

原型设计

第一章 技术介绍

前台页面设计选用现在比较流行的 Easyui 前台框架。Easyui 基于 JQuery 开发，提供了大量经常用的 UI 插件，例如选项卡，折叠样式，树，对话框等等。另外和 Bootstrap 不同的是 Easyui 提供了良好的通过 ajax 方式和后台做交互，所以 Easyui 更适合做后台管理界面。Easyui 提供了两种对页面格式渲染的方式，一种是 js 的方式（如：`$('#p').panel({...})`）和 html 标记方式（如：`class="easyui-panel"`）。另外登录页面的话可以选用 h-ui 前台框架，实现响应式的布局方式。

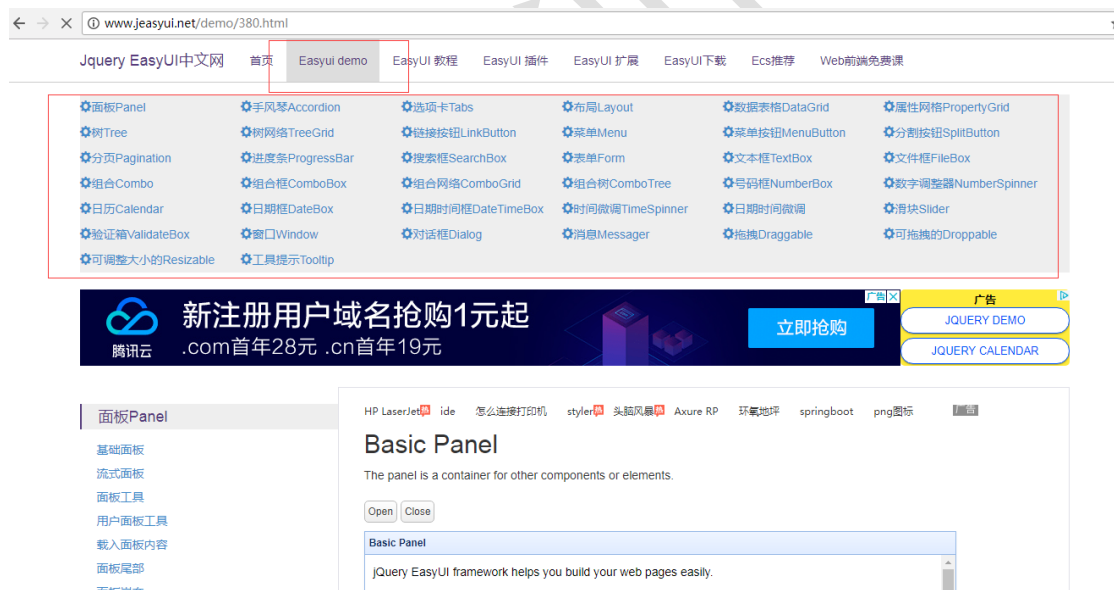
Easyui 的官方网站地址为：<http://www.jeasyui.net/>。在官方网站上可以下载 easyui，平时使用用 GPL 版本即可。



网站上还提供了常用插件的 API，方便我们参考。

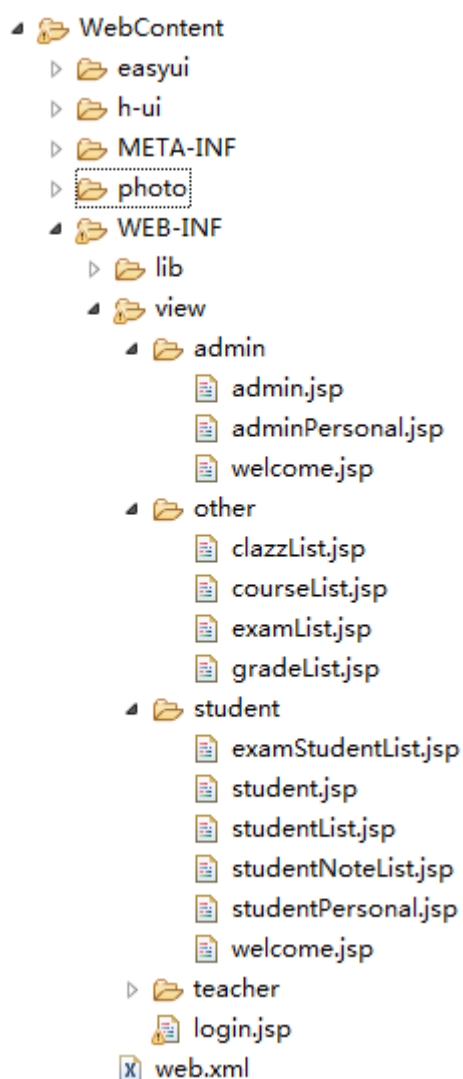


另外也提供了一些常用的 Demo，方便我们开发时直接使用和参考。



第二章 前台文件结构

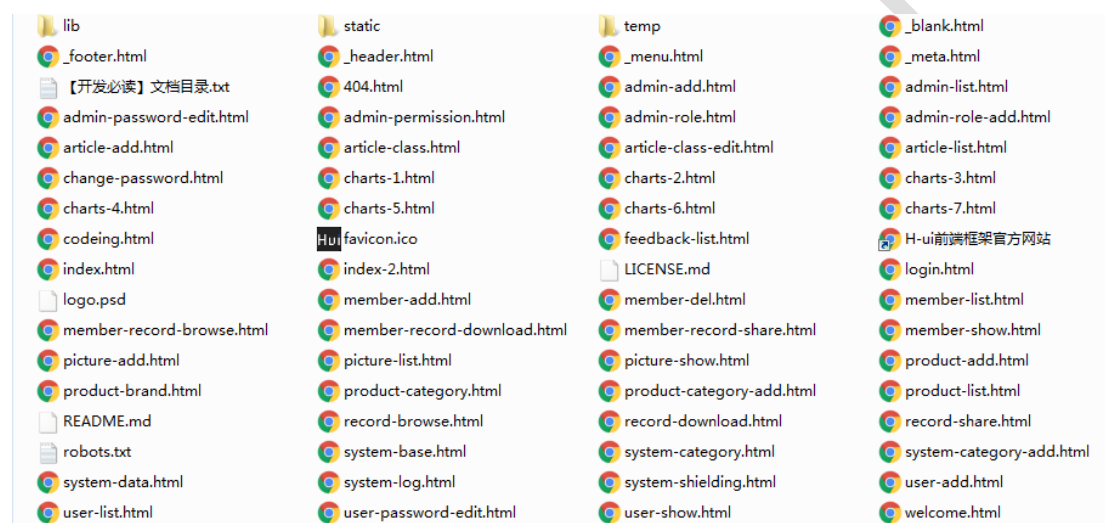
因为需要用框架，所以框架中的相关文件直接放在 WebContent 文件夹中就可以了。其他项目实现的文件为了安全起见，放在了 WEB-INF 文件夹中。为了阅读清晰，每一个模块单独列了相关的文件夹。文件结构如图所示：



第三章 登录页面的实现

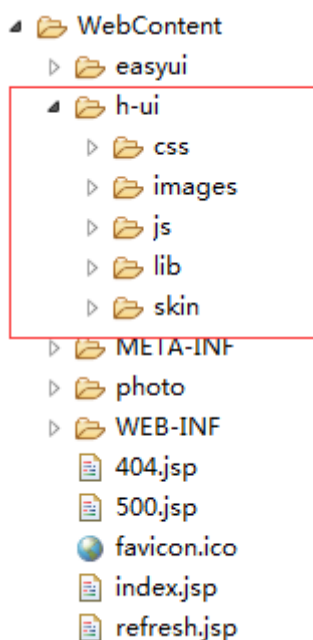
登录页面使用 h-ui 框架，它是一个基于 Bootstrap 框架并整合了一些常用 Web 组件的一个响应式的前台框架。

进 H-ui 的官网：<http://www.h-ui.net/H-ui.admin.shtml>，即可下载。下载后，解压文件，可以看到如下的目录结构：



其中，static 目录为 H-ui 框架的核心组件，lib 目录为 H-ui 框架整合的一些其它 Web 组件，temp 目录为临时目录，其余文件都是纯 HTML 页面，是演示用的，我们做页面时，可以从演示页面找到相似的界面并进行模仿。

1. 相关文件的引用



h-ui css 文件的引入:

```
<link href="h-ui/css/H-ui.min.css" rel="stylesheet"
type="text/css" />
<link href="h-ui/css/H-ui.Login.css" rel="stylesheet"
type="text/css" />
<link href="h-ui/lib/icheck/icheck.css" rel="stylesheet"
type="text/css" />
<link href="h-ui/lib/Hui-iconfont/1.0.1/iconfont.css"
rel="stylesheet" type="text/css" />
```

第一个样式表, H-ui.min.css, 这个样式表是最基本的样式表, 由于 H-ui 是基于 Bootstrap 框架的, 所以这个样式表中也包含了 Bootstrap 框架常用的栅格系统的样式。

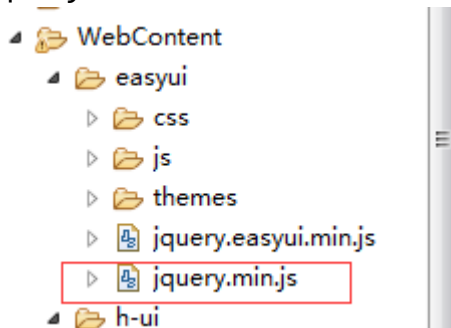
第二个样式表, H-ui.login.css, 这个样式表是登录页用到的样式表。

第四个样式表, iconfont.css, 这个样式表引入了 H-ui 的图标字体功能, 它里面定义了一系列图标, 每个图标用特殊文字来表示。

h-ui js 文件的引入:

```
<script type="text/javascript" src="h-ui/js/H-ui.js"></script>
<script type="text/javascript"
src="h-ui/lib/icheck/jquery.icehck.min.js"></script>
```

jQuery文件的引入:



```
<script type="text/javascript"
src="easyui/jquery.min.js"></script>
```

2. 登录页面布局实现

这时，只需将 H-ui 框架中自带的 login.html 中的所有代码复制到 login.jsp 中即可。

```
<form id="form" class="form form-horizontal" method="post">
  <div class="row cl">
    <label class="form-label col-3"><i class="Hui-iconfont">&#xe60d;</i>
    <div class="formControls col-8">
      <input id="" name="account" type="text" placeholder="账户" class=
    </div>
  </div>
  <div class="row cl">
    <label class="form-label col-3"><i class="Hui-iconfont">&#xe60e;</i>
    <div class="formControls col-8">
      <input id="" name="password" type="password" placeholder="密码" c
```

(1) 表单通常添加类样式：form。还有一个类样式

form-horizontal，表示水平表单，即表单元素是水平排列的。通常 Bootstrap 的表单元素是垂直排列的。

(2) 每一行表单元素都放在一个 div 中，类样式为：row cl。row 样式说明这是一行，它使用浮动让内部元素水平排成一行。cl 表示

clearfix，它使用了清除浮动，做到了左右清除浮动并重新计算盒子高度。

(3) 标题使用 `form-label` 样式，输入框使用 `formControls` 样式，`form-label` 样式使得标签右对齐，`formControls` 样式使得表单元素的宽度充满容器。要了解详情，可以打开 `static/h-ui/css/H-ui.css`，使用文字查找，根据样式名找到样式的定义，进行详细了解。

(4) 栅格系统，`col-xs-3` 与 `col-x-8`，这两个样式都是 Bootstrap 的栅格系统样式。Bootstrap 栅格系统将页面分成十二等分，如果你想让一个元素占用 $1/12$ 页面，可以设置样式 `col-xs-1`，占用 $2/12$ 页面，设置样式 `col-xs-2`，.....。然后 Bootstrap 还可以针对不同尺寸的设备进行定制，它总共有 4 种设备尺寸：`xs` (extra small, 超小)、`sm` (small, 小)、`md` (middle, 中等)、`bg` (big, 大)，例如你想在手机上一行显示 2 张图片，而在较大设备上一行显示 4 张图片，你就可以对一个列设置样式：`col-xs-6 col-bg-3`，这个样式表示此列在超小设备上，占用 $6/12$ ，在大设备上，占用 $3/12$ 。如果只指定一种尺寸类型例如 `xs`，那么其它类型的设备沿用这个比例。

还有一个样式 `col-xs-offset-3`，这个样式表示该元素缩进 3 个栅格显示。在不使用表格布局时，为了对齐，应该善用 Bootstrap 的栅格系统。

第四章 主页面实现



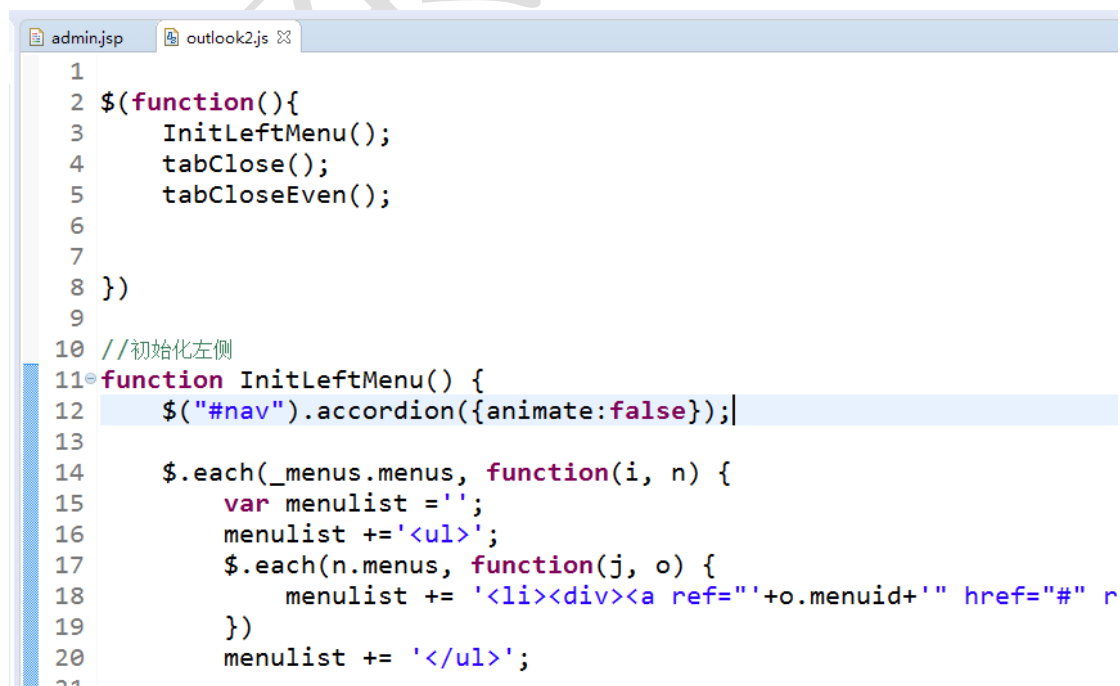
首先来看一下页面布局怎么出来的。看 easyui 的帮助文档，可以用 easyui 的布局，class="easyui-layout"。分四块，region="north", region="south", region="west", region="

center”。

然后左边导航栏可以使用 easyui 中的手风琴 (easyui-accordion) 样式。其中的数据使用 js 加载进来。

```
var _menus = {"menus":[
    {"menuid":"1","icon":"","menuname":"成绩统计分析",
      "menus":[
        {"menuid":"11","menuname":"考试列表","icon":"i
      ]
    },
    {"menuid":"2","icon":"","menuname":"学生信息管理",
      "menus":[
        {"menuid":"21","menuname":"学生列表","icon":"i
      ]
    },
    {"menuid":"3","icon":"","menuname":"教师信息管理",
      "menus":[
        {"menuid":"31","menuname":"教师列表","icon":"i
      ]
    },
    {"menuid":"4","icon":"","menuname":"基础信息管理",
      "menus":[
        {"menuid":"41","menuname":"年级列表","icon":"i
        {"menuid":"42","menuname":"班级列表","icon":"i
      ]
    }
  ]
}
```

<script type="text/javascript" src='easyui/js/outlook2.js'> </script>



首先定义一个 json 格式的变量，然后在 outlook2.js 中定义页

面刚加载的时候就初始化导航栏的函数，并且在菜单上定义点击事件。

山东师创

第五章 学生列表页面实现



The screenshot displays the implementation of the student list page. On the left, a file explorer shows the project structure: WEB-INF > lib > view > student > studentList.jsp. The main area shows the HTML and JavaScript code for the student list page.

HTML Code:

```
<!-- 学生列表 -->
<table id="dataList" cellspacing="0" cellpadding="0">

</table>
<!-- 工具栏 -->
<div id="toolbar">
    <div style="float: left;"><a id="add" href="javascript:;" class="easyui-linkbutton">添加</a>
    <div style="float: left;" class="datagrid-btn-separator"></div>
    <div style="float: left;"><a id="edit" href="javascript:;" class="easyui-linkbutton">编辑</a>
    <div style="float: left;" class="datagrid-btn-separator"></div>
    <div style="float: left;"><a id="delete" href="javascript:;" class="easyui-linkbutton">删除</a>
    <div style="float: left; margin: 0 10px 0 10px">年级: <input id="gradeList" class="easyui-textbox" type="text">
    <div style="margin-left: 10px;">班级: <input id="clazzList" class="easyui-textbox" type="text">
</div>
```

JavaScript Code:

```
// 初始化 datagrid
$( '#dataList' ).datagrid({
    title: '学生列表',
    iconCls: 'icon-more', // 图标
    border: true,
    collapsible: false, // 是否可折叠的
    fit: true, // 自动大小
    method: "post",
    url: "StudentServlet?method=StudentList&t="+new Date().getTime(),
    idField: 'id',
    singleSelect: false, // 是否单选
    pagination: true, // 分页控件
    rownumbers: true, // 行号
    sortName: 'id',
    sortOrder: 'DESC',
    remoteSort: false,
    columns: [[
        {field: 'chk', checkbox: true, width: 50},
        {field: 'id', title: 'ID', width: 50, sortable: true},
        {field: 'number', title: '学号', width: 200, sortable: true},
        {field: 'name', title: '姓名', width: 200},
    ]]
```

```

        {field: 'name', title: '姓名', width: 200},
        {field: 'sex', title: '性别', width: 100},
        {field: 'phone', title: '电话', width: 150},
        {field: 'qq', title: 'QQ', width: 150},
        {field: 'clazz', title: '班级', width: 150,
          formatter: function(value, row, index){
            if (row.clazz){
              return row.clazz.name;
            } else {
              return value;
            }
          }
        },
        {field: 'grade', title: '年级', width: 150,
          formatter: function(value, row, index){
            if (row.grade){
              return row.grade.name;
            } else {
              return value;
            }
          }
        }
      ],
      toolbar: "#toolbar"
    });
    //设置分页控件
    var p = $('#dataList').datagrid('getPager');
    $(p).pagination({
      pageSize: 10, //每页显示的记录条数, 默认为10
      pageList: [10, 20, 30, 50, 100], //可以设置每页记录条数的列表
      beforePageText: '第', //页数文本框前显示的汉字
      afterPageText: '页 共 {pages} 页',
      displayMsg: '当前显示 {from} - {to} 条记录 共 {total} 条记录',
    });
  });

```

列表使用 easyui 中的插件 datagrid, 然后增删改的工具栏中的三个功能使用 easyui 中的 linkbutton 插件。通过在 datagrid 中设置 pagination:true 将分页栏显示在表格中, 然后通过 js 设置 pagination 插件的一些属性。通过设置 toolbar: "#toolbar", 来将工具栏放到表格中。

第六章 查询功能实现

学生列表中的数据可以根据年级班级进行选择查询。

```
// 年级下拉框
$("#gradeList").combobox({
    width: "150",
    height: "25",
    valueField: "id",
    textField: "name",
    multiple: false, // 可多选
    editable: false, // 不可编辑
    method: "post",
    url: "GradeServlet?method=GradeList&t="+new Date().getTime(),
    onChange: function(newValue, oldValue){
        // 加载该年级下的学生
        $('#dataList').datagrid("options").queryParams = {gradeid: newVa
        $('#dataList').datagrid("reload");

        // 加载该年级下的班级
        $("#clazzList").combobox("clear");
        $("#clazzList").combobox("options").queryParams = {gradeid: newVa
        $("#clazzList").combobox("reload")
    }
}
```

工具栏中的年级班级选择下拉框使用 easyui 中的 combobox 插件。其中下拉框中的数据是动态获取的，是通过设置 combobox 的 url 属性来访问的具体数据获取方法。当选择的年级发生改变的时候，对应的班级下拉框中的数据也会重新加载，通过设置 `$("#clazzList").combobox("reload")`，并且学生列表中的数据也会发生改变，需要重新加载，通过设置 `$('#dataList').datagrid("reload")`；

```
// 班级下拉框
$("#clazzList").combobox({
    width: "150",
    height: "25",
    valueField: "id",
    textField: "name",
    multiple: false, // 可多选
    editable: false, // 不可编辑
    method: "post",
    url: "ClazzServlet?method=ClazzList&t="+new Date().getTime(),
    onChange: function(newValue, oldValue){
        // 加载班级下的学生
        $('#dataList').datagrid("options").queryParams = {clazzid: newVal
        $('#dataList').datagrid("reload");
    }
});
```

班级下拉框和年级下拉框的区别就是班级下拉框的值改变的时候，只需要重新加载学生表的数据就行了。

第七章 学生添加页面实现

```

$( "#add" ).click(function(){
    $( "#addDialog" ).dialog("open");
});

<!-- 添加学生窗口 -->
<div id="addDialog" style="padding: 10px">
<div style="float: right; margin: 20px 20px 0 0; width: 200px; border: 1px solid #ccc; padding: 5px; text-align: center;">
<img alt="照片" style="max-width: 200px; max-height: 400px;" title="照片" />
</div>
<form id="addForm" method="post">
<table cellpadding="8">
<tr>
<td>学号:</td>
<td>
<input id="add_number" class="easyui-textbox" style="width: 200px;" type="text">
</td>
</tr>
<tr>
<td>姓名:</td>
<td><input id="add_name" style="width: 200px; height: 30px;" type="text">
</td>
</tr>
<tr>
<td>性别:</td>
<td><select id="add_sex" class="easyui-combobox" data-options="{labelWidth: 100px}">
</td>
</tr>
<tr>
<td>电话:</td>
<td><input id="add_phone" style="width: 200px; height: 30px;" type="text">
</td>
</tr>
<tr>
<td>QQ:</td>
<td><input id="add_qq" style="width: 200px; height: 30px;" type="text">
</td>
</tr>
<tr>
<td>年级:</td>
<td><input id="add_gradeList" style="width: 200px; height: 30px;" type="text">
</td>
</tr>
<tr>
<td>班级:</td>
<td><input id="add_clazzList" style="width: 200px; height: 30px;" type="text">
</td>
</tr>
</table>
</form>

```

点击完添加按钮后，打开添加学生页面的对话框。添加学生的页面用了 easyui 的 dialog 插件，其中性别，年级和班级因为都是下拉框，都用了 combobox 插件。

```

//设置添加学生窗口
$("#addDialog").dialog({
    title: "添加学生",
    width: 650,
    height: 460,
    iconCls: "icon-add",
    modal: true,
    collapsible: false,
    minimizable: false,
    maximizable: false,
    draggable: true,
    closed: true,
    buttons: [
        {
            text: '添加',
            plain: true,
            iconCls: 'icon-user_add',
            handler: function(){
                var validate = $("#addForm").form("validate");
                if(!validate){
                    $.messager.alert("消息提醒", "请检查你输入的数据!", "warning");
                    return;
                } else{
                    var gradeid = $("#add_gradeList").combobox("getValue")
                    var clazzid = $("#add_clazzList").combobox("getValue")
                    $.ajax({
                        type: "post",
                        url: "StudentServlet?method=AddStudent",
                        data: $("#addForm").serialize(),
                        success: function(msg){
                            if(msg == "success"){
                                $.messager.alert("消息提醒", "添加成功!", "info")
                                //关闭窗口
                                $("#addDialog").dialog("close");
                                //清空原表格数据
                                $("#add_number").textbox('setValue', "");
                                $("#add_name").textbox('setValue', "");
                                $("#add_sex").textbox('setValue', "男");
                                $("#add_phone").textbox('setValue', "");
                                $("#add_qq").textbox('setValue', "");

                                //重新刷新页面数据
                                $('#dataList').datagrid("options").queryP
                                $('#dataList').datagrid("reload");
                                $("#gradeList").combobox('setValue', grad
                                setTimeout(function(){
                                    $("#clazzList").combobox('setValue',
                                }, 100);
                            } else{
                                $.messager.alert("消息提醒", "添加失败!", "warni
                                return;
                            }
                        }
                    });
                }
            }
        }
    ],
});

```

添加页面中重新定义了点击添加按钮之后的处理事件。通过设置 handler:function() 来定义的。点击完添加按钮之后先进行各个输入

框的验证，通过 easyui 的表单验证方式：
`$("#addForm").form("validate");` 输入框的验证用的是 easyui 自带的验证，比如学号的必填验证实现方式是：
`data-options="required:true,validType:'repeat',missingMessage:'请输入学号'"`。如果验证不通过不能走添加的后台方法；验证通过的话，通过 ajax 请求方式走添加学生的后台方法。添加数据到数据库成功后刷新学生列表页面上的数据。

```
{
  text: '重置',
  plain: true,
  iconCls: 'icon-reload',
  handler: function(){
    $("#add_number").textbox('setValue', "");
    $("#add_name").textbox('setValue', "");
    $("#add_phone").textbox('setValue', "");
    $("#add_qq").textbox('setValue', "");
    //重新加载年级
    $("#add_gradelist").combobox("clear");
    $("#add_gradelist").combobox("reload");
  },
},
];
```

重置按钮点击之后清空添加学生页面上输入框的数据。实现方式是通过设置 textbox 的值为空。

第八章 学生修改页面实现

学生修改页面和学生添加页面的区别就是修改学生信息的时候，首先在学生列表上选中时修改的哪个学生，否则提示还未选择学生。

```
//修改
$("#edit").click(function(){
    var selectRows = $("#dataList").datagrid("getSelections");
    if(selectRows.length != 1){
        $.messager.alert("消息提醒", "请选择一条数据进行操作!", "warning");
    } else{
        $("#editDialog").dialog("open");
    }
});
```

在代码实现的时候判断一下学生列表上是否有选中的数据，没有的话就弹出消息提醒框。有且仅有一条的话情况下直接打开学生修改页面。

修改页面的布局实现方式以及点击完修改页面上的提交按钮后需要干的事和学生添加页面是一样的。不同的是在打开学生修改页面之前需要将选择的学生列表上的某个学生信息放到修改页面的文本框中。

```
onBeforeOpen: function(){
    var selectRow = $("#dataList").datagrid("getSelected");
    //设置值
    $("#edit_number").textbox('setValue', selectRow.number);
    $("#edit_name").textbox('setValue', selectRow.name);
    $("#edit_sex").textbox('setValue', selectRow.sex);
    $("#edit_phone").textbox('setValue', selectRow.phone);
    $("#edit_qq").textbox('setValue', selectRow.qq);
    $("#edit_photo").attr("src", "PhotoServlet?method=GetPhoto&type=2");
    var gradeid = selectRow.gradeid;
    var clazzid = selectRow.clazzid;
    $("#edit_gradeList").combobox('setValue', gradeid);
    setTimeout(function(){
        $("#edit_clazzList").combobox('setValue', clazzid);
    }, 100);
}
});
```

设置对话框的 onBeforeOpen 属性函数，依次将值放入文本框和下拉框。

第九章 学生删除页面实现

学生信息删除的时候也同样先判断学生列表上是否有已经选中的数据，如果没有，则提醒。如果已经选中要删除的学生，（可以是多条要删除的数据），则弹出是否确定要删除的对话框，并将尧舜出数据的 number 数组值和 id 数组值作为后天删除方法的参数传过去，进行数据删除。数据成功从数据库删除之后，对学生列表页的数据重新加载。

```
//删除
$("#delete").click(function(){
    var selectRows = $("#dataList").datagrid("getSelections");
    var selectLength = selectRows.length;
    if(selectLength == 0){
        $.messager.alert("消息提醒", "请选择数据进行删除!", "warning");
    } else{
        var numbers = [];
        $(selectRows).each(function(i, row){
            numbers[i] = row.number;
        });
        var ids = [];
        $(selectRows).each(function(i, row){
            ids[i] = row.id;
        });
        $.messager.confirm("消息提醒", "将删除与学生相关的所有数据(包括成绩)，确认继续?",
            function(r){
                $.ajax({
                    type: "post",
                    url: "StudentServlet?method=DeleteStudent",
                    data: {numbers: numbers, ids: ids},
                    success: function(msg){
                        if(msg == "success"){
                            $.messager.alert("消息提醒", "删除成功!", "info");
                            //刷新表格
                            $("#dataList").datagrid("reload");
                            $("#dataList").datagrid("uncheckAll");
                        } else{
                            $.messager.alert("消息提醒", "删除失败!", "warning");
                            return;
                        }
                    }
                });
            });
    });
});
```