# 通用格式

## 数据表

用json格式表示一张二维表

{

//字段列表

fs:[

//第1个字段

{

value:以逗号分隔的值,string,格式为字段名,字段类型,长度,小数点位数,是否是主键,是否允许为空,是否是扩展字段

//日期类型才有

date:Y,string, 固定值Y

},

//第2个字段

…

],

//数据行

rows:[

//第1行

字符串表示的该行每个字段的值,字段值之间用逗号分割。对特使符号做转译，转译表如下值为空用%2表示

//第2行

…

]

}

* 数据类型表

|  |  |
| --- | --- |
| 类型代号 | 类型名称 |
| 0 | String |
| 1 | Integer |
| 2 | double |

* 转译表

|  |  |
| --- | --- |
| 符号 | 替换符号 |
| 值为空时 | %2 |
| % | %1 |
| , | %0 |

范例

{

"fs":[{"value":"r\_id,0,-1,0,True,True,False"},{"value":"r\_name,0,10,0,False,True,False"},{"value":"r\_desc,0,20,0,False,True,False"},{"value":"r\_owner,0,10,0,False,True,False"},{"value":"@pms,0,-1,0,False,True,False"}],

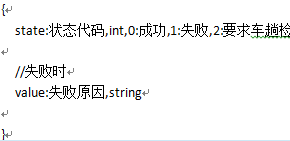
"rows":["1,Guest,Guest,DEMO,%2"]

}

数据表的字段都以小写开头

## JSON命名

在json格式里，除如下格式特殊外，属性以小写开头，其他属性都以大写开头



## HTTP参数

http参数名都以小写开头

## 声明

如果下文中有违背上文所列的命名规则，一般属于笔误，以上文所列命名规则为准

# 气瓶档案

## 气瓶档案上传

* POST

http://对接平台地址/Steel/Info/PutBottle?data=json格式数据&token=对接口令

参数data的json格式

{

Barcode:钢瓶条码,string

Printno:钢印号,string

Style:钢瓶型号,string,代码有:YSP118,YSP35.5,YSP12

Factory:钢瓶制造商编码,int

ProductDate:钢瓶制造日期,string

DiscardDate:报废日期,string,yyyyMMdd

CheckDate:上次年检,string,yyyyMMdd

NextCheckDate:下次年检,string,yyyyMMdd

FillDate:上次充装日期,string,yyyyMMdd,选填

State:状态,int, 0:一般,4:故障,5:流失,6:停用,200:作废

RegisterNo:登记代码,string

//产权单位

Equity:{

Id:系统ID,int

Name:名称,string

}

}

* 响应

Json格式

{

State:状态代码,int,0:成功,1:失败

//失败时

Value:失败原因,string

}

## 气瓶档案下载

* POST

http://对接平台地址/Steel/Info/BottleList?lastDate=上次时间&token=对接口令

* 响应

json格式

{

PlanCount:预计瓶数,int,如果实际返回的瓶数比该值小,表示已经全部下载完毕

//气瓶列表

Items: [

//第1笔气瓶档案

{

Barcode:钢瓶条码,string

Printno:钢印号,string

Style:钢瓶型号,string,代码有:YSP118,YSP35.5,YSP12

Valve:阀门,int,0:未知, 1:角阀液相, 2:角阀气相, 3:直阀气相

ProductDate:钢瓶制造日期,string,yyyyMMdd

DiscardDate:报废日期,string,yyyyMMdd

CheckDate:上次年检,string,yyyyMMdd

NextCheckDate:下次年检,string,yyyyMMdd

UpdateDate:档案更新时间,string,yyyyMMddHHmmss

State:状态,int,0:正常,其他值表示有异常

Id:气瓶系统ID,int

//气瓶生产厂家,

Factory:{

Id:厂家Id,int

Name: 厂家名称,string

}

},

…

]

}

注意：当气瓶档案的State!=0、下检日期或报废日期已过期这几种情况下都不能充装

* 更新同步流程

客户端采用间隔轮询下载有更新的气瓶档案

1.访问上面接口，下载有变更的气瓶档案。如果客户端是第1次启动,参数lastDate传入空值；如果是重新运行或下次轮询，则传入上一次本地保存的lastDate

2.服务器返回一批有变更的气瓶档案列表

3.客户端循环每笔气瓶档案，更新本地气瓶档案库，本批气瓶档案都循环处理完毕后，持久保存最后一笔气瓶档案的UpdateDate。UpdateDate即第1步所说的lastDate，保存的目的是为了防止程序崩溃重启后，能够从上一次中断处继续下载

4.继续访问上面接口，此时lastDate传入第3步保存的UpdateDate

4.1.如果服务器返回列表为空，则表示全部更新完毕,下次lastDate=UpdateDate加1秒

4.2. 如果服务器返回列表不为空，表示还有记录, 重复第1步

# 气瓶流转

## 充前预检上传

* POST

http://对接平台地址/Steel/Trace/BeforeFill?data=json格式数据&token=令牌

参数data的json格式

{

//钢瓶记录

Items:[

//第1项

{

Barcode:钢瓶条形码,string

Damage:缺陷部位,int,用bit位表示,每位用1/0表示, 1表示相应的位置打勾,从低位到高位顺序: 可疑气瓶,护罩损坏,底座损坏,瓶阀损坏,瓶体裂纹,瓶体焊疤,缺防震圈,瓶体变形,颜色不符,瓶号不符,介质不符,瓶体腐蚀,油脂污损,瓶体火烧,外观凹坑,瓶阀缺损

ThirdId:第三方系统唯一ID,int,用于防止重复上传

},

…

]

//操作员

Man:{

Id: 系统ID,int

Name:姓名,string

}

TraceDate:本环节产生时间,,string, yyyyMMddHHmmss

}

* 响应

Json格式

{

State:状态代码,int,0:成功,1:失败

//失败时

Value:失败原因,string

}

## 充前预检下载

* POST

http://对接平台地址/Steel/Trace/BeforeFillList?lastId=上次系统ID &token=对接口令

* 响应

json格式

{

PlanCount:预计笔数,int,如果实际返回的笔数比该值小,表示已经全部下载完毕

//记录列表

Items: [

//第1笔记录

{

Id:系统ID,int

Barcode:钢瓶条码,string

State:状态,int,0:不合格,1:合格

Damage:缺陷部位,int,用bit位表示,每位用1/0表示, 1表示相应的位置打勾,从低位到高位顺序: 可疑气瓶,护罩损坏,底座损坏,瓶阀损坏,瓶体裂纹,瓶体焊疤,缺防震圈,瓶体变形,颜色不符,瓶号不符,介质不符,瓶体腐蚀,油脂污损,瓶体火烧,外观凹坑,瓶阀缺损

EndDate: 检查时间,string,yyyyMMddHHmmss

ManId: 检查人系统Id,int

ManName: 检查人姓名,string

},

…

]

}

## 充装记录上传

* POST

http://对接平台地址/Steel/Trace/Fill321?data=json格式数据&token=令牌

参数data的json格式

{

//充装记录

Items:[

//第1项

{

Barcode:钢瓶条形码,string

NetWeight:净重, float(2),小数点2位

IdleWeight:瓶重, float(2),小数点2位

GrossWeight:毛重, float(2),小数点2位

BeginDate:开始时间,string,yyyyMMddHHmmss

EndDate: 结束时间,string,yyyyMMddHHmmss

Machine:充装秤号,int

Temperature:摄氏温度,float(1)

CustomerNo:客户编号,string

ThirdId:第三方系统唯一ID,int,用于防止重复上传

},

…

]

//充装秤厂家

Factory:{

Id:厂家ID,int,1:祥康,2:兰洋,3:长龙,4:兰洋旧秤

Name:厂家名称,string

}

//操作员

Man:{

Id: 系统ID,int

Name:姓名,string

}

TraceDate:本环节产生时间,,string, yyyyMMddHHmmss

}

* 响应

Json格式

{

State:状态代码,int,0:成功,1:失败

//失败时

Value:失败原因,string

}

## 充装记录下载

* POST

http://对接平台地址/Steel/Trace/FillList?lastId=上次系统ID &token=对接口令

* 响应

json格式

{

PlanCount:预计笔数,int,如果实际返回的笔数比该值小,表示已经全部下载完毕

//记录列表

Items: [

//第1笔记录

{

Id:系统ID,int

Barcode:钢瓶条码,string

NetWeight:净重, float(2),小数点2位

IdleWeight:瓶重, float(2),小数点2位

GrossWeight:毛重, float(2),小数点2位

BeginDate:开始时间,string,yyyyMMddHHmmss

EndDate: 结束时间,string,yyyyMMddHHmmss

FactoryId:秤厂家ID,int

Machine:充装秤号,int

ManId: 充装员系统Id,int

ManName: 充装员姓名,string

},

…

]

}

## 充后复检上传

* POST

http://对接平台地址/Steel/Trace/AfterFill?data=json格式数据&token=令牌

参数data的json格式

{

//钢瓶记录

Items:[

//第1项

{

Barcode:钢瓶条形码,string

Security:安全检查,int,用bit位表示,每位用1/0表示, 1表示相应的位置打勾,从低位到高位顺序: 密封检查,瓶温检查,瓶体检查,警示标签

Weight:复检重,float，默认抓验证钢瓶里的毛重

ThirdId:第三方系统唯一ID,int,用于防止重复上传

},

…

]

//操作员

Man:{

Id: 系统ID,int

Name:姓名,string

}

TraceDate:本环节产生时间, string, yyyyMMddHHmmss

}

* 响应

Json格式

{

State:状态代码,int,0:成功,1:失败

//失败时

Value:失败原因,string

}

## 充后复检下载

* POST

http://对接平台地址/Steel/Trace/AfterFillList?lastId=上次系统ID &token=对接口令

* 响应

json格式

{

PlanCount:预计笔数,int,如果实际返回的笔数比该值小,表示已经全部下载完毕

//记录列表

Items: [

//第1笔记录

{

Id:系统ID,int

Barcode:钢瓶条码,string

State:状态,int,0:不合格,1:合格

Security:安全检查,int,用bit位表示,每位用1/0表示, 1表示相应的位置打勾,从低位到高位顺序: 密封检查,瓶温检查,瓶体检查,警示标签

Weight:复检重,float

EndDate: 检查时间,string,yyyyMMddHHmmss

ManId: 检查人系统Id,int

ManName: 检查人姓名,string

},

…

]

}

# 报修单

## 下单

* POST

http://中间件地址/SteelVAS/Maintain/AddOrder?data=*json格式*&token=对接口令

data参数json格式

{

No:订单号码,string,为空的话系统会自动产生

//客户资料

Consumer:{

Id:客户ID,int

Name:客户名称,string

Address:客户地址,string

Tel:联系电话,string

}

//报修内容

Content:{

//报修类型列表

Items:[

{

Id:系统ID,int

Name:项目名称,string

},

…

]

Remark:具体内容,string

}

Priority:优先级,int,0:普通单,1:加急单

RequireDate:建议处理日期,string,yyyyMMdd

RequireOccasion:全天、上午、中午、下午、晚上等,string

//处理人,为空表示自动分配

Man:{

Id:系统ID,int,0:自动分配,-1:客服派单

Name:姓名,string

Tel:电话,string

}

//下单渠道

User:{

Id:系统ID,int,0:呼叫中心,1:微信

Name:名称,string

}

//第3方系统

Third:{

Id:在第3方系统上的ID,int

}

}

* 响应

Json格式

{

State:状态代码,int,0:成功,1:失败

//成功时

Value:字符串的json格式,string

{

Id:系统ID,int

}

//失败时

Value:失败原因,string

}

## 查询

* POST

http://中间件地址/SteelVAS/Maintain/GetOrder?filter=*json格式*&token=对接口令

filter参数json格式

{

//第3方系统

Third:{

Id:在第3方系统上的ID,int

}

}

* 响应

Json格式

{

Id:系统ID,int,-1表示不存在

State: 订单状态,int, 1:待调度/2:待处理/4:已处理/8:已取消

DoDate:完成时间,string,yyyyMMddHHmmss

//处理人,为空表示未派单

Man:{

Id:系统ID,int,0: 未派单

Name:姓名,string

Tel:电话,string

}

}

## 退单

* POST

http://中间件地址/SteelVAS/Maintain/RefundOrder?data=*json格式*&token=对接口令

data参数格式

{

//第3方系统

Third:{

Id:在第3方系统上的ID,int

}

Remark:原因备注,string

}

* 响应

Json格式

{

State:状态代码,int,0:成功,1:失败

//失败时

Value:失败原因,string

}

## 报修类型列表

* GET

http://中间件地址/SteelVAS/Code/MaintainTypeList?token=对接口令

* 响应

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| id | int | 代码 |
| name | String | 名称 |
|  |  |  |