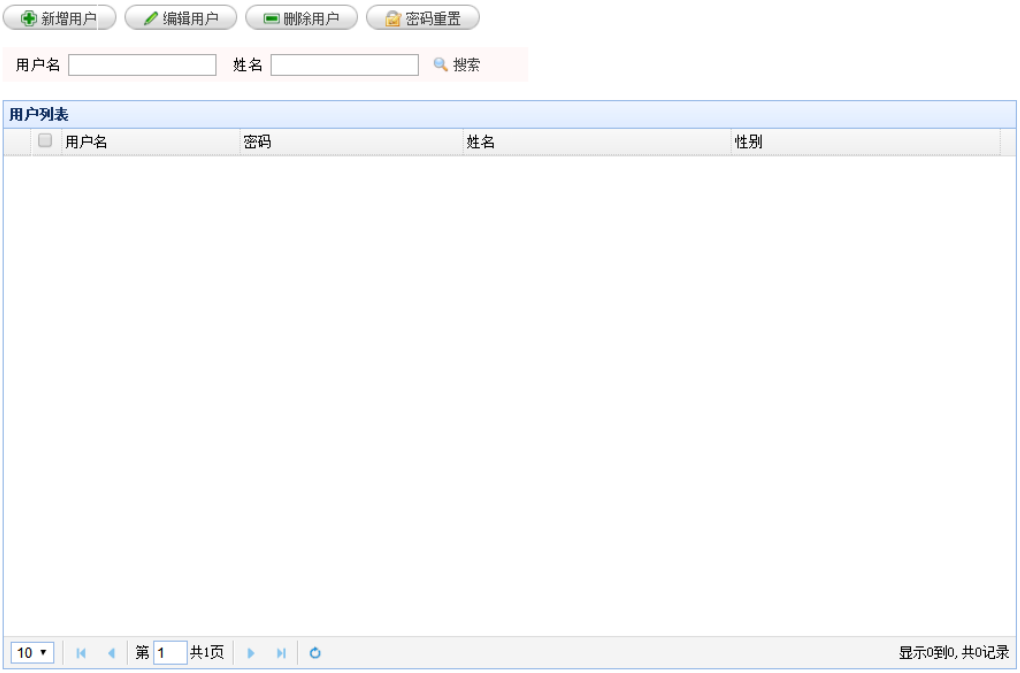
【Java框架型项目从入门到装逼】第十一节 用户新增之把数据传递到后台

让我们继续来做“主线任务”，这一节，我们来做具体的用户新增功能。首先，为了简单起见，我把主页面改了一些，改的是列表那一块。删去了一些字段，和数据库表对应一致：



现在，我们要实现一个效果，就是当我点击新增用户的按钮时，就弹出一个框来。因为我们使用的是EasyUI组件，所以我们可以用dialog组件来完成那个弹窗界面。

对话框（dialog）是一个特殊类型的窗口，它在顶部有一个工具栏，在底部有一个按钮栏。默认情况下，对话框（dialog）只有一个显示在头部右侧的关闭工具。用户可以配置对话框行为来显示其他工具（比如：可折叠 collapsible、可最小化 minimizable、可最大化 maximizable，等等）。

代码：

```
<div id="dlg" class="easyui-dialog" title="新增用户" style="width: 300px;height: 300px;padding: 10px 20px" closed="true" buttons="#dlg-buttons">
    <form id="fm" method="post">
        <!-- 这里画了一个table -->
        <table cellspacing="8px">
            <!-- 这个是table的第一行 -->
            <tr>
                <td>用户名：</td>
                <td>
                    <input type="text" id="username" name="username" class="easyui-validatebox" required="true" />
                </td>
            </tr>
            <tr>
                <td>密码：</td>
                <td>
                    <input type="text" id="password" name="password" class="easyui-validatebox" required="true" />
                </td>
            </tr>
            <!-- 这个是table的第二行 -->

            <tr>
                <td>姓名：</td>
                <td>
                    <input type="text" id="name" name="name" class="easyui-validatebox" required="true"/>
                </td>
            </tr>

            <tr>
                <td>性别：</td>
                <td>
                    <select class="easyui-combobox" data-options="value:'男'"  style="width: 144px" id="sex"  required="true"
                        editable="false" panelHeight="auto">
                        <option value="男">男</option>
                        <option value="女">女</option>
                    </select>
                </td>
            </tr>

        </table>
    </form>
</div>

<div id="dlg-buttons">
    <a href="javascript:saveUser()" class="easyui-linkbutton" iconCls="icon-ok">保存</a>
</div>
```

效果：

接下来，给新增按钮编写对应的点击事件：

```
function openUserAddPage(){
    $("#dlg").dialog("open");
}
```

这样一来，当我们点击按钮的时候，那个对话框就会自己跳出来哦。

画好了对话框，我们需要给保存按钮写对应的函数：

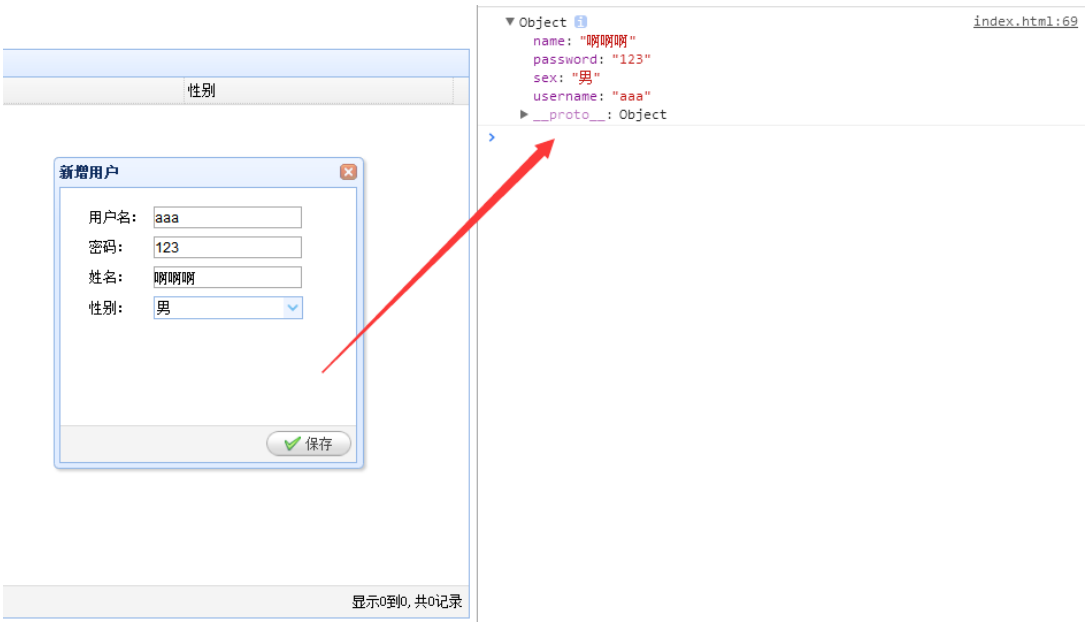
```
function saveUser(){
    var username = $('#username').val();
    var password = $('#password').val();
    var name = $('#name').val();
    var sex = $('#sex').combobox('getValue');


    if(!username){
        alert("用户名不能为空！");
        return;
    }

    if(!password){
        alert("密码不能为空！");
        return;
    }

    if(!name){
        alert("姓名不能为空！");
        return;
    }

    if(!password){
        alert("性别不能为空！");
        return;
    }



}
```

这个saveUser函数中，首先是用jQuery去获取每个文本框或者下拉框的值，然后依次判断是否为空，如果为空，就给出对应的提示。

然后，我们还需要用一个json数据将这些内容保存起来，到时候传递给后台的就是一个json数据。

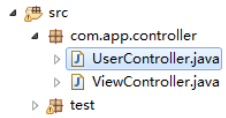
```
//开始拼接json数据，为了传递给后台
var json = {};
json.username = username;
json.password = password;
json.name = name;
json.sex = sex;

console.log(json);
```

效果：



这样一来，我们是不是可以拿到表单数据啦？OK，那么下一步，就是把这些数据传递到后台。

写一个UserController，作为控制器：



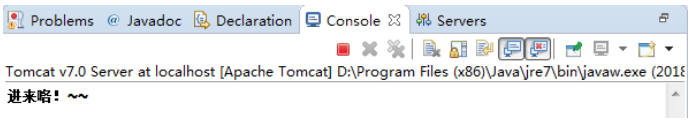

```
@Controller
public class UserController {

    @RequestMapping("/addUser")
    public void addUser(HttpServletRequest request , HttpServletResponse response){

    }
}
```

我们先不急着写代码，先看下这个Controller是否编写正确，于是乎，我们在里面打印一句话即可：

```
@Controller
public class UserController {

    @RequestMapping("/addUser")
    public void addUser(HttpServletRequest request , HttpServletResponse response){
        System.out.println("进来咯！~~");
    }
}
```

让我们启动Tomcat服务器，然后打开浏览器，在地址栏输入：
http://localhost/student/addUser.do
为什么是.do呢？那是因为我们在web.xml中进行了配置，让springMVC只拦截*.do的请求。

效果：



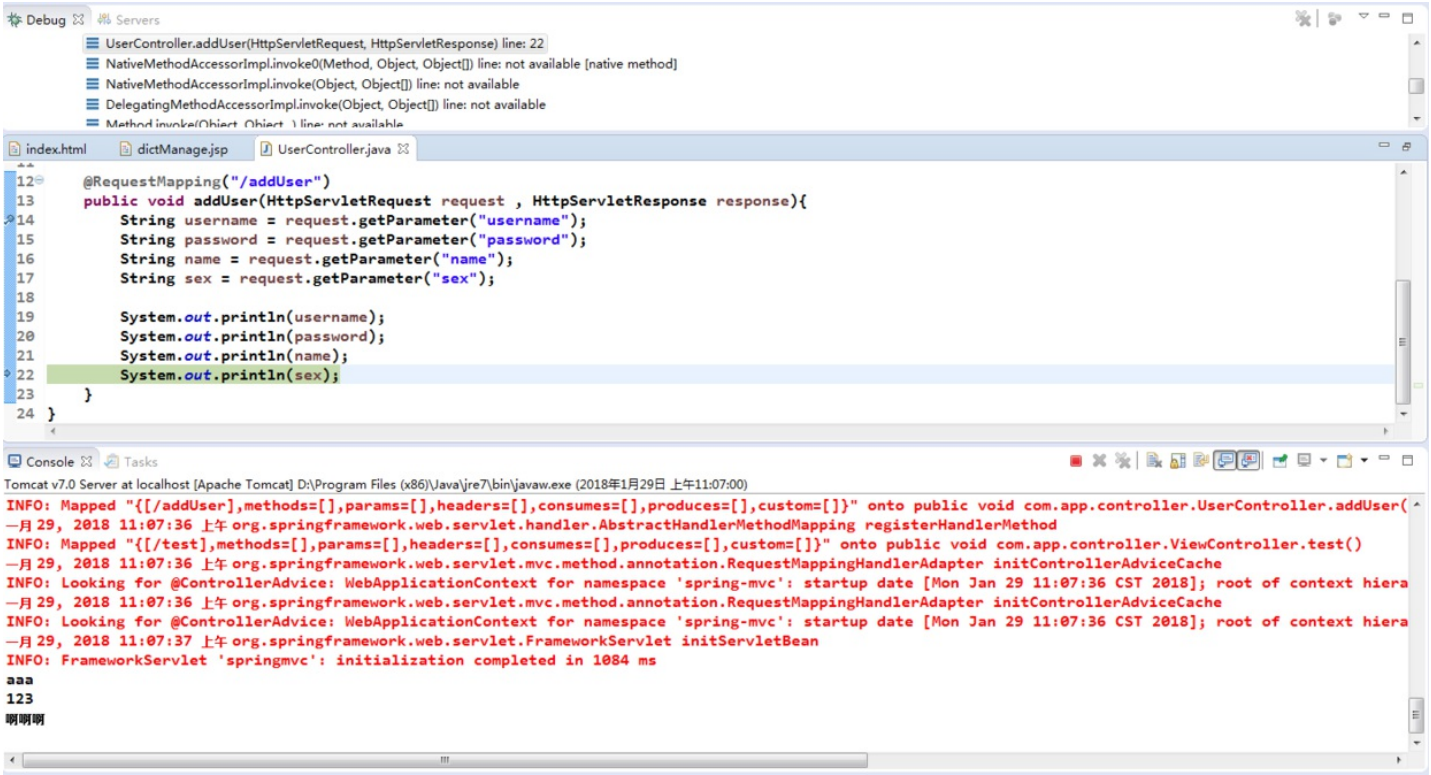
成功了。

接下来，用ajax传递数据给Controller

```
//使用ajax传递到后台
$.post("addUser.do",json,function(data){
    //这里是处理返回数据的回调函数

},"json");
```

填写表单，点击保存按钮：

发现进来了：



好的，这一节我们先聊到这里。