springmvc+jsp框架模式中：

内容主要：

引入jquery.js；

页面js；

数据传递；

·提交：页面实体类变量自动，form使用；

·提交：ajax的data中；

·返回实体类；

·返回json；| 简单json；jackson；

返回数据：

·实体类； ---先使用这个；

·Json； ---然后是更强大的json使用；

---------------------------------

1，提交数据：页面实体类变量自动，form使用；

//注册实现，

引入js：*jquery.form.js*，

页面，

<form action=*""* id=*"register\_form"* >

<input type=*"text"* id=*"username"* name=*"username"* />

<input onclick="goto\_register()" type=*"button"* name=*"Submit1"* id=*"subbtn"* class=*"login\_btn"* value=*"提交注册"* />

</form>

Js,

**function** goto\_register(){

$("#register\_form").ajaxSubmit({

url : '${path }/member/registMemberInfoCollect.do',

data:{},

type : 'post',

conrtoller中， //实体类自动获得变量的值；

**public** MessageBox registMemberInfoCollect(Member member,

member.getUsername();

//使用form，实体类名自动传参时；ajax的data中也可以另加入参数，(实体类名，没有页面控件的，data中手写参数和值)，controller中实体类中获取；

2，提交：ajax的data中；

页面， //不使用form，一个超链按钮，

<span>用户名：<input type=*"text"* id=*"uname"* /></span>

<a id=*"registerHref2"* href=*"javascript:void(0);"* onclick="goto\_register2();">

Js， //date中参数名为实体类变量名

**function** goto\_register2(){

$.ajax({

type : 'post',

url : '${portalPath}/member/registMemberInfoCollectHb.do',

data : {

'username' : encodeURI($.trim($('#uname').val())),

'logpassword' : $('#upassword').val()

},

controller中， //实体类自动获得变量的值；---同上；

**public** MessageBox registMemberInfoCollectHb(Member member, HttpServletRequest request, HttpSession session, HttpServletResponse response) {

member.setUsername(CharacterEncoder.*decodeURL*(member.getUsername(), "UTF-8"));

member.getUsername();

//不实用form，data中传递参数，参数名为实体类变量名时；

也可以自定义参数名，controller中使用request取值，

'novalidatecode':'1'

request.getParameter("novalidatecode")

----------------------------------

3，返回实体类：---返回操作结果提示信息；

·声明一个消息实体类，MessageBox；

·Action，方法处理完业务逻辑后，返回MessageBox；

·ajax中，success:**function**(data){

data.code，data.message，显示内容；

实现：

js引入，

jquery-2.1.1.js

页面，

//无form，

<input class=*"safe\_button01"* type=*"button"* value=*"锁定新增"* onclick="bankCardLock();" id=*"btnBankCardLock"* style="cursor:*pointer*;">

**function** bankCardLock() {

layer.confirm("确定锁定新增银行卡功能？", **function**() {

$.ajax({

url : "${path}/bankinfo/lockBankCard.do",

data : {},

type : 'post',

success : **function**(data) {

**if** (data.code=="1") {

$(".nbcard").css("display","none");

$("#mes").text("新增银行卡功能已手动锁定");

$("#mes").css("font-size","70%");

$("#btnBankCardLock").attr("disabled","true");

layer.msg(data.message, 1, 1);

} **else** {

layer.msg(data.message, 1, 5);

}

},

error : **function**(data) {

layer.msg('网络连接异常，请刷新页面或稍后重试！', 1, 5);

}

});

});

}

实体类：

**public** **class** MessageBox **implements** Serializable {

**private** **static** **final** **long** ***serialVersionUID*** = -770558557256525814L;

**private** String code;

**private** String message;

**public** MessageBox() {

}

**public** MessageBox(String code, String message) {

**this**.code = code;

**this**.message = message;

}

controller中，

@ResponseBody

**public** MessageBox lockBankCard(HttpServletRequest request) {

bankInfoService.insertBankCardLock(shiroUser.getUserId());

**return** new MessageBox("1", "锁定成功.");

-------------

4，返回json:

·简单的，手写json串返回； | 不需要json插件、spring的json配置；

·页面js中，接收json对象，对象.属性使用，如：data.totalMoney；

conrtoller中， ---返回string，

@RequestMapping(value = "queryCount")

@ResponseBody

//查询统计数据

**public** String queryCount(HttpServletRequest request) {

String str2 = "{\"totalMoney\":\"1\",\"income\":\"2\",\"riskMargin\":\"3\",\"rate\":\"5\"}";

return str2;

页面，

$(**function**(){

$.ajax({

url : '${path}/home/queryCount.do',

data : {},

type : 'post',

dataType : 'json',

success : **function**(data) {

$("#totalMoney").html(data.totalMoney);

$("#income").html(data.income);

$("#riskMargin").html(data.riskMargin);

$("#rate").html(data.rate);

}

});

});

<div class=*"deal-all"* id=*"totalMoney"*></div>

5，复杂的，使用json插件，如：jackson使用：

·页面js中的使用方式不变；

·见“2json使用.doc”文档-json使用2：；