

【Java框架型项目从入门到装逼】第六节 - 用ajax请求后台数据

这一节我们来说一下如何用ajax提交请求？

我们先不讲ajax的原理，还是先以实战为主，看一下这个东西到底怎么用的？

form表单：

```
<!-- 采用post表单提交 -->
<form style="margin-left:200px;" id="myform" name="myform" method="post" onsubmit="return sumbitTest();"
    action="login.do">
    <table>
        <tr>
            <td>用户名:</td>
            <td> <input type="text" name="username" id="username"/> </td>
        </tr>
        <tr>
            <td>密码:</td>
            <td> <input type="password" name="password" id="password"/> </td>
        </tr>
        <tr>
            <td colspan="2">
                <input style="display:none" id="login_btn" type="submit" value="提交">
            </td>
        </tr>

        <div id="errMsg" style="color:red">${errMsg}</div>

        <input id="loginType" type="text" name="loginType" hidden='true' /> </td>

    </table>
</form>
```

修改提交方法：

```
function doLogin(type){
    if(type == 'QQ'){
        alert('您选择的是QQ登录');
    }

    if(type == 'weixin'){
        alert('您选择的是微信登录');
    }

    $("#loginType").val(type);

    //$("#login_btn").click();

    //如果验证通过，就开始进行提交
    if(submitTest()){
        $.ajax({
            type: "GET",
            url: "login.do",
            data: {username:$("#username").val(), password:$("#password").val()},
            dataType: "json", //预期服务器返回的数据
            success: function(data){
                if(data.errCode < 0){
                    $("#errMsg").show().html(data.errMsg).stop(true,true).fadeOut(3000);
                }else{
                    //登录成功，做其他处理
                    alert("恭喜你，登录成功！")
                }
            }
        });
    }
}

//验证登录信息
function submitTest(){

    return true;

    //在这个方法中可以对登录信息进行校验 （作业，用户名和密码都不能为空）
    if(!$("#username").val()){
        alert("用户名不能为空！");
        return false;
    }
}
```

```

    }

    if(!$("#password").val()){
        alert("密码不能为空!");
        return false;
    }
}

```

ajax是异步操作，和同步操作的form表单提交不同。

例子一：同步就是打电话，对方不接就一直等。异步就是发微信，对方回不回我都可以去做别的事情。

例子二：同步就像玩RPG回合制游戏，比如口袋妖怪，你打我一下，我打你一下。异步就像玩红色警戒，我可以先派20辆天启坦克去轰炸对面的兵工厂，下完指令后还可以立刻回到主基地造一座雷达。

OK，刚才我们看到ajax提交有一个属性是text，这就表示返回的数据格式是文本。实际上，我们一般用的数据格式并非text，而是JSON。我们需要在Servlet中手动给我们的JSP页面返回一个json数据。

```
WebUtil.writeObject(resp, JSONObject.fromObject(resultData) );
```

其中，resultData是一个包含错误码和错误信息的专用类：

```

public class ResultData {

    private int errCode = 0;
    private String errMsg;

    public int getErrCode() {
        return errCode;
    }
    public void setErrCode(int errCode) {
        this.errCode = errCode;
    }
    public String getErrMsg() {
        return errMsg;
    }
    public void setErrMsg(String errMsg) {
        this.errMsg = errMsg;
    }
}

```

因为最终转换的时候肯定默认调用resultData对象的toString方法，所以我们不妨直接重写ResultData类的toString方法，返回JSON格式。