Evan
Only let oneself become strong enough, good enough, can afford the life that you want to.

EasyUI Tree递归方式获取JSON

标签： easyui tree json

2016-12-21 18:43 452人阅读 评论(0) 收藏 编辑 删除

分类： JQuery (9)

最近需要用到EASYUI中的TREE功能，以前我是直接拼接成<UL><LI>发现这样拼完之后在更改树后对树的刷新不是很理想，现改用JSON格式，首先分析TREE中JOSN格式如下：

```
[{
        "id":1,
        "text":"流程分类",
        "children":[{
                "id":11,
                "text":"门禁流程分类",
                "checked":true
        },{
                "id":113,
                "text":"子门禁流程分类",
                "children":[{
                        "id":1131,
                        "text":"子子门禁流程分类"
                },{
                        "id": 8,
                        "text":"Async Folder",
                        "state":"closed"
                }]
        }]
},{
        "id":3
        "text":"行政",
        "children":[{
                "id":"31",
                "text":"加班"
```

```
26        },{
27                "id":"33",
28                "text":"请假"
29        }]
30}]
```

可以看出这种模式是由三个属性所组成，ID TEXT 集合，根据分析 我们需要对此模式建立一个BEAN的结构模型，建立TREENODE：

```
1 packagecom.odbpo.beans;
2
3 importjava.util.List;
4
5 publicclassTreeNode {
6
7         privateintid;
8
9         privateString text;
10
11        privateintpid;
12
13        privateList<TreeNode> children;
14
15        publicintgetPid() {
16                returnpid;
17        }
18
19        publicvoidsetPid(intpid) {
20                this.pid = pid;
21        }
22
23        publicintgetId() {
24                returnid;
25        }
26
27        publicvoidsetId(intid) {
28                this.id = id;
29        }
30
31        publicString getText() {
32                returntext;
33        }
34
```

```
35        publicvoidsetText(String text) {
36                this.text = text;
37        }
38
39        publicList<TreeNode> getChildren() {
40                returnchildren;
41        }
42
43        publicvoidsetChildren(List<TreeNode> children) {
44                this.children = children;
45        }
46}
```

BEAN构建完成，那么接下来分析如何往BEAN里传数据，首先分析 **数据库**表中结构

```
1createtabledepatment(
2id,--当前ID
3pid,--父ID
4name--显示名称
5)
```

接下来我们要建立一个COM.UTIL包，所递归方法放置在这个包下，以便后续多次调用方便

建立类名为：JSONFACTORY

```
1 /*
2           * 以对象形式传回前台
3           */
4         publicstaticList<TreeNode> buildtree(List<TreeNode> nodes,intid){
5                 List<TreeNode> treeNodes=newArrayList<TreeNode>();
6                 for(TreeNode treeNode : nodes) {
7                         TreeNode node=newTreeNode();
8                         node.setId(treeNode.getId());
9                         node.setText(treeNode.getText());
10                        if(id==treeNode.getPid()){
11                                node.setChildren(buildtree(nodes, node.getId()));
12                                treeNodes.add(node);
13                        }
14
15                }
16                returntreeNodes;
17        }
18
```

完成以上工作后我们就要在控制器中使用在DAO中建立好的查询方法，这里DAO中写法就不细说了；

控制器写法如下：

```
@RequestMapping("/flow_tree")
@ResponseBody
publicList<TreeNode> getTree(){
        List<TreeNode> nodes=newArrayList<TreeNode>();
        List<FlowSortTable> list_all=flowSortTableServiceImpl.findAll();
        for(FlowSortTable flowSortTable : list_all) {
                TreeNode treeNode=newTreeNode();
                treeNode.setId(flowSortTable.getSortId());
                treeNode.setPid(flowSortTable.getSortPartmentId());
                treeNode.setText(flowSortTable.getSortName());
                nodes.add(treeNode);
        }
        List<TreeNode> treeNodes=JsonTreeFactory.buildtree(nodes,0);
        returntreeNodes;
}
```

以上工作结束，我们就可以在前台使用EASYUITREE模式了

将此代码 放在$(document).ready(function(){})中

```
$("#tt1").tree({
                        url:'${contextPath}/main/flow/flow_tree.html',
                        onClick:function(node){
                                $("#sort").css("display","block");
                                $("#save").hide();
                                $("#update").show();
                                odbpo_combobox("#flowType",'${contextPath}/main/flow/flowSelect.html',"flowId","flowName");
                                varpnode=$(this).tree('getParent',node.target);
                                $("#flowType").combobox('setValue', pnode.id);
                                $("#node_id").val(node.id);
                                $("#node_text").val(node.text);
                                console.debug(node.id);
                                console.debug(node.text);
                        }
                })
```

HTML构建：

```
<ulid="tt1">

            </ul>
```

启动TOMCAT预览就可以看到一个树形图的效果了！