*<%--  
 ></script>  
 此时这个关闭标签不能省略  
--%>*<**script type="text/javascript" src="${**pageContext.request.contextPath**}/statics/js/jquery-3.2.1.min.js"**></**script**>  
<**script type="text/javascript"**>  
 */\*\*  
 \* 这里需要注意的是contentType要写上 contentType:"application/json;charset=utf-8",  
 \* jsonr有两种情况  
 \* 1、直接是数组  
 \* 这个时候要使用JSON.stringify(jsonr)把数据转换成json格式  
 \* 2、直接是字符串包着数组  
 \* 这个时候就不用转换了  
 \* 强调EL表达式 <%--${}--%>  
 \*/* **$**(**function**(){  
*// alert("{pageContext.request.contextPath}");* **$**(**"#jsontest"**).**click**(**function** () {  
*// var jsonr='{"id":12,"password":"asd","username":"卢少"}';* **var** jsonr={**"id"**:12,**"password"**:**"asd"**,**"username"**:**"卢少"**};  
 **$**.**ajax**({  
 **type**:**"post"**,  
 **url**:**'${**pageContext.request.contextPath**}/u/getJson.do'**,  
 **data**:***JSON***.stringify(jsonr),  
*// data:jsonr,* **dataType**:**"JSON"**,  
 **contentType**:**"application/json;charset=utf-8"**,  
 success:**function**(data){  
 alert(data.**username**);  
 }  
 });  
 });  
 */\*  
 \* 多选操作s  
 \* \*/* **$**(**"#all"**).**click**(**function**(){  
*// alert($("#all").attr("checked"));  
// alert($("#all :radio:checked"));* **if**(**this**.**checked**){  
 **$**(**"table :checkbox"**).prop(**"checked"**, **true**);  
 }**else**{  
 **$**(**"table :checkbox"**).prop(**"checked"**, **false**);  
 }  
 */\*$(":checkbox").each(function(){  
 $(this).attr("checked",!$(this).attr("checked"))  
 });\*/* });  
 *//获得多选个数* **$**(**"#getCheckedNum"**).**click**(**function**(){  
 **var** len=**$**(**":checkbox[name='id']:checked"**).**length**;  
 alert(len);  
 });  
 *//多选框总个数* **$**(**"#getAllNum"**).**click**(**function**(){  
 **var** len= **$**(**"table :checkbox[name='id']"**).**length**;  
 alert(len);  
 });  
*// 判断当其中一个没有选中，则全选框不勾选，当全部选中的时候，全选框勾选* **$**(**"table :checkbox"**).**click**(**function**(){  
 **var** len1= **$**(**"table :checkbox[name='id']"**).**length**;  
 **var** len2=**$**(**":checkbox[name='id']:checked"**).**length**;  
*// alert(len1+"~~~~~~"+len2);* **if**(len1==len2){  
 **$**(**"#all"**).prop(**"checked"**, **true**);  
 }**else**{  
 **$**(**"#all"**).prop(**"checked"**, **false**);  
 }  
 });  
  
 *//获得选中的数据* **$**(**"#sendValue"**).**click**(**function** () {  
 **var** boxList=**$**(**":checkbox[name='id']:checked"**);  
 **var** len=boxList.**length**;  
*// alert("选中的有："+len+"个")* **var** s=**''**;  
 **for**(**var** i=0;i<len;i++){  
*// alert(boxList[i].value);* **var** j=boxList[i].**value**;  
 s+=**',{"uid":"'**+**$**(**"#uid"**+j).val()+**'","username":"'**+**$**(**"#username"**+j).val()+**'","password":"'**+**$**(**"#password"**+j).val()+**'"}'**;  
 }  
 s=**'['**+s.substring(1)+**']'**;  
*// alert(JSON.stringify(s));  
// var jsonr='{"id":12,"password":"asd","username":"卢少"}';* **$**.**ajax**({  
 **type**:**"post"**,  
 **url**:**"${**pageContext.request.contextPath**}/u/getMultiValue.do"**,  
 **dataType**:**"JSON"**,  
 **contentType**:**"application/json;charset=utf-8"**,  
 **data**:s,  
 success:**function** (data) {  
*// alert(data[0].username.toString());* **$**.each(data,**function** (i,o) {  
 alert(o.**username**);  
 });  
*// alert(data.username);* }  
 });  
*// s=s.substring()* });  
 });  
</**script**>

<**form action="${**pageContext.request.contextPath**}/u/getMultiValue.do" method="post"**>  
 <**table**>  
 <**thead**>  
 <**tr**>  
 <**th**><**input type="checkbox" id="all"**></**th**>  
 <**th**>编号：</**th**>  
 <**th**>用户名：</**th**>  
 <**th**>密码：</**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 <**tr**>  
 <**td**><**input type="checkbox" name="id" value="1"**></**td**>  
 <**td**><**input type="text" id="uid1" name="uid" value="a1"**></**td**>  
 <**td**><**input type="text" id="username1" name="username" value="a1"**></**td**>  
 <**td**><**input type="text" id="password1" name="password" value="a1"**></**td**>  
 </**tr**>  
 <**tr**>  
 <**td**><**input type="checkbox" name="id" value="2"**></**td**>  
 <**td**><**input type="text" id="uid2" name="uid" value="a2"**></**td**>  
 <**td**><**input type="text" id="username2" name="username" value="a2"**></**td**>  
 <**td**><**input type="text" id="password2" name="password" value="a2"**></**td**>  
 </**tr**>  
 <**tr**>  
 <**td**><**input type="checkbox" name="id" value="3"**></**td**>  
 <**td**><**input type="text" id="uid3" name="uid" value="a3"**></**td**>  
 <**td**><**input type="text" id="username3" name="username" value="a3"**></**td**>  
 <**td**><**input type="text" id="password3" name="password" value="a3"**></**td**>  
 </**tr**>  
 </**tbody**>  
 </**table**>  
 <**input type="button" value="发送数据" id="sendValue"**>  
</**form**>  
<**button id="getCheckedNum"**>获得选中个数</**button**><**button id="getAllNum"**>获得多选总个数</**button**>

*/\*\*  
 \* 如果前台的是数组，则使用用对象数组接收  
 \* 不能使用List<T> list只支持简单属性类型 不然会报错不能封装  
 \* 重点：此时的接收类型是数组，则前台封装的数据也得是数组 格式： [data]  
 \** ***@param users*** *\*/* @RequestMapping(**"/getMultiValue"**)  
 **public** @ResponseBody List<User> getMultiValue(@RequestBody User[] users){  
 System.***out***.println(users[0]);  
 List<User> listUser=**new** ArrayList<>();  
 **for** (User user1 : users) {  
 System.***out***.println(**"循环"**+user1);  
 listUser.add(user1);  
 }  
*// System.out.println(user);* **return** listUser;  
 }

*/\*\*  
 \* 这里需要注意的是contentType要写上 contentType:"application/json;charset=utf-8",  
 \* jsonr有两种情况  
 \* 1、直接是数组  
 \* 这个时候要使用JSON.stringify(jsonr)把数据转换成json格式  
 \* 2、直接是字符串包着数组  
 \* 这个时候就不用转换了  
 \* 强调EL表达式 <%--${}--%>  
 \*/*

**$**(**"#jsontest"**).**click**(**function** () {  
*// var jsonr='{"id":12,"password":"asd","username":"卢少"}';* **var** jsonr={**"id"**:12,**"password"**:**"asd"**,**"username"**:**"卢少"**};  
 **$**.**ajax**({  
 **type**:**"post"**,  
 **url**:**'${**pageContext.request.contextPath**}/u/getJson.do'**,  
 **data**:***JSON***.stringify(jsonr),  
*// data:jsonr,* **dataType**:**"JSON"**,  
 **contentType**:**"application/json;charset=utf-8"**,  
 success:**function**(data){  
 alert(data.**username**);  
 }  
 });  
 });

*/\*\*  
 \* 这里的@RequestBody 是前台传入的数据  
 \*  
 \** ***@param user*** *\** ***@return*** *\** ***@ResponseBody*** *是往前台发送响应的数据  
 \* 需要导入jackson-databind jackson-annotations jackson-core 2.9.4  
 \*/* @RequestMapping(**"/getJson"**)  
 **public** @ResponseBody User getJson(@RequestBody User user) {  
 System.***out***.println(**"前台传到后台获得suer："** + user);  
 **return** user;  
 }

<**a href="javascript:void(0)" id="jsontest"**>测试前台JSON数据发送到后台</**a**>

**package** com.mvc.entry;  
  
**import** javax.persistence.Basic;  
**import** javax.persistence.Column;  
**import** javax.persistence.Entity;  
**import** javax.persistence.Id;  
  
@Entity  
**public class** User {  
 **private** Integer **id**;  
 **private** String **uid**;  
 **private** String **username**;  
 **private** String **password**;  
 **private** String **email**;  
 **private** Integer **state**;  
 **private** String **code**;  
  
 @Id  
 @Column(name = **"id"**, nullable = **false**)  
 **public** Integer getId() {  
 **return id**;  
 }  
  
 **public void** setId(Integer id) {  
 **this**.**id** = id;  
 }  
  
 @Basic  
 @Column(name = **"uid"**, nullable = **true**, length = 255)  
 **public** String getUid() {  
 **return uid**;  
 }  
  
 **public void** setUid(String uid) {  
 **this**.**uid** = uid;  
 }  
  
 @Basic  
 @Column(name = **"username"**, nullable = **true**, length = 255)  
 **public** String getUsername() {  
 **return username**;  
 }  
  
 **public void** setUsername(String username) {  
 **this**.**username** = username;  
 }  
  
 @Basic  
 @Column(name = **"password"**, nullable = **true**, length = 255)  
 **public** String getPassword() {  
 **return password**;  
 }  
  
 **public void** setPassword(String password) {  
 **this**.**password** = password;  
 }  
  
 @Basic  
 @Column(name = **"email"**, nullable = **true**, length = 255)  
 **public** String getEmail() {  
 **return email**;  
 }  
  
 **public void** setEmail(String email) {  
 **this**.**email** = email;  
 }  
  
 @Basic  
 @Column(name = **"state"**, nullable = **true**)  
 **public** Integer getState() {  
 **return state**;  
 }  
  
 **public void** setState(Integer state) {  
 **this**.**state** = state;  
 }  
  
 @Basic  
 @Column(name = **"code"**, nullable = **true**, length = 255)  
 **public** String getCode() {  
 **return code**;  
 }  
  
 **public void** setCode(String code) {  
 **this**.**code** = code;  
 }  
  
 @Override  
 **public boolean** equals(Object o) {  
 **if** (**this** == o) **return true**;  
 **if** (o == **null** || getClass() != o.getClass()) **return false**;  
  
 User user = (User) o;  
  
 **if** (**id** != **null** ? !**id**.equals(user.**id**) : user.**id** != **null**) **return false**;  
 **if** (**uid** != **null** ? !**uid**.equals(user.**uid**) : user.**uid** != **null**) **return false**;  
 **if** (**username** != **null** ? !**username**.equals(user.**username**) : user.**username** != **null**) **return false**;  
 **if** (**password** != **null** ? !**password**.equals(user.**password**) : user.**password** != **null**) **return false**;  
 **if** (**email** != **null** ? !**email**.equals(user.**email**) : user.**email** != **null**) **return false**;  
 **if** (**state** != **null** ? !**state**.equals(user.**state**) : user.**state** != **null**) **return false**;  
 **if** (**code** != **null** ? !**code**.equals(user.**code**) : user.**code** != **null**) **return false**;  
  
 **return true**;  
 }  
  
 @Override  
 **public int** hashCode() {  
 **int** result = **id** != **null** ? **id**.hashCode() : 0;  
 result = 31 \* result + (**uid** != **null** ? **uid**.hashCode() : 0);  
 result = 31 \* result + (**username** != **null** ? **username**.hashCode() : 0);  
 result = 31 \* result + (**password** != **null** ? **password**.hashCode() : 0);  
 result = 31 \* result + (**email** != **null** ? **email**.hashCode() : 0);  
 result = 31 \* result + (**state** != **null** ? **state**.hashCode() : 0);  
 result = 31 \* result + (**code** != **null** ? **code**.hashCode() : 0);  
 **return** result;  
 }  
  
 @Override  
 **public** String toString() {  
 **return "User{"** +  
 **"id="** + **id** +  
 **", uid='"** + **uid** + **'\''** +  
 **", username='"** + **username** + **'\''** +  
 **", password='"** + **password** + **'\''** +  
 **", email='"** + **email** + **'\''** +  
 **", state="** + **state** +  
 **", code='"** + **code** + **'\''** +  
 **'}'**;  
 }  
}