$.val

1.$.val()，无参数，取值

2.$.val(a)，有参数，赋值

下面这个例子，

在前端的角度，通过一个参数获取到数据，数据中有个caseInfo的大对象，大对象中的小对象有需要的属性数据。

其中，对象的属性在前端，统一都是name类型的，所以用name选择器 [ ]

在后端的角度，传进来的参数大多是数据库某个表的ID，通过ID取到一条记录，然后可能还有从表，于是在这条记录中找到从表的ID再找到记录…… 最后把这些记录整合，封装成各自的bean，或者继续将各个bean整合成一个大对象。如果前后端使用Ajax和json数据交互，那么可使用流行的json--对象转换工具（如jackson）进行转换，此时需要将对象放入一个map中。 转换成json字符串后，通过http响应发回给调用者。前端接受到数据，如果使用jq的$.ajax，那么还可以通过响应数据触发相应的回调函数。

$(**function**(){

BOMF.server.ajax("get", "/rest/assist/CpCheckPageService/queryChaKanInfo?subRegistId=1763", **null**,

**function** (data) {

// 案件信息拼装开始

**var** caseInfo = data['caseInfo'];

$('#caseRegistNo').val(caseInfo['custRegistInfo']['registNo']);

$('#caseReportTime').val(caseInfo['custRegistInfo']['reportDate'].substring(0,11)+caseInfo['custRegistInfo']['reportHour']);

$('#caseCaseType').val(caseInfo['custRegistInfo']['caseType']);

$('#caseLicenseNo').val(caseInfo['custRegistInfo']['licenseNo']);

$('#caseDamageTime').val(caseInfo['custRegistInfo']['damageDate'].substring(0,11)+caseInfo['custRegistInfo']['damageHour']);

$('#caseDamageAddress').val(caseInfo['custRegistInfo']['damageAddress']);

$('#caseDamageReason').val(caseInfo['custRegistInfo']['damageCode']);

$('#caseRemark').val(caseInfo['custRegistInfo']['remark']);

$('#caseClaimType').val(caseInfo['checkInfo']['ClaimType']);

$('#caseDamageCaseCode').val(caseInfo['checkInfo']['damageCaseCode']);

$('#caseSolutionUnit').val(caseInfo['checkInfo']['solutionUnit']);

$('#caseLossKindCode').val(caseInfo['checkInfo']['lossType']);

$('#caseCheckSite').val(caseInfo['checkTaskInfo']['checkSite']);

$('#caseCiDutyFlag').val(caseInfo['checkDutyInfo']['ciDutyFlag']);

$('#caseClaimFlag').val(caseInfo['checkDutyInfo']['claimFlag']);

$('#caseIndemnityDuty').val(caseInfo['checkDutyInfo']['indemnityDuty']);

$('#caseIndemnityDutyRate').val(caseInfo['checkDutyInfo']['indemnityDutyRate']);

$('#caseContext').val(caseInfo['checkTaskInfo']['context']);

// 案件信息拼装结束

});

});