id="6100" icon="fa fa-usd" url="/obtainEmploymentManager" title="就业管
" function="2">
      <mappings>
        <mapping id="6110" icon="fa fa-usd" url="/studentEmploymentManager/unEmpl
yed" title="未就业" function="3">
          <mappings>
            <mapping id="6120" icon="approval" url="studentEmploymentManager/unE
ployedList.ftl"
                title="未就业学员信息表"
                function="1">
              <mappings>
                <mapping id="6130" icon="seeUnEmploy" title="详情" url="seeUnEmplo
" function="0">
                  <mappings/>
                </mapping>
                <mapping id="6131" icon="employmentTrack" title="追踪" url="employ
entTrack" function="0">
                  <mappings/>
                </mapping>
                <mapping id="6132" icon="employmentApproval" title="就业" url="emp
oymentApproval" function="0">
                  <mappings/>
                </mapping>
                <mapping id="6133" icon="employmentApproval" title="查看" url="seeD
tail" function="0">
                  <mappings/>
                </mapping>
              </mappings>
            </mapping>
          </mappings>
        </mapping>
        <mapping id="6200" icon="fa fa-usd" url="/studentEmploymentManager/Employ
d" title="已就业" function="3">
          <mappings>
            <mapping id="6210" icon="approval" url="studentEmploymentManager/emp
oyedList.ftl"
                title="已就业学员信息表"
                function="1">
              <mappings>
                <mapping id="6220" icon="seeEmploy" title="查看详情" url="seeEmploy"
function="0">
                  <mappings/>
                </mapping>
              </mappings>
            </mapping>
          </mappings>
        </mapping>
        <mapping id="6300" icon="fa fa-usd" url="/studentEmploymentManager/UnEmpl
ymentApply" title="就业审批" function="3">
          <mappings>
            <mapping id="6310" icon="approval" url="studentEmploymentManager/emp
oyApproveList.ftl"
                title="审批列表"
                function="1">
              <mappings>
                <mapping id="6320" icon="seeApproval" title="查看" url="seeEmpApp" f
nction="0">
                  <mappings/>
                </mapping>
                <mapping id="6321" icon="seeApproval" title="同意" url="agreeEmpApp
 function="0">
                  <mappings/>
                </mapping>
                <mapping id="6322" icon="seeApproval" title="不同意" url="disagreeEm
App" function="0">
                  <mappings/>
                </mapping>
              </mappings>
            </mapping>
          </mappings>
        </mapping>
      </mappings>
    </mapping>
    <mapping id="2330" icon="fa fa-usd" url="/repaymentManager" title="还款管理" functi
n="2">
      <mappings>
        <mapping id="3700" icon="repaymentList" title="还款管理" url="/repaymentMana
er/repaymentList.ftl"
            function="1">
          <mappings>
            <mapping id="3701" icon="addRepayment" url="addRepayment" title="添加"
unction="0">
```

```
                            <mappings/>
                        </mapping>
                        <mapping id="3702" icon="updateRepayment" url="updateRepayment" title=
修改" function="0">
                            <mappings/>
                        </mapping>
                        <mapping id="3703" icon="deleteRepayment" url="deleteRepayment" title=
删除" function="0">
                            <mappings/>
                        </mapping>
                    </mappings>
                </mapping>
            </mappings>
        </mapping>
        <mapping id="3851" icon="fa fa-crosshairs" url="/attendanceManager/personalAttend
nce" title="我的考勤"
                 function="2">
            <mappings>
                <mapping id="3852" icon="personalAttendanceList" url="/attendanceManager/per
onalAttendanceList.ftl"
                        title="考勤信息列表" function="1">
                    <mappings/>
                </mapping>
                <mapping id="3853" icon="personalAttendanceChart" url="/attendanceManager/
ersonalAttendanceChart.ftl"
                        title="考勤信息图表" function="1">
                    <mappings/>
                </mapping>
            </mappings>
        </mapping>
        <mapping id="2331" icon="fa fa-crosshairs" url="/attendanceManager/attendance" titl
="出勤管理" function="2">
            <mappings>
                <mapping id="2344" icon="attendanceList" url="/attendanceManager/attendanceL
st.ftl" title="考勤信息列表"
                        function="1">
                    <mappings>
                        <mapping id="2364" icon="addAttendance" url="addAttendance" title="添加
情" function="0">
                            <mappings/>
                        </mapping>
                        <mapping id="2365" icon="updateAttendance" url="updateAttendance" title
"查看详情" function="0">
                            <mappings/>
                        </mapping>
                        <mapping id="2366" icon="deleteStudent" url="deleteStudent" title="删除" f
nction="0">
                            <mappings/>
                        </mapping>
                    </mappings>
                </mapping>
                <mapping id="3854" icon="attendanceChart" url="/attendanceManager/attendanc
Chart.ftl" title="考勤信息图表"
                        function="1">
```

```xml
          <mappings/>
        </mapping>
      </mappings>
    </mapping>
    <mapping id="2332" icon="fa fa-book" url="/certificateManager" title="证书管理" funct
on="2">
      <mappings>
        <mapping id="2004" icon="certificateList" url="/certificateManager/certificateList.ft
" title="证书管理"
             function="1">
          <mappings>
            <mapping id="2005" icon="addCertificate" url="addCertificate" title="添加" fu
ction="0">
              <mappings/>
            </mapping>
            <mapping id="2006" icon="updateCertificate" url="updateCertificate" title="
改" function="0">
              <mappings/>
            </mapping>
            <mapping id="2007" icon="deleteCertificate" url="deleteCertificate" title="删
" function="0">
              <mappings/>
            </mapping>
          </mappings>
        </mapping>
      </mappings>
    </mapping>
    <mapping id="2333" icon="fa fa-pencil" url="/test" title="我的考试" function="2">
      <mappings>
        <mapping id="2345" icon="testHistory" url="/test/history" title="考试历史" functio
="3">
          <mappings>
            <mapping id="2367" icon="historyList" url="/test/historyList.ftl" title="考试历
" function="1">
              <mappings/>
            </mapping>
          </mappings>
        </mapping>
        <mapping id="2346" icon="toTest" url="/test/to" title="参加考试" function="3">
          <mappings>
            <mapping id="2368" icon="testing" url="/test/testing.ftl" title="正在进行的考
" function="1">
              <mappings/>
            </mapping>
          </mappings>
        </mapping>
        <mapping id="3000" icon="toQuestions" title="题库" url="/test/questions" functio
="3">
          <mappings>
            <mapping id="3001" icon="questions" url="/test/questions.ftl" function="1" ti
le="中心题库">
              <mappings>
                <mapping id="3007" icon="addTestQuestion" url="addTestQuestion" fu
ction="0" title="添加">
```

```
                  <mappings/>
                </mapping>
                <mapping id="3008" icon="updateTestQuestion" url="updateTestQuesti
n" function="0"
                    title="修改">
                  <mappings/>
                </mapping>
                <mapping id="3009" icon="deleteTestQuestion" url="deleteTestQuestion
 function="0"
                    title="删除">
                  <mappings/>
                </mapping>
                <!--<mapping id="3010" icon="addToTest" url="addToTest" function="0
 title="添加到试卷">
                  <mappings/>
                </mapping>
                <mapping id="3011" icon="testManage" url="testManage" function="0"
itle="当前试卷管理">
                  <mappings/>
                </mapping>-->
              </mappings>
            </mapping>
          </mappings>
        </mapping>
        <mapping id="105" icon="tagManager" title="标签分类管理" url="/test/tagManage
" function="3">
          <mappings>
            <mapping id="106" icon="domainList" title="大分类列表" url="/test/domainLis
.ftl" function="1">
              <mappings>
                <mapping id="107" icon="addDomain" title="添加" url="addDomain" fu
ction="0">
                  <mappings/>
                </mapping>
                <mapping id="108" icon="updateDomain" title="修改" url="updateDoma
n" function="0">
                  <mappings/>
                </mapping>
                <mapping id="109" icon="deleteDomain" title="删除" url="deleteDomai
" function="0">
                  <mappings/>
                </mapping>
              </mappings>
            </mapping>
          </mappings>
        </mapping>
        <mapping id="3002" icon="testList" title="试卷列表" url="/test/testList" function="
">
          <mappings>
            <mapping id="3003" icon="testList" title="试卷列表" url="/test/testList.ftl" fun
tion="1">
              <mappings>
                <mapping id="3020" icon="addTestQuestionnaire" title="添加试卷"
                    url="addTestQuestionnaire"
                    function="0">
                  <mappings/>
                </mapping>
                <!--<mapping id="3028" icon="distributeTestQuestionnaire" title="分配
卷"
                    url="distributeTestQuestionnaire"
                    function="0">
                  <mappings/>
                </mapping>-->
                <mapping id="3021" icon="updateTestQuestionnaire" title="修改"
                    url="updateTestQuestionnaire"
                    function="0">
                  <mappings/>
                </mapping>
                <mapping id="3022" icon="deleteTestQuestionnaire" title="删除"
                    url="deleteQuestionnaire"
                    function="0">
                  <mappings/>
                </mapping>
              </mappings>
            </mapping>
            <mapping id="2025" icon="disList" title="已分配试卷" url="/test/disList.ftl" fun
tion="1">
              <mappings/>
            </mapping>
          </mappings>
        </mapping>
        <mapping id="3004" icon="toCount" title="成绩统计" url="/test/count" function="
">
          <mappings>
            <mapping id="3005" icon="count" title="试卷列表" url="/test/gradeList.ftl" fu
ction="1">
              <mappings>
                <mapping id="1035" icon="delayTest" url="delayTest" title="开启提交" f
nction="0">
                  <mappings/>
                </mapping>
                <mapping id="1036" icon="closeTest" url="closeTest" title="关闭提交" fu
ction="0">
                  <mappings/>
                </mapping>
                <mapping id="237" icon="updateReply" url="updateReply" title="提交
卷信息" function="0">
                  <mappings/>
                </mapping>
              </mappings>
            </mapping>
            <mapping id="3006" icon="count" title="统计图表" url="/test/gradeChart.ftl" f
nction="1">
              <mappings/>
            </mapping>
          </mappings>
        </mapping>
      </mappings>
```

```
    </mapping>
    <mapping id="2334" icon="fa fa-file-text" url="/#" title="我的调查" function="2">
      <mappings>
        <mapping id="2347" icon="toSurveying" url="/surveys/surveyList" title="调查列表"
unction="3">
          <mappings>
            <mapping id="2369" icon="surveying" url="/surveys/surveying.ftl" title="正
进行的调查" function="1">
              <mappings/>
            </mapping>
          </mappings>
        </mapping>
        <mapping id="2348" icon="toInputBigType" url="/surveys/inputBigType" title="
加大类型" function="3">
          <mappings>
            <mapping id="2370" icon="inputBigType" url="/surveys/inputBigType.ftl" titl
="添加大类型"
                 function="1">
              <mappings/>
            </mapping>
          </mappings>
        </mapping>
        <mapping id="2349" icon="toInputQuestion" url="/surveys/inputQuestion" title=
添加问题" function="3">
          <mappings>
            <mapping id="2371" icon="inputQuestion" url="/surveys/inputQuestion.ftl" tit
e="添加问题"
                 function="1">
              <mappings/>
            </mapping>
          </mappings>
        </mapping>
        <mapping id="2350" icon="toInputQuestionnaire" url="/surveys/inputQuestionnai
e" title="组卷"
             function="3">
          <mappings>
            <mapping id="2372" icon="inputQuestionnaire" url="/surveys/inputQuestion
aire.ftl" title="组卷"
                 function="1">
              <mappings/>
            </mapping>
          </mappings>
        </mapping>
        <mapping id="2351" icon="toCheckResult" url="/surveys/checkResult" title="查看
查结果" function="3">
          <mappings>
            <mapping id="2373" icon="checkResult" url="/surveys/checkResult.ftl" title=
查看调查结果"
                 function="1">
              <mappings/>
            </mapping>
          </mappings>
        </mapping>
      </mappings>
```

```
        </mapping>
        <mapping id="2335" icon="fa fa-gear" url="/option" title="通用设置" function="2">
            <mappings>
                <mapping id="2352" icon="updatePurikura" url="/option/updatePurikura.ftl" title=
修改头像" function="1">
                    <mappings/>
                </mapping>
                <mapping id="2353" icon="updatePassword" url="/option/updatePassword.ftl" titl
="修改密码" function="1">
                    <mappings/>
                </mapping>
            </mappings>
        </mapping>
        <mapping id="2336" icon="fa fa-thumbs-up" url="/enrollManager" title="报名学生" fun
tion="2">
            <mappings>
                <mapping id="2354" icon="enrollList" url="/enrollManager/enroll.ftl" title="报名
生列表" function="1">
                    <mappings>
                        <mapping id="2374" icon="addEnroll" url="addEnroll" title="添加报名学生" fu
ction="0">
                            <mappings/>
                        </mapping>
                        <mapping id="2375" icon="updateEnroll" url="updateEnroll" title="修改" funct
on="0">
                            <mappings/>
                        </mapping>
                        <mapping id="2376" icon="deleteEnroll" url="deleteEnroll" title="删除" functi
n="0">
                            <mappings/>
                        </mapping>
                    </mappings>
                </mapping>
            </mappings>
        </mapping>
    </mappings>
</mapping>
[b96984f14cd146c496e40871d973f594-permission.xml](https://b3logfile.com/file/2016/12/b
6984f14cd146c496e40871d973f594-permission.xml)
```

BaseController完成基础权限地图处理操作

```
//基础渲染方法
public void index() {
    fillHeaderAndFooter();
    if (!fillContent()) {
        renderError(403);
        return;
    }
    render("/index.ftl");
}

/**
 * @return 返回一级菜单的mapping
 */
 public abstract Mapping init();

public void fillHeader() {
    //三个地址：servePath用于得到去掉参数的网址、holdPath为带参数网址
  String uri = getRequest().getRequestURI();
    String url = String.valueOf(getRequest().getRequestURL());
    String staticPath = getAttr(Common.LABEL_STATIC_SERVE_PATH);

    //将不需要的参数忽略掉
  String para = StrPlusKit.ignoreQueryString(getRequest().getQueryString(), "_pjax", "list_p", "c
art_p", "p");
    if (!StrPlusKit.isEmpty(para))
        para = "?" + para;
    String actionKey = getAttr(Common.LABEL_ACTION_KEY);
    String servePath = staticPath + actionKey;
    if (para != null)
        url += para;
    setAttr(Common.LABEL_SERVE_PATH, servePath);
    setAttr(Common.LABEL_HOLD_PATH, url);
    setAttr(Common.LABEL_STATIC_RESOURCE_VERSION, new Date().getTime());
    User currentUser = getCurrentUser(this);

    setAttr(Common.LABEL_IS_LOGIN, currentUser == null);
    setAttr(Common.LABEL_LOGIN_ROLE, currentUser != null ? currentUser.getUserRole() : "");
    setAttr(Common.LABEL_USER, currentUser);
    Prop langProp = LangConfig.getLangProp();
    setAttr(Common.LABEL_LOGIN_NAME_ERROR, langProp.get(Common.LABEL_LOGIN_NAME
ERROR));
    setAttr(Common.LABEL_INVALID_PASSWORD, langProp.get(Common.LABEL_INVALID_PAS
WORD));
}

public void fillFooter() {
    /*Prop langProp = LangConfig.getLangProp();
 Enumeration elements = langProp.getProperties().elements(); while(elements.hasMoreEleme
ts()) { Object o = elements.nextElement(); //System.out.println(o); }*/}

//页面测试
protected void fillTest() {

    List<Mapping> mappings = mappingService.getTree().getList();
    mappings.remove(0);

    setAttr(Common.LABEL_SIDES, mappings);

}

protected boolean fillContentParent() {
    User user = getCurrentUser(getRequest());
    if (user == null) {
        forwardAction("/user/showLogin");
        return false;
    }
    if (mapping == null) {
        renderError(403);
        return false;
    }

    ArrayTree<Mapping> roleTree = roleService.getRoleTree(roleService.getRoleByName(user.
etUserRole()));
    List<Mapping> sides = new ArrayList<>();
    List<Mapping> childSides = new ArrayList<>();
    List<Mapping> views = new ArrayList<>();
    final int[] size = new int[Common.MAX_DEGREE];
    roleTree.checkTreePreorder(now -> {
        if (roleTree.getParent(now) == null)
            return true;
        //子菜单计数,只支持二级菜单。。。子菜单下继续遍历子视图
  if (now.getFunction() > Mapping.FUNCTION_MENUITEM) {
            size[sides.indexOf(now.getParent())]++;
            childSides.add(now);
            //子菜单为当前点击菜单才会继续向子视图遍历
  return now == mapping;
        }
        //一级菜单
  if (now.getFunction() == Mapping.FUNCTION_MENUITEM) {
            sides.add(now);
            return true;
        }
        //视图遍历，遍历到一级视图停止遍历,并添加到视图链表，以便后续功能或子视图的遍历处理
  if (now.getParent() == mapping && now.getFunction() == Mapping.FUNCTION_VIEW) {
            views.add(now);
            return false;
        }
        //如果同层不同访问，则其他同层节点子节点放弃遍历
  /*if(mappingService.getBaseMenu(now) == mapping)
 return true; return mappingService.getBaseMenu(now) == mapping;*/  return false;
    });
    setAttr(Common.LABEL_SIDES, sides);
    setAttr(Common.LABEL_SIDES_SIZE, size);
    setAttr(Common.LABEL_SIDES_CHILD, childSides);
    setAttr(Common.LABEL_VIEWS, views);
    setAttr(Common.LABEL_NOW_VISIT, mapping);
    //base处理通用的，其他处理继续下放
  setAttr(Common.LABEL_ROLE_TREE, roleTree);
    setAttr(Common.LABEL_ROOT_MAPPING, roleTree.root());
    return true;
}

//如果没有子视图模块，则可以使用通用的操作遍历
protected boolean fillContentChild() {
    User user = getCurrentUser(getRequest());
    if (user == null) {
        forwardAction("/user/showLogin");
        return false;
    }
    if (mapping == null) {
        renderError(403);
        return false;
    }
    ArrayTree<Mapping> roleTree = roleService.getRoleTree(roleService.getRoleByName(user.
etUserRole()));
    List<Mapping> sides = getAttr(Common.LABEL_VIEWS);
    Map<String, List<Mapping>> map = new HashMap<>();
    for (Mapping side : sides) {
        List<Mapping> operators = new ArrayList<>();
        roleTree.checkTree(side, now -> {
            if (now.getFunction() == Mapping.FUNCTION_OPERATE) {
                operators.add(now);
                return false;
            }
            return true;
        });
        map.put("operators" + side.getId(), operators);
    }
    setAttr("map", map);
    return true;
}

protected boolean fillContent() {
    return fillContentParent() && fillContentChild();
}

public void fillHeaderAndFooter() {
    fillHeader();
    fillFooter();
}
```

然后controller几乎可以直接使用super.index()，完成大部分权限地图处理操作（同时也生成了菜单视图，后来的开发人员也能够很简单的使用，直接专注于功能，而要达到这些，只需要在xml文件中单添加菜单

视图信息就行了）

classController示例

```
public class ClassController extends BaseController {
    public ClassService classService;
    public StudentService studentService;
    public UserService userService;
    public RegionService regionService;

    public void index() {
        //注意此方法~
        super.index();
        List<Mapping> views = new ArrayList<>();
        Mapping mapping = mappingService.getMappingByUrl("/classManager/classList.ftl");
        views.add(mapping);
        setAttr(Common.LABEL_VIEWS, views);
        Integer p = getParaToInt("p", 1);
        Page<Class> allClass = classService.getAllClass(p);
        setAttr("classes", allClass);

        List<Region> regionList = regionService.getAllRegions();
        setAttr("regionList", regionList);
    }

    /**
 * @return 返回一级菜单的mapping
 */  @Override
  public Mapping init() {
        return mappingService.getMappingByUrl("/classManager");
    }

    @Before(POST.class)
    public void addClass() {
        Class model = getModel(Class.class);
        boolean flag;
        if (model.getId() == null)
            flag = model.save();
        else {
            flag = model.update();
        }

        if (!flag) {
            RenderKit.renderError(this, "保存班级失败");
            return;
        }
        RenderKit.renderSuccess(this, "保存班级成功");
    }

    public void getClassStudents() {
        Integer classId = getParaToInt("classId");
        List<Student> studentByClassId = studentService.getStudentByClassId(classId, Student.S
ATUS_STUDYING);

    }

    @Before(POST.class)
    public void deleteClass() {
        Integer id = getParaToInt(0);
        if (id == null) {
            RenderKit.renderError(this, "该班级不存在或已被删除");
            return;
        }
        Class aClass = classService.getClassById(id);
        if (aClass == null) {
            RenderKit.renderError(this, "该班级不存在或已被删除");
            return;
        }
        if (aClass.delete()) {
            List<Student> studentList = studentService.getAllStudentByClassId(aClass.getId());
            if (studentList != null) {
                for (Student student : studentList) {
                    User stuAccount = userService.getUserByStuPhone(student);
                    studentService._deleteStudentById(student.getId());
                    userService._deleteUser(stuAccount);
                }
            }
            RenderKit.renderSuccess(this, "班级以及学生信息删除成功！");
            return;
        }
        RenderKit.renderError(this, "删除不成功！");
    }

    public void letGraduate() {
        String jsonStuIdList = getPara("classStuIdList");
        int clsId = getParaToInt("clsId");
        System.out.println(jsonStuIdList);
        JSONArray jsonStuIdArray = JSON.parseArray(jsonStuIdList);
        boolean allComplete = true;
        for (Object o : jsonStuIdArray) {
            if (o == null)
                continue;
            int id = Integer.valueOf(o.toString());
            Student student = studentService.getUnEmpStudentById(id);
            if (student != null) {
                student.setStatus(Student.STATUS_GRADUATION);
                student.setEmploymentStatus(Student.EMPLOYMENTSTATUS_UN_EMPLOYED);
                student.setRemark("毕业啦！");
                boolean b = studentService._updateStudentById(student);
                if (!b) {
                    allComplete = false;
                }
            }
        }
        Class aClass = classService.getClassById(clsId);
        boolean b = true;
        boolean isG = false;
        if (aClass != null) {
            if (aClass.getStatus() == Class.CLASS_STATUS_GRADUATED) {
                isG = true;
            }
            aClass.setStatus(Class.CLASS_STATUS_GRADUATED);
            aClass.setRemark("毕业班！");
            b = classService._updateClass(aClass);
        }
        if (allComplete && b) {
            if (isG) {
                RenderKit.renderError(this, "该班级已经毕业，不建议反复提交！");
            } else {
                RenderKit.renderSuccess(this, "操作成功！");
            }
        } else {
            RenderKit.renderError(this, "操作存在异常！");
        }
    }

}
```

而在前台（这里使用的freemarker），就拥有一个遍历的机会了，这样把每个模块也就分开了，最主还是，不能显示的导航也去掉了，在单独的模块里面可以继续做类似的遍历。

填充侧边导航

```
  <div class="tip-container">
        <ul class="nav" id="main-menu">

            <#list sides as side>
                <li>
                        <a href="${side.url}"><i class="${side.icon}"></i>${side.title}</a>
                </li>
            </#list>
        </ul>
    </div>
```

填充内容

```
<div id="page-wrapper">
    <div id="page-inner">
    <#list content as c>
     <#include c.url>
   </#list>
   </div>
</div>
```

以上使用的JFinal框架，大同小异，我二次开发过sym，简单说明一下，这里没有用latke那种先取模再在model中设置参数（因为JFinal在这方面没有专门的使用json），而是直接在request里面设置参，然后渲染模板（这是基类，渲染在子类里，这里没有）。

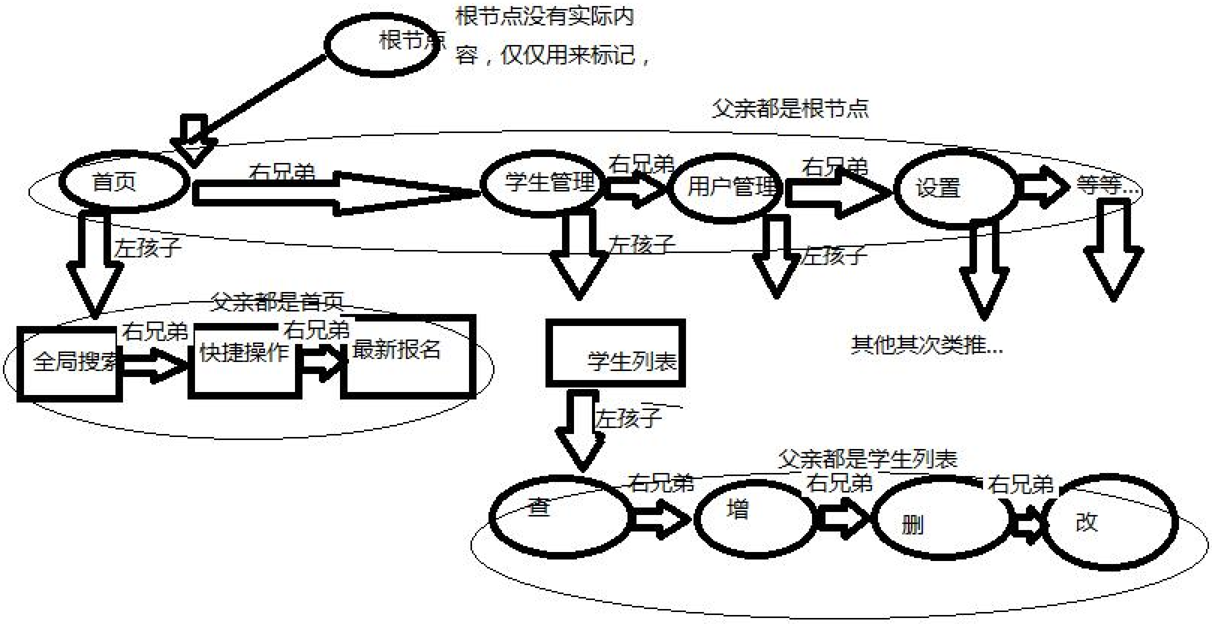
子类重写渲染示例

```
    /**
     * 重写index方法，渲染为自己的首页
     */
    @Override
    public void index() {
        super.index();

        //对roleTree继续做处理
        ...

        //----处理结束----

        render("index.ftl");
    }
```

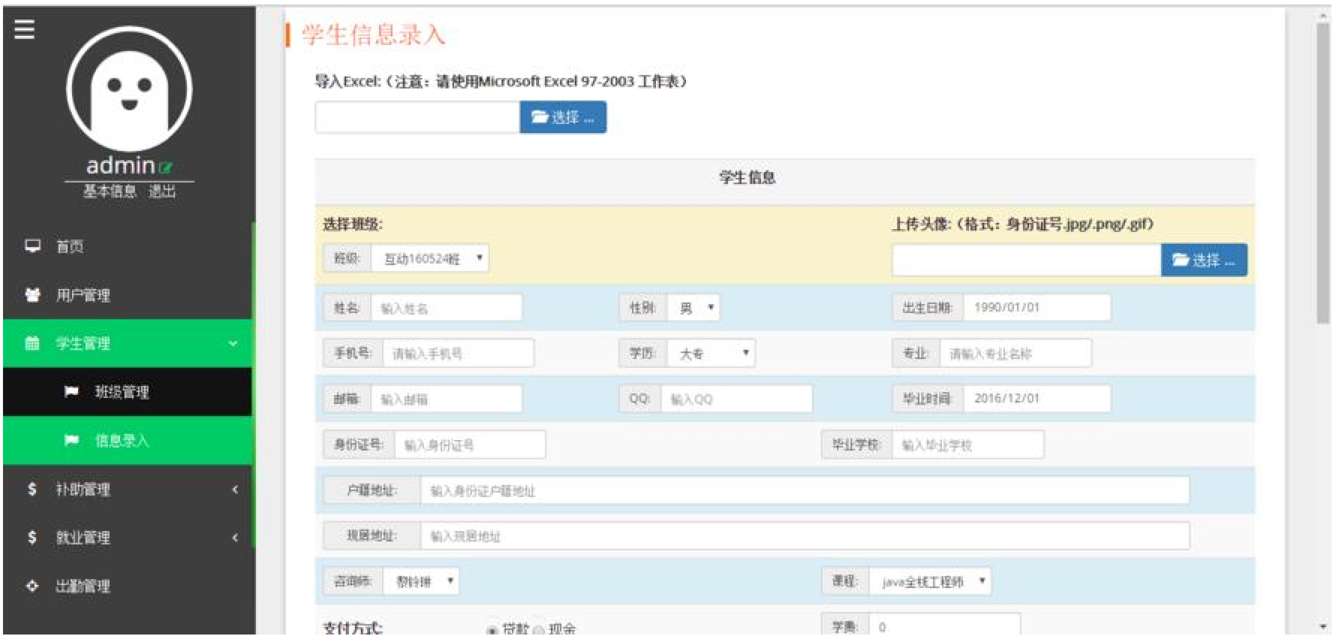


鉴于有人看不懂代码，那么明确的给出一个树结构图。
但是画的丑，不要介意~

强调一下，这不是二叉树...是用的二叉链表存储树结构只是~~一种存储方式而已~~

前端界面截图

管理员视图



学生信息录入

导入Excel:（注意：请使用Microsoft Excel 97-2003 工作表）

学生信息

选择班级:

班级: 互动160524班

上传头像:（格式：身份证号.jpg/.png/.gif）

姓名: 输入姓名 性别: 男 出生日期: 1990/01/01

手机号: 请输入手机号 学历: 大专 专业: 请输入专业名称

邮箱: 输入邮箱 QQ: 输入QQ 毕业时间: 2016/12/01

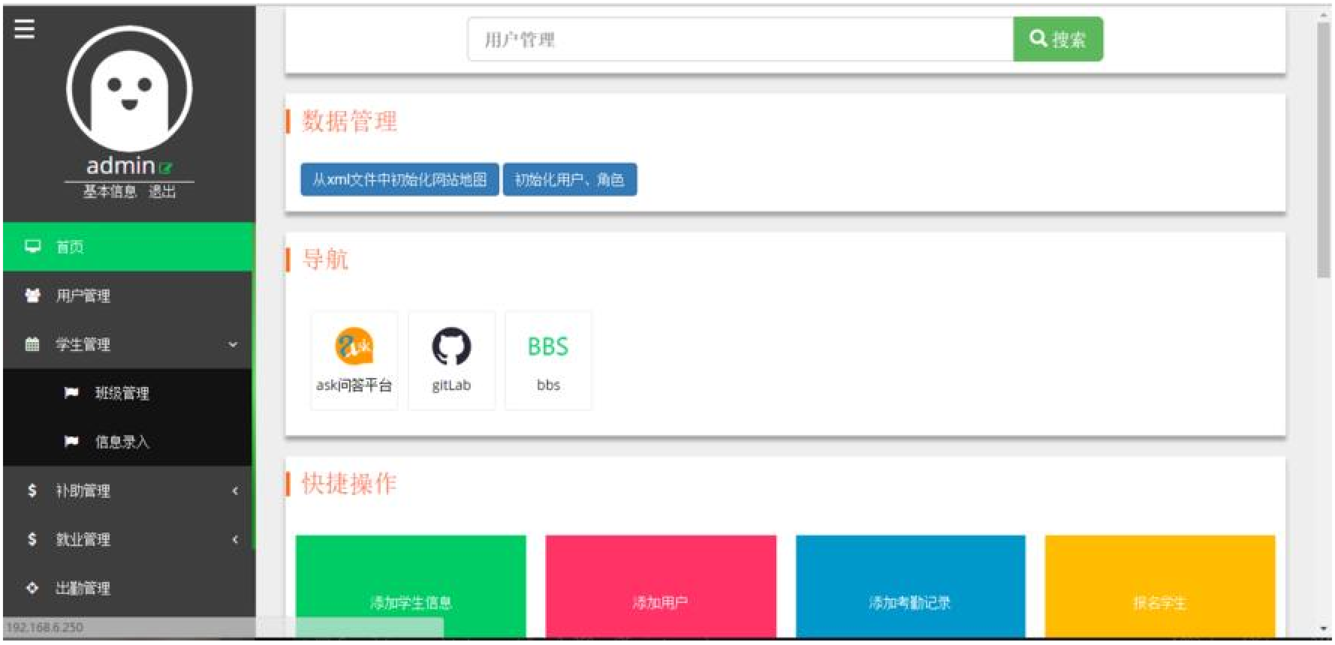
身份证号: 输入身份证号 毕业学校: 输入毕业学校

户籍地址: 输入身份证户籍地址

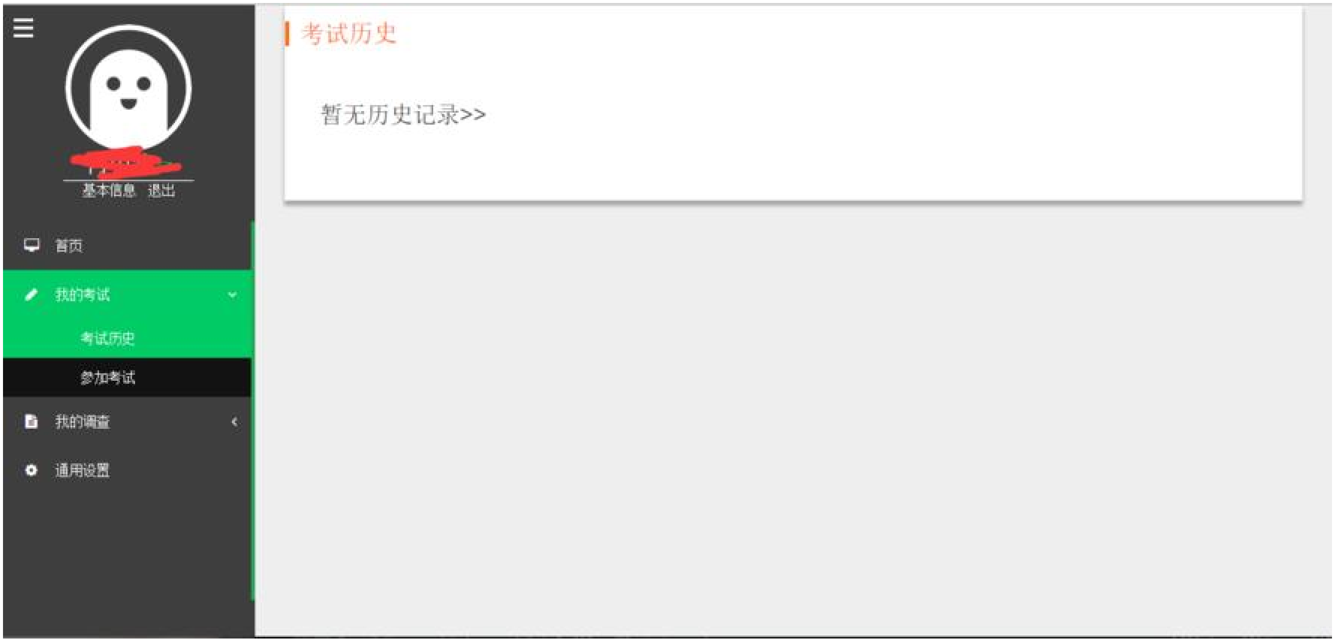
现居地址: 输入现居地址

咨询师: 黎特珊 课程: java全栈工程师

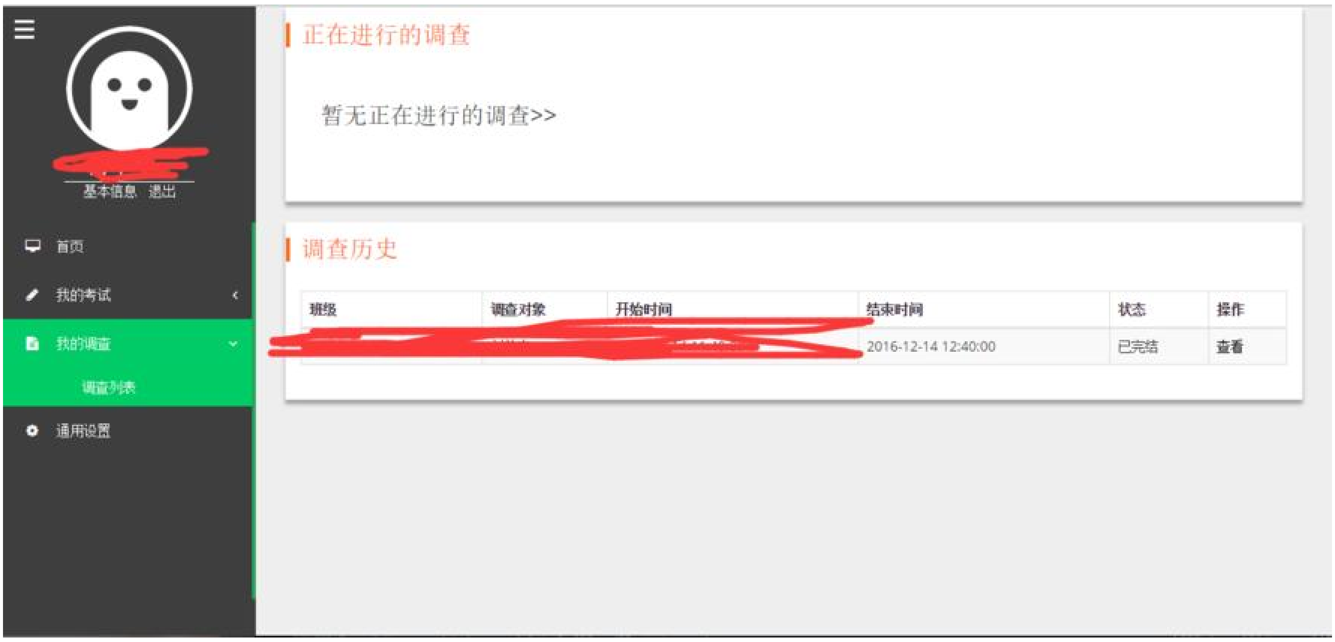
支付方式: ● 贷款 ○ 现金 学费: 0



普通学生视图



原文链接：怎样构建一个灵活的权限系统



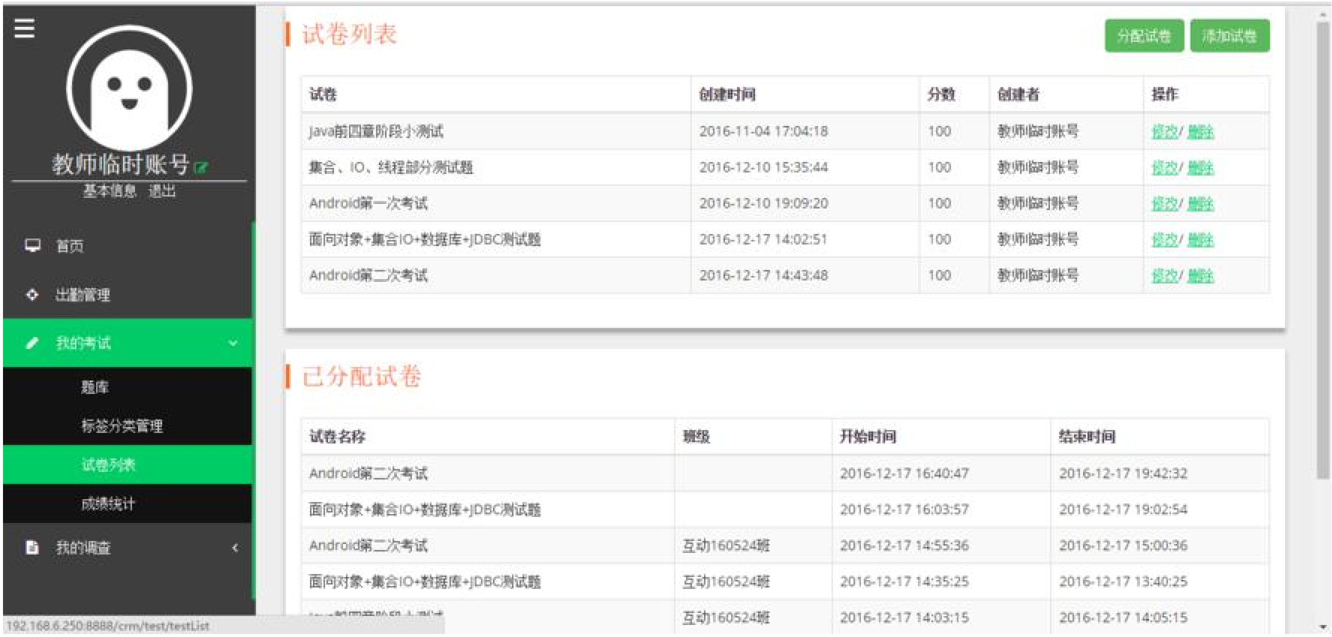
正在进行的调查

暂无正在进行的调查>>

调查历史

班级	调查对象	开始时间	结束时间	状态	操作
[redacted]	[redacted]	[redacted]	2016-12-14 12:40:00	已完结	查看

教师视图



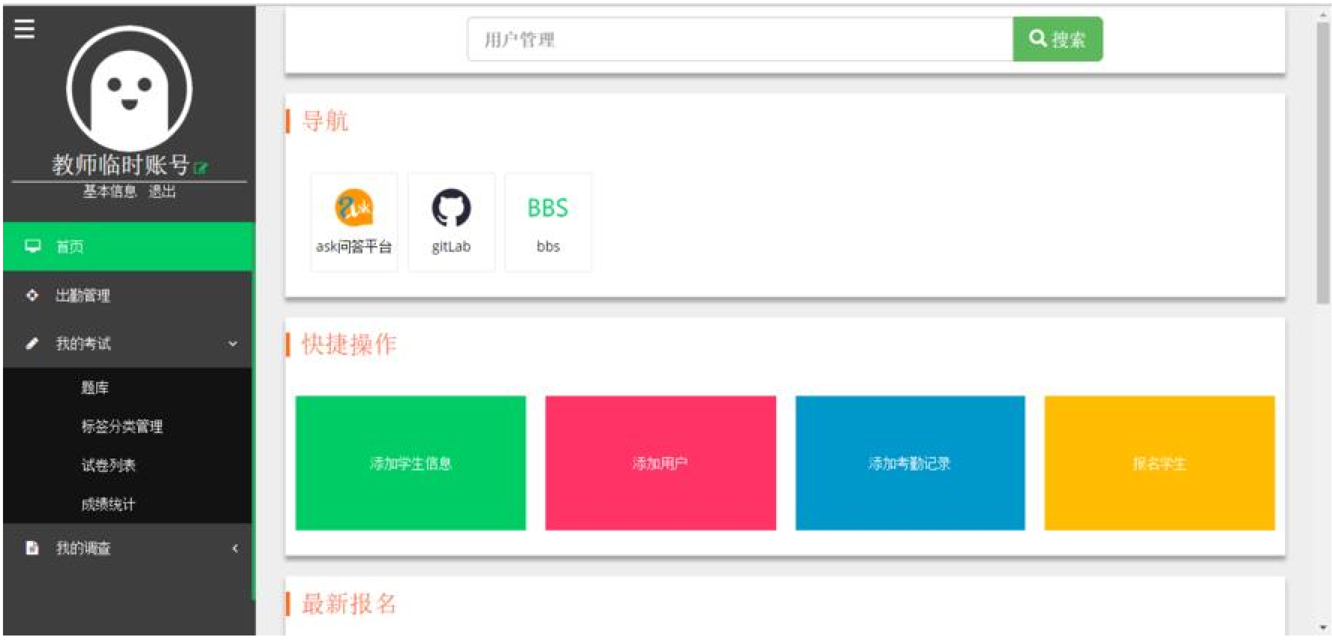
试卷列表

试卷	创建时间	分数	创建者	操作
Java前四章阶段小测试	2016-11-04 17:04:18	100	教师临时账号	修改/ 删除
集合、IO、线程部分测试题	2016-12-10 15:35:44	100	教师临时账号	修改/ 删除
Android第一次考试	2016-12-10 19:09:20	100	教师临时账号	修改/ 删除
面向对象+集合IO+数据库+JDBC测试题	2016-12-17 14:02:51	100	教师临时账号	修改/ 删除
Android第二次考试	2016-12-17 14:43:48	100	教师临时账号	修改/ 删除

已分配试卷

试卷名称	班级	开始时间	结束时间
Android第二次考试		2016-12-17 16:40:47	2016-12-17 19:42:32
面向对象+集合IO+数据库+JDBC测试题		2016-12-17 16:03:57	2016-12-17 19:02:54
Android第二次考试	互动160524班	2016-12-17 14:55:36	2016-12-17 15:00:36
面向对象+集合IO+数据库+JDBC测试题	互动160524班	2016-12-17 14:35:25	2016-12-17 13:40:25
Java前四章阶段小测试	互动160524班	2016-12-17 14:03:15	2016-12-17 14:05:15

原文链接：怎样构建一个灵活的权限系统



原文链接：怎样构建一个灵活的权限系统