

第一章 EasyUI 框架使用

1.1 ComboBox 组建的使用

1、JSP 页面

```
<input id="politicalstate" name=" politicalstate "/>
```

2、JavaScript 操作

```
$("#unit").combobox({
    url: 'SystemEmployeeService?caozuo=init politicalstate ',
    valueField: politicalstateid,
    textField: politicalstatename,
    panelHeight: 'auto'
});
```

3、Servlet 操作

从 Servlet 返回 json 数组:

```
[{"politicalstatename":"中共党员","politicalstateid":"001"},
 {"politicalstatename":"共青团员","politicalstateid":"002"}]
```

注意:

表单序列化时, 取 valueField 属性对应得值。

1.2 DataGrid 组件使用

1、JSP 页面构建

(1) 导入相应的 JS、CSS 文件:

```
<link type="text/css" rel="stylesheet" href="<%=basePath %>css/easyui.css" />
<link type="text/css" rel="stylesheet" href="<%=basePath %>css/icon.css" />

<script type="text/javascript" src="<%=basePath %>js/jquery.min.js"></script>
<script type="text/javascript" src="<%=basePath %>js/jquery.easyui.min.js"></script>
<script type="text/javascript" src="<%=basePath %>js/easyui-lang-zh_CN.js"></script>
```

(2) 加入 DataGrid 组件

■ 加入并设置: 创建JSP文件

```
<table id="info" class="easyui-datagrid" width="100%" style="height:100%;" border="0" cellpadding="0"
    cellspacing="0" data-options=" toolbar:'#tb'">
<thead>
<tr>
<th data-options="field:'sno',width:120,align:'center'">学 号</th>
<th data-options="field:'name',width:120,align:'center'">姓 名</th>
<th data-options="field:'age',width:100,align:'center'">年 龄</th>
```

```

<th data-options="field:'address',width:200,align:'center'">地 址</th>
<th data-options="field:'phone',width:100,align:'center',hidden:'true'">联系方式</th>
<th data-options="field:'institute',width:200,align:'center'">学 院</th>
<th data-options="field:'demo',width:180,align:'center'">备 注</th>
</tr>
</thead>
</table>

```

```

[{"sno":"20150001","name":"ahngsan"},
 {"sno":"20150002","name":"lisi"}]

```

■ 工具栏设置（操作按钮和查询部分）

```

<div id="tb" style="padding:5px;height:auto">
  <!-- 按钮部分 -->
  <div style="margin-bottom:5px">
    <a id="add" class="easyui-linkbutton" iconCls="icon-add" plain="true" >增 加</a>
    <a id="edit" class="easyui-linkbutton" iconCls="icon-edit" plain="true" >编 辑</a>
    <a id="delete" class="easyui-linkbutton" iconCls="icon-remove" plain="true">删 除</a>
  </div>
  <!-- 查询部分 -->
  <div>
    &nbsp;&nbsp;&nbsp;姓名: <input id="name" style="width:150px"></input>
    &nbsp;&nbsp;&nbsp;所在学院: <input class="easyui-combobox" id="institute" style="width:150px"></input>
    &nbsp;&nbsp;&nbsp;<a id="query" class="easyui-linkbutton" iconCls="icon-search" plain="true">查 询</a>
  </div>
</div>

```

2、DataGrid初始化：JavaScript实现，下面代码放在\$(function(){-----})中。

```

$("#studentinfo").datagrid({
  loadMsg: "数据加载中，请等待...",
  iconCls: 'icon-issue',
  nowrap: false,
  striped: true,
  collapsible: true,
  rownumbers: true,
  pagination: true,
  singleSelect: true,
  autoRowHeight: true,
  fitColumns: false,
  pageSize: 10,
  pageList: [10, 20, 30, 40],
  cache:false,
  url:"SystemStudentService?caozuo=init"
});

```

3、Servlet（SystemStudentService）中的实现：

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,

```
IOException {  
    String caozuo=request.getParameter("caozuo");  
    switch(caozuo){  
        case "init":  
            String row=request.getParameter("rows");  
            String page=request.getParameter("page");  
            getStudentInfo(response,"",page,row);  
        }  
    }  
}
```

```
private String getStudentInfo (HttpServletResponse response,String con,String page,String row){  
    String result="";  
    Map<String,Object> map=new HashMap<String,Object>();  
    ArrayList<HashMap<String,String>> dt=null;  
    String sql;  
    int rowscount=0;  
    if(con==null)con="";  
    if(row==null)row="0";  
    if(page==null)page="0";  
    try{  
        int r=Integer.parseInt(row);  
        int p=Integer.parseInt(page);  
        if(!con.equals("")){  
            sql="select * from student where "+con;  
        }else{  
            sql="select * from student";  
        }  
        dt=DBUtil.getDataSetInfoByCon(sql, r, p);  
        rowscount=DBUtil.getRowCount(sql);  
        map.put("total", rowscount);  
        map.put("rows", dt);  
        result=JSON.toJSONString(map);  
        response.setCharacterEncoding("utf-8");  
        PrintWriter out=response.getWriter();  
    }  
}
```

```

        out.print(result);
        out.close();
    }catch(Exception ex){
        ex.printStackTrace();
    }
    return result;
}

```

4、数据库操作类DBUtil:

添加**getDataSetInfoByCon**和**getRowCount**两个方法。

/**

*@ 函数名称: getDataSetInfoByCon

*@ 功能描述: 根据查询SQL语句、页码及页数返回部分多条记录。

*@ 传入参数: 用于查询的SQL语句、页码、页数

*@ 返回类型: (ArrayList<HashMap<String,Object>>)

*@ 文件作者: tjut.yangwenjun

*@ 创建时间: 2018-05-6

*@ 版本编号: 1.00

*/

```

public static ArrayList<HashMap<String,String>> getDataSetInfoByCon(String sql,int rowCount,int page){
    Connection conn=null;
    ArrayList<HashMap<String,String>> result=null;
    Statement st=null;
    ResultSet rs=null;
    ResultSetMetaData rsmd=null;

    try{
        conn=getConn();
        st=conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_READ_ONLY);
        if(rowCount>0)
            st.setMaxRows(page*rowCount);
        rs=st.executeQuery(sql);
        if(page>=0 && rowCount>0)
            rs.absolute((page-1)*rowCount);
        rsmd=rs.getMetaData();
        result=new ArrayList<HashMap<String,String>>();
        while(rs.next()){
            int columnCount=rsmd.getColumnCount();
            HashMap<String,String> record=new HashMap<String,String>();
            for(int i=1;i<=columnCount;i++){
                record.put(rsmd.getColumnName(i),rs.getString(i));
            }
            result.add(record);
        }
    }
}

```

```

    }catch(Exception e){
        e.printStackTrace();
    }finally{
        finallyHandle(conn,st,rs);
    }
    return result;
}
//获得满足条件的记录数
public static int getRowCount(String sql){
    Connection conn=null;
    Statement st=null;
    ResultSet rs=null;
    int length=0;
    try{
        conn=getConn();
        st=conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_READ_ONLY);r
        s=st.executeQuery(sql);
        rs.last();
        length=rs.getRow();
    }catch(Exception e){
        e.printStackTrace();
    }finally{
        finallyHandle(conn,st,rs);
    }
    return length;
}

```

5、增加及编辑记录操作:

(1) 在JSP页面中添加表单信息:

```

<div id="dlgStudentInfo" class="easyui-dialog" style="width:360px;height:450px;padding:10px 10px"
    closed="true" buttons="#dlg-buttons">
    <div class="ftitle">学生信息</div>
    <form id="fm" method="post" novalidate>
        <div class="fitem">
            <label>学 号</label>
            <input id="sno" name="sno"></input>
        </div>
        <div class="fitem">
            <label>学生姓名</label>
            <input id="name" name="name"></input>
        </div>
        <div class="fitem">
            <label>年 龄</label>
            <input id="age" name="age"/>
        </div>
    </form>
</div>

```

```

<div class="fitem">
    <label>政治面貌</label>
    <input id="politicalstate" name="politicalstate" />
</div>

<div class="fitem">
    <label>出生日期:</label>
    <input id="birthday" name="birthday" />
</div>
</form>
</div>
<div id="dlg-buttons">
    <a href="javascript:void(0)" id="save" class="easyui-linkbutton" iconCls="icon-ok"
    style="width:90px">保 存</a>
    <a href="javascript:void(0)" class="easyui-linkbutton" iconCls="icon-cancel"
    onclick="javascript:$('#dlgStudentInfo').dialog('close')" style="width:90px">关 闭</a>
</div>

```

(2) JavaScript操作:

■ 增加记录操作，打开对话框

```

$("#add").click(function() {
    Operation="add";
    $('#dlgStudentInfo').dialog('open').dialog('center').dialog('setTitle','添加学生信息');
    $('#fm').form('clear');
});

```

■ 编辑记录操作

```

$("#edit").click(function () {
    var row=$('#studentinfo').datagrid('getSelected');
    if(row){
        sno=row.sno;
        $('#fm').form('clear');
        operation="edit";
        $('#dlgstudentinfo').dialog('open').dialog('center').dialog('setTitle','编辑学生信息');

        $("#sno").val(row.sno);
        $("#name").val(row.name);
        $("#age").val(row.age);
        $("#politiicalstate").combobox("setValue",row.politicalstate);
        $("#dbirthday").datebox("setValue", row.birthday);
    }else{
        $.messager.alert("提示", "请选择要编辑的学生信息!", 'info');
        return;
    }
});

```

■ 保存操作

//对话框中的保存按钮操作:完成添加和编辑操作

```

$("#save").click(function(){
    if($("#sno").val()==""){
        $.messager.alert("提示", "学生编码不能为空! ", 'info');
        return;
    }

    if(operation=="add"){
        $.post("SystemStudentService?caozuo=add",
            $("#fm").serialize(),
            function (data) {
                var value=eval("(" +data+")")
                if (value.ret == "0") {
                    $.messager.alert("提示", "添加操作成功! ", 'info');
                    initstudentinfo ();
                    return;
                }else {
                    $.messager.alert("提示",value.reason, 'info');
                    return;
                }
            });
    }else if(operation=="edit"){
        $.post("SystemStudentService?caozuo=edit&oldsno="+sno,
            $("#fm").serialize(),
            function (data) {
                var value=eval("(" +data+")")
                if (value.ret == "0") {
                    $.messager.alert("提示", "修改操作成功! ", 'info');
                    initstudentinfo ();
                    return;
                }else {
                    $.messager.alert("提示",value.reason, 'info');
                    return;
                }
            });
    }
});

```

■ 删除操作

```

$("#delete").click(function(){
    var row=$("#employeeinfo").datagrid('getSelected');
    if(row==" " || row==null){
        $.messager.alert("提示", "请选择要删除的员工信息! ", 'info');
        return;
    }
}

```

```
$.post("SystemEmployeeService?caozuo=delete&sno=" + row.sno,
function (data) {
    var value=eval("(" + data + ")")
    if (value.ret == "0") {
        $.messager.alert("提示", "此操作成功! ", 'info');
        initstudentinfo();
        return;
    } else {
        $.messager.alert("提示", "本次操作失败! ", 'info');
        return;
    }
});
```

■ 刷新DataGrid组件

```
function initstudentinfo(){
    $("#studentinfo").datagrid({
        loadMsg: "数据加载中，请等待...",
        url: 'SystemStudentService?caozuo=init&con='+getQueryCondition()
    });
}
```

1.3 日期组件

1、JSP页面

```
<input id="birthday" name="birthday">
```

2、JavaScript设置

```
$("#birthday").datebox();
```


第二章 系统部分

2.1 角色权限管理

1、用户--角色--权限

(1) 权限管理（模块管理）

增加 编辑 删除							
模块编码:		模块名称:	上级模块编码:		上级模块名称:	查询	
	模块编码	模块名称	上级模块编码	上级模块名称	图标	模块地址	备注
1	0100	系统管理			icon-system		
2	0101	用户管理	0100	系统管理	icon-users	jsp/System/users.jsp	
3	0102	角色管理	0100	系统管理	icon-roles	jsp/System/role.jsp	
4	0103	模块信息	0100	系统管理	icon-modules	jsp/System/modules.jsp	
5	0104	密码修改	0100	系统管理	icon-password	jsp/System/password.jsp	
6	0105	部门信息	0100	系统管理	icon-department	jsp/System/department.jsp	
7	0106	员工管理	0100	系统管理	icon-employee	jsp/System/employees.jsp	
8	0107	库站用户	0100	系统管理	icon-user	jsp/System/osusers.jsp	
9	0108	访问日志	0100	系统管理	icon-general	jsp/System/log.jsp	
10	0200	任务管理			icon-task		

(2) 角色管理

角色名称: 角色说明:		
保存 删除 刷新 查询		
	角色名称	角色说明
1	admin	管理员角色, 不能删除
2	admin0	管理员角色
3	role	实验室普通角色
4	station	库站用户

(3) 用户管理

增加 编辑 删除 密码重置			
所属公司:		所属单位:	员工名称:
所属角色:			
	用户名	所属分公司	所属角色
1	50233211	仓储分公司	station
2	50227829	仓储分公司	station
3	00603369	仓储分公司	station
4	00599360	仓储分公司	station
5	50380452	仓储分公司	station
6	00624493	仓储分公司	station
7	00624481	仓储分公司	station
8	00605612	市区分公司	station
9	00604913	市区分公司	station
10	00602791	市区分公司	station

添加用户信息

用户信息

所属分公司: 所属单位: 员工名称: 用户名: 所属角色: 用户类型: admin admin0 role station 备注:

保存 关闭

2、相应的表的结构

(1) 模块表结构

字段名	类型及长度	是否主键	是否空	字段说明
mid	varchar(32)	是	否	模块 ID
pid	varchar(32)			父模块 ID
mname	varchar(50)		否	模块名称
url	varchar(100)			链接地址（jsp）
image	varchar(100)			模块显示的图标
demo	varchar(150)			备注

(2) 角色表结构

字段名	类型及长度	是否主键	是否空	字段说明
rolename	varchar(100)	是	否	角色名称
demo	varchar(150)			备注

(3) 角色权限表

字段名	类型及长度	是否主键	是否空	字段说明
rolename	varchar(100)	是	否	角色名称
mid	varchar(32)			模块 ID

(4) 用户表结构

字段名	类型及长度	是否主键	是否空	字段说明
username	varchar(100)	是	否	用户名称
password	varchar(32)			用户密码
rolename	varchar(32)			用户角色
....				