通用数据提交

操作手册

目　　录

1. 功能概述

1.1 概述

业务系统中要面对很多数据提交，常用的方案针对每个表或者实体进行数据操作，包括增加、修改或者删除，甚至执行存储过程等。原则上每个这样的操作都要对应一个具体的方法或者接口，但是这样做在开发中会大大降低效率，为了提高开发效率，可以定义一个标准，提供一个统一的方法，提交的时候按照这个标准来保存数据。当然我们只很对没有复杂逻辑的业务这样做，例如物资基本信息，销售单的保存，对复杂的重要的业务，还是要专门编写方法或者接口，例如支付接口。

2. 通用数据提交

通用数据提交是指使用数据提交引擎提交数据的方法。在数据提交引擎中配置数据提交参数，主要包括提交 Id，提交的单表或者多表结构。

2.1 通用提交接口

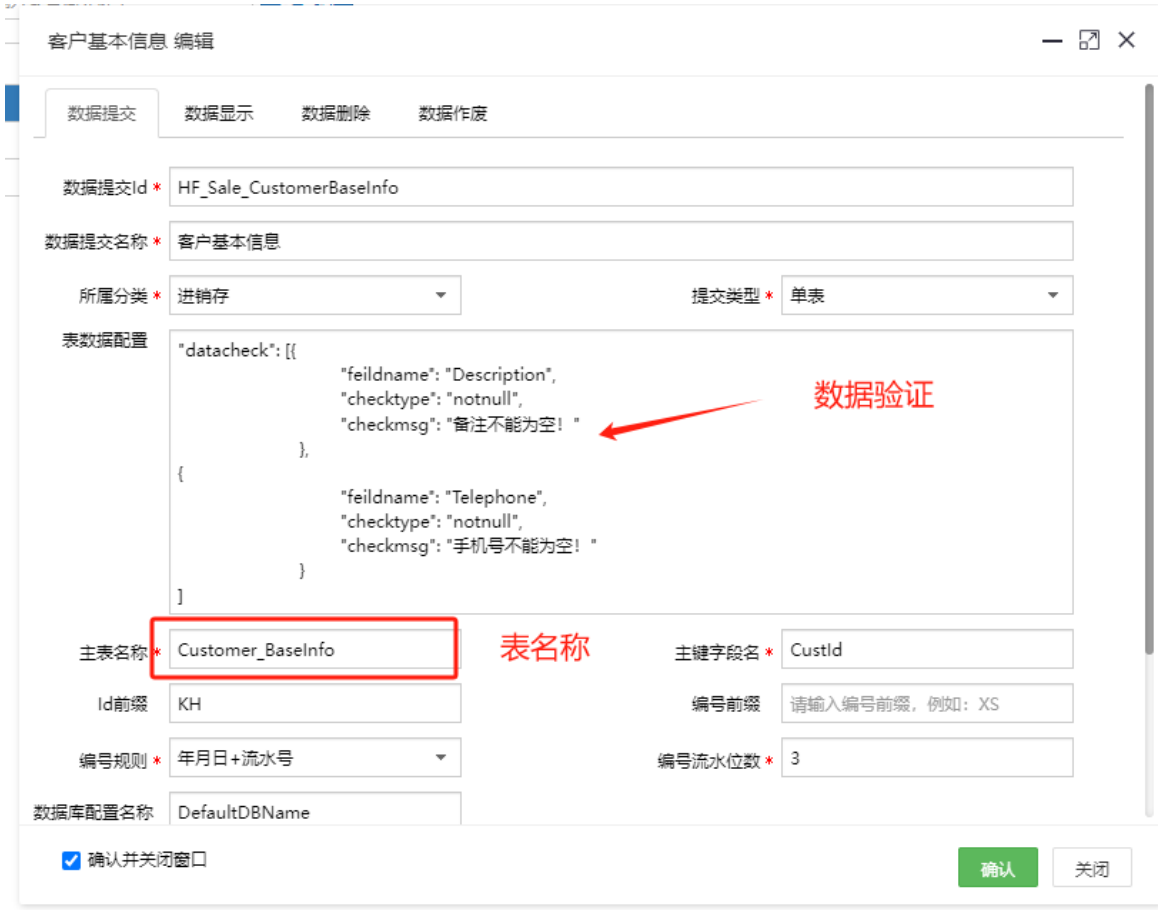
通用提交包括：单表提交、多表提交和存储过程提交，接口地址如下：

地址	Application/ SubmitDataLib/ SaveObjBySubmitId			
登录	需要登录			
输入	参数	是否必填	说明	实例值
1.	submitId	是	提交 Id	
2.	id	否	更新的主键值，为空时表示新建，不为空时表示修改。	
3.	jsondata	是	要更新的数据，包括: 1 单表不需要的数据; 2 多表需要的数据; 3 存储过程需要的数据。	
输出	{ Success:true			

	Message:“提交成功” }

2.2 单表提交

配置如下图：



单表数据配置主要是数据校验，如果没有数据验证时，必须为空，实例如下：

```
"datacheck": [{
        "feildname": "Description",
        "checktype": "notnull",
        "checkmsg": "备注不能为空！"
    },
    {
        "feildname": "Telephone",
        "checktype": "notnull",
```

```
        "checkmsg": "手机号不能为空！"
    }
]
```

序号	参数名（注意参数名全部为小写）	说明
1.	feildname	校验的字段名
2.	checktype	校验类型目前支持一下类型： notnull：不能为空 >0：大于零
3.	checkmsg	消息提示

前台调用如下图：

```
                );
            }
        });
    }
    //保存表单
    function AcceptClick() {
        if (!$('#form1').Validform()) {
            return false;
        }
        var postData = $("#form1").GetWebControls(keyValue);
        if (!!keyValue) {
            postData["ModifyUserId"] = "@userid";
            postData["ModifyUserName"] = "@username";
            postData["ModifyTime"] = "@getdate";
        }
        else {
            postData["CreateUserId"] = "@userid";
            postData["CreateUserName"] = "@username";
        }
        $.SaveForm({
            url: rootpath + "Application/SubmitDataLib/SaveObjBySubmitId",
            param: { "submitId": "HF_Sale_CustomerBaseInfo", "id": keyValue, "jsondata": JSON.stringify(postData) },
            loading: "正在保存数据...",
            success: function (data) {
                if (data.Success) {
                    $.currentIframe().$("#gridTable").trigger("reloadGrid");
                    dialogClose();
                    //initControl();
                }
            }
        })
    }
```

提交Id

提交的jsondata

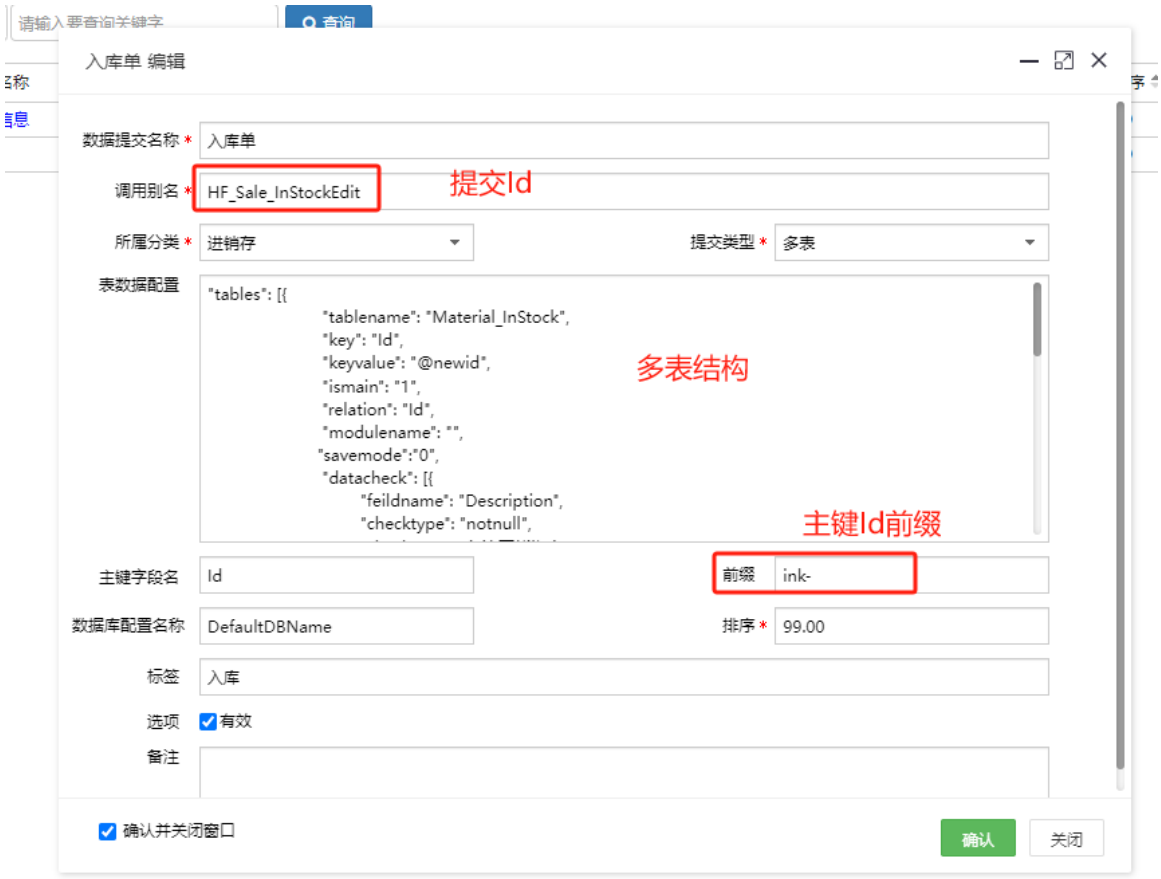
单表 jsondata 为要提交的数据对象，系统自动更加 id 判断是修改还是新增数据。无论是修改还是新增，只影响 jsondata 中包含的数据项。切记不更新的数据不要包含在 jsondata 中。jsondata 中可以使用系统变量，支持的系统变量如下表：

序号	变量名	值
1.	@userid	当前用户 userid
2.	@username	当前用户 username
3.	@organizeid	当前用户所属组织 id

4.	@newid	创建一个 guid
5.	@getdate	取数据库服务器当前时间

2.3 多表提交

配置如下图：



多表数据配置，主要是对要保存的表进行定义，不能为空，通常为主细表，实例如下：

```
"tables": [{
        "tablename": "Material_InStock",
        "key": "Id",
        "keyvalue": "@newid",
        "ismain": "1",
        "relation": "Id",
        "modulename": "",
        "savemode": "1",
        "datacheck": [{
```

```
            "feildname": "Description",
            "checktype": "notnull",
            "checkmsg": "备注不能为空！"
        }]
    },
    {
        "tablename": "Material_InStockDet",
        "key": "Id",
        "keyvalue": "@newid",
        "ismain": "0",
        "relation": "BillId",
        "savemode": "0",
        "modulename": "",
        "datacheck": [{
                "feildname": "TypeName",
                "checktype": "notnull",
                "checkmsg": "规格型号不能为空！"
            },
            {
                "feildname": "Quantity",
                "checktype": ">0",
                "checkmsg": "数量必须大于 0！"
            }
        ]
    }
]
```

表数据配置如下图：

```
"tables": [{
        "tablename": "Material_InStock",
        "key": "Id",
        "keyvalue": "@newid",
        "ismain": "1",
        "relation": "Id",
        "modulename": "",
        "savemode": "0",
        "datacheck": [{
            "feildname": "Description",
            "checktype": "notnull",
            "checkmsg": "备注不能为空！"
        }]
    },
    {
        "tablename": "Material_InStockDet",
        "key": "Id",
        "keyvalue": "@newid",
        "ismain": "0",
        "relation": "BillId",
        "savemode": "0",
        "modulename": "",
        "datacheck": [{
                "feildname": "TypeName",
                "checktype": "notnull",
                "checkmsg": "规格型号不能为空！"
            },
            {
                "feildname": "Quantity",
                "checktype": ">0",
                "checkmsg": "数量必须大于0！"
            }
        ]
    }
]
```

表1的配置

表2的配置

参数含义：

序号	参数名（注意参数名全部为小写）	说明
6.	tablename	数据库中的表名称
7.	key	表主键字段名
8.	keyvalue	主键值，为空时自动创建
9.	ismain	1 主表，0 明细表
10.	relation	关联字段，当为明细表示，必须设置与主表关联的字段名称
11.	modulename	模块名称
12.	savemode	更新数据时，0 先删除再创建（主键值可能会变化），1 判断是否存在，存在则修改，不存

		在则创建（主键值不会变化）。2 只添加数据（不管是否存在）。
13.	datacheck	数据校验
14.	datacheck. feildname	校验的字段名
15.	datacheck. checktype	校验类型目前支持一下类型： notnull：不能为空 >0：大于零
16.	datacheck. checkmsg	消息提示
17.		

前台提交如下图：

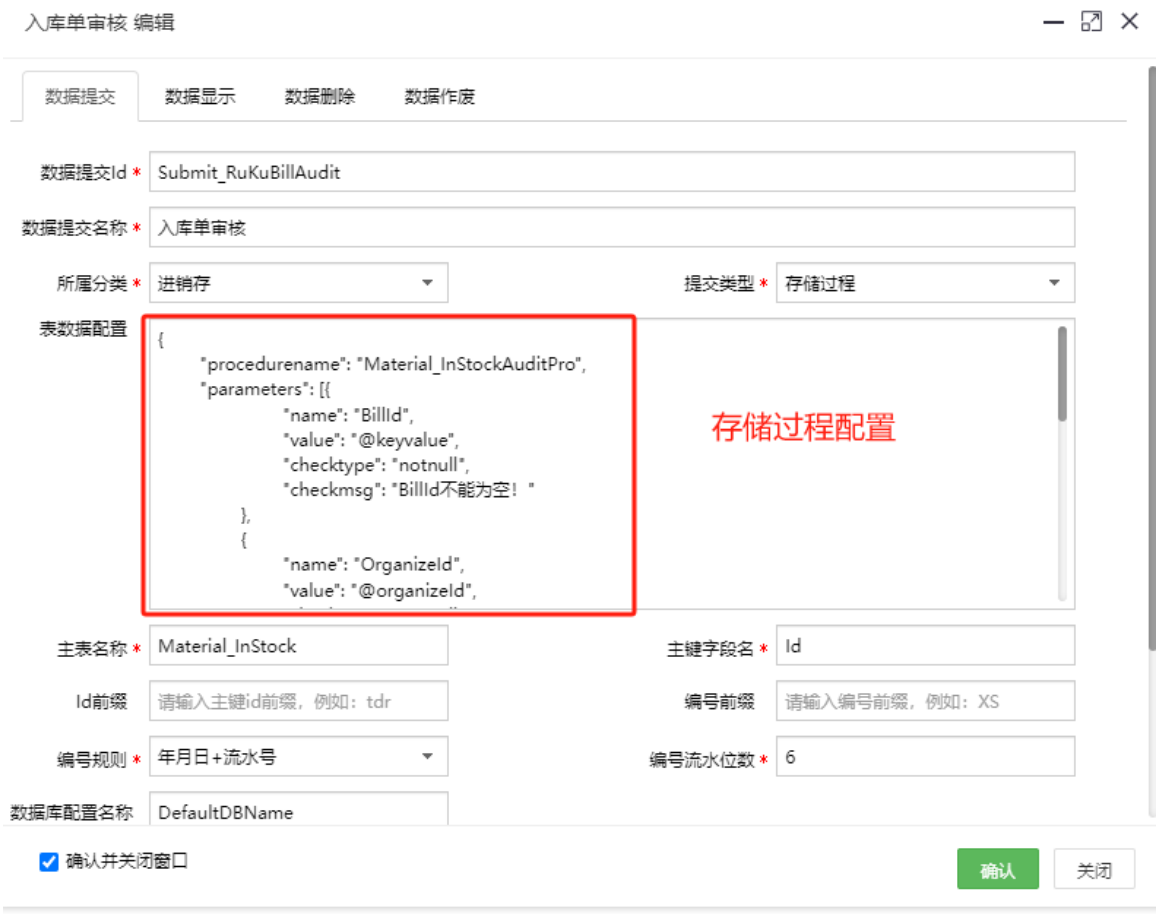
```
}
//保存表单
function AcceptClick() {
    if (!$('#form1').Validform()) {
        return false;
    }
    var ckData = checkData();
    if (ckData != true) {
        dialogMsg(ckData, 0);
        return;
    }

    var postData = getData();
    if (postData.Material_InStockDet.length == 0) {
        dialogMsg('明细数据不能为空！', 0);
        return;
    }
    $.ConfirmAjax({
        msg: "注：您确认要保存此操作吗？",
        url: rootpath + "Application/SubmitDataLib/SaveObjBySubmitId",
        param: { "submitId": "HF_Sale_InStockEdit", "id": keyValue, "jsondata": JSON.stringify(postData) },
        success: function (data) {
            keyValue = data.Id;
            if (data.Success) {
                top.$.removeTab('closeCurrent');
            }
        }
    });
}
//审核入库单
function auditBill() {
```

提交Id

提交的jsondata

2.4 存储过程提交

配置如下图：



存储过程的数据配置，主要是存储过程名称以及参数的配置，不能为空，实例如下：

```
{
    "procedurename": "Material_InStockAuditPro",
    "parameters": [{
            "name": "BillId",
            "value": "@keyvalue",
            "checktype": "notnull",
            "checkmsg": "BillId 不能为空！"
        },
        {
            "name": "OrganizeId",
            "value": "@organizeId",
            "checktype": "notnull",
            "checkmsg": "organizeId 不能为空！"
        },
```

```
        {
            "name": "AuditUserId",
            "value": "@userId"
        },
        {
            "name": "AuditUserName",
            "value": "@username"
        },
        {
            "name": "EnCode",
            "value": "EnCode",
            "checktype": "notnull",
            "checkmsg": "EnCode 不能为空！"
        }
    ]
}
```

序号	参数名（注意参数名全部为小写）	说明
1.	procedurename	存储过程名
2.	parameters	参数列表
3.	parameters. name	参数名称
4.	parameters. value	参数值取值方式，@表示从系统中获取。 其它从 jsondata 中按同名获取。区分大小写。
5.	parameters. checktype	参数验证类型
6.	parameters. checkmsg	验证消息

前台提交如下图：

```
//审核单据
function auditBill(keyValue) {
    var postData = {};
    postData["EnCode"] = keyValue;
    $.ConfirmAjax({
        msg: "注：您确定要【审核】入库单？",
        /* url: rootpath + "Material/InStockBLL/BillAudit?Id=" + keyValue,*/
        url: rootpath + "Application/SubmitDataLib/SaveObjBySubmitId",
        param: { "submitId": "Submit_RuKuBillAudit", "id": keyValue, "jsondata": JSON.stringify(postData) },
        success: function (data) {
            $("#gridTable").trigger("reloadGrid");
        }
    })
}
//作废单据
```

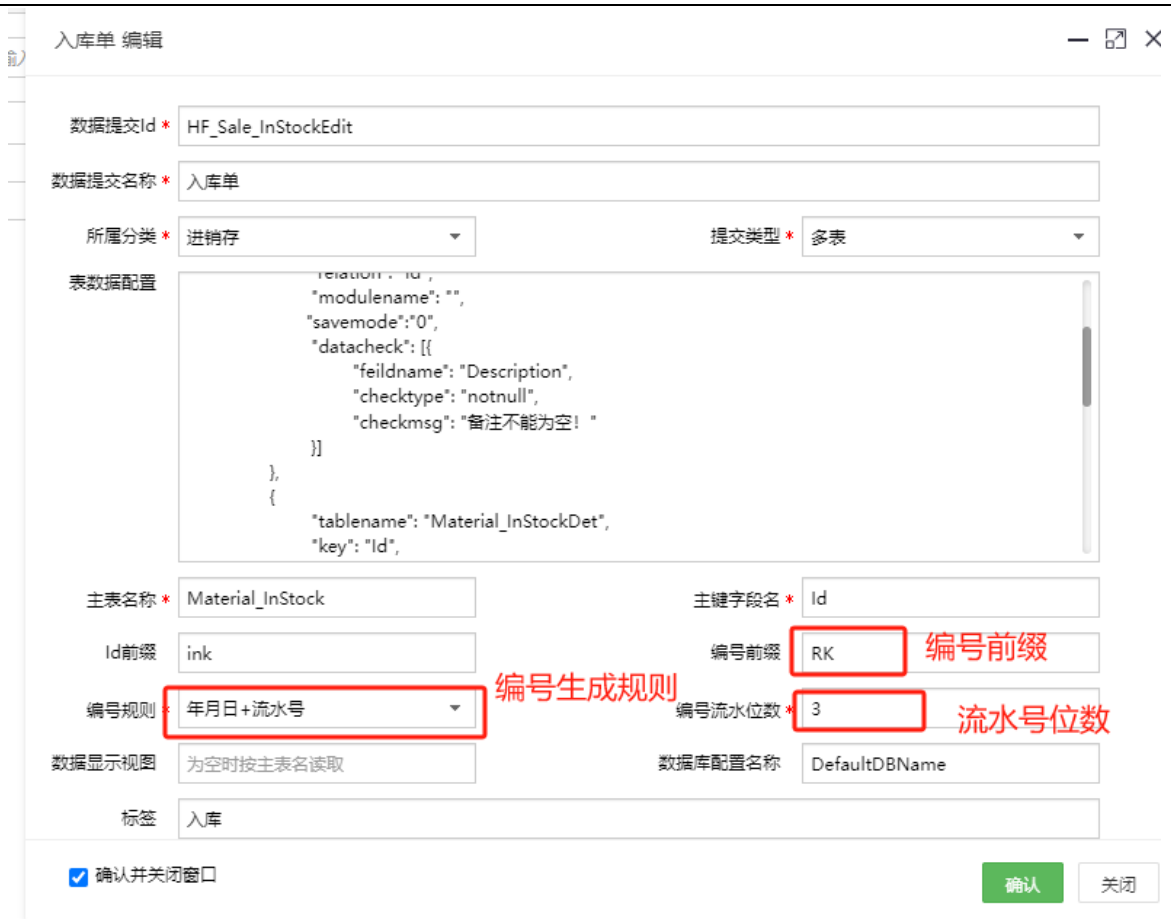
存储过程提交，jsondata 主要为存储过程需要的参数，除系统变量类型的参数外，存储过程需要的参数必须包含在 jsondata 中。支持的系统变量类型的参数如下表：

序号	变量名	值
1.	@userid	当前用户 userid
2.	@username	当前用户 username
3.	@organizeid	当前用户所属组织 id
4.	@keyvalue	取 SaveObjBySubmitId 接口传递的参数 id

2.5 生成编号

地址	Application/ SubmitDataLib/ GetNewEnCodeBySubmitId			
登录	需要登录			
输入	参数	是否必填	说明	实例值
1.	submitId	是	提交 Id	
2.	prefix	否	编号前缀，例如 XS，代表销售单，为空时按配置中读取。	XS20024050600001
输出	XS20024050600001			

配置如下图：



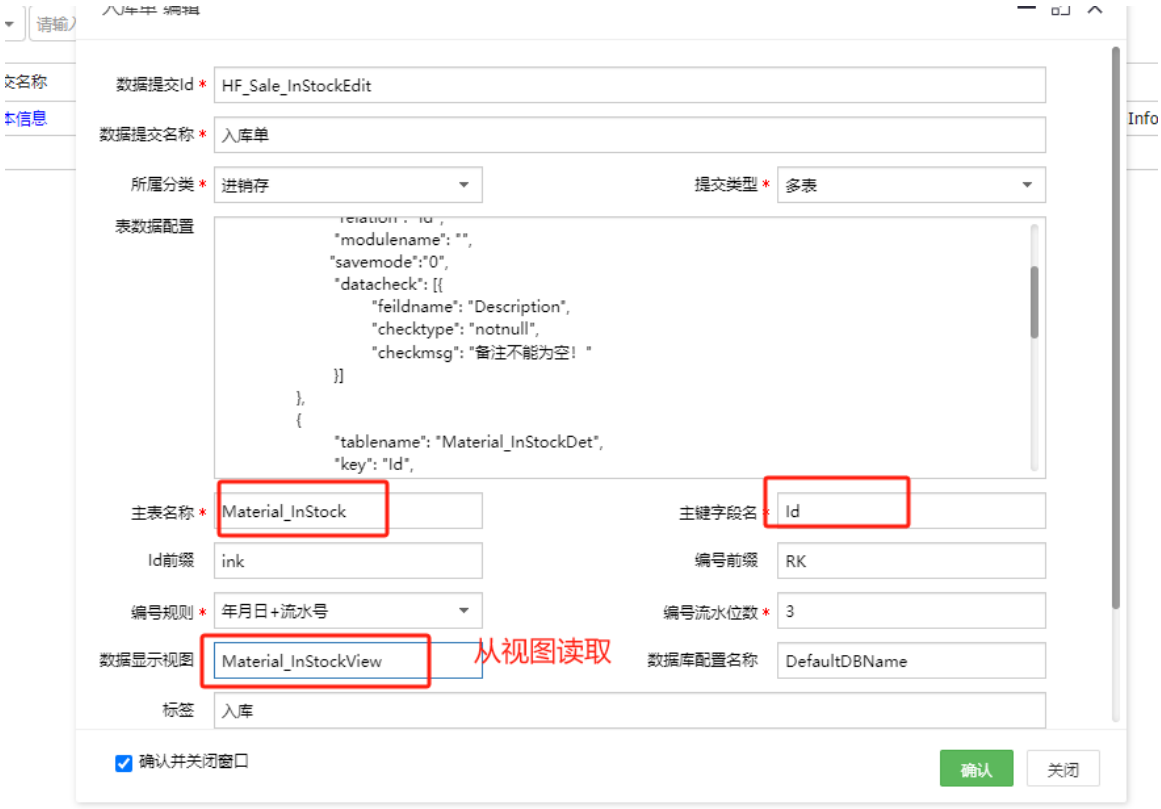
支持的编号规则如下表：

标识	编号规则	实例
yeardatenumber	年月日+流水号	XS20240625000001
yearmonthnumber	年月+流水号	XS202406000001
yearnumber	年+流水号	XS2024000001
number	流水号	100001

2.6 读取数据

地址	Application/ SubmitDataLib/ GetInfoBySubmitId			
登录	需要登录			
输入	参数	是否必填	说明	实例值
1.	submitId	是	数据提交 Id	
2.	id	是	读取信息的主键值	
输出	json 对象			

配置如下图：



如果配置了显示视图，从视图中读取基本信息。注意视图中的主键要与主表中的主键字段一致。

2.7 删除数据

地址	Application/ SubmitDataLib/ DeleteBySubmitId			
登录	需要登录			
输入	参数	是否必填	说明	实例值
1.	submitId	是	数据提交 Id	
2.	id	是	删除信息的主键值	

输出	{
	Success:true
	Message:“提交成功”
	}			

注意：删除数据库时要对业务数进行如下验证：

1、已经审核的不能删除；

2、只能管理员和创建人才可以删除；

3、业务数据删除成功后，删除流程实例数据；

所以必须保证业务主表中包含以下字段，才可以使用统一删除接口：

序号	字段名	说明
1、	AuditFlag	审核标志，只有为 0 时可以删除。
2、	CreateUserId	创建人 Id
3、	WorkFlowInstanceId	流程实例 Id

2.8 作废数据

地址	Application/ SubmitDataLib/ CancelBySubmitId			
登录	需要登录			
输入	参数	是否必填	说明	实例值
1.	submitId	是	数据提交 Id	
2.	id	是	作废信息的主键值	
3.	msg	是	作废原因	

输出	{
	Success:true
	Message:“作废成功”
	}			

注意：作废数据库时要对业务数进行如下验证：

1、只能管理员和创建人才可以作废；

2、业务数据作废成功后，作废流程实例数据；

所以必须保证业务主表中包含以下字段，才可以使用统一作废接口：

序号	字段名	说明
1、	CreateUserId	创建人 Id
2、	WorkFlowInstanceId	流程实例 Id
3、	EnableRemark	作废原因