

HAIQ 设备调度平台 (ESS)

API 接口文档

(产品版本: 24.4.1.0)

文档版本: V3.2

发布日期: 2024-12-30

前言

感谢您选择深圳市海柔创新科技有限公司（以下简称“海柔创新”）的产品，本手册记录了正确使用本产品的所需事项，请在使用本产品之前仔细阅读本手册内容。请妥善保管本手册，以便日后随时取阅。



内容概要

本手册主要描述了 HAIQ 智慧管理平台的设备调度系统（Equipment Scheduling System，以下简称“ESS”）与客户上游系统对接的接口，包括上游系统如何下发业务任务，如何查询机器人和工作位状态，以及 ESS 如何响应的内容。






阅读对象

- 开发工程师
- 系统工程师
- 方案集成商
- 产品经理



标识约定

本手册中与安全或关键信息相关的内容，使用下列标识。当手册中出现以下标识的内容，请务必留意和遵守。

标识	说明
 警告	表示会出现潜在的危險情形，若不可避免，可能会造成人员伤害、机器人毁坏等情况。
 注意	表示会出现不可预知的情形，若不可避免，可能会导致机器人损坏、性能降低、数据丢失等情况。
 提示	表示关键信息阐述、操作技巧提示。



版权声明

© 2024 深圳市海柔创新科技有限公司，版权所有。

海柔创新拥有本产品及其软件的专利、版权和其它知识产权，未经本公司书面授权，任何单位和个人不得擅自摘抄、复制本手册内容；不得直接或间接复制、制造、加工、使用本产品及其相关部分，不得将本手册的部分或全部内容以任何形式用于商业用途。



商标声明

HAIROBOTICS 为深圳市海柔创新科技有限公司的注册商标，海柔创新拥有此商标的所有权，未经本公司书面许可不得用于其它商业用途。



免责声明

由于产品的更新、改良或其它原因，本手册中内容和产品会不定期进行版本更新；本公司保留在不事先通知的情况下，修改本手册中产品和产品规格等文件的权利；本手册中的资料仅用于参考，实际情况以产品实物为准，恕不另行通知。

海柔创新不承担由于使用本手册或本产品不当，或与其相关的任何直接的、间接的、特殊的、附带的或相应产生的损失和责任。



联系方式

深圳市海柔创新科技有限公司
深圳市宝安区西乡街道南昌社区固戍二路 99 号

联系电话: +86-0755-23208112

代理合作: partnership@hairobotics.com

技术支持: support@hairobotics.com

公司网站: www.hairobotics.com

版本记录

文档版本	发布日期	修订内容
V1.0	2024-03-31	第一次正式发布。
V2.0	2024-06-30	第二次正式发布。
V3.0	2024-09-30	第三次正式发布。
V3.1	2024-10-10	第四次正式发布。
V3.2	2024-12-30	第五次正式发布。 更新称重盘点中的“taskDescribe 参数说明”。

目录

前言	I
版本记录	III
目录	IV
第 1 章 接口概述	1
1.1 接入方式	1
1.2 接口 header	1
1.3 接口清单	1
1.4 参数说明	2
第 2 章 常用接口	3
2.1 ESS 提供的接口	3
2.1.1 任务下发请求	3
2.1.1.1 入库	3
2.1.1.2 出库	8
2.1.1.3 移库	13
2.1.1.4 点对点搬运	18
2.1.1.5 扫描盘点	23
2.1.1.6 称重盘点	26
2.1.1.7 RFID 盘点	30
2.1.1.8 模板任务	33
2.1.2 任务取消请求	37
2.1.2.1 接口描述	37
2.1.2.2 请求参数说明	38
2.1.2.3 请求示例	38
2.1.2.4 响应参数说明	38
2.1.2.5 响应示例	39
2.2 上游系统提供的接口	40
2.2.1 任务状态回调	40
2.2.1.1 接口描述	40
2.2.1.2 回调参数说明	41
2.2.1.3 回调示例	42
2.2.1.4 响应参数说明	46
2.2.1.5 响应示例	47
第 3 章 可选接口	48

3.1 ESS 提供的接口	48
3.1.1 任务优先级调整	48
3.1.1.1 接口描述	48
3.1.1.2 请求参数说明	48
3.1.1.3 请求示例	49
3.1.1.4 响应参数说明	49
3.1.1.5 响应示例	50
3.1.2 容器创建	51
3.1.2.1 接口描述	51
3.1.2.2 请求参数说明	51
3.1.2.3 请求示例	52
3.1.2.4 响应参数说明	52
3.1.2.5 响应示例	53
3.1.3 容器移除	55
3.1.3.1 接口描述	55
3.1.3.2 请求参数说明	55
3.1.3.3 请求示例	55
3.1.3.4 响应参数说明	55
3.1.3.5 响应示例	56
3.1.4 容器入场	58
3.1.4.1 接口描述	58
3.1.4.2 请求参数说明	58
3.1.4.3 请求示例	58
3.1.4.4 响应参数说明	59
3.1.4.5 响应示例	59
3.1.5 容器离场	61
3.1.5.1 接口描述	61
3.1.5.2 请求参数说明	61
3.1.5.3 请求示例	62
3.1.5.4 响应参数说明	62
3.1.5.5 响应示例	63
3.1.6 工作位锁定	64
3.1.6.1 接口描述	64
3.1.6.2 请求参数说明	65
3.1.6.3 请求示例	65

3.1.6.4 响应参数说明	65
3.1.6.5 响应示例	66
3.1.7 工作位解锁	67
3.1.7.1 接口描述	67
3.1.7.2 请求参数说明	67
3.1.7.3 请求示例	68
3.1.7.4 响应参数说明	68
3.1.7.5 响应示例	68
3.1.8 呼叫机器人至工作位	70
3.1.8.1 接口描述	70
3.1.8.2 请求参数说明	70
3.1.8.3 请求示例	70
3.1.8.4 响应参数说明	71
3.1.8.5 响应示例	71
3.1.9 让机器人离开工作位	71
3.1.9.1 接口描述	71
3.1.9.2 请求参数说明	72
3.1.9.3 请求示例	72
3.1.9.4 响应参数说明	72
3.1.9.5 响应示例	73
3.1.10 系统暂停	73
3.1.10.1 接口描述	73
3.1.10.2 请求参数说明	73
3.1.10.3 请求示例	74
3.1.10.4 响应参数说明	74
3.1.10.5 响应示例	74
3.1.11 系统恢复	74
3.1.11.1 接口描述	74
3.1.11.2 请求参数说明	75
3.1.11.3 请求示例	75
3.1.11.4 响应参数说明	75
3.1.11.5 响应示例	75
3.1.12 容器查询	75
3.1.12.1 接口描述	75
3.1.12.2 请求参数说明	76

3.1.12.3 请求示例.....	76
3.1.12.4 响应参数说明.....	76
3.1.12.5 响应示例.....	77
3.1.13 机器人查询.....	78
3.1.13.1 接口描述.....	78
3.1.13.2 请求参数说明.....	78
3.1.13.3 请求示例.....	78
3.1.13.4 响应参数说明.....	78
3.1.13.5 响应示例.....	81
3.1.14 工作位查询.....	84
3.1.14.1 接口描述.....	84
3.1.14.2 请求参数说明.....	85
3.1.14.3 请求示例.....	85
3.1.14.4 响应参数说明.....	85
3.1.14.5 响应示例.....	87
3.2 上游系统提供的接口.....	88
3.2.1 任务申请.....	88
3.2.1.1 接口描述.....	88
3.2.1.2 回调参数说明.....	88
3.2.1.3 请求示例.....	89
3.2.1.4 响应参数说明.....	89
3.2.1.5 响应示例.....	89
3.2.2 异常上报.....	90
3.2.2.1 接口描述.....	90
3.2.2.2 回调参数说明.....	90
3.2.2.3 回调示例.....	91
3.2.2.4 响应参数说明.....	91
3.2.2.5 响应示例.....	91
第 4 章 周边设备控制接口.....	92
4.1 电子标签接口.....	92
4.1.1 ESS 提供的接口.....	92
4.1.1.1 电子标签控制请求.....	92
4.1.2 上游系统提供的接口.....	94
4.1.2.1 电子标签信号回传.....	94
4.2 输送线接口.....	96

4.2.1 ESS 提供的接口	96
4.2.1.1 容器流动通知	96
4.2.1.2 输送线线体状态查询	98
4.2.2 上游系统提供的接口	103
4.2.2.1 容器到达上报	103
第 5 章 异常码	106
第 6 章 典型场景示例	110
6.1 AGV 搬运容器出库场景	110
6.2 AVG 搬运容器入库场景	111

第1章 接口概述

1.1 接入方式

- ESS 标准接口采用 HTTP 协议，数据交换格式为 JSON，接口访问端口统一为 9046。
- 编码格式为 UTF-8，输入参数严格区分大小写。
- 中文输入使用 `URLEncoder.encode("string", "UTF-8")`。
- 获取中文字段使用 `URLDecoder.decode("string", "UTF-8")`。

1.2 接口 header

接口 header 用于描述接口版本，ESS 24.1.1.0 及之后的版本支持对接口头部进行校验。后文中，不同的接口只描述请求体和响应体。

接口 header 示例

```
"api-version": "v2.0"
```

字段说明

参数名	必选	类型	说明
api-version	Y	string	接口版本。

1.3 接口清单

接口类型	接口名称	调用方向
常用接口	任务下发请求	上游系统→ESS
	任务取消请求	
	任务状态回调	ESS→上游系统
可选接口	任务优先级调整	上游系统→ESS
	容器创建/移除	
	容器入场/离场	
	工位锁定/解锁	
	机器人呼叫/离开	
	系统暂停/恢复	
	容器查询	
	机器人查询	
	工位查询	

接口类型	接口名称		调用方向
	任务申请		ESS→上游系统
	异常上报		ESS→上游系统
周边设备控制接口	电子标签	电子标签控制请求	上游系统→ESS
		电子标签信号回传	ESS→上游系统
	输送线	容器流动通知	上游系统→ESS
		输送线线体状态查询	上游系统→ESS
		容器到达上报	ESS→上游系统

1.4 参数说明

- 本文中 int 和 long 类型的参数，除特殊说明以外，均为非负值。
- 本文中 Map 类型的参数按以下定义：
 - 键：string 类型，长度不超过 64。
 - 值：string 类型，长度不超过 255。

第2章 常用接口

2.1 ESS 提供的接口

2.1.1 任务下发请求

接口 URL	http://{IP:PORT}/task/create
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	上游系统通过此接口，向 ESS 下发任务。 为达到最佳性能，建议单次请求不超过 100 行明细，请求频率不超过 20 次/秒。
备注	该接口单次调用最多可下发 200 条任务。

2.1.1.1 入库

入库任务类型为 putaway，用于以下两种上架场景：

- 场景一：机器人扫码取箱需要校验容器是否有上架任务
- 场景二：输送线规则校验容器是否存在上架任务

请求参数说明

表 2-1 请求参数说明

参数名	必选	类型	说明
taskType	Y	string	任务类型，取值为 putaway：用于需要校验容器是否有上架任务的入库场景。
taskGroupCode	N	string	任务组号，若不传值则由系统自动生成。
groupPriority	N	int	任务组优先级，取值范围为【0~2147483647】，2147483647 为 int 类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级组的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
tasks	Y	list<task>	任务列表。详情请参见“tasks 参数说明”。

表 2-2 tasks 参数说明

参数名	必选	类型	说明
taskCode	Y	string	业务任务号。

参数名	必选	类型	说明
taskPriority	N	int	<p>任务优先级，取值范围为【0~2147483647】，2147483647为int类型的最大值。</p> <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
taskDescribe	Y	object<taskDescribe>	任务描述，描述任务所需要的参数。详情请参见“taskDescribe 参数说明”。

表 2-3 taskDescribe 参数说明

参数名	必选	类型	说明
containerCode	C	string	<p>容器编码。</p> <ul style="list-style-type: none"> ● 容器编码和起始工位编码二选一必填。 ● 容器编码与起始工位均有值时，系统取容器编码的值。 ● 容器编码在系统内不存在时，系统将自动创建型号为“CT_KUBOT_STANDARD”的容器。 ● 工作站入库任务不传容器编码，只传源位置和目标位置，上游需控制任务下发，相同源位置的入库任务一次只能下发一条，机器人扫码后更新容器。
containerType	N	string	<p>容器类型，默认值为“CT_KUBOT_STANDARD”。</p> <ul style="list-style-type: none"> ● 填写容器类型时： <ul style="list-style-type: none"> ■ 容器编码在系统中存在，原容器类型和现容器类型一致时，容器类型不变。 ■ 容器编码在系统中存在，原容器类型和现容器类型不一致时，更新为现容器类型。 ■ 容器编码在系统中不存在，使用该容器类型创建容器。 ● 若不填容器类型： <ul style="list-style-type: none"> ■ 容器编码在系统中存在，使用原容器类型。 ■ 容器编码在系统中不存在，使用系统默认的容器类型“CT_KUBOT_STANDARD”创建容器。
storageTag	N	string	工位标签，系统会根据标签为容器分配对应的工作位。适用于逻辑分区、大小库位等场景。
fromLocationCode	N	string	起始工位。

参数名	必选	类型	说明
			<ul style="list-style-type: none"> 容器编码和起始工位编码二选一必填。 容器编码与起始工位均有值时，系统取容器编码的值。 对于上游系统不调用容器入场接口的场景，开启容器自动入场配置，系统创建入库任务时，会自动将容器入场到起始工位fromLocationCode；若系统创建入库任务失败，会将此容器从起始工位fromLocationCode移出。fromLocationCode不能为空，但containerCode可以为空，containerCode为空时会入场一个默认容器到fromLocationCode。
toLocationCode	C	string	<ul style="list-style-type: none"> toLocationCode: 目标工位编码。
toStationCode	C	string	<ul style="list-style-type: none"> toStationCode: 目标工作站编码。 灰盒入库存在如下两种场景： <ol style="list-style-type: none"> 目标工位和目标工作站均不填。 目标工位和目标工作站二选一必填。 白盒入库场景下，目标工位必填。 目标工位和目标工作站均有值时，系统取目标工位的值。 目前工位仅支持填写一个。 目前工作站支持填写多个，需要以逗号隔开。 若填写多个目标工作站，表明从多个目标工作站中分配一个启用的工作站。
containerFace	N	string	<p>目标容器朝向，取值为枚举值 A、B、C、D。常用于出库 HAIFLEX 需要翻面的场景。</p> <ul style="list-style-type: none"> A: 0 B: 90 C: 180 D: 270
deadline	N	long	<p>截单时间，格式为时间戳，单位为 ms。</p> <p>截单时间规则可做如下配置：距离截单时间 N ms 时，任务在原来的基础上提高 P1 个优先级；可配置多条规则。</p>

请求示例

```
{
  "taskType": "putaway",
  "taskGroupCode": "taskGroupCode-001",
```

```

"groupPriority": 0,
"tasks": [
  {
    "taskCode": "taskCode-001",
    "taskPriority": 0,
    "taskDescribe": {
      "containerCode": "container001",
      "containerType": "CT_KUBOT_STANDARD",
      "storageTag": "",
      "fromLocationCode": "S07-001-008-02",
      "toStationCode": "haiport_station_1",
      "toLocationCode": "HAI-001-001-01",
      "deadline": 0
    }
  }
]
}

```

响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 1: 表示未完全成功, 即存在部分任务/容器/工位响应成功, 部分响应失败。 ● 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时, msg值为success。 ● 当code为1时, msg值为partial response failure。 ● 当code为其他值时, msg值为error。
data	Y	object	返回任务处理详情, 详情请参见“tasks 参数说明”。

表 2-4 tasks 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示处理成功。 ● 其他值: 表示处理失败, 详见异常码。
message	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当errorCode为0时, message值为OK。 ● 当errorCode为其他值时, message值为错误码对应的错

字段	必选	定义	备注
			误描述。
taskcode	Y	string	任务编码。

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "tasks": [
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-26-572"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-261-501"
      }
    ]
  }
}
```

失败响应示例一

```
{
  "code": 1010100001,
  "msg": "error",
  "data": {
    "tasks": [
      {
        "errorCode": "1030600022",
        "message": "toLocationCode[L08SR-007-002-15] is not exist",
        "taskCode": "task-16601-683"
      }
    ]
  }
}
```

失败响应示例二

```
{
  "code": 1030300007,
  "msg": "error",
```

```
"data": null
}
```

失败响应示例三

```
{
  "code": 1,
  "msg": "partial response failure",
  "data": {
    "tasks": [
      {
        "errorCode": "1030600022",
        "message": "toLocationCode[L08SR-007-002-14] is not exist",
        "taskCode": "task-16651-105"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-1665-887"
      }
    ]
  }
}
```

2.1.1.2 出库

出库任务，任务类型为 carry。

请求参数说明

表 2-5 请求参数说明

参数名	必选	类型	说明
taskType	Y	string	任务类型，取值为 carry。
taskGroupCode	N	string	任务组号，若不传值则由系统自动生成。
groupPriority	N	int	任务组优先级，取值范围为【0~2147483647】，2147483647 为 int 类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级组的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
tasks	Y	list<task>	任务列表。详情请参见“tasks 参数说明”。

表 2-6 tasks 参数说明

参数名	必选	类型	说明
taskCode	Y	string	业务任务号。
taskPriority	N	int	<p>任务优先级，取值范围为【0~2147483647】，2147483647为int类型的最大值。</p> <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
taskDescribe	Y	object<taskDescribe>	任务描述，描述任务所需要的参数。详情请参见“taskDescribe 参数说明”。

表 2-7 taskDescribe 参数说明

参数名	必选	类型	说明
containerCode	C	string	<p>容器编码。</p> <ul style="list-style-type: none"> ● 容器编码和起始工位编码二选一必填。 ● 容器编码与起始工位均有值时，系统取容器编码的值。 ● 容器编码在系统内不存在时，系统将自动创建型号为“CT_KUBOT_STANDARD”的容器。
containerType	N	string	<p>容器类型，默认值为“CT_KUBOT_STANDARD”。</p> <ul style="list-style-type: none"> ● 填写容器类型时： <ul style="list-style-type: none"> ■ 容器编码在系统中存在，原容器类型和现容器类型一致时，容器类型不变。 ■ 容器编码在系统中存在，原容器类型和现容器类型不一致时，更新为现容器类型。 ■ 容器编码在系统中不存在，使用该容器类型创建容器。 ● 若不填容器类型： <ul style="list-style-type: none"> ■ 容器编码在系统中存在，使用原容器类型。 ■ 容器编码在系统中不存在，使用系统默认的容器类型“CT_KUBOT_STANDARD”创建容器。
storageTag	N	string	工位标签，系统会根据标签为容器分配对应的工作位。适用于逻辑分区、大小库位等场景。
fromLocationCode	N	string	<p>起始工位。</p> <ul style="list-style-type: none"> ● 容器编码和起始工位编码二选一必填，容器编码为空则取起始工位的值。

参数名	必选	类型	说明
			<ul style="list-style-type: none"> 容器编码与起始工位均有值时，系统取容器编码的值。 开启容器自动入场配置时，系统自动将容器入场到起始工位fromLocationCode，所以fromLocationCode不能为空，但containerCode可以为空，containerCode为空时会入场一个默认容器到fromLocationCode。适用于客户不调用容器入场接口的场景。
toLocationCode	C	string	<ul style="list-style-type: none"> toLocationCode: 目标工位编码。
toStationCode	C	string	<ul style="list-style-type: none"> toStationCode: 目标工作站编码。 灰盒出库场景下，目标工位和目标工作站二选一必填。 白盒出库存在如下两种场景： <ol style="list-style-type: none"> 缓存货架工作站：目标工位必填。 非缓存货架工作站：目标工位和目标工作站二选一。 目标工位和目标工作站均有值时，系统取目标工位的值。 目前工位仅支持填写一个。 目前工作站支持填写多个，需要以逗号隔开。 若填写多个目标工作站，表明从多个目标工作站中分配一个启用的工作站。
containerFace	N	string	<p>目标容器朝向，取值为枚举值 A、B、C、D。常用于 HAIFLEX 需要翻面的场景。</p> <ul style="list-style-type: none"> A: 0 B: 90 C: 180 D: 270
deadline	N	long	<p>截单时间，格式为时间戳，单位为 ms。</p> <p>截单时间规则可做如下配置：距离截单时间 N ms 时，任务在原来的基础上提高 P1 个优先级；可配置多条规则。</p>

请求示例

```
{
  "taskType": "carry",
  "taskGroupCode": "taskGroupCode-001",
  "groupPriority": 0,
  "tasks": [
    {
      "taskCode": "taskCode-001",
```

```

    "taskPriority": 0,
    "taskDescribe": {
      "containerCode": "Container001",
      "containerType": "CT_KUBOT_STANDARD",
      "storageTag": "",
      "fromLocationCode": "S07-001-008-02",
      "toStationCode": "haiport_station_1,haiport_station_2",
      "toLocationCode": "HAI-001-001-01",
      "deadline":0
    }
  }
]
}

```

响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 1: 表示未完全成功,即存在部分任务/容器/工位响应成功,部分响应失败。 ● 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时, msg值为success。 ● 当code为1时, msg值为partial response failure。 ● 当code为其他值时, msg值为error。
data	Y	object	返回任务处理详情, 详情请参见“tasks 参数说明”。

表 2-8 tasks 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示处理成功。 ● 其他值: 表示处理失败, 详见异常码。
message	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当errorCode为0时, message值为OK。 ● 当errorCode为其他值时, message值为错误码对应的错误描述。
taskcode	Y	string	任务编码。

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "tasks": [
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-26-572"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-261-501"
      }
    ]
  }
}
```

失败响应示例一

```
{
  "code": 1010100001,
  "msg": "error",
  "data": {
    "tasks": [
      {
        "errorCode": "1030600022",
        "message": "toLocationCode[L08SR-007-002-15] is not exist",
        "taskCode": "task-16601-683"
      }
    ]
  }
}
```

失败响应示例二

```
{
  "code": 1030300007,
  "msg": "error",
  "data": null
}
```

失败响应示例三

```

{
  "code": 1,
  "msg": "partial response failure",
  "data": {
    "tasks": [
      {
        "errorCode": "1030600022",
        "message": "toLocationCode[L08SR-007-002-14] is not exist",
        "taskCode": "task-16651-105"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-1665-887"
      }
    ]
  }
}

```

2.1.1.3 移库

搬运容器移库任务，任务类型为 carry。

请求参数说明

表 2-9 请求参数说明

参数名	必选	类型	说明
taskType	Y	string	任务类型，取值为 carry。
taskGroupCode	N	string	任务组号，若不传值则由系统自动生成。
groupPriority	N	int	任务组优先级，取值范围为【0~2147483647】，2147483647 为 int 类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级组的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
tasks	Y	list<task>	任务列表。详情请参见“tasks 参数说明”。

表 2-10 tasks 参数说明

参数名	必选	类型	说明
taskCode	Y	string	业务任务号。
taskPriority	N	int	任务优先级，取值范围为【0~2147483647】，

参数名	必选	类型	说明
			2147483647 为 int 类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
taskDescribe	Y	object<taskDescribe>	任务描述，描述任务所需要的参数。详情请参见“taskDescribe 参数说明”。

表 2-11 taskDescribe 参数说明

参数名	必选	类型	说明
containerCode	C	string	容器编码。 <ul style="list-style-type: none"> ● 容器编码和起始工位编码二选一必填。 ● 容器编码与起始工位均有值时，系统取容器编码的值。 ● 容器编码在系统内不存在时，系统将自动创建型号为“CT_KUBOT_STANDARD”的容器。
containerType	N	string	容器类型，默认值为“CT_KUBOT_STANDARD”。 <ul style="list-style-type: none"> ● 填写容器类型时： <ul style="list-style-type: none"> ■ 容器编码在系统中存在，原容器类型和现容器类型一致时，容器类型不变。 ■ 容器编码在系统中存在，原容器类型和现容器类型不一致时，更新为现容器类型。 ■ 容器编码在系统中不存在，使用该容器类型创建容器。 ● 若不填容器类型： <ul style="list-style-type: none"> ■ 容器编码在系统中存在，使用原容器类型。 ■ 容器编码在系统中不存在，使用系统默认的容器类型“CT_KUBOT_STANDARD”创建容器。
storageTag	N	string	工位标签，系统会根据标签为容器分配对应的工作位。适用于逻辑分区、大小库位等场景。
fromLocationCode	N	string	起始工位。 <ul style="list-style-type: none"> ● 容器编码和起始工位编码二选一必填，容器编码为空则取起始工位的值。 ● 容器编码与起始工位均有值时，系统取容器编码的值。 ● 开启容器自动入场配置时，系统自动将容器入场到起始工位 fromLocationCode，所以 fromLocationCode 不能为空，但

参数名	必选	类型	说明
			containerCode 可以为空， containerCode 为空时会入场一个默认容器到 fromLocationCode 。适用于客户不调用容器入场接口的场景。
toLocationCode	C	string	<ul style="list-style-type: none"> ● toLocationCode: 目标工位编码: ● toStationCode: 目标工作站编码。 ● 灰盒移库场景下，目标工位和目标工作站二选一必填。 ● 白盒移库场景下，目标工位必填。 ● 目标工位和目标工作站均有值时，系统取目标工位的值。 ● 目前工位仅支持填写一个。 ● 目前工作站支持填写多个，需要以逗号隔开。 ● 若填写多个目标工作站,表明从多个目标工作站中分配一个启用的工作站。
toStationCode	C	string	
containerFace	N	string	目标容器朝向，取值为枚举值 A、B、C、D。常用于 HAIFLEX 需要翻面的场景。 <ul style="list-style-type: none"> ● A: 0 ● B: 90 ● C: 180 ● D: 270
deadline	N	long	截单时间，格式为时间戳，单位为 ms。 截单时间规则可做如下配置：距离截单时间 N ms 时，任务在原来的基础上提高 P1 个优先级；可配置多条规则。

请求示例

```
{
  "taskType": "carry",
  "taskGroupCode": "taskGroupCode-001",
  "groupPriority": 0,
  "tasks": [
    {
      "taskCode": "taskCode-001",
      "taskPriority": 0,
      "taskDescribe": {
        "containerCode": "Container001",
        "containerType": "CT_KUBOT_STANDARD",
        "storageTag": "",
        "fromLocationCode": "S07-001-008-02",
        "toStationCode": "haiport_station_1",
        "toLocationCode": "HAI-001-001-01",

```

```

        "deadline": 0
    }
}
]
}

```

响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 1: 表示未完全成功, 即存在部分任务/容器/工作位响应成功, 部分响应失败。 ● 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时, msg值为success。 ● 当code为1时, msg值为partial response failure。 ● 当code为其他值时, msg值为error。
data	Y	object	返回任务处理详情, 详情请参见“tasks 参数说明”。

表 2-12 tasks 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示处理成功。 ● 其他值: 表示处理失败, 详见异常码。
message	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当errorCode为0时, message值为OK。 ● 当errorCode为其他值时, message值为错误码对应的错误描述。
taskcode	Y	string	任务编码。

成功响应示例

```

{
  "code": 0,
  "msg": "success",
  "data": {
    "tasks": [

```

```
{
  {
    "errorCode": "0",
    "message": "OK",
    "taskCode": "task-26-572"
  },
  {
    "errorCode": "0",
    "message": "OK",
    "taskCode": "task-261-501"
  }
]
}
```

失败响应示例一

```
{
  "code": 1010100001,
  "msg": "error",
  "data": {
    "tasks": [
      {
        "errorCode": "1030600022",
        "message": "toLocationCode[L08SR-007-002-15] is not exist",
        "taskCode": "task-16601-683"
      }
    ]
  }
}
```

失败响应示例二

```
{
  "code": 1030300007,
  "msg": "error",
  "data": null
}
```

失败响应示例三

```
{
  "code": 1,
  "msg": "partial response failure",
  "data": {
    "tasks": [
      {
        "errorCode": "1030600022",
```

```

        "message": "toLocationCode[L08SR-007-002-14] is not exist",
        "taskCode": "task-16651-105"
    },
    {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-1665-887"
    }
]
}
}

```

2.1.1.4 点对点搬运

点对点搬运任务，泛指工作位到工作位、工作位到库位等的任务，任务类型为 carry。

请求参数说明

表 2-13 请求参数说明

参数名	必选	类型	说明
taskType	Y	string	任务类型，取值为 carry。
taskGroupCode	N	string	任务组号，若不传值则由系统自动生成。
groupPriority	N	int	任务组优先级，取值范围为【0~2147483647】，2147483647 为 int 类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级组的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
tasks	Y	list<task>	任务列表。详情请参见“tasks 参数说明”。

表 2-14 tasks 参数说明

参数名	必选	类型	说明
taskCode	Y	string	业务任务号。
taskPriority	N	int	任务优先级，取值范围为【0~2147483647】，2147483647 为 int 类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。

参数名	必选	类型	说明
taskDescribe	Y	object<taskDescribe>	任务描述，描述任务所需要的参数。详情请参见“taskDescribe 参数说明”。

表 2-15 taskDescribe 参数说明

参数名	必选	类型	说明
containerCode	C	string	<p>容器编码。</p> <ul style="list-style-type: none"> ● 容器编码和起始工位编码二选一必填 ● 容器编码与起始工位均有值时，系统取容器编码的值。 ● 容器编码在系统内不存在时，系统将自动创建型号为“CT_KUBOT_STANDARD”的容器。 ● 不传容器编码，只传源位置和目标位置，上游需控制任务下发，相同源位置的搬运任务一次只能下发一条。
containerType	N	string	<p>容器类型，默认值为“CT_KUBOT_STANDARD”。</p> <ul style="list-style-type: none"> ● 填写容器类型时： <ul style="list-style-type: none"> ■ 容器编码在系统中存在，原容器类型和现容器类型一致时，容器类型不变。 ■ 容器编码在系统中存在，原容器类型和现容器类型不一致时，更新为现容器类型。 ■ 容器编码在系统中不存在，使用该容器类型创建容器。 ● 若不填容器类型： <ul style="list-style-type: none"> ■ 容器编码在系统中存在，使用原容器类型。 ■ 容器编码在系统中不存在，使用系统默认的容器类型“CT_KUBOT_STANDARD”创建容器。
storageTag	N	string	<p>工位标签，系统会根据标签为容器分配对应的工作位。适用于逻辑分区、大小库位等场景。</p>
fromLocationCode	N	string	<p>起始工位。</p> <ul style="list-style-type: none"> ● 容器编码和起始工位编码二选一必填，容器编码为空则取起始工位的值。 ● 容器编码与起始工位均有值时，系统取容器编码的值。 ● 开启容器自动入场配置时，系统自动将容器入场到起始工位 fromLocationCode，所以 fromLocationCode 不能为空，但 containerCode 可以为空，containerCode 为空时会入场一个默认容器到 fromLocationCode。适用于客户不调用容器入场接口的场景。

参数名	必选	类型	说明
toLocationCode	C	string	<ul style="list-style-type: none"> ● toLocationCode: 目标工位编码。
toStationCode	C	string	<ul style="list-style-type: none"> ● toStationCode: 目标工作站编码。 ● 灰盒点对点搬运场景下, 目标工位和目标工作站二选一必填。 ● 白盒点对点搬运场景下, 目标工位必填。 ● 目标工位和目标工作站均有值时, 系统取目标工位的值。 ● 目前工位仅支持填写一个。 ● 目前工作站支持填写多个, 需要以逗号隔开。 ● 若填写多个目标工作站, 表明从多个目标工作站中分配一个启用的工作站。
containerFace	N	string	<p>目标容器朝向, 取值为枚举值 A、B、C、D。常用于 HAIFLEX 需要翻面的场景。</p> <ul style="list-style-type: none"> ● A: 0 ● B: 90 ● C: 180 ● D: 270
deadline	N	long	<p>截单时间, 格式为时间戳, 单位为 ms。</p> <p>截单时间规则可做如下配置: 距离截单时间 N ms 时, 任务在原来的基础上提高 P1 个优先级; 可配置多条规则。</p>

请求示例

```
{
  "taskType": "carry",
  "taskGroupCode": "taskGroupCode-001",
  "groupPriority": 0,
  "tasks": [
    {
      "taskCode": "taskCode-001",
      "taskPriority": 0,
      "taskDescribe": {
        "containerCode": "Container001",
        "containerType": "CT_KUBOT_STANDARD",
        "storageTag": "",
        "fromLocationCode": "S07-001-008-02",
        "toStationCode": "haiport_station_1",
        "toLocationCode": "HAI-001-001-01",
        "deadline": 0
      }
    }
  ]
}
```

```

    }
  ]
}

```

响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 1: 表示未完全成功, 即存在部分任务/容器/工位响应成功, 部分响应失败。 ● 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时, msg值为success。 ● 当code为1时, msg值为partial response failure。 ● 当code为其他值时, msg值为error。
data	Y	object	返回任务处理详情, 详情请参见“tasks 参数说明”。

表 2-16 tasks 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示处理成功。 ● 其他值: 表示处理失败, 详见异常码。
message	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当errorCode为0时, message值为OK。 ● 当errorCode为其他值时, message值为错误码对应的错误描述。
taskcode	Y	string	任务编码。

成功响应示例

```

{
  "code": 0,
  "msg": "success",
  "data": {
    "tasks": [
      {
        "errorCode": "0",

```

```
        "message": "OK",
        "taskCode": "task-26-572"
    },
    {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-261-501"
    }
]
}
```

失败响应示例一

```
{
  "code": 1010100001,
  "msg": "error",
  "data": {
    "tasks": [
      {
        "errorCode": "1030600022",
        "message": "toLocationCode[L08SR-007-002-15] is not exist",
        "taskCode": "task-16601-683"
      }
    ]
  }
}
```

失败响应示例二

```
{
  "code": 1030300007,
  "msg": "error",
  "data": null
}
```

失败响应示例三

```
{
  "code": 1,
  "msg": "partial response failure",
  "data": {
    "tasks": [
      {
        "errorCode": "1030600022",
        "message": "toLocationCode[L08SR-007-002-14] is not exist",
        "taskCode": "task-16651-105"
      }
    ]
  }
}
```

```

    },
    {
      "errorCode": "0",
      "message": "OK",
      "taskCode": "task-1665-887"
    }
  ]
}
}

```

2.1.1.5 扫描盘点

扫描盘点任务，任务类型为 scan。

请求参数说明

表 2-17 请求参数说明

参数名	必选	类型	说明
taskType	Y	string	任务类型，取值为 scan。
taskGroupCode	N	string	任务组号，若不传值则由系统自动生成。
groupPriority	N	int	任务组优先级，取值范围为【0~2147483647】，2147483647 为 int 类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级组的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
tasks	Y	list<task>	任务列表。详情请参见“tasks 参数说明”。

表 2-18 tasks 参数说明

参数名	必选	类型	说明
taskCode	Y	string	业务任务号。
taskPriority	N	int	任务优先级，取值范围为【0~2147483647】，2147483647 为 int 类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
taskDescribe	Y	object<taskDescribe>	任务描述，描述任务所需要的参数。详情请参见“taskDescribe 参数说明”。

表 2-19 taskDescribe 参数说明

参数名	必选	类型	说明
containerCode	N	string	容器编码。
locationCode	Y	string	目标工作位，即要进行盘库的目标库位。

请求示例

```
{
  "taskType": "scan",
  "taskGroupCode": "taskGroupCode-001",
  "groupPriority": 0,
  "tasks": [
    {
      "taskCode": "taskCode-001",
      "taskDescribe": {
        "containerCode": "Container001",
        "locationCode": "SUY-002-012-01"
      }
    }
  ]
}
```

响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 1: 表示未完全成功, 即存在部分任务/容器/工作位响应成功, 部分响应失败。 ● 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时, msg值为success。 ● 当code为1时, msg值为partial response failure。 ● 当code为其他值时, msg值为error。
data	Y	object	返回任务处理详情, 详情请参见“tasks 参数说明”。

表 2-20 tasks 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。

字段	必选	定义	备注
			<ul style="list-style-type: none">0: 表示处理成功。其他值: 表示处理失败, 详见异常码。
message	Y	string	响应消息补充说明。 <ul style="list-style-type: none">当errorCode为0时, message值为OK。当errorCode为其他值时, message值为错误码对应的错误描述。
taskcode	Y	string	任务编码。

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "tasks": [
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-1664-222"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-1665-811"
      }
    ]
  }
}
```

失败响应示例一

```
{
  "code": 1,
  "msg": "partial response failure",
  "data": {
    "tasks": [
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-1665-512"
      },
      {
        "errorCode": "1030600015",
```

```

        "message": "NO_SPECIFIED_CONTAINER",
        "taskCode": "task-1664-988"
    }
  ]
}

```

失败响应示例二

```

{
  "code": 1010100001,
  "msg": "error",
  "data": {
    "tasks": [
      {
        "errorCode": "1030600015",
        "message": "NO_SPECIFIED_CONTAINER",
        "taskCode": "task-1665-177"
      }
    ]
  }
}

```

2.1.1.6 称重盘点

称重盘点任务，任务类型为 weight。

请求参数说明

表 2-21 请求参数说明

参数名	必选	类型	说明
taskType	Y	string	任务类型，取值为 weight。
taskGroupCode	N	string	任务组号，若不传值则由系统自动生成。
groupPriority	N	int	任务组优先级，取值范围为【0~2147483647】，2147483647 为 int 类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级组的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
tasks	Y	list<task>	任务列表。详情请参见“tasks 参数说明”。

表 2-22 tasks 参数说明

参数名	必选	类型	说明
taskCode	Y	string	业务任务号。
taskPriority	N	int	任务优先级，取值范围为【0~2147483647】，2147483647为int类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
taskDescribe	Y	object<taskDescribe>	任务描述，描述任务所需要的参数。详情请参见“taskDescribe 参数说明”。

表 2-23 taskDescribe 参数说明

参数名	必选	类型	说明
containerCode	N	string	<ul style="list-style-type: none"> ● containerCode: 容器编码。
locationCode	C	string	<ul style="list-style-type: none"> ● locationCode: 目标工位，即要进行盘库的目标库位。 ● 灰盒场景下，容器编码和目标工位二选一必填 ● 白盒场景下，目标工位必填

请求示例

```
{
  "taskType": "weight",
  "taskGroupCode": "taskGroupCode-001",
  "groupPriority": 0,
  "tasks": [
    {
      "taskCode": "taskCode-001",
      "taskDescribe": {
        "containerCode": "Container001",
        "locationCode": "SUY-002-012-01"
      }
    }
  ]
}
```

响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 1: 表示未完全成功, 即存在部分任务/容器/工作位响应成功, 部分响应失败。 ● 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时, msg值为success。 ● 当code为1时, msg值为partial response failure。 ● 当code为其他值时, msg值为error。
data	Y	object	返回任务处理详情, 详情请参见“tasks 参数说明”。

表 2-24 tasks 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示处理成功。 ● 其他值: 表示处理失败, 详见异常码。
message	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当errorCode为0时, message值为OK。 ● 当errorCode为其他值时, message值为错误码对应的错误描述。
taskcode	Y	string	任务编码。

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "tasks": [
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-26-572"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-261-501"
      }
    ]
  }
}
```

```
    ]  
  }  
}
```

失败响应示例一

```
{  
  "code": 1010100001,  
  "msg": "error",  
  "data": {  
    "tasks": [  
      {  
        "errorCode": "1030600022",  
        "message": "toLocationCode[L08SR-007-002-15] is not exist",  
        "taskCode": "task-16601-683"  
      }  
    ]  
  }  
}
```

失败响应示例二

```
{  
  "code": 1030300007,  
  "msg": "error",  
  "data": null  
}
```

失败响应示例三

```
{  
  "code": 1,  
  "msg": "partial response failure",  
  "data": {  
    "tasks": [  
      {  
        "errorCode": "1030600022",  
        "message": "toLocationCode[L08SR-007-002-14] is not exist",  
        "taskCode": "task-16651-105"  
      },  
      {  
        "errorCode": "0",  
        "message": "OK",  
        "taskCode": "task-1665-887"  
      }  
    ]  
  }  
}
```

2.1.1.7 RFID 盘点

RFID 盘点任务，任务类型为 rfid。

请求参数说明

表 2-25 请求参数说明

参数名	必选	类型	说明
taskType	Y	string	任务类型，取值为 rfid。
taskGroupCode	N	string	任务组号，若不传值则由系统自动生成。
groupPriority	N	int	任务组优先级，取值范围为【0~2147483647】，2147483647 为 int 类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级组的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
tasks	Y	list<task>	任务列表。详情请参见“tasks 参数说明”。

表 2-26 tasks 参数说明

参数名	必选	类型	说明
taskCode	Y	string	业务任务号。
taskPriority	N	int	任务优先级，取值范围为【0~2147483647】，2147483647 为 int 类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
taskDescribe	Y	object<taskDescribe>	任务描述，描述任务所需要的参数。详情请参见“taskDescribe 参数说明”。

表 2-27 taskDescribe 参数说明

参数名	必选	类型	说明
containerCode	N	string	容器编码。
locationCode	Y	string	目标工位，即要进行盘库的目标库位。

请求示例

```

{
  "taskType": "rfid",
  "taskGroupCode": "taskGroupCode-001",
  "groupPriority": 0,
  "tasks": [
    {
      "taskCode": "taskCode-001",
      "taskDescribe": {
        "containerCode": "Container001",
        "locationCode": "SUY-002-012-01"
      }
    }
  ]
}

```

响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 1: 表示未完全成功, 即存在部分任务/容器/工作位响应成功, 部分响应失败。 ● 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时, msg值为success。 ● 当code为1时, msg值为partial response failure。 ● 当code为其他值时, msg值为error。
data	Y	object	返回任务处理详情, 详情请参见“tasks 参数说明”。

表 2-28 tasks 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示处理成功。 ● 其他值: 表示处理失败, 详见异常码。
message	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当errorCode为0时, message值为OK。 ● 当errorCode为其他值时, message值为错误码对应的错误描述。
taskcode	Y	string	任务编码。

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "tasks": [
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-26-572"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-261-501"
      }
    ]
  }
}
```

失败响应示例一

```
{
  "code": 1010100001,
  "msg": "error",
  "data": {
    "tasks": [
      {
        "errorCode": "1030600022",
        "message": "toLocationCode[L08SR-007-002-15] is not exist",
        "taskCode": "task-16601-683"
      }
    ]
  }
}
```

失败响应示例二

```
{
  "code": 1030300007,
  "msg": "error",
  "data": null
}
```

失败响应示例三

```

{
  "code": 1,
  "msg": "partial response failure",
  "data": {
    "tasks": [
      {
        "errorCode": "1030600022",
        "message": "toLocationCode[L08SR-007-002-14] is not exist",
        "taskCode": "task-16651-105"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-1665-887"
      }
    ]
  }
}

```

2.1.1.8 模板任务

模板任务，任务类型为 template。

请求参数说明

表 2-29 请求参数说明

参数名	必选	类型	说明
taskType	Y	string	任务类型，取值为 template：用于下发模板任务。
taskGroupCode	N	string	任务组号，若不传值则由系统自动生成。
groupPriority	N	int	任务组优先级，取值范围为【0~2147483647】，2147483647 为 int 类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级组的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
tasks	Y	list<task>	任务列表。详情请参见“tasks 参数说明”。

表 2-30 tasks 参数说明

参数名	必选	类型	说明
taskCode	Y	string	业务任务号。
taskPriority	N	int	任务优先级，取值范围为【0~2147483647】，

参数名	必选	类型	说明
			2147483647 为 int 类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级的任务。 ● 取值超过int类型最大值系统会报错，任务下发失败。
taskDescribe	Y	object<taskDescribe>	任务描述，描述任务所需要的参数。详情请参见“taskDescribe 参数说明”。

表 2-31 taskDescribe 参数说明

参数名	必选	类型	说明
flowTaskTemplateCode	Y	string	模板编码。
actions	N	list<action>	指令，详情请参见“actions 参数说明”。

表 2-32 actions 参数说明

参数名	必选	类型	说明
containerFace	N	string	目标容器朝向，取值为枚举值 A、B、C、D。常用于出库 HaiFlex 需要翻面的场景。 <ul style="list-style-type: none"> ● A: 0 ● B: 90 ● C: 180 ● D: 270
containerCode	N	string	容器编码。
toLocationCode	N	string	目标工位，即机器人要进行操作的目标位置。
waitTimeMs	N	long	等待时间。
containerType	N	string	容器类型，默认值为“CT_KUBOT_STANDARD”。 <ul style="list-style-type: none"> ● 填写容器类型时： <ul style="list-style-type: none"> ■ 容器编码在系统中存在，原容器类型和现容器类型一致时，容器类型不变。 ■ 容器编码在系统中存在，原容器类型和现容器类型不一致时，更新为现容器类型。 ■ 容器编码在系统中不存在，使用该容器类型创建容器。 ● 若不填容器类型：

参数名	必选	类型	说明
			<ul style="list-style-type: none"> 容器编码在系统中存在，使用原容器类型。 容器编码在系统中不存在，使用系统默认的容器类型“CT_KUBOT_STANDARD”创建容器。

请求示例

```
{
  "taskType": "template",
  "taskGroupCode": "taskGroupCode011011",
  "groupPriority": 0,
  "tasks": [
    {
      "taskCode": "taskCode16x5a281201",
      "taskDescribe": {
        "flowTaskTemplateCode": "Template-1",
        "actions": [
          {
            "containerCode": "test_010",
            "toLocationCode": ""
          },
          {
            "containerCode": "test_011",
            "toLocationCode": ""
          },
          {
            "toLocationCode": "location-11",
            "waitTimeMs": 10000
          }
        ]
      }
    }
  ]
}
```

响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> 0: 表示接口响应成功。 其他值: 表示响应失败，详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> 当code为0时，msg值为success。

字段	必选	定义	备注
			<ul style="list-style-type: none">当code为其他值时，msg值为ERROR。
data	Y	object	返回每条任务的处理结果。 <ul style="list-style-type: none">处理成功：OK处理失败：详细的错误描述。

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "tasks": [
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "1718862117"
      }
    ]
  }
}
```

失败响应示例一

```
{
  "code": 1010100001,
  "msg": "error",
  "data": {
    "tasks": [
      {
        "errorCode": "1150100001",
        "message": "flowTaskTemplateTaskPositionNum is not match",
        "taskCode": "556666"
      },
      {
        "errorCode": "1150100001",
        "message": "flowTaskTemplateTaskPositionNum is not match",
        "taskCode": "1718862255"
      }
    ]
  }
}
```

失败响应示例二

```

{
  "code": 1,
  "msg": "partial response failure",
  "data": {
    "tasks": [
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "1718862380"
      },
      {
        "errorCode": "1150100001",
        "message": "flowTaskTemplateTaskPositionNum is not match",
        "taskCode": "5566665"
      }
    ]
  }
}

```

2.1.2 任务取消请求

2.1.2.1 接口描述

接口 URL	http://{IP:PORT}/task/cancel
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	上游系统通过此接口，将已下发待处理的任务取消。 为达到最佳性能，建议单次请求不超过 100 行明细，请求频率不超过 140 次/秒。
备注	<ul style="list-style-type: none"> ● 该接口单次调用最多可取消200条任务。 ● 无法取消完成、取消和失败状态的任务。 ● 对于入库任务，当任务状态为待办或执行中时： <ol style="list-style-type: none"> 1) 机器人取箱前，可取消成功； 2) 机器人在取箱中或取箱后，则支持配置是否允许取消业务任务，默认不允许。 ● 对于模板任务，当任务状态为待办和执行中，支持取消待办和执行中的任务，任务取消后，若机器人身上有容器： <ol style="list-style-type: none"> 1) 白盒：放回原来的位置，当原来的位置被占用时，机器人去维修区等待（有维修区时）或在原地等待人工处理（无维修区时）。 2) 灰盒：放回原来的位置，当原来的位置被占用时，就近放回空闲库位。 ● 对于除入库任务和模板任务之外的其他任务，当任务状态为待办或执行中时： <ol style="list-style-type: none"> 1) 可直接取消待办状态的任务。

2) 对于状态为执行中的任务，若机器人正在执行取放动作则取消失败；若机器人未执行取放箱动作，则取消成功。

2.1.2.2 请求参数说明

参数名	必选	类型	说明
taskCodes	Y	list<string>	任务编码。业务任务和系统任务都可取消。

2.1.2.3 请求示例

```
{
  "taskCodes": [
    "taskCode-001"
  ]
}
```

2.1.2.4 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> 0: 表示接口响应成功。 1: 表示未完全成功, 即存在部分任务/容器/工作位响应成功, 部分响应失败。 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> 当code为0时, msg值为success。 当code为1时, msg值为partial response failure。 当code为其他值时, msg值为error。
data	Y	object	返回任务处理详情, 详情请参见“tasks 参数说明”。

表 2-33 tasks 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。 <ul style="list-style-type: none"> 0: 表示处理成功。 其他值: 表示处理失败, 详见异常码。
message	Y	string	响应消息补充说明。

字段	必选	定义	备注
			<ul style="list-style-type: none">当errorCode为0时，message值为OK。当errorCode为其他值时，message值为错误码对应的错误描述。
taskcode	Y	string	任务编码。

2.1.2.5 响应示例

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "tasks": [
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-26-572"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-261-501"
      }
    ]
  }
}
```

失败响应示例一

```
{
  "code": 1010100001,
  "msg": "error",
  "data": {
    "tasks": [
      {
        "errorCode": "1030600044",
        "message": "task cancel fail",
        "taskCode": "task-2665554996-609"
      }
    ]
  }
}
```

失败响应示例二

```

{
  "code": 1,
  "msg": "partial response failure",
  "data": {
    "tasks": [
      {
        "errorCode": "1030600044",
        "message": "task cancel fail",
        "taskCode": "task-1660-882"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "taskCode": "task-1660-880"
      },
      {
        "errorCode": "1030600044",
        "message": "task cancel fail",
        "taskCode": "task-1660-881"
      }
    ]
  }
}

```

2.2 上游系统提供的接口

2.2.1 任务状态回调

2.2.1.1 接口描述

接口 URL	上游系统提供
接口形式	HTTP POST+JSON
调用方向	ESS→上游系统
功能描述	任务状态发生变化时，ESS 调用此接口将任务状态变更信息回传给上游系统。
备注	<ul style="list-style-type: none"> ● 任务状态回调有以下8种： <ol style="list-style-type: none"> 1. 任务完成时回调 2. 任务挂起时回调 3. 任务取消时回调 4. 任务失败时回调 5. 任务分配时回调 6. 扫描任务完成时回调

	<ul style="list-style-type: none"> 7. 称重任务完成时回调 8. rfid 任务完成时回调 ● 机器人取放箱回调有以下4种： <ul style="list-style-type: none"> 1. 机器人取箱完成时回调 2. 机器人取箱失败时回调 3. 机器人放箱完成时回调 4. 机器人放箱失败时回调 ● 机器人到达工作站时回调
--	--

2.2.1.2 回调参数说明

表 2-34 回调参数说明

参数名	必选	类型	说明
callId	Y	string	一次回调唯一标识。
taskCode	C	string	业务任务号。 ● 当eventType为robot_reach时不填。 ● 当eventType为其他值时必须填。
eventType	Y	string	上报事件类型： ● task: 上报任务状态发生变化的事件。 ● task_allocated: 上报任务分配给机器人。 ● tote_load: 上报取箱状态。 ● tote_unload: 上报放箱状态。 ● robot_reach: 机器人到达工作站。
status	Y	string	任务状态： ● success: 成功。 ● fail: 失败。 ● cancel: 取消。 ● suspend: 挂起。
robotCode	N	string	机器人编码。
containerCode	N	string	容器编码。
locationCode	N	string	工作位编码。
stationCode	N	string	工作站编码。
message	N	string	描述信息，描述异常原因/挂起原因等。
sysTaskCode	N	string	系统任务编码，用于任务挂起回调。可通过系统任务编码

参数名	必选	类型	说明
			恢复挂起的系统任务。业务任务没有挂起状态。
isLocationHasContainer	C	boolean	库位是否有容器，当 eventType 为 task 且为盘点任务时才会返回该值。
trayLevel	C	int	背篓层号，从 0 层开始，从下往上编号。64 表示放在了货叉上。当 eventType 为 task 且为称重盘点任务或 rfid 盘点任务时才会返回该值。
weight	C	int	重量，单位为 g。当 eventType 为 task 且为称重盘点任务才会返回该值。
rfidInfo	C	list< string >	rfid 盘点信息。当 eventType 为 task 且为 rfid 盘点任务时才会返回该值。
robotTypeCode	C	string	机器人类型编码。当 eventType 为 robot_reach 时才会返回该值。
trays	C	List<TrayItem>	背篓信息。当 eventType 为 robot_reach 时才会返回该值，详见“trays 参数说明”。

表 2-35 trays 参数说明

参数名	必选	类型	说明
containerCode	N	string	容器编码。
trayLevel	Y	int	背篓层号，从 0 层开始，从下往上编号，64 表示容器放在了货叉上。
positionCode	Y	string	位置编码，由机器人编码和背篓层数组成，即机器人 id# 背篓层数。
containerFace	N	string	目标容器朝向，取值为枚举值 A、B、C、D。常用于出库 HAIFLEX 需要翻面的场景。 <ul style="list-style-type: none"> ● A: 0 ● B: 90 ● C: 180 ● D: 270

2.2.1.3 回调示例

任务状态回调

1. 任务完成

```
{
```

```
"callId": "1785343693261111296",
"taskCode": "Outside01652361018",
"eventType": "task",
"status": "success",
"containerCode": "bin0009",
"locationCode": "LT_CONVEYOR_INPUT:POINT:29940:8710",
"robotCode": "kubot-7",
"stationCode": "2_01"
}
```

2. 任务挂起

```
{
"callId": "1785343693261111296",
"taskCode": "Outside01652363222",
"eventType": "task",
"status": "suspend",
"containerCode": "bin0006",
"locationCode": "LT_CONVEYOR_INPUT:POINT:29940:8710",
"robotCode": "kubot-5",
"stationCode": "2_01",
"sysTaskCode": "return:task-1749167617665406208",
"message": "box tag is not detected at the specified location!"
}
```

3. 任务取消

```
{
"callId": "1785343693261111296",
"taskCode": "task-1653381132",
"eventType": "task",
"status": "cancel",
"containerCode": "A0000001",
"locationCode": null,
"robotCode": null,
"stationCode": null
}
```

4. 任务失败

```
{
"callId": "1785343693261111296",
"taskCode": "task-1653381132",
"eventType": "task",
"status": "fail",
"containerCode": "bin0009",
"locationCode": "CH08-25-03",
"robotCode": "kubot-7",
"sysTaskCode": "return:task-1749167617665406208",
}
```

```
"message": "NO_AVAILABLE_ROBOT"
}
```

5. 任务分配

```
{
  "callId": "1785343693261111296",
  "taskCode": "task-1653381132",
  "eventType": "task_allocated",
  "status": "success",
  "containerCode": "A0000001",
  "locationCode": "4-05-10",
  "robotCode": "kubt-01"
}
```

6. 扫描任务完成

```
{
  "callId": "1785343693261111296",
  "taskCode": "task001",
  "eventType": "task",
  "status": "success",
  "containerCode": "bin0009",
  "locationCode": "CH08-25-03",
  "robotCode": "kubot-7",
  "stationCode": "2_01",
  "isLocationHasContainer": true
}
```

7. 称重任务完成

```
{
  "callId": "1785343693261111296",
  "taskCode": "task001",
  "eventType": "task",
  "status": "success",
  "containerCode": "bin0009",
  "locationCode": "CH08-25-03",
  "robotCode": "kubot-7",
  "stationCode": "2_01",
  "weight": 500,
  "trayLevel": 64
}
```

8. rfid 任务回调

```
{
  "callId": "1785343693261111296",
  "taskCode": "task001",
}
```

```
"eventType": "task",
"status": "success",
"containerCode": "bin0009",
"locationCode": "CH08-25-03",
"robotCode": "kubot-7",
"stationCode": "2_01",
"rfidInfo": [
  "663164",
  "303169"
],
"trayLevel": 64
}
```

机器人取放箱回调

1. 取箱完成

```
{
  "callId": "1785343693261111296",
  "taskCode": "Outside01652361018",
  "eventType": "tote_load",
  "status": "success",
  "containerCode": "bin0009",
  "locationCode": "HAI-005-003-01",
  "robotCode": "kubot-7",
  "stationCode": "LA_SHELF_STORAGE"
}
```

2. 取箱失败

```
{
  "callId": "1785343693261111296",
  "taskCode": "Outside01652361018",
  "eventType": "tote_load",
  "status": "fail",
  "containerCode": "bin0009",
  "locationCode": "HAI-005-003-01",
  "robotCode": "kubot-7",
  "stationCode": "LA_SHELF_STORAGE"
}
```

3. 放箱完成

```
{
  "callId": "1785343693261111296",
  "taskCode": "Outside01652361018",
  "eventType": "tote_unload",
  "status": "success",
  "containerCode": "bin0009",
}
```

```

"locationCode": "LT_CONVEYOR_INPUT:POINT:29940:8710",
"robotCode": "kubot-7",
"stationCode": "2_01"
}

```

4. 放箱失败

```

{
  "callId": "1785343693261111296",
  "taskCode": "Outside01652361018",
  "eventType": "tote_unload",
  "status": "fail",
  "containerCode": "bin0009",
  "locationCode": "LT_CONVEYOR_INPUT:POINT:29940:8710",
  "robotCode": "kubot-7",
  "stationCode": "2_01"
}

```

机器人到达工作站

```

{
  "callId": "1785343693261111296",
  "taskCode": null,
  "eventType": "robot_reach",
  "status": "success",
  "containerCode": null,
  "locationCode": "LT_LABOR:POINT:11660:39850",
  "robotCode": "kubot-7",
  "stationCode": "labor01",
  "trays": [
    {
      "containerCode": "G0980",
      "trayLevel": 64,
      "positionCode": "kubot-7#64",
      "containerFace": "C"
    }
  ]
}

```

2.2.1.4 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0：正常。 ● 非0：异常。

字段	必选	定义	备注
msg	Y	string	响应消息补充说明, success 或其他异常消息。
data	Y	json 对象	返回的数据。

2.2.1.5 响应示例

```
{  
  "code": 0,  
  "msg": "success",  
  "data": {}  
}
```

第3章 可选接口

3.1 ESS 提供的接口

3.1.1 任务优先级调整

3.1.1.1 接口描述

接口 URL	http://{IP:PORT}/task/updateTaskPriority
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	<p>将 ESS 业务任务队列中待办状态的业务任务优先级提高。业务任务优先级默认为 0，数值越高，优先级越高。</p> <p>为达到最佳性能，建议单次请求不超过 100 行明细，请求频率不超过 400 次/秒。</p>
备注	<ul style="list-style-type: none"> ● 支持业务任务组优先级调整。 ● 支持业务任务优先级调整。 ● 该接口单次调用最多可调整200条任务的优先级。

3.1.1.2 请求参数说明

表 3-1 请求参数说明

参数名	必选	类型	说明
updateTaskPriority	Y	list	需要调整任务优先级的任务列表。详情请参见“updateTaskPriority 参数说明”。

表 3-2 updateTaskPriority 参数说明

参数名	必选	类型	说明
type	Y	string	调整类型： <ul style="list-style-type: none"> ● UPDATE_GROUP_PRIORITY=调整组优先级。 ● UPDATE_TASK_PRIORITY=调整任务优先级。
code	Y	string	任务编号。
priority	Y	int	调整后的优先级，由上游系统指定，取值范围为【0~2147483647】，2147483647 为 int 类型的最大值。 <ul style="list-style-type: none"> ● 默认值为0，代表不区分优先级。 ● 从1开始，数值越大，优先级越高，系统会优先执行高优先级的任务。

参数名	必选	类型	说明
			<ul style="list-style-type: none"> 取值超过int类型最大值系统会报错，任务优先级调整失败。

3.1.1.3 请求示例

```
{
  "updateTaskPriority": [
    {
      "type": "UPDATE_TASK_PRIORITY",
      "code": "taskCode001",
      "priority": 5
    }
  ]
}
```

3.1.1.4 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> 0: 表示接口响应成功。 1: 表示未完全成功,即存在部分任务/容器/工位响应成功,部分响应失败。 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> 当code为0时, msg值为success。 当code为1时, msg值为partial response failure。 当code为其他值时, msg值为error。
data	Y	object	返回任务处理详情, 详情请参见“codes 参数说明”。

表 3-3 codes 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。 <ul style="list-style-type: none"> 0: 表示处理成功。 其他值: 表示处理失败, 详见异常码。
message	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> 当errorCode为0时, message值为OK。

字段	必选	定义	备注
			<ul style="list-style-type: none">当errorCode为其他值时，message值为错误码对应的错误描述。
code	Y	string	任务编码或任务组编码。

3.1.1.5 响应示例

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "codes": [
      {
        "errorCode": "0",
        "message": "OK",
        "code": "task-16651-473"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "code": "task-1665-617"
      }
    ]
  }
}
```

失败响应示例一

```
{
  "code": 1010100001,
  "msg": "error",
  "data": {
    "codes": [
      {
        "errorCode": "1030600205",
        "message": "updateTaskPriority fail",
        "code": "task-1665-859"
      }
    ]
  }
}
```

失败响应示例二

```
{
  "code": 1,
  "msg": "partial response failure",
  "data": {
    "codes": [
      {
        "errorCode": "1030600205",
        "message": "updateTaskPriority fail",
        "code": "task-16651-470"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "code": "task-1665-617"
      }
    ]
  }
}
```

3.1.2 容器创建

3.1.2.1 接口描述

接口 URL	http://{IP:PORT}/container/create
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	在 ESS 系统中新增或更新容器基础信息。 为达到最佳性能，建议单次请求不超过 100 行明细，请求频率不超过 400 次/秒。
备注	<ul style="list-style-type: none"> ● 容器创建成功后则被ESS认为是合法的容器。 ● 支持根据入库容器类型分配对应能取该型号的机器人执行取箱任务。 ● 灰盒场景下可根据容器标签分配对应逻辑区的库位。 ● 该接口单次调用最多可创建200个容器。

3.1.2.2 请求参数说明

表 3-4 请求参数说明

参数名	必选	类型	说明
containerAdds	Y	list<containerAdd>	容器创建操作。详情请参见“containerAdds 参数说明”。

表 3-5 containerAdds 参数说明

参数名	必选	类型	说明
containerCode	Y	string	容器编码。 <ul style="list-style-type: none"> ESS系统中不存在的编码，创建此容器； ESS系统中存在的编码，更新此容器的容器类型或标签。
containerTypeCode	C	string	ESS 系统中维护的容器类型编码。
storageTag	N	string	ESS 系统中维护的工作位标签，通常也被称作逻辑区名称。

3.1.2.3 请求示例

```
{
  "containerAdds": [
    {
      "containerCode": "container001",
      "containerTypeCode": "CT_KUBOT_STANDARD"
    }
  ]
}
```

3.1.2.4 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> 0: 表示接口响应成功。 1: 表示未完全成功,即存在部分任务/容器/工作位响应成功,部分响应失败。 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> 当code为0时, msg值为success。 当code为1时, msg值为partial response failure。 当code为其他值时, msg值为error。
data	Y	object	返回任务处理详情, 详情请参见“containerCodes 参数说明”。

表 3-6 containerCodes 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。 ● 0: 表示处理成功。 ● 其他值: 表示处理失败, 详见 异常码 。
message	Y	string	响应消息补充说明。 ● 当errorCode为0时, message值为OK。 ● 当errorCode为其他值时, message值为错误码对应的错误描述。
containerCode	Y	string	容器编码。

3.1.2.5 响应示例

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "containerCodes": [
      {
        "errorCode": "0",
        "message": "OK",
        "containerCode": "A0000011"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "containerCode": "A0000012"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "containerCode": "A0000013"
      }
    ]
  }
}
```

失败响应示例一

```
{
  "code": 1010100001,
```

```
"msg": "error",
"data": {
  "containerCodes": [
    {
      "errorCode": "1030300004",
      "message": "NOT_CONTAINER",
      "containerCode": "HAI-038-035-06"
    },
    {
      "errorCode": "1030300005",
      "message": "NOT_NORMAL_CONTAINER",
      "containerCode": "PT0009"
    }
  ]
}
```

失败响应示例二

```
{
  "code": 1,
  "msg": "partial response failure",
  "data": {
    "containerCodes": [
      {
        "errorCode": "1030300004",
        "message": "NOT_CONTAINER",
        "containerCode": "HAI-038-035-06"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "containerCode": "container008"
      },
      {
        "errorCode": "1030300005",
        "message": "NOT_NORMAL_CONTAINER",
        "containerCode": "PT0009"
      }
    ]
  }
}
```

3.1.3 容器移除

3.1.3.1 接口描述

接口 URL	http://{IP:PORT}/container/delete
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	清除记录在 ESS 系统中容器的基础信息。 为达到最佳性能，建议单次请求不超过 100 行明细，请求频率不超过 400 次/秒。
备注	在校验容器编码合法性时被系统判定为不合法的容器编码（即无法执行取箱动作的容器），可以使用该接口将其清除。 限制条件： <ul style="list-style-type: none"> ● 容器需为离场状态。 ● 容器没有关联的任务（包含系统未清除的已完成、失败、取消的任务）。 ● 该接口单次调用最多可移除200个容器。

3.1.3.2 请求参数说明

参数名	必选	类型	说明
containerCodes	Y	list<string>	容器编码。

3.1.3.3 请求示例

```
{
  "containerCodes": [
    "container001"
  ]
}
```

3.1.3.4 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 1: 表示未完全成功, 即存在部分任务/容器/工作位响应成功, 部分响应失败。 ● 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。

字段	必选	定义	备注
			<ul style="list-style-type: none"> 当code为0时，msg值为success。 当code为1时，msg值为partial response failure。 当code为其他值时，msg值为error。
data	Y	object	返回任务处理详情，详情请参见“containerCodes 参数说明”。

表 3-7 containerCodes 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。 <ul style="list-style-type: none"> 0：表示处理成功。 其他值：表示处理失败，详见异常码。
message	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> 当errorCode为0时，message值为OK。 当errorCode为其他值时，message值为错误码对应的错误描述。
containerCode	Y	string	容器编码。

3.1.3.5 响应示例

成功响应示例

```

{
  "code": 0,
  "msg": "success",
  "data": {
    "containerCodes": [
      {
        "errorCode": "0",
        "message": "OK",
        "containerCode": "A0000010"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "containerCode": "A0000011"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "containerCode": "AGEEF00000111"
      }
    ]
  }
}
    
```

```
}
```

失败响应示例一

```
{
  "code": 1010100001,
  "msg": "error",
  "data": {
    "containerCodes": [
      {
        "errorCode": "1030300004",
        "message": "NOT_CONTAINER",
        "containerCode": "HAI-038-035-06"
      },
      {
        "errorCode": "1030300005",
        "message": "NOT_NORMAL_CONTAINER",
        "containerCode": "PT0009"
      }
    ]
  }
}
```

失败响应示例二

```
{
  "code": 1,
  "msg": "partial response failure",
  "data": {
    "containerCodes": [
      {
        "errorCode": "1030300004",
        "message": "NOT_CONTAINER",
        "containerCode": "HAI-038-035-06"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "containerCode": "container008"
      },
      {
        "errorCode": "1030300005",
        "message": "NOT_NORMAL_CONTAINER",
        "containerCode": "PT0009"
      }
    ]
  }
}
```

3.1.4 容器入场

3.1.4.1 接口描述

接口 URL	http://{IP:PORT}/container/movelIn
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	将容器位置更新至系统工作位上。 为达到最佳性能，建议单次请求不超过 100 行明细，请求频率不超过 140 次/秒。
备注	<ul style="list-style-type: none"> ● 支持将容器更新至不同类型的工作位上。 ● 在缓存货架工作位入场一个容器，系统会自动呼叫机器人过来。 ● 在输送线入库工作位入场一个容器，系统会自动呼叫机器人过来。 ● 该接口单次调用最多可使200个容器入场。

3.1.4.2 请求参数说明

表 3-8 请求参数说明

参数名	必选	类型	说明
containerMoveIns	Y	list<containerMoveIn>	容器入场。详情请参见“containerMoveIns 参数说明”。

表 3-9 containerMoveIns 参数说明

参数名	必选	类型	说明
containerCode	N	string	容器编码，若不填写，则系统认为是一个 UNKNOWN 状态的容器占用了该位置，在客户使用非扫描容器进行上架时使用。 <ul style="list-style-type: none"> ● 人工工作站操作机器人上的容器：必填。 ● 缓存货架工作站操作容器：非必填。 ● 输送线工作站：非必填。
positionCode	Y	string	位置编码，可以是库位工作位编码、工作站工作位编码、机器人背篓（机器人 id#背篓序号）。

3.1.4.3 请求示例

```
{
  "containerMoveIns": [
    {
      "containerCode": "container001",
```

```

        "positionCode": "kubot-1#1"
    }
]
}
    
```

3.1.4.4 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 1: 表示未完全成功, 即存在部分任务/容器/工位响应成功, 部分响应失败。 ● 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时, msg值为success。 ● 当code为1时, msg值为partial response failure。 ● 当code为其他值时, msg值为error。
data	Y	object	返回任务处理详情, 详情请参见“positionCodes 参数说明”。

表 3-10 positionCodes 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示处理成功。 ● 其他值: 表示处理失败, 详见异常码。
message	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当errorCode为0时, message值为OK。 ● 当errorCode为其他值时, message值为错误码对应的错误描述。
positionCode	Y	string	工位编码。

3.1.4.5 响应示例

成功响应示例

```

{
  "code": 0,
  "msg": "success",
  "data": {
    "positionCodes": [
    
```

```
{
  {
    "errorCode": "0",
    "message": "OK",
    "positionCode": "L08SR-004-001-08"
  },
  {
    "errorCode": "0",
    "message": "OK",
    "positionCode": "L08SR-004-001-09"
  },
  {
    "errorCode": "0",
    "message": "OK",
    "positionCode": "L08SR-004-001-10"
  }
}
]
```

失败响应示例一

```
{
  "code": 1010100001,
  "msg": "error",
  "data": {
    "positionCodes": [
      {
        "errorCode": "1030800002",
        "message": "position unknown",
        "positionCode": "L08SR-002-004-021"
      }
    ]
  }
}
```

失败响应示例二

```
{
  "code": 1,
  "msg": "partial response failure",
  "data": {
    "positionCodes": [
      {
        "errorCode": "0",
        "message": "OK",
        "positionCode": "L08SR-004-001-08"
      }
    ],
  }
}
```

```

{
  {
    "errorCode": "0",
    "message": "OK",
    "positionCode": "L08SR-004-001-09"
  },
  {
    "errorCode": "1030600032",
    "message": "CONTAINER_NOT_EXIST",
    "positionCode": "L08SR-004-001-10"
  }
}
]
}
}

```

3.1.5 容器离场

3.1.5.1 接口描述

接口 URL	http://{IP:PORT}/container/moveOut
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	<p>将容器从 ESS 系统中删除位置信息，置为在库外的状态，离场后，ESS 将不能再操作该容器。</p> <p>为达到最佳性能，建议单次请求不超过 100 行明细，请求频率不超过 140 次/秒。</p>
备注	<ul style="list-style-type: none"> ● 确认为工作位异常后需要移除容器。 ● 容器自身原因出现异常时，需要人工将容器取出。 ● 该接口单次调用最多可使200个容器离场。

3.1.5.2 请求参数说明

表 3-11 请求参数说明

参数名	必选	类型	说明
containerMoveOuts	Y	list<containerMoveOuts>	容器离场。详情请参见“containerMoveouts 参数说明”。

表 3-12 containerMoveouts 参数说明

参数名	必选	类型	说明
containerCode	N	string	<ul style="list-style-type: none"> ● containerCode: 容器编码。 ● positionCode: 位置编码, 可以是库位工作位
positionCode	N	string	

参数名	必选	类型	说明
			编码、工作站工位编码、机器人背篓（机器人id#背篓序号）。 <ul style="list-style-type: none"> ● 容器编码和位置编码二选一必填。

3.1.5.3 请求示例

```
{
  "containerMoveOuts": [
    {
      "containerCode": "Container001",
      "positionCode": "kubot-1#1"
    }
  ]
}
```

3.1.5.4 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 1: 表示未完全成功, 即存在部分任务/容器/工位响应成功, 部分响应失败。 ● 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时, msg值为success。 ● 当code为1时, msg值为partial response failure。 ● 当code为其他值时, msg值为error。
data	Y	object	返回任务处理详情, 详情请参见“codes 参数说明”。

表 3-13 codes 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示处理成功。 ● 其他值: 表示处理失败, 详见异常码。
message	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当errorCode为0时, message值为OK。

字段	必选	定义	备注
			<ul style="list-style-type: none"> 当errorCode为其他值时，message值为错误码对应的错误描述。
code	Y	string	工作位编码或容器编码。

3.1.5.5 响应示例

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "codes": [
      {
        "errorCode": "0",
        "message": "OK",
        "code": "L08SR-004-001-08"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "code": "L08SR-004-001-09"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "code": "L08SR-004-001-10"
      }
    ]
  }
}
```

失败响应示例一

```
{
  "code": 1010100001,
  "msg": "error",
  "data": {
    "codes": [
      {
        "errorCode": "1030600032",
        "message": "CONTAINER_NOT_EXIST",
        "code": "HAI-001-002-01"
      }
    ]
  }
}
```

```
}
}
```

失败响应示例二

```
{
  "code": 1,
  "msg": "partial response failure",
  "data": {
    "codes": [
      {
        "errorCode": "0",
        "message": "OK",
        "code": "L08SR-004-001-08"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "code": "L08SR-004-001-09"
      },
      {
        "errorCode": "1030600032",
        "message": "CONTAINER_NOT_EXIST",
        "code": "L08SR-004-001-10"
      }
    ]
  }
}
```

3.1.6 工作位锁定

3.1.6.1 接口描述

接口 URL	http://{IP:PORT}/location/lock
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	当某些工作位暂时不需要使用时，调用此接口将工作位锁定，锁定后机器人只能在该工作位取容器，不能放容器到该工作位。 为达到最佳性能，建议单次请求不超过 100 行明细，请求频率不超过 140 次/秒。
备注	该接口单次调用最多可锁定 200 个工作位。

3.1.6.2 请求参数说明

参数名	必选	类型	说明
locationCodes	Y	list<string>	工作位编码，需为未锁定的工作位。

3.1.6.3 请求示例

```
{
  "locationCodes": [
    "L010100017.01"
  ]
}
```

3.1.6.4 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 1: 表示未完全成功，即存在部分任务/容器/工作位响应成功，部分响应失败。 ● 其他值: 表示响应失败，详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时，msg值为success。 ● 当code为1时，msg值为partial response failure。 ● 当code为其他值时，msg值为error。
data	Y	object	返回任务处理详情，详情请参见“locationCodes 参数说明”。

表 3-14 locationCodes 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示处理成功。 ● 其他值: 表示处理失败，详见异常码。
message	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当errorCode为0时，message值为OK。 ● 当errorCode为其他值时，message值为错误码对应的错误描述。
locationCode	Y	string	工作位编码。

3.1.6.5 响应示例

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "locationCodes": [
      {
        "errorCode": "0",
        "message": "OK",
        "locationCode": "L08SR-004-003-08"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "locationCode": "L08SR-004-003-09"
      },
      {
        "errorCode": "0",
        "message": "OK",
        "locationCode": "L08SR-004-003-07"
      }
    ]
  }
}
```

失败响应示例一

```
{
  "code": 1010100001,
  "msg": "error",
  "data": {
    "locationCodes": [
      {
        "errorCode": "1030200007",
        "message": "LOCATION_NOT_EXIST",
        "locationCode": "L08SR-004-004-011"
      }
    ]
  }
}
```

失败响应示例二

```
{
  "code": 1,
```

```

"msg": "partial response failure",
"data": {
  "locationCodes": [
    {
      "errorCode": "0",
      "message": "OK",
      "locationCode": "L08SR-004-003-08"
    },
    {
      "errorCode": "0",
      "message": "OK",
      "locationCode": "L08SR-004-003-09"
    },
    {
      "errorCode": "1030200007",
      "message": "LOCATION_NOT_EXIST",
      "locationCode": "L08SR-004-003-17"
    }
  ]
}
}

```

3.1.7 工作位解锁

3.1.7.1 接口描述

接口 URL	http://{IP:PORT}/location/unlock
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	<p>当需要恢复使用已被锁定的工作位时，可以调用此接口将工作位解锁，解锁后机器人可以在该工作位取放容器。</p> <p>为达到最佳性能，建议单次请求不超过 100 行明细，请求频率不超过 140 次/秒。</p>
备注	该接口单次调用最多可解锁 200 个工作位。

3.1.7.2 请求参数说明

参数名	必选	类型	说明
locationCodes	Y	list<string>	工作位编码，必须为已锁定的工作位。

3.1.7.3 请求示例

```
{
  "locationCodes": [
    "L010100017.01"
  ]
}
```

3.1.7.4 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 1: 表示未完全成功，即存在部分任务/容器/工位响应成功，部分响应失败。 ● 其他值: 表示响应失败，详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时，msg值为success。 ● 当code为1时，msg值为partial response failure。 ● 当code为其他值时，msg值为error。
data	Y	object	返回任务处理详情，详情请参见“locationCodes 参数说明”。

表 3-15 locationCodes 参数说明

字段	必选	定义	备注
errorCode	Y	string	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示处理成功。 ● 其他值: 表示处理失败，详见异常码。
message	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当errorCode为0时，message值为OK。 ● 当errorCode为其他值时，message值为错误码对应的错误描述。
locationCode	Y	string	工位编码。

3.1.7.5 响应示例

成功响应示例

```
{
```

```
"code": 0,
"msg": "success",
"data": {
  "locationCodes": [
    {
      "errorCode": "0",
      "message": "OK",
      "locationCode": "L08SR-004-003-08"
    },
    {
      "errorCode": "0",
      "message": "OK",
      "locationCode": "L08SR-004-003-09"
    },
    {
      "errorCode": "0",
      "message": "OK",
      "locationCode": "L08SR-004-003-07"
    }
  ]
}
```

失败响应示例一

```
{
  "code": 1010100011,
  "msg": "error",
  "data": {
    "locationCodes": [
      {
        "errorCode": "1030200007",
        "message": "LOCATION_NOT_EXIST",
        "locationCode": "L08SR-004-004-011"
      }
    ]
  }
}
```

失败响应示例二

```
{
  "code": 1,
  "msg": "partial response failure",
  "data": {
    "locationCodes": [
      {
```

```

        "errorCode": "0",
        "message": "OK",
        "locationCode": "L08SR-004-003-08"
    },
    {
        "errorCode": "0",
        "message": "OK",
        "locationCode": "L08SR-004-003-09"
    },
    {
        "errorCode": "1030200007",
        "message": "LOCATION_NOT_EXIST",
        "locationCode": "L08SR-004-003-17"
    }
]
}

```

3.1.8 呼叫机器人至工位

3.1.8.1 接口描述

接口 URL	http://{IP:PORT}/station/callRobot
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	将机器人呼叫至工作站/工位等待，主要应用于人工工作站场景。 为达到最佳性能，建议请求频率不超过 400 次/秒。
备注	支持由 ESS 分配空闲机器人去往上游系统指定的工作站/工位。

3.1.8.2 请求参数说明

参数名	必选	类型	说明
stationCode	N	string	<ul style="list-style-type: none"> stationCode: 工作站编码。 locationCode: 工位编码。 工作站编码和工位编码二选一必填；当工位编码和工作站编码都填时，取工位编码的值。
locationCode	N	string	

3.1.8.3 请求示例

```

{
  "locationCode": "LT_CONVEYOR_OUTPUT:POINT:54900:1960",

```

```
"stationCode": ""
}
```

3.1.8.4 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时, msg值为success。 ● 当code为其他值时, msg值为error。
data	Y	object	响应内容。 <ul style="list-style-type: none"> ● 当code为0时, data值为OK。 ● 当code为其他值时, data值为错误描述。

3.1.8.5 响应示例

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": "OK"
}
```

失败响应示例

```
{
  "code": 1030600022,
  "msg": "error",
  "data": "TO_LOCATION_NOT_EXIST"
}
```

3.1.9 让机器人离开工作位

3.1.9.1 接口描述

接口 URL	http://{IP:PORT}/station/letRobotGo
接口形式	HTTP POST+JSON

调用方向	上游系统→ESS
功能描述	让机器人离开某个位置。 为达到最佳性能，建议请求频率不超过 400 次/秒。
备注	<ul style="list-style-type: none"> ● 支持让在工作位等待的机器人离开。 ● 支持让在工作站等待的机器人离开。 ● 支持指定编码的机器人离开当前位置。

3.1.9.2 请求参数说明

参数名	必选	类型	说明
stationCode	N	string	<ul style="list-style-type: none"> ● stationCode: 工作站编码。 ● locationCode: 工作位编码。 ● robotCode: 机器人编码。 ● 工作站编码、工作位编码和机器人编码三选一必填；若填写了工作位编码，取工作位的值。
locationCode	N	string	
robotCode	N	string	

3.1.9.3 请求示例

```
{
  "locationCode": "LT_CONVEYOR_OUTPUT:POINT:54900:1960",
  "stationCode": "",
  "robotCode": ""
}
```

3.1.9.4 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 其他值: 表示响应失败，详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时，msg值为success。 ● 当code为其他值时，msg值为error。
data	Y	object	响应内容。 <ul style="list-style-type: none"> ● 当code为0时，data值为OK。 ● 当code为其他值时，data值为错误描述。

3.1.9.5 响应示例

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": "OK"
}
```

失败响应示例

```
{
  "code": 1030600005,
  "msg": "error",
  "data": "LOCATION_ROBOT_NOT_EXIST"
}
```

3.1.10 系统暂停

3.1.10.1 接口描述

接口 URL	http://{IP:PORT}/system/pause
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	上游系统调用该接口时，ESS 系统将暂停任务处理，不再下发新的指令给机器人，全场机器人暂停。 为达到最佳性能，建议请求频率不超过 400 次/秒。
备注	<ul style="list-style-type: none"> ● 触发系统暂停后，机器人执行完当前指令后暂停。 ● 由于本体机器人可能会合并移动指令执行，所以暂停时机器人不一定立即停在当前点位。

3.1.10.2 请求参数说明

参数名	必选	类型	说明
zone	N	string	库区： <ul style="list-style-type: none"> ● 不填库区，暂停全场。 ● 填写正确库区，暂停该库区。 ● 填写错误库区，系统中只有一个库区，系统暂停。 ● 填写错误库区，系统中存在多个库区，系统不做任

参数名	必选	类型	说明
			何处理。

3.1.10.3 请求示例

```
{
  "zone": ""
}
```

3.1.10.4 响应参数说明

参数名	必选	类型	说明
code	Y	int	响应状态码：0，表示接口请求成功。
msg	Y	string	返回信息说明：success。
data	Y	string	响应消息补充说明：OK。

3.1.10.5 响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": "OK"
}
```

3.1.11 系统恢复

3.1.11.1 接口描述

接口 URL	http://{IP:PORT}/system/resume
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	上游系统调用该接口，将暂停状态的 ESS 系统进行恢复，恢复成功后 ESS 继续处理任务。 为达到最佳性能，建议请求频率不超过 400 次/秒。

3.1.11.2 请求参数说明

参数名	必选	类型	说明
zone	N	string	库区： <ul style="list-style-type: none"> ● 不填库区，恢复全场。 ● 填写正确库区，恢复该库区。 ● 填写错误库区，系统中只有一个库区，系统恢复。 ● 填写错误库区，系统中存在多个库区，系统不做任何处理。

3.1.11.3 请求示例

```
{
  "zone": ""
}
```

3.1.11.4 响应参数说明

参数名	必选	类型	说明
code	Y	int	响应状态码：0，表示接口请求成功。
msg	Y	string	返回信息说明：success。
data	Y	string	响应消息补充说明：OK。

3.1.11.5 响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": "OK"
}
```

3.1.12 容器查询

3.1.12.1 接口描述

接口 URL	http://{IP:PORT}/container/queryContainer
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	通过容器编号数组或容器类型数组，查询对应容器集合的容器状态（位置、可用状态、入场状态、使用状态）；若有多个查询条件，则按逻辑关系筛选。 为达到最佳性能，建议单次请求不超过 100 行明细，请求频率不超过 400 次/秒。

3.1.12.2 请求参数说明

参数名	必选	类型	说明
containerCodes	N	list<string>	容器编码，表示按容器编码筛选容器，不写值则表示选择全部容器。
containerTypes	N	list<string>	容器类型，表示按容器类型筛选容器，不写值则表示选择全部容器。

3.1.12.3 请求示例

```
{
  "containerCodes": [
    "container001"
  ],
  "containerTypes": [
    "CT_KIVA_STANDARD"
  ]
}
```

3.1.12.4 响应参数说明

表 3-16 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码，0表示接口响应成功。
msg	Y	string	响应消息补充说明，值为 success。
data	Y	object	返回容器列表 containers，详情请参见“containers 参数说明”。

表 3-17 containers 参数说明

参数名	必选	类型	说明
containerCode	Y	string	容器编码。
containerType	Y	string	容器类型。
positionCode	Y	string	位置编码，位置编码可以为库位工作位编码、工作站工作位编码、机器人背篓（机器人 id#背篓序号）。
storageTag	Y	string	容器标识，用于和库位匹配。系统可以根据容器标签为容器分配相应标签的工作位。适用于逻辑分区、大小库位等场景。
admissionStatus	Y	string	入场状态： <ul style="list-style-type: none"> ● OUTSIDE: 场外。

参数名	必选	类型	说明
			● INSIDE: 场内。

3.1.12.5 响应示例

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "containers": [
      {
        "containerCode": "L08TB-0003",
        "containerType": "CT_KUBOT_STANDARD",
        "positionCode": "",
        "storageTag": "",
        "enableStatus": "ENABLE",
        "admissionStatus": "OUTSIDE",
        "workStatus": "IDLE"
      },
      {
        "containerCode": "L08TB-0004",
        "containerType": "CT_KUBOT_STANDARD",
        "positionCode": "L08SR-007-001-04",
        "storageTag": "",
        "enableStatus": "ENABLE",
        "admissionStatus": "INSIDE",
        "workStatus": "IDLE"
      }
    ]
  }
}
```

失败响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "containers": []
  }
}
```

3.1.13 机器人查询

3.1.13.1 接口描述

接口 URL	http://{IP:PORT}/robot/query
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	按机器人编号数组查询当前 ESS 系统中对应机器人的相关信息：基础信息、当前位置、当前姿态、当前状态、当前背篓容器、当前背篓容器任务、当前背篓容器任务目标位置等。 为达到最佳性能，建议单次请求不超过 100 行明细，请求频率不超过 20 次/秒。

3.1.13.2 请求参数说明

参数名	必选	类型	说明
robotCodes	N	list<string>	机器人编码，可传多个，若不写值则查询全部。

3.1.13.3 请求示例

```
{
  "robotCodes": [ //机器人编码列表
    "kubot-1"
  ]
}
```

3.1.13.4 响应参数说明

表 3-18 响应参数说明

参数名	必选	类型	说明
code	Y	int	响应状态码，0表示接口响应成功。
msg	Y	string	响应消息补充说明，值为 success。
data	Y	object	返回机器人列表 robots，详情请参见“robots 参数说明”。

表 3-19 robot 参数说明

参数名	必选	类型	说明
robotCode	Y	string	机器人编码，机器人唯一识别码。

参数名	必选	类型	说明
robotTypeCode	Y	string	机器人类型，例如 RT_KUBOT_KIVA。
pointCode	Y	string	坐标点编码，机器人当前所处坐标点，可以为空。
positionX	Y	long	机器人当前 x 坐标。
positionY	Y	long	机器人当前 y 坐标。
energyLevel	Y	int	机器人当前电量。
theta	Y	int	机器人车头方向相对于地图 X 轴正方向的角度。 备注：对于 HaiPick，机器人车头指有背篓的一侧。
forkHeight	Y	long	当前货叉提升高度。
forkLength	Y	long	当前货叉伸展长度。
forkTheta	Y	int	货叉朝向相对于机器人车头方向的角度（以机器人车头方向为 0 度）。
stationCode	N	string	工作站编码，当机器人在工作站工作时有值。
locationCode	N	string	工作位编码，当机器人控制权在工作位时有值。
state	Y	string	机器人状态： <ul style="list-style-type: none"> ● UNAVAILABLE: 移除机器人后，机器人在场外的状态。 ● UNKNOWN: 未知或待初始化。 ● ERROR: 错误，表示机器人状态更新超时。 ● IDLE: 空闲。 ● EXECUTING: 执行任务中。 ● AWAITING: 原地等待。
hardwareState	Y	string	机器人硬件状态： <ul style="list-style-type: none"> ● ROBOT_READY_TO_INIT: 机器人启动以后的初始状态，等待初始化指令。 ● ROBOT_IDLE: 空闲状态，等待指令/任务行为 (MOVE、BIN_OP)。 ● ROBOT_RUNNING: 运行状态（正在执行指令/任务行为）。 ● ROBOT_ABNORMAL: 异常状态（本体内部动作执行出现异常）。 ● ROBOT_RECOVERY: 恢复状态。 ● ROBOT_PAUSED: 暂停状态。
workState	Y	string	机器人作业状态：

参数名	必选	类型	说明
			<ul style="list-style-type: none"> ● OTHER: 空闲 ● WORKING: 工作中 ● OFFLINE: 下线 ● CHARGING: 充电 ● EXCEPTION: 异常
isCharging	Y	boolean	是否充电: <ul style="list-style-type: none"> ● true: 正在充电。 ● false: 未在充电。
lockedStatePointCode	Y	list<string>	锁定点位。
trays	Y	list<tray>	背篓描述。详情请参见“trays 参数说明”。
rfids	Y	list< rfid>	rfid 信息。无 rfid 信息时, 值为 null。详情请参见“rfids 参数说明”。
executingWmsTaskCode	N	string	正在执行的业务任务编码。
assignedTaskCodes	N	list<string>	分配的任务编码。
paused	Y	boolean	机器人是否暂停。 <ul style="list-style-type: none"> ● 是: true ● 否: false
zone	N	string	机器人所在区域
softwareVersion	N	string	机器人固件版本, 默认为 version1.0.0 。
cycle	N	int	电池充电循环次数。
temperature	N	double	电池温度。
alleyNo	N	Int	机器人所在巷道编码。
ip	N	string	机器人 IP。

表 3-20 trays 参数说明

参数名	必选	类型	说明
trayLevel	Y	long	背篓层数, 从 0 开始, 从下往上编号。货叉编码为 64。
containerCode	N	string	容器编码, 若背篓为空则不写值。
taskCodes	N	list< string >	业务任务编码, 仅在出库任务时有值。
positionCode	Y	string	坐标点编码, 格式为: id#背篓序号。

表 3-21 rfids 参数说明

参数名	必选	类型	说明
layer	N	int	背篓层数，从 1 开始，从下往上编号。货叉编码为 0。
rfidData	N	list<Object<string>>	RFID 读取到的标签 id，可以读取多个标签 ID。为 NULL 时表示未读取到标签。
state	N	boolean	传感器状态是否正常。 <ul style="list-style-type: none"> ● true: 正常 ● false: 不正常

3.1.13.5 响应示例

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "robots": [
      {
        "robotCode": "kubot-1",
        "robotTypeCode": "RT_KUBOT",
        "pointCode": "POINT:39491:7405",
        "positionX": 39491,
        "positionY": 7405,
        "energyLevel": 89,
        "theta": 0,
        "forkHeight": 0,
        "forkLength": 0,
        "forkTheta": 0,
        "stationCode": null,
        "locationCode": "",
        "state": "IDLE",
        "hardwareState": "ROBOT_IDLE",
        "isCharging": false,
        "lockedStatePointCode": [
          "POINT:39491:7405#0"
        ],
      },
      {
        "trays": [
          {
            "trayLevel": 0,
            "containerCode": null,
            "taskCodes": [],
            "positionCode": "kubot-1#0"
          }
        ]
      }
    ]
  }
}
```

```
        "trayLevel": 1,  
        "containerCode": null,  
        "taskCodes": [],  
        "positionCode": "kubot-1#1"  
    },  
    {  
        "trayLevel": 2,  
        "containerCode": null,  
        "taskCodes": [],  
        "positionCode": "kubot-1#2"  
    },  
    {  
        "trayLevel": 3,  
        "containerCode": null,  
        "taskCodes": [],  
        "positionCode": "kubot-1#3"  
    },  
    {  
        "trayLevel": 4,  
        "containerCode": null,  
        "taskCodes": [],  
        "positionCode": "kubot-1#4"  
    },  
    {  
        "trayLevel": 5,  
        "containerCode": null,  
        "taskCodes": [],  
        "positionCode": "kubot-1#5"  
    },  
    {  
        "trayLevel": 64,  
        "containerCode": null,  
        "taskCodes": [],  
        "positionCode": "kubot-1#64"  
    }  
],  
"rfids": null,  
"executingWmsTaskCode": null,  
"assignedTaskCodes": null,  
"paused": true,  
"zone": "",  
"softwareVersion": "version1.0.0",  
"cycle": 0,  
"temperature": 30.0,  
"workState": "OTHER",  
"alleyNo": 2  
},  
{
```

```
"robotCode": "kubot-2",
"robotTypeCode": "RT_KUBOT",
"pointCode": "POINT:43320:13835",
"positionX": 43320,
"positionY": 13835,
"energyLevel": 100,
"theta": 90,
"forkHeight": 0,
"forkLength": 0,
"forkTheta": 0,
"stationCode": null,
"locationCode": "",
"state": "IDLE",
"hardwareState": "ROBOT_IDLE",
"isCharging": true,
"lockedStatePointCode": [
  "POINT:43320:13835#90"
],
"trays": [
  {
    "trayLevel": 0,
    "containerCode": null,
    "taskCodes": [],
    "positionCode": "kubot-2#0"
  },
  {
    "trayLevel": 1,
    "containerCode": null,
    "taskCodes": [],
    "positionCode": "kubot-2#1"
  },
  {
    "trayLevel": 2,
    "containerCode": null,
    "taskCodes": [],
    "positionCode": "kubot-2#2"
  },
  {
    "trayLevel": 3,
    "containerCode": null,
    "taskCodes": [],
    "positionCode": "kubot-2#3"
  },
  {
    "trayLevel": 4,
    "containerCode": null,
    "taskCodes": [],
    "positionCode": "kubot-2#4"
  }
]
```

```
    },
    {
      "trayLevel": 5,
      "containerCode": null,
      "taskCodes": [],
      "positionCode": "kubot-2#5"
    },
    {
      "trayLevel": 64,
      "containerCode": null,
      "taskCodes": [],
      "positionCode": "kubot-2#64"
    }
  ],
  "rfids": null,
  "executingWmsTaskCode": null,
  "assignedTaskCodes": null,
  "paused": true,
  "zone": "",
  "softwareVersion": "version1.0.0",
  "cycle": 0,
  "temperature": 30.0,
  "workState": "CHARGING",
  "alleyNo": -1,
  "ip": "123.12.12.12"
}
]
}
```

失败响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "robots": []
  }
}
```

3.1.14 工作位查询

3.1.14.1 接口描述

接口 URL	http://{IP:PORT}/location/query
接口形式	HTTP POST+JSON

调用方向	上游系统→ESS
功能描述	通过工位编码、容器编码或工位类型编码查询工位的详细数据信息：工位位置信息、是否被占用等。 为达到最佳性能，建议单次请求不超过 100 行明细，请求频率不超过 400 次/秒。

3.1.14.2 请求参数说明

参数名	必选	类型	说明
locationCodes	N	list<string>	工位编码。可写入多个工位编码，若不写值则默认查询全部。若容器编码、工位类型编码均填写，则互为且关系进行过滤。
containerCode	N	string	容器编码，查看容器所在工位。系统会过滤容器编码，不可传递错误编码。
locationTypeCodes	N	list<string>	工位类型编码，可写入多个工位类型编码，若不写值则默认查询全部。系统会过滤工位类型，不可传递空字符串或是错误编码。

3.1.14.3 请求示例

```
{
  "locationCodes": [
    "L010100018.01"
  ],
  "containerCode": "container001",
  "locationTypeCodes": [
    "LT_SHELF_STORAGE"
  ]
}
```

3.1.14.4 响应参数说明

表 3-22 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码，0 表示接口响应成功。
msg	Y	string	响应消息补充说明，值为 success。
data	Y	object	返回工位列表 locations，详情请参见“locations 参数说明”。

表 3-23 locations 参数说明

参数名	必选	类型	说明
locationCode	Y	string	工作位编码，工作位唯一识别码。
locationTypeCode	Y	string	工作位类型编码。
stationCode	N	string	工作站编码，若是库位，则无工作站编码。
stationModelCode	N	string	工作站类型： <ul style="list-style-type: none"> ● LABOR_STATION: 人工工作站。 ● CONVEYOR_STATION: 输送线工作站。 ● RACK_BUFFER_STATION: 缓存货架工作站。 ● HAIPORT_STATION: 卸料机工作站。 ● STORAGE_LOCATION: 库位。 ● NOT_SET: 其他。
positionX	Y	string	工作位的 x 坐标。
positionY	Y	string	工作位的 y 坐标。
positionZ	Y	string	工作位的 z 坐标。
loadContainerCode	N	string	当前工作位存放的容器编码，若没有容器则返回空字符串。
parentContainerCode	N	string	容器关联的父容器编码，若没有关联的父容器则返回 NULL。
level	N	int	工作位所处的层，若没有则返回 NULL。
bay	N	int	工作位所处的列，若没有则返回 NULL。
storageTag	N	string	工作位标识，系统会根据标签为容器分配对应的工作位，适用于逻辑分区、大小库位等场景。
isLocked	Y	boolean	工作位是否被禁用： <ul style="list-style-type: none"> ● TURE: 禁用。 ● FALSE: 未禁用。
isAbnormal	Y	boolean	工作位是否异常： <ul style="list-style-type: none"> ● TURE: 异常。 ● FALSE: 正常。
abnormalReason	Y	string	异常原因。
updateTime	Y	string	更新时间，单位：ms。

3.1.14.5 响应示例

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "locations": [
      {
        "locationCode": "L08SR-006-004-10",
        "locationTypeCode": "LT_SHELF_STORAGE",
        "stationCode": "LA_SHELF_STORAGE",
        "stationModelCode": "STORAGE_LOCATION",
        "positionX": "38301",
        "positionY": "7405",
        "positionZ": "4000",
        "loadContainerCode": "",
        "parentContainerCode": null,
        "level": null,
        "bay": null,
        "storageTag": "",
        "isLocked": false,
        "isAbnormal": false,
        "abnormalReason": "NONE",
        "updateTime": "1709006251354"
      },
      {
        "locationCode": "L08SR-006-004-13",
        "locationTypeCode": "LT_SHELF_STORAGE",
        "stationCode": "LA_SHELF_STORAGE",
        "stationModelCode": "STORAGE_LOCATION",
        "positionX": "38301",
        "positionY": "7405",
        "positionZ": "5200",
        "loadContainerCode": "",
        "parentContainerCode": null,
        "level": null,
        "bay": null,
        "storageTag": "",
        "isLocked": false,
        "isAbnormal": false,
        "abnormalReason": "NONE",
        "updateTime": "1709006251353"
      }
    ]
  }
}
```

失败响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "locations": []
  }
}
```

3.2 上游系统提供的接口

3.2.1 任务申请

3.2.1.1 接口描述

接口 URL	上游系统提供
接口形式	HTTP POST+JSON
调用方向	ESS→上游系统
功能描述	容器入库时需要向上游系统申请入库任务
备注	<p>上游系统管理库位时，ESS 支持以下两种任务申请场景：</p> <ul style="list-style-type: none"> 上游系统没有提前给容器下发入库任务，机器人取箱之后需根据容器编码向上游申请入库任务，上游调用任务下发接口创建入库任务，这种方式下需要提前初始化容器基础信息。 上游下发的容器入库任务没有目标库位，机器人取箱之后需根据容器编码和入库任务编码向上游申请入库任务，上游调用任务下发接口创建入库任务，更新容器的入库任务信息。

3.2.1.2 回调参数说明

表 3-24 请求参数说明

参数名	必选	类型	说明
mapNo	N	string	地图编码。
robotCode	Y	string	机器人编码。
robotTypeCode	Y	string	根据机器人类型填写，默认值为 RT_KUBOT。
containerTaskDtoList	Y	List<ContainerInfo>	需要申请库位的容器列表。

表 3-25 containerTaskDtoList 参数说明

参数名	必选	类型	说明
containerCode	Y	string	需要申请目标库位的容器编码。
taskCode	C	string	任务编码，根据系统配置选填： <ul style="list-style-type: none"> ● 必填：上游下发的入库任务没有目标库位，向上游申请入库任务。 ● 不填：上游没有下发入库任务，需要申请入库任务下发。

3.2.1.3 请求示例

```
{
  "mapNo": "1",
  "robotCode": "robot-1",
  "robotTypeCode": "RT_KUBOT",
  "containerTaskDtoList": [
    {
      "containerCode": "container001",
      "taskCode": "taskCode-001"
    }
  ]
}
```

3.2.1.4 响应参数说明

参数名	必选	类型	说明
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 1：表示接口请求成功。 ● 其他值：表示接口请求失败。
msg	Y	string	返回信息说明。 <ul style="list-style-type: none"> ● 成功："success"。 ● 其他：详细描述。
data	Y	string	返回处理成功的任务编码列表，未返回任务编码表示处理失败。

3.2.1.5 响应示例

```
{
  "code": 1,
  "msg": "success",
  "data": {}
}
```

3.2.2 异常上报

3.2.2.1 接口描述

接口 URL	上游系统提供
接口形式	HTTP POST+JSON
调用方向	ESS→上游系统
功能描述	ESS 调用此接口将工位异常信息和机器人异常信息上报至上游系统。
备注	<p>工位异常原因：</p> <ul style="list-style-type: none"> ● 取箱失败次数达到上限 ● 放箱失败次数达到上限 ● 容器位置不匹配 ● 容器不存在 ● 扫码失败 ● 设备异常 <p>机器人异常包含：</p> <ul style="list-style-type: none"> ● 机器人状态异常 ● 机器人指令超时 ● 机器人初始化异常 ● 机器人充电异常 ● 机器人位置异常 ● 机器人货叉异常

3.2.2.2 回调参数说明

参数名	必选	类型	说明
eventType	Y	string	事件类型： <ul style="list-style-type: none"> ● location_abnormal: 工位异常。 ● robot_abnormal: 机器人异常。
robotCode	N	string	机器人编码。
stationCode	N	string	工作站编码。
locationCode	N	string	工位编码。
containerCode	N	string	容器编码。
message	N	string	异常描述。
updateTime	Y	long	更新时间，时间单位：ms。

3.2.2.3 回调示例

机器人异常回调

```
{
  "eventType": "robot_abnormal",
  "robotCode": "robot001",
  "stationCode": "",
  "locationCode": "location001",
  "containerCode": "container001",
  "message": "something happened",
  "updateTime": 1637141519277
}
```

工作位异常回调

```
{
  "eventType": "location_abnormal",
  "robotCode": null,
  "locationCode": "LT_CONVEYOR_INPUT:POINT:29940:8710",
  "stationCode": "2_01",
  "containerCode": "",
  "message": "UNLOAD_FAILED_COUNT_EXCEEDED_THE_LIMIT",
  "updateTime": 1652412924367
}
```

3.2.2.4 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 ● 0: 正常。 ● 非0: 异常。
msg	Y	string	响应消息补充说明, success 或其他异常消息。
data	Y	json 对象	返回数据。

3.2.2.5 响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {}
}
```

第4章 周边设备控制接口

4.1 电子标签接口

以下接口适用于由 ESS 系统提供电子标签设备底层对接能力的场景。上游系统调用 ESS 接口控制电子标签的亮灯、闪烁及灭灯等，同时接收 ESS 回传电子标签灭灯、数字改变、光电信号变化及异常等信息，以配合完成业务操作。

4.1.1 ESS 提供的接口

4.1.1.1 电子标签控制请求

接口描述

接口 URL	http://{IP:PORT}/ptl/sendCommand
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	发送命令给电子标签，可以控制电子标签亮灯颜色及灭灯等。

请求参数说明

参数名	必选	类型	说明
controllerCode	N	string	控制器 code, 需要与控制器配置 ptlController 里的 code 对应。
index	N	string	电子标签序号, 需要与电子标签 ptlTag 里的 index 对应。
tagCode	Y	string	电子标签编号, 需要与电子标签 ptlTag 里的 code 对应。
color	Y	string	颜色: <ul style="list-style-type: none"> ● RED ● GREEN ● BLUE
mode	Y	string	工作模式: <ul style="list-style-type: none"> ● LIGHT: 亮灯 ● DARK: 灭灯 ● FLASH: 闪灯
display	N	string	常用于显示数字, 支持 3 位数。
displayText	N	string	显示文本内容。

请求示例

```
{
  "controllerCode": "group01",
  "index": "0001",
  "tagCode": "CS-001-001-01",
  "color": "RED", //GREEN BLUE YELLOW PINK CYAN WHITE
  "mode": "LIGHT DARK FLASH",
  "display": "777",
  "displayText": "Text"
}
```

响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 其他值: 表示响应失败, 详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时, msg值为success。 ● 当code为其他值时, msg值为error。
data	Y	object	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时, data值为OK。 ● 当code为其他值时, data值为错误描述。

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": "OK"
}
```

失败响应示例

```
{
  "code": 2001001008,
  "msg": "error",
  "data": null
}
```

4.1.2 上游系统提供的接口

4.1.2.1 电子标签信号回传

接口描述

接口 URL	上游系统提供
接口形式	HTTP POST+JSON
调用方向	ESS→上游系统
功能描述	信号回传接口，回传电子标签状态变化内容给上游系统。 当人为操作（比如拍灯，更改数字，放箱等）导致电子标签的状态和信号等发生变化、或设备本身发生异常时，ESS 接收到信号后，通过该接口将信号回传给上游系统。

回调参数说明

参数名	必选	类型	说明
eventCode	Y	string	事件名称： <ul style="list-style-type: none"> ● LIGHT_OFF: 灭灯。 ● DISPLAY_CHANGE: 显示数字的变化。 ● TAG_ABNORMAL: 电子标签异常。 ● SIGNAL: 光电信号。
controllerCode	Y	string	控制器 code，需要与控制器配置 ptlController 里 code 对应。
tagCode	Y	string	电子标签编号，需要与电子标签 ptlTag 里的 code 对应。
display	N	long	显示信息，数量改变时回传。
status	N	int	电子标签状态： <ul style="list-style-type: none"> ● 0: 作业完成。 ● 1: 电子标签异常。 ● 2: 作业取消。 ● 3: 命令错误。 ● 4: 连接失败。
signal	N	int	是否有箱子的信号： <ul style="list-style-type: none"> ● 1: 有箱。 ● 2: 无箱。

灯灭回调示例

```
{
  "controllerCode": "group01",
  "tagCode": "CK-001-001-01",
  "display": "",
  "eventCode": "LIGHT_OFF",
  "status": 0
}
```

电子标签异常回调示例

```
{
  "controllerCode": "group02",
  "tagCode": "CK-002-001-03",
  "display": "",
  "eventCode": "TAG_ABNORMAL",
  "status": 1
}
```

数字改变回调示例

```
{
  "controllerCode": "group01",
  "tagCode": "CK-001-001-02",
  "display": "22122",
  "eventCode": "DISPLAY_CHANGE",
  "status": 0
}
```

光电信号回调示例

```
{
  "controllerCode": "12_controller",
  "tagCode": "CS-002-005-03",
  "display": "",
  "signal": 1,
  "eventCode": "SIGNAL"
}
```

响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 正常。 ● 非0: 异常。

字段	必选	定义	备注
msg	Y	string	响应消息补充说明, success 或其他异常消息。
data	Y	json 对象	返回数据。

响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {}
}
```

4.2 输送线接口

以下接口适用于由 ESS 对接输送线, 且需要在拣选位等节点将容器身份、重量等信息上报给上游系统进行业务操作或校验的场景。ESS 读取输送线线体上关键节点的信息, 并通过接口上报给上游系统, 上游系统操作或校验完成后, 调用 ESS 接口通知容器流动。

4.2.1 ESS 提供的接口

4.2.1.1 容器流动通知

接口描述

接口 URL	http://{IP:PORT}/conveyor/moveContainer
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	业务操作完成容器可以离开时, 上游系统调用 ESS 接口通知输送线将容器取出或让容器流动等。

请求参数说明

参数名	必选	类型	说明
slotCode	Y	string	输送线节点编码。
containerCode	Y	string	容器编码。
direction	N	string	容器在拣选位时的流动方向, 拣选任务完成后由上游系统控制箱子流向或空箱取出。流动信号的取值需要与输送线厂商约定好, 上下游系统保持一致, 例如: <ul style="list-style-type: none"> 100: 向前流动, 离开拣选位 (只有一个流向时, 可

参数名	必选	类型	说明
			以不传，默认向前流动)。 ● 200: 取出容器。

请求示例

```
{
  "slotCode": "c1-1",
  "containerCode": "container001",
  "direction": "100"
}
```

响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 ● 0: 表示接口响应成功。 ● 其他值: 表示响应失败, 详见 异常码 。
msg	Y	string	响应消息补充说明。 ● 当code为0时, msg值为success。 ● 当code为其他值时, msg值为error。
data	Y	object	响应消息补充说明。 ● 当code为0时, data值为OK。 ● 当code为其他值时, data值为错误描述。

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": "OK"
}
```

失败响应示例

```
{
  "code": 2001001008,
  "msg": "ERROR",
  "data": null
}
```

4.2.1.2 输送线线体状态查询

接口描述

接口 URL	http://{IP:PORT}/conveyor/queryConveyor
接口形式	HTTP POST+JSON
调用方向	上游系统→ESS
功能描述	<p>上游系统调用该接口，根据输送线线体编码查询输送线线体上各个点位是否有容器、容器编码、容器属性等信息。</p> <p>适用的业务场景：上游系统根据输送线上容器数量及入库任务情况，判断下发空箱出库任务的数量等。</p>

请求参数说明

参数名	必选	类型	说明
conveyorCodes	N	list<string>	输送线编码列表，根据输送线编码列表查询状态。

请求示例

```
{
  "conveyorCodes": [
    "1",
    "2"
  ]
}
```

响应参数说明

表 4-1 响应参数说明

字段	必选	定义	备注
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0: 表示接口响应成功。 ● 其他值: 表示响应失败，详见异常码。
msg	Y	string	响应消息补充说明。 <ul style="list-style-type: none"> ● 当code为0时，msg值为success。 ● 当code为其他值时，msg值为error。
data	Y	object	返回处理结果。 <ul style="list-style-type: none"> ● 当code为0时，data值为输送线列表conveyors，详情请参见“conveyors参数说明”。

字段	必选	定义	备注
			<ul style="list-style-type: none"> 当code为其他值时，data值为错误描述。

表 4-2 conveyors 参数说明

参数名	必选	类型	说明
code	Y	string	输送线编码
nodeStates	Y	object	输送线节点状态信息，详情请参见“nodeStates 参数说明”。
workState	Y	string	输送线通讯连接状态。 <ul style="list-style-type: none"> ONLINE：输送线与ESS通信连接正常。 OFFLINE：输送线与ESS通信连接中断。

表 4-3 nodeStates 参数说明

参数名	必选	类型	说明
slotCode	Y	string	输送线上的位置编码
hasContainer	Y	boolean	是否有容器： <ul style="list-style-type: none"> 有容器：true。 无容器：false。
containerCode	N	string	容器编码，没有读码器时可能为空。
lastReadTime	Y	long	最近一次读取容器状态的时间，时间单位：ms。
lastReportTime	Y	long	最后一次上报容器到达的时间，时间单位：ms。
lastHasContainerTime	Y	long	最后一次有容器的时间，时间单位：ms。
attribute	N	object	容器属性，详情请参见“attribute 参数说明”。

表 4-4 attribute 参数说明

字段	必选	定义	备注
height	N	long	容器高度
length	N	long	容器长度
width	N	long	容器宽度
weight	N	long	容器重量
material	N	long	容器材料 <ol style="list-style-type: none"> 料箱 纸箱
orientation	N	string	容器朝向
qrCode	N	string	料箱二维码
skuInfo	N	string	sku 信息

字段	必选	定义	备注
abnormalReason	N	string	输送线节点异常原因
groupabnormalReason	N	string	输送线节点所在组异常原因

成功响应示例

```
{
  "code": 0,
  "msg": "success",
  "data": {
    "conveyors": [
      {
        "code": "1",
        "nodeStates": [
          {
            "slotCode": "1-1",
            "hasContainer": false,
            "containerCode": "",
            "lastReadTime": 0,
            "lastReportTime": 0,
            "lastHasContainerTime": 0,
            "lastLeaveSuccessTime": 0,
            "groupCode": "",
            "attribute": {
              "height": 0,
              "length": 0,
              "width": 0,
              "weight": 0,
              "material": 0,
              "orientation": "",
              "skuInfo": "",
              "abnormalReason": "",
              "groupAbnormalReason": ""
            }
          }
        ],
      },
      {
        "slotCode": "1-2",
        "hasContainer": false,
        "containerCode": "",
        "lastReadTime": 0,
        "lastReportTime": 0,
        "lastHasContainerTime": 0,
        "lastLeaveSuccessTime": 0,
        "groupCode": "",
        "attribute": {
          "height": 0,
          "length": 0,
          "width": 0,
```

```
        "weight": 0,
        "material": 0,
        "orientation": "",
        "skuInfo": "",
        "abnormalReason": "",
        "groupAbnormalReason": ""
    }
},
{
    "slotCode": "1-3",
    "hasContainer": false,
    "containerCode": "",
    "lastReadTime": 0,
    "lastReportTime": 0,
    "lastHasContainerTime": 0,
    "lastLeaveSuccessTime": 0,
    "groupCode": "",
    "attribute": {
        "height": 0,
        "length": 0,
        "width": 0,
        "weight": 0,
        "material": 0,
        "orientation": "",
        "skuInfo": "",
        "abnormalReason": "",
        "groupAbnormalReason": ""
    }
}
],
"workState": "OFFLINE"
},
{
    "code": "2",
    "nodeStates": [
        {
            "slotCode": "2-1",
            "hasContainer": false,
            "containerCode": "",
            "lastReadTime": 0,
            "lastReportTime": 0,
            "lastHasContainerTime": 0,
            "lastLeaveSuccessTime": 0,
            "groupCode": "",
            "attribute": {
                "height": 0,
                "length": 0,
                "width": 0,
```

```
        "weight": 0,  
        "material": 0,  
        "orientation": "",  
        "skuInfo": "",  
        "abnormalReason": "",  
        "groupAbnormalReason": ""  
    },  
    {  
        "slotCode": "2-2",  
        "hasContainer": false,  
        "containerCode": "",  
        "lastReadTime": 0,  
        "lastReportTime": 0,  
        "lastHasContainerTime": 0,  
        "lastLeaveSuccessTime": 0,  
        "groupCode": "",  
        "attribute": {  
            "height": 0,  
            "length": 0,  
            "width": 0,  
            "weight": 0,  
            "material": 0,  
            "orientation": "",  
            "skuInfo": "",  
            "abnormalReason": "",  
            "groupAbnormalReason": ""  
        }  
    },  
    {  
        "slotCode": "2-3",  
        "hasContainer": false,  
        "containerCode": "",  
        "lastReadTime": 0,  
        "lastReportTime": 0,  
        "lastHasContainerTime": 0,  
        "lastLeaveSuccessTime": 0,  
        "groupCode": "",  
        "attribute": {  
            "height": 0,  
            "length": 0,  
            "width": 0,  
            "weight": 0,  
            "material": 0,  
            "orientation": "",  
            "skuInfo": "",  
            "abnormalReason": "",  
            "groupAbnormalReason": ""  
        }  
    }  
}
```

```

        }
      }
    ],
    "workState": "OFFLINE"
  }
]
}
}

```

失败响应示例

```

{
  "code": 2001001008,
  "msg": "error",
  "data": null
}

```

4.2.2 上游系统提供的接口

4.2.2.1 容器到达上报

接口描述

接口 URL	上游系统提供
接口形式	HTTP POST+JSON
调用方向	ESS→上游系统
功能描述	容器到达输送线某个点位时，ESS 定时向上游系统发送(时间间隔可配置)容器到达的通知。
备注	<ul style="list-style-type: none"> ● 默认只上报一次。 ● 支持配置无限次上报，可配置上报时间间隔，但会影响效率，不建议使用。

回调参数说明

表 4-5 回调参数说明

参数名	必选	类型	说明
callId	Y	string	一次回调唯一标识。
slotCode	Y	string	输送线节点编码。
containerCode	Y	string	容器编码，输送线有读码器可以识别或已知容器身份时，该字段传容器编码，否则为空。
containerAttribute	N	object	容器属性。

表 4-6 containerAttribute 参数说明

参数名	必选	类型	说明
height	N	long	容器高度，例如：300（mm）。
length	N	long	容器长度，例如：650（mm）。
width	N	long	容器宽度，例如：400（mm）。
weight	N	long	容器重量。
material	N	long	容器材料： 1：料箱 2：纸箱
orientation	N	string	容器朝向，单位为“°”，例如：0、90、180、270。
qrcode	N	string	料箱二维码。
skuInfo	N	string	sku 信息。
abnormalReason	N	string	异常原因。

回调示例

```
{
  "callid": "1726476859022"
  "slotCode": "c1-1",
  "containerCode": "container001",
  "containerAttribute": {
    "height": 148,
    "length": 300,
    "width": 400,
    "weight": 10,
    "material": 1,
    "orientation": "",
    "qrcode": "",
    "abnormalReason": ""
  }
}
```

响应参数说明

表 4-7 响应参数说明

参数名	必选	类型	说明
code	Y	int	响应状态码。 <ul style="list-style-type: none"> ● 0：表示接口请求成功。 ● 其他值：表示接口请求失败。

参数名	必选	类型	说明
msg	Y	string	接口调用状态。
data	N	string	返回数据对象。

表 4-8 data 参数说明

参数名	必选	类型	说明
direction	N	string	容器在拣选位时的流动方向，上游直接校验任务，没有拣选需求的容器直接流走。例如： <ul style="list-style-type: none">● 100: 向前流动，离开拣选位（只有一个流向的话，可以不传，默认向前流动）。● 200: 取出容器。

响应示例

```
{
  "code": 0,
  "msg": "",
  "data": {
    "direction": "100"
  }
}
```

第5章 异常码

序号	异常码	异常描述
1	429	请求太频繁，稍后再试
2	1030100002	编码[XXX]在系统中已被其他类型使用，创建容器失败
3	1030200000	工位[XXX]不允许机器人取箱，容器入场失败
4	1030200001	工位[XXX]已存在容器，容器入场失败
5	1030200002	容器的状态为在场内，容器入场失败
6	1030200003	工位[XXX]不允许呼叫机器人，创建呼叫任务失败
7	1030200005	工位[XXX]不允许机器人放箱，容器离场失败
8	1030200006	容器编码与工位记录的容器编码不一致，容器离场失败
9	1030200007	工位编码[XXX]在系统中不存在
10	1030300000	容器类型不存在，创建容器失败
11	1030300001	容器编码不能为空
12	1030600032	容器编码[XXX]在系统中不存在
13	1030400000	[XXX]工作站不支持调用让机器人离开接口，任务下发失败
14	1030400003	工作站编码[XXX]不存在
15	1030500006	任务状态不支持取消
16	1030500007	机器人状态异常，与该机器人绑定的容器不允许操作
17	1030600005	[XXX]工位上无机器人
18	1030600009	系统不支持任务模板[XXX]
19	1030600015	容器编码[XXX]在系统中不存在，创建任务失败
20	1030600016	系统暂无可用的机器人，创建任务失败
21	1030600017	任务在系统中已存在，创建任务失败
22	1030600021	目标工作站或目标工位不能均为空

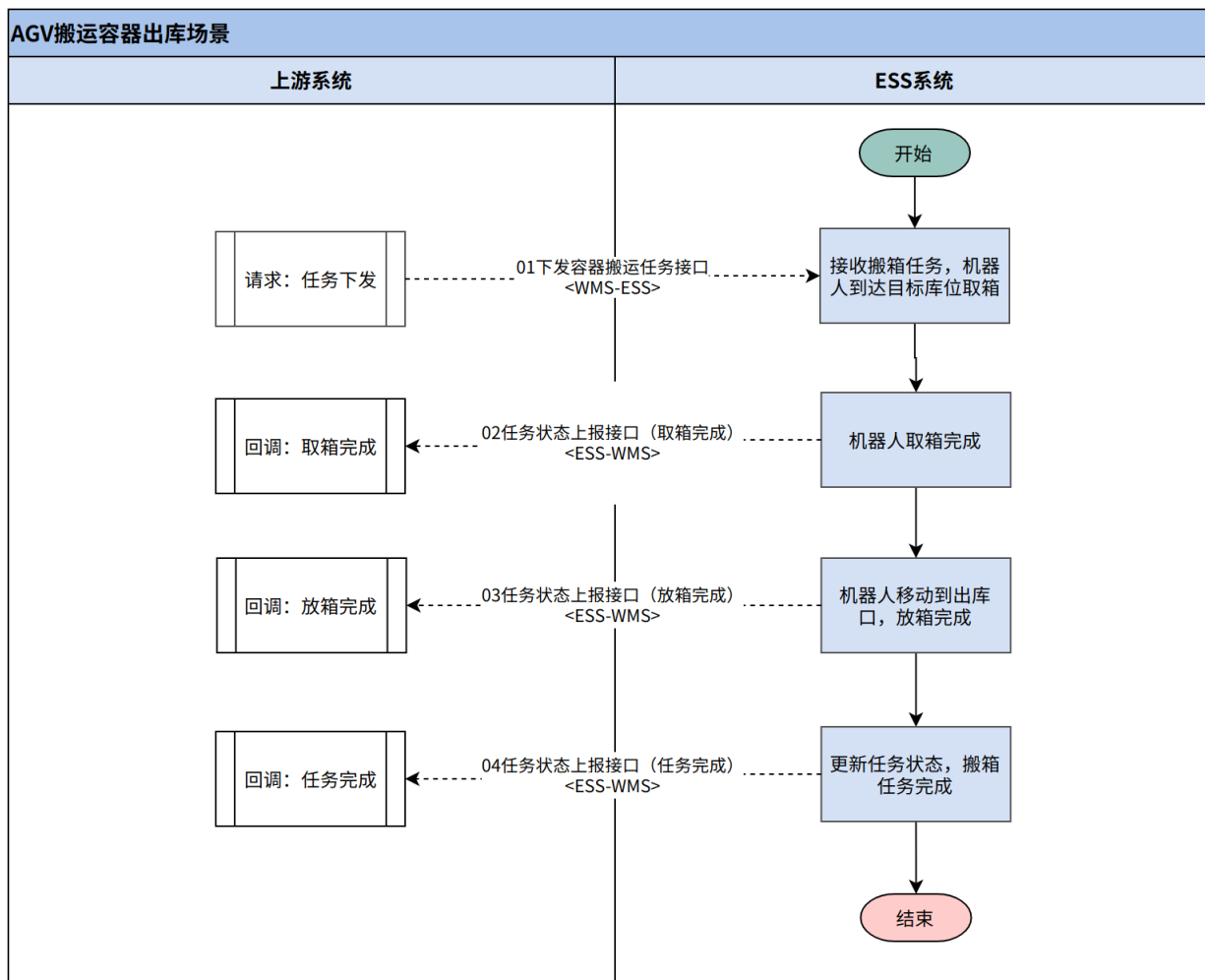
序号	异常码	异常描述
23	1030600022	目标工位 [XXX] 不存在
24	1030600023	目标工位 [XXX] 未绑定工作站
25	1030600024	容器编码或起始工位不能均为空
26	1030600027	未知的容器朝向 [XXX]
27	1030600028	起始工位 [XXX] 在系统中不存在，创建任务失败
28	1030600029	起始工位[XXX] 未关联工作站，创建任务失败
29	1030600204	未指定任务优先级，优先级调整失败
30	1030800002	工位[XXX]在系统中不存在
31	1060100208	目标背篓[XXX]不存在容器[XXX]
32	1060100209	目标背篓[XXX]存在其他容器[XXX]
33	1060100210	目标背篓 [XXX] 在系统中不存在
34	1060100211	容器不在场外，容器入场失败
35	1070200004	电子标签没有绑定控制器，下发控制请求失败
36	1070200005	电子标签不支持[XXX]颜色
37	1070200007	系统不支持[XXX]电子标签类型
38	1070200008	下发电子标签指令失败
39	1070300001	未知异常
40	1070300006	输送线未配置相关节点，查询失败
41	1070300009	[XXX]节点编码在系统中不存在
42	1070300010	[XXX]节点编码关联的输送线在系统中不存在
43	1030600030	任务下发的容器编码和系统记录的工作位容器信息不一致，创建任务失败
44	1030300004	下发的容器编码不是容器类型的编码，操作失败，请下发正确的容器编码

序号	异常码	异常描述
45	1030300005	异常容器或在场内容器不能移除
46	1030300006	容器有任务，无法移除
47	1030200013	工位[XXX]未关联工作站
48	2001001007	请求超时
49	2001001008	系统内部异常
50	2001001009	参数错误
51	2004001000	权限认证失败，无法访问
52	2004001001	该用户不存在
53	2004001002	该账号不存在
54	2004001003	该用户已存在
55	2004001004	密码输入错误
56	2004001005	token 无效，请重新申请 token 鉴权
57	2004001006	账号异常
58	2004001007	创建 token 失败
59	2004001008	APPID 在系统中不存在，请求生成 token 失败，请配置正确的 APPID
60	2004001009	密钥错误
61	2004001010	账户被锁定，三分钟后再试
62	2004001011	账户信息过期
63	2004002000	角色不存在
64	2004002001	角色已存在
65	2004002002	角色有关联用户，不允许删除
66	2007001017	请求失败，请求参数 isShelfStorage 字段必填

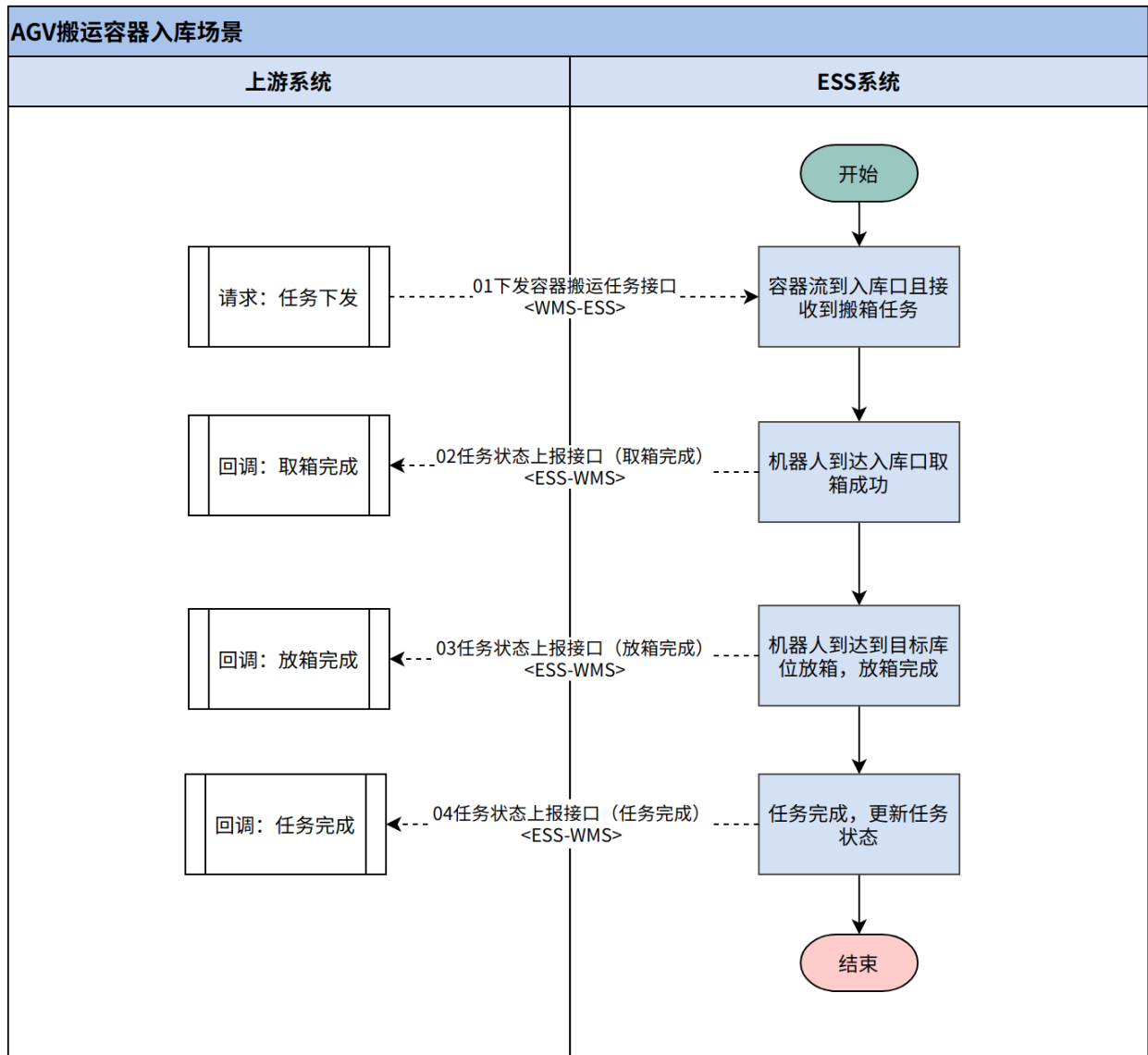
序号	异常码	异常描述
67	2007001018	请求失败, 请求参数 taskType 字段必填
68	2007001019	任务下发时系统自动创建容器失败, 导致任务创建失败
69	1030600034	工作站编码在系统中不存在, 呼叫机器人任务下发失败
70	1030600035	工作站[XXX]不支持呼叫机器人, 呼叫机器人任务下发失败
71	1030600036	工作位未绑定工作站, 呼叫机器人任务下发失败
72	1030600037	工作位不支持呼叫机器人, 呼叫机器人任务下发失败
73	1010100011	未知的位置编码, 操作失败
74	1060100000	机器人编码[xxx]在系统中不存在, 让机器人离开任务下发失败
75	1030600006	机器人有未完成的任务, 让机器人离开任务下发失败
76	1030600010	目的地编码[xxx]在系统中不存在, 入库任务下发失败
77	1030600018	容器有正在进行的上架任务, 任务下发失败
78	1030600012	任务下发的参数数组大小不匹配, 任务下发失败
79	1030600013	目的地不存在, 任务下发失败
80	1030600026	目标工作站不可用, 搬运任务下发失败
81	1030600011	任务模板不存在, 任务下发失败
82	1030600031	创建的异常容器编码已用完, 任务下发失败。请创建异常容器编码后再下发任务
83	1030600033	目标工作位不支持移库, 移库任务下发失败
84	1010100000	ESS 系统未启动, 任务下发失败
85	2007001020	容器有待办或执行中的任务, 任务下发失败
86	2007001021	容器无位置信息, 下发任务失败

第6章 典型场景示例

6.1 AGV 搬运容器出库场景



6.2 AVG 搬运容器入库场景



HAI ROBOTICS

— 箱式仓储机器人系统的首创者与领航者 —

www.hairobotics.com



© 深圳市海柔创新科技有限公司