

WMS_WCS交互协议

基础文档版本号: `R4.0

更新时间: 2024年5月20日

更新人: 葛林强

江苏菲达宝开电气股份有限公司

内部资料, 未经本公司书面授权禁止外泄

统一返回:

接口定义中标注统一返回的均按照下面方式返回。

键名	名称	数据类型	长度	是否必填	备注
code	返回码	int	4	是	操作正常返回 0, 数据重复返回 200, 其他为异常代码
message	说明信息	string	255	是	
returnData	返回数据	object		否	在需要返回数据的时候带出数据

堆垛机:

1、WMS发送堆垛机任务给WCS

客户端: WMS

服务端: WCS

调用方式: POST

接口地址: /api/wms/stacker/setStackerTask

使用场景: WMS在需要WMS执行任务时调用此接口发送需要执行的任务

键名	名称	数据类型	长度	是否必填	是否主键	备注
	WMS任务	List		是		发送的任务数组

键名	名称	数据类型	长度	是否必填	是否主键	备注
taskId	WMS任务编号	string	64	是	是	唯一任务号, 作为WCS上报任务凭证
taskType	任务类型	int		是	否	1: 入库 2: 出库 4: 拣选 9: 移库
priority	任务优先级	int		是	否	数字越大
origin	任务起点	string	32	是	否	
midpoint	任务中间点	string	32	否	否	预留点, 特殊情况需要
destination	任务终点	string	32	否	否	入库: 必填, 终点一般为库位 出库: 若填写, 则Wcs按照填写的位置出库 若不填, Wcs寻找空闲出库站台出库 移库: 必填, 终点一般为库位
vehicleNo	载具编号	string	32	是	否	
vehicleSize	载具尺寸	int		是	否	若没有则填 0
weight	总重量	decimal	(10,2)	是	否	若没有则填 0, 单位: KG

请求示例:

```
[  
  {  
    ...  
  }  
]
```

```
"taskId": "12345aaabbb",
"taskType": "1",
"priority": 0,
"origin": "101",
"midpoint": null,
"destination": "A-2-5-1",
"vehicleNo": "TP999",
"vehicleSize": 0,
"weight": 2.00,
}
]
```

// 这些数据中只要有一个存在问题，都会返回错误，所有任务就都无效

返回示例:

```
{
  "code": 0,
  "message": "任务创建成功"
}
```

2、WMS请求变更任务状态

客户端: WMS

服务端: WCS

调用方式: POST

接口地址: /api/wms/stacker/changeTaskStatus

使用场景: WMS系统内任务状态发生变更并且需要通知WCS时调用此接口

请求格式:

键名	名称	数据类型	长度	是否必填	是否主键	备注
taskId	WMS任务编号	string	64	是	是	唯一任务号, 下发任务时使用的
taskStatus	任务状态	int	4	是	否	0: 重新执行任务 1: 取消/删除任务 2: 完成任务 注意: 变更状态后若任务正在执行则设备可能仍然继续执行任务

请求示例:

```
{
  "taskId": "12423452345",
  "taskStatus": 1
}
```

响应示例:

```
{
  "code": 0,
  "message": "任务取消成功"
}
```

3、WCS反馈WMS任务状态

客户端: WCS

服务端: WMS

调用方式: POST

接口地址:

使用场景: WCS在任务执行的时间节点通知WMS任务状态

注意: 该接口WCS会尝试多次, 若多次未调用成功或WMS返回异常则不再调用

请求格式:

键名	名称	数据类型	长度	是否必填	是否主键	备注
taskId	WMS任务编号	string	64	是	是	唯一任务号，下发任务时使用的
taskStatus	任务状态	int	4	是	否	1: 任务排队中 2: 任务开始执行 3: 任务已经离开初始位置 4: 任务到达中间点 5: 任务到达目的地 100: 任务完成 998: 任务取消 999: 任务异常
destination	任务终点	string	32	是	否	入库任务填库位，出库任务填出库站台
message	任务信息	string	128	否	否	一般用作异常时填写信息

请求示例:

```
{
  "taskId": "235345634534",
  "taskStatus": 2,
  "destination": "102",
  "message": "任务已经开始执行"
}
```

响应示例:

```
{
  "code": 0,
  "message": "成功"
}
```

4、WCS请求载具入库

客户端: WCS
服务端: WMS
调用方式: POST
接口地址:

使用场景: WCS在扫码入库/盘点回库等情况下需要向WMS发送请求信号，WMS收到此信号后若正常则通过任务下发接口下发任务给WCS

请求格式:

键名	名称	数据类型	长度	是否必填	是否主键	备注
point	点位	string	32	是	否	
vehicleNo	载具编号	string	32	是	否	
codeMessage	条码信息	string	512	否	否	若条码含物料信息或者其他需要一并传递的信息将会出现在此处
remark	备注	string	256	否	否	

响应格式(returnData):

键名	名称	数据类型	长度	是否必填	是否主键	备注
inStand	入库站台	string	64	否	否	入库站台
remark	备注信息	string	128	否	否	

请求示例:

```
{
  "point": "101",
  "vehicleNo": "TP1001",
  "codeMessage": "010001@03423423@手榴弹@100@枚",
  "remark": ""
}
```

响应示例:

```
{
  "code": 0,
  "message": "请求成功",
  "returnData": {
    "inStand": "101",
    "reamrk": ""
  }
}
```

5、WMS向WCS推送新的目的地

客户端: WMS

服务端: WCS

调用方式: POST

接口地址: /api/wms/stacker/setStackerTaskNewDestination

使用场景: WMS在设备重复入库时推送新终点任务给WCS

键名	名称	数据类型	长度	是否必填	是否主键	备注
taskId	WMS任务编号	string	64	是	是	唯一任务号, 和卸货位置有货时的任务号一致
destination	任务终点	string	32	是	否	
vehicleNo	载具编号	string	32	是	否	

请求示例:

```
{
  "taskId": "12334534",
  "destination": "010101",
  "vehicleNo": "T0001"
}
```

箱式线:

1、WMS向WCS发送输送任务

客户端: WMS

服务端: WCS

调用方式: POST

接口地址: /api/wms/convey/setConveyTask

使用场景: WMS推送拣选任务给WCS

请求格式:

键名	名称	数据类型	长度	是否必填	是否主键	备注
taskGroup	任务组编号	string	40	否	否	不填的情况下WCS自动生成
vehicleNo	载具号	string	32	是	否	

键名	名称	数据类型	长度	是否必填	是否主键	备注
taskType	任务类型	int		是	否	任务类型: 1: 拣选 2: 补货 3: 发货 4: 复核
location	点位	List	128	是	否	
remark	备注	string	256	否	否	

请求示例:

```
{
  "taskGroup": "q231231231432",
  "vehicleNo": "TP1001",
  "taskType": 1,
  "location": ["2001", "2002"],
  "remark": ""
}
```

响应示例:

```
{
  "code": 0,
  "message": "请求成功"
}
```

```
{
  "code": 1,
  "message": "请求失败, 当前任务未全部完结"
}
```

2、WCS向WMS上报箱子到达

客户端: WMS

服务端: WCS

调用方式: POST

接口地址:

使用场景: WCS向WMS上报箱子到达

请求格式:

键名	名称	数据类型	长度	是否必填	是否主键	备注
taskGroup	任务组编号	string	4	是	否	
vehicleNo	载具号	string	32	是	否	
location	点位	string	32	是	否	
remark	备注	string	256	否	否	

请求示例:

```
{
  "taskGroup": "q231231231432",
  "vehicleNo": "TP1001",
  "location": "2001",
  "remark": ""
}
```

响应示例:

```
{
  "code": 0,
  "message": "请求成功"
}
```

电子标签

1、WMS向WCS发送电子标签任务

客户端：WMS

服务端：WCS

请求方式：POST

接口地址：

请求格式：

键名	名称	数据类型	长度	是否必填	是否主键	备注
taskGroup	任务组	string	40	否	否	不填情况下由WCS自动生成 每次发送任务时必须都不一样
taskType	任务类型	int		是	否	任务类型： 1：拣选/配件 2：存储任务
vehicleNo	载具号	string	32	是	否	箱号
orderId	订单号	string	32	否	否	
taskData	任务数据	List		是	否	任务数据列表

taskData 数据

键名	名称	数据类型	长度	是否必填	是否主键	备注
taskId	任务号	string	40	否	是	不填情况下由WCS自动生成
location	点位	string	32	是	否	需要拣选或者存储的点位
goodsId	物料编号	string	32	否	否	
goodsName	物料名称	string	64	否	否	
needNum	需求数量	int		是	否	部分情况下若不需要数量请填写 0

请求示例：

```
{
  "taskGroup": "Boom1234345",
  "taskType": 1,
  "vehicleNo": "P0001",
  "orderId": "DFG1121312312",
  "taskData": [
    {
      "taskId": "T231231",
      "location": "1-L-001",
      "goodsId": "B0001",
      "goodsName": "宝开牌手榴弹",
      "needNum": 20
    },
    {
      "taskId": "T243534553453451",
      "location": "1-L-040",
      "goodsId": "R0801",
      "goodsName": "火箭筒",
      "needNum": 13
    }
  ]
}
```

响应示例：

```
{
  "code": 0,
  "message": "请求成功"
}
```

2、WCS向WMS反馈电子标签确认

客户端: WCS

服务端: WMS

请求方式: POST

请求地址:

请求格式:

键名	名称	数据类型	长度	是否必填	是否主键	备注
taskGroup	任务组	string	40	否	否	
taskType	任务类型	int		是	否	任务类型: 1: 拣选/配件 2: 存储任务
vehicleNo	载具号	string	32	是	否	箱号
orderId	订单号	string	32	否	否	
taskId	任务号	string	40	否	是	
location	点位	string	32	是	否	需要拣选或者存储的点位
goodsId	物料编号	string	32	否	否	
goodsName	物料名称	string	64	否	否	
needNum	需求数量	int		是	否	部分情况下若不需要数量请填写 0
confirmNum	确认数量	int		是	否	电子标签的按下确认时显示的数量

请求示例:

```
{
  "taskGroup": "123123123",
  "taskType": 1,
  "vehicleNo": "V12123",
  "orderId": "234213",
  "taskId": "456456",
  "location": "1-L-231",
  "goodsId": "7867867",
  "goodsName": "宝开牌迫击炮",
  "needNum": 34,
  "confirmNum": 12
}
```

响应示例:

```
{
  "code": 0,
  "message": "请求成功"
}
```

