Geek+
机器人智能物流专家

北京极智嘉科技有限公司

目录

版本及修订

在此部分管理 API 文档的历史版本及修改记录。

时间	版本	修订人	修订内容
2020-04-02	V3.3.0	RMS 团队	初始化，适用系统版本 3.5.0

1 RMS API 简介

1.1 RMS 系统

极智嘉机器人管理系统（Robot Manage System，简称 RMS，下同）为结构化环境下的多机器人应用提供完整的解决方案，例如仓库内的订单拣选和包裹分拣，以及工厂内的物料搬运。RMS 通过无线网络与系统中所有机器人进行连接、更新机器人状态信息、将任务分配给合适的机器人、确定哪一个机器人自动充电、同时为机器人规划路径并控制机器人正确执行任务、调度机器人路径以避免碰撞。此外，RMS 还可以管理系统中货架的位置。

1.1.1 RMS 的 API 是什么

RMS 的标准 API，是 RMS 系统预先设定好的一系列应用程序接口(Application Programming Interface，简称 API，下同)，用户按照指定的格式与方法向 RMS 发送信息，即可使自己的应用程序与 RMS 进行交互，进而达到命令机器人完成相关任务的目的。

RMS 的标准 API 文档为用户将 RMS 系统集成到用户自己的应用程序中提供指导，在文档中用户可以了解到 RMS 与主机系统的集成方法，包括接口协议，消息格式和一些相关示例等。

1.2 基本概念简介

1.2.1 机器人

机器人（Robot）是指在一个可以在平面内运动的、受 RMS 系统管理的、独立工作的最小执行单元。根据不同的业务场景，目前本公司已推出拣选（P、C）、搬运（M、F）、分拣（S）等多类型、多型号的一系列机器人。不同类型的机器人所能接收的任务类型也不一致，用户在后续使用 API 进行任务下发时需要注意。

1.2.2 货架

货架（Shelf）是可以被 RMS 及机器人识别、由机器人进行搬运的、独立的物资存放设备。在不同的业务场景中，货架的表现形式可能并不一致，多层储物架、钢托、笼车在不同的场景下都可以被 RMS 认为是一个货架。

1.2.3 地图

地图（Map）是实际运行场地在 RMS 中的虚拟映射概念。

在二维码类型的 RMS 中，场地会被规划实施人员划分为一个二维平面矩阵，由一个个单元格组成，RMS 通过指定地图上的索引坐标控制机器人在实际场地中的运动位置。

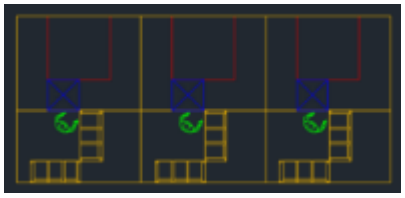
1.2.4 工作站

1.2.4.1 标准工作站

标准布局类型的工作站有一个排队区（田字格区域），机器人可以在这里进行排队，因此工作人员只需呆在原地就可以完成工作，并且需要有一个旋转区域来保证机器人安全地旋转货架，两种典型标准布局类型的工作站如下：

“E”形布局类型：

“U”形“布局类型：



除了这两种典型的拣选工作站类型外，还有一些典型的工作站布局类型，此处便不一一展示。

客户在购买 RMS 的整体解决方案时，极智嘉的规划人员会根据用户的实际场地进行地图及工作站的整体设计，保证满足功能的前提下达到利用率和效率的最大化。

1.2.4.2 单点工作站

单点工作站（又称虚拟工作站），在地图中只占据一个货架位置，同一时间一个单点工作站只能容纳一个机器人及货架。

1.2.5 充电站

充电站是指地图上机器人可以充电的位置，同一时间一个充电站只能容纳一个机器人进行充电。

1.2.6 货箱到人场景

1.2.6.1 固定货架

一个单元格上一个固定货架（Rack，非实物货架，仅为 RMS 系统概念），不可移动，系统初始化后其属性不变，包含多个格子（Lattice），可用于前端展示。

1.2.6.2 格子

格子（Lattice），也称为货格、货位。一个固定货架上会有多个格子（目前的设计为：5 层，每层 1-2 列，一个货架上最多有 10 个格子），每个格子上只允许放置一个箱子。

1.2.6.3 货箱

一个货箱（Box）只能被一个机器人装载。

1.2.7 容器

RMS 中的货架及货箱都属于容器。容器是一个虚拟的概念，不一定是仓库中的一个实体。

1.2.8 其他

1.2.8.1 主机系统

给 RMS 发送具体任务消息的用户自定义应用程序（即 RMS 的上游程序系统）在本文中统一称之为主机系统。

1.2.8.2 业务场景

拣选：即货到人模式，机器人搬运装满不同类型的商品到标准工作站由工人按订单进行拣选配单。

搬运：即点到点的内物流模式，可以是在传统的工厂生产线进行空满交换，也可以是在仓库、商场、医院等场地内进行集货、缓存、运输等操作。

分拣：在快递物流行业的大型集散中心，对快递件按其目的地进行自动化分类集中。

1.3 RMS 自定义数据类型

除八种 JAVA 的基本类型及 String 类型外，部分数据类型为 RMS 自定义，主机系统对此类数据进行填写或解析时需要注意。

下方表格中列出了几种典型的 RMS 自定义数据，用户在实际填写或解析时还需参考文档中相关说明，或咨询极智嘉技术人员。

数据类型	含义	类型说明	示例
IPoint	整数型三维坐标索引，"z"表示楼层，"x"、"y"表示平面中的坐标索引	"x"，"y"，"z"三个参数均为 Integer	{"z":1,"x":45,"y":42}
DPoint	小数型三维实际坐标点，"z"表示楼层，"x"、"y"表示平面中的实际坐标	"z"为 Integer，其余为 Double	{"z": 1,"x": 29.5,"y": 10.5}
mapArea	整数型三维坐标索引集合，"z"表示楼层，"x"、"y"表示平面中的坐标索引	"x"，"y"，"z"三个参数均为 Integer	[{"z": 1,"x": 29,"y": 10}, {"z": 1,"x": 30,"y": 10}]
sortStation	分拣投递格口信息	见 3.11 章节	[{"cellCode": "04050345","cellNo": "102","allowDirs": [1,2,3], "available": true, "spare": false}]
TaskFailure	任务失败附加信息，目前只支持取货架时校验货架码失败时上报实际扫码值	附加信息为 String	{shelfCodeActuallyScanned":"00002"}

1.4 各章节简述

本文档涵盖了多种业务场景下 RMS API 所支持的所有功能， 不同的用户可以根据适应场景选择合适的功能。因此接下来的这一小节我们将简述每一章的主要内容，用户可根据需要及兴趣，直接阅读相应章节。

1.4.1 第 1 章 RMS API 简介

本章简述了 RMS 以及 RMS API 的基础概念。看完本章，用户应了解通过调用 RMS API 能够实现什么样的功能。

1.4.2 第 2 章 总体概述

本章对消息结构及使用方法进行介绍，并对 RMS 目前所支持的三种协议进行了说明。看完本章，用户应了解如何使用 API 发送消息请求。

1.4.3 第 3 章 任务与业务场景

本章对任务从请求、执行回调、结束的全流程进行了介绍，按照任务类型对 RMS 支持的所有任务进行了阐述，是 API 的核心部分。看完本章，用户应了解如何使用不同的任务请求去实现不同的目的。

1.4.4 第 4 章 仓库资源更新请求

本章对 RMS 仓库的各类资源更新请求进行了介绍，按照不同资源的参数进行了阐述，看完本章，用户应了解如何对仓库中的资源进行更新以达到资源的管控目的。

1.4.5 第 5 章 仓库控制请求

本章对仓库的各类控制指令进行了介绍，按照不同维度和对象进行了阐述，看完本章，用户应了解如何对仓库的运行进行干预控制以达到资源、流程管理控制的目的。

1.4.6 第 6 章 查询请求

本章对查询请求这种类型的消息进行了介绍。看完本章，用户应了解如何查询 RMS 所管理的各项资源的信息。

1.4.7 第 7 章 仓库事件通知回调

本章对仓库中某些事件发生后回调通知给主机系统的接口进行了介绍，看完本章，用户可有选择地关注某些事件回调以配合自身业务。

1.4.8 第 8 章 任务转移逻辑

本章对机器人因异常或是其他原因在 RMS 系统中被移除，RMS 对该机器人正在执行的任务的处理逻辑进行了介绍，看完本章，用户应了解自身业务怎么配合 RMS 系统中机器人移除的事件。

1.4.9 附录

本章为附录，用户可以在此查看 API 的响应码等信息详情。

2 总体概述

上一章我们介绍了 RMS 的相关基本概念，以及阐述了 RMS 的标准 API 是什么、能够做什么。

在本章中，我们将首先对消息结构及其使用方法进行了介绍，然后对 RMS 标准 API 支持的三种协议进行了说明，接下来对 RMS 的支持的不同类型的消息进行了简要介绍，最后以一个简单的任务请求示例来结束本章。

阅读完本章内容，用户应该能够了解如何使用 API 发送消息请求，并且进行相应的开发后，可以按照给出的任务示例来对 RMS 发起任务请求。

2.1 消息介绍

2.1.1 RMS-API 中的消息

主机系统和 RMS 之间的每次通信都被称之为一个消息，消息是 RMS 所定义的一组用于外部系统和 RMS 通信的数据结构，它包含消息类型、消息头、消息体等。

2.1.2 消息分类

根据消息的方向性，可以分为 6 种类型，如下表所示：

消息类型	功能描述	消息发送方向
请求消息	请求消息用于外部系统向 RMS 发送请求，它包含任务指令下发、任务指令更新、查询指令、仓库指令、参数更新指令等等。	主机系统 -> RMS
响应消息	响应消息是针对主机系统的一次请求，RMS 做出的回应。	RMS -> 主机系统
回调请求消息	RMS 会在任务的阶段、仓库资源、系统运行等状态发生变化时，向外部系统发送回调请求消息。	RMS -> 主机系统
回调应答消息	RMS 的回调消息是严格保证送达，因此主机系统收到回调消息后必须给与一个回调应答消息。	主机系统 -> RMS
心跳消息	只适用于主机系统基于 TCP SOCKET 与 RMS 建立连接的场景（其他协议忽略），用于保持长连接，具体见 2.4.1.2。	主机系统 <-> RMS
未知信息响应消息	当 RMS 收到的消息无法正确解析时，将给主机系统返回未知信息响应消息（主机系统需要技术人员介入检查消息的合法性）。	RMS -> 主机系统

2.2 消息结构

目前 RMS API 采用标准的 JSON 消息格式，不同分类的消息在结构上略有区别。

2.2.1 请求消息

当主机系统想使用 RMS 实现某些功能时，则需要给 RMS 发送一条请求消息。

例如当主机系统希望 RMS 调度机器人执行任务时，主机系统应当向 RMS 发送一个任务请求消息，RMS 收到后将会对此消息将进行合法性校验。校验通过后 RMS 将发送任务响应消息告知主机系统任务已接收，校验不合格时 RMS 也将发送任务响应消息告知主机系统任务请求消息有误。

又或者当主机系统希望了解某些资源的状态信息时，可以向 RMS 发送查询请求消息，RMS 收到后会将相应资源的状态信息以查询响应消息的形式返回给主机系统。

注意事项：发送请求消息，如果不传某些参数，则直接在请求体里面去掉改参数即可，而不允许传默认值，如字符串传空串("")，int 值传 0 等。

RMS 目前支持的请求消息类型有：

序号	值	类型	说明	RMS 应答时响应消息类型
1	RobotTaskRequestMsg	任务请求消息	主机系统通过此类型消息下发任务	RobotTaskResponseMsg
2	RobotTaskUpdateRequestMsg	任务更新消息	主机系统通过此类型消息更新任务指令	RobotTaskUpdateResponseMsg
3	QueryInstructionRequestMsg	查询消息	主机系统通过此类型消息查询 RMS 内部状态	QueryInstructionResponseMsg
4	WarehouseInstructionRequestMsg	仓库控制消息	主机系统通过此类型消息下发仓库指令	WarehouseInstructionResponseMsg
5	ParameterInstructionRequestMsg	参数设置消息	主机系统通过此类型消息参数设置	ParameterInstructionResponseMsg

请求消息格式如下所示：

```
{
   "id": "xxx",
   "msgType": "xxx",
```

```
  "request": {
    "header": {
      "channelId": "xxx",
      "requestId": "xxx",
      "clientCode": "xxx",
      "warehouseCode": "xxx",
      "userId": "xxx",
      "userKey": "xxx",
      "language": "xxx",
      "version": "xxx"
    },
    "body": {
      "field1": "xxx",
      "field2": "xxx"
    }
  }
}
```

字段说明：

序号	字段	功能描述
1	id	当前链接通道的唯一标识，必填。 不同通道需要保证 id 唯一，同一个通道则需要保证 id 一致。 用于标识消息的链路来源，用于在多客户端同时连接 RMS 时，RMS 根据此 ID 选择正确连接通道发送回调消息。 当用户有多个链接通道向 RMS 发送消息时，建议命名规则：clientCode_warehouseCode_001、clientCode_warehouseCode_002、clientCode_warehouseCode_003…。
2	msgType	当前的请求消息类型， 不同类型的请求消息涉及的消息正文（body）会不一样。RMS 支持的请求消息详见后文。
3	request	请求消息的消息内容，包括消息头和消息正文。
4	header	包含在“request”中，消息头。在请求消息中定义，它包含基本的验证信息字段。
5	requestId	包含在消息头中，用作对一次请求及其响应进行唯一标识。 该字段的值由主机系统定义，并且响应消息应具有相同的值。 RMS 需要根据此 ID 来保证幂等性，因此需要保证该 ID 的全局唯一，推荐使用 UUID。
6	clientCode	包含在消息头中，用户编码，是上游客户的一个标识。 此 ID 由 RMS 颁发给上游客户，在同一个项目服务中使用同一个 clientCode。
7	warehouseCode	包含在消息头中，仓库编码，是一个仓库在 RMS 中的唯一标识。 此 ID 由 RMS 颁发给上游客户，不同仓库需要保证 warehouseCode 唯一，同一个仓库则需要保证 warehouseCode 一致。
8	channelId	包含在消息头中，链路通道的标识，意义等同于 id。

序号	字段	功能描述
		注意：此字段仅供极智嘉内部 RPC 使用，非极智嘉的用户请忽略此字段，即使填写也不会生效。 后文的消息示例中也将隐藏此字段。
9	version	包含在消息头中，本次消息的版本信息，目前支持 3.3.0。
10	language	包含在消息头中，响应消息的语言设定。 设置此字段后，RMS 会根据这个字段选择返回响应消息的语言。 目前支持简体中文（zh_cn）、英文(en_us)、日文（ja_jp）、繁体中文（zh_hk/zh_tw）等，默认英文。
11	userId	包含在消息头中，用户认证 id（目前不起作用）
12	userKey	包含在消息头中，用户认证 key（目前不起作用）
13	body	包含在“request”中，消息正文及其包含的详细字段。 此部分内容将在后续章节中对每个不同的请求消息类型进行解释。

2.2.2 响应消息

针对主机系统的每一次有效的请求消息，RMS 都会发送一个响应消息作为应答。

响应消息的格式如下所示：

```
{
   "id": "xxx",
   "msgType": "xxx",
   "response": {
      "header": {
         "responseId": "xxx",
         "code": xxx,
         "msg": "xxx"
      },
      "body": {
         "field1": xxx,
         "field2":"xxx"
      }
   }
}
```

字段说明：

序号	字段	功能描述
1	id	值必须与响应消息所应答的请求消息中的 id 字段相同。
2	msgType	当前响应消息的消息类型，与请求消息一一对应。

序号	字段	功能描述
3	response	响应消息的消息正文，包括消息头和正文（如果需要）。
4	header	包含在“response”中，消息头。其中包含基本验证信息字段。
5	responseId	包含在消息头中，与当前响应消息对应的请求消息的 requestId 保持一致。
6	code	包含在消息头中，响应代码。详细的响应码及消息说明请见附录。
7	msg	包含在消息头中，响应消息说明。该值用来描述对请求消息的响应说明。
8	body	包含在“response”中，包含详细字段的消息正文。部分响应消息没有正文字段。

2.2.3 回调请求

当 RMS 开始调度某个机器人执行任务后，会在任务的不行执行阶段（例如机器人举起货架、机器人开始运送货架、机器人到达目的地等）向主机系统发送任务回调请求消息，告知主机系统此任务目前的执行状态。主机系统在收到此消息后，应当向 RMS 发送任务回调应答消息进行告知。另外 RMS 系统发生仓库资源发生变化时，也会向主机系统发送回调消息。

RMS 目前支持的回调请求消息类型有：

序号	值	类型	说明	主机系统应答时响应消息类型
1	RobotTaskCallbackMsg	任务回调请求消息	RMS 通过此类型消息通知主机系统任务状态变化	RobotTaskCallbackResponseMsg
2	WarehouseCallbackMsg	仓库事件回调消息	RMS 通过此类型消息通知主机系统仓库资源状态变化	WarehouseCallbackResponseMsg

回调请求消息格式如下：

```
{
   "id": "xxx",
   "msgType": "xxx",
   "request": {
       "header": {
            "warehouseCode": "xxx",
            "requestId": "xxx",
            "version": "xxx"
        },
       "body": {
            "field1": xxx,
            "field2": "xxx",
        }
    }
```

}

字段说明：

序号	字段	功能描述
1	id	当前链接通道的唯一标识，必填。
2	msgType	当前回调消息的消息类型。
3	request	响应消息的消息正文，包括消息头和正文字段（如果需要）。
4	header	包含在“request”中，消息头。其中包含基本验证信息字段。
5	warehouseCode	包含在消息头中，当前回调事件所产生的仓库。
6	requestId	包含在消息头中，当前回调消息的唯一标识。用于主机系统保证幂等性。
7	version	包含在消息头中，本次消息的版本信息，目前支持 3.3.0。
8	body	包含在“request”中，包含详细字段的消息正文。任务回调消息体说明见 3.1.7 章节。仓库事件回调消息体说明见第七章。

2.2.4 回调应答消息

主机系统的回调应答消息格式如下：

```
{
   "id": "xxx",
   "msgType": "xxxxx",
   "response": {
      "header": {
         "responseId": "xxx",
         "code": xxx,
         "msg": "xxx"
      }
   }
}
```

字段说明：

序号	字段	功能描述
1	id	此字段不为空即可。
2	msgType	当前回调应答消息的消息类型，与回调消息一一对应。
3	response	回调应答消息的消息正文，包括消息头和正文字段（如果需要）。
4	header	包含在“response”中，消息头，其中包含基本验证信息字段。

序号	字段	功能描述
5	responseId	包含在消息头中，当前回调应答消息的唯一标识。用于保证消息的幂等性，当前回调应答消息与对应的回调消息的 requestId 保持一致。
6	code	包含在消息头中，回调应答代码，默认填“0”。
7	msg	包含在消息头中，返回消息说明，默认填写“Success”。

2.2.5 心跳消息

心跳消息示例见 2.4.1.2 长连接部分。

2.2.6 未知信息响应消息

当主机系统发送给 RMS 的消息不能被正确解析时，RMS 将发送给未知信息响应消息给主机系统，表示 RMS 无法解析此消息，请用户检查请求。

主机系统收到此消息无需应答，此消息的“msgType”值为“UnparsableResponseMsg”。

消息示例：

```
{
   "id": "xxx",
   "msgType": "UnparsableResponseMsg",
   "response": {
      "header": {
         "responseId": "xxx",
         "code": 1001,
         "msg": "Illegal request message, parsing msg failed"
      }
   }
}
```

2.3 消息相关补充

2.3.1 请求原则

RMS 基于消息中的 requestId 来确定唯一一次消息请求，当 RMS 收到多次相同 requestId 的请求时，RMS 会回复完全一致的响应消息，即使两次请求的内容完全不一样，因此，主机系统在向 RMS 发送消息时，不同的消息一定要使用不一样的 requestId。

主机系统向 RMS 发送的任务请求、任务更新、仓库控制、查询请求等功能性请求消息时，RMS 会对这些请求消息进行严格的校验。若校验不合格，则 RMS 不会产生相应任务，并返回给主机系统一个包含错误代码和错误信息的响应消息，主机系统应根据 RMS 反馈的响应消息，更改自己的请求消息并重新发送，直到 RMS 返回表示成功接收的响应消息为止。

2.3.2 响应代码及消息说明

RMS 在接收到主机系统的请求消息之后，将会对请求消息中内容信息进行校验，校验通过后才会生成相关请求任务并且执行。

通常情况下，无论校验通过与否，RMS 都会在校验后立即向主机系统发送响应消息（不考虑网络延迟，响应时间大约是毫秒级别）。如果长时间没有收到 RMS 的响应（如 30 秒），主机系统应该考虑重新发送请求消息。

请求消息在 RMS 中校验通过后，RMS 给主机系统发送的响应代码为 0，响应消息说明为“Success”，说明 RMS 已经接受此请求。

请求消息在 RMS 中校验不合格时，RMS 给主机系统发送的响应消息中响应代码不为零且可能各不相同，每一种响应代码及消息说明都会指出当前请求消息无法通过校验的原因。客户应根据响应代码及消息说明修改主机系统的请求消息，然后再次发送。

因此主机系统收到来自 RMS 的响应消息后，需要验证响应消息中的“code”字段，以确定请求状态。0 表示请求成功，非零则表示请求失败。

响应代码及消息说明详见附录。

2.3.3 版本介绍

RMS 的标准 API 目前已经历了多个版本的迭代，目前的最新版 API 为“3.3.0”，此文档也是根据“3.3.0”版本的 API 进行撰写。

因此用户在参考本文档编写请求消息时，须使用“version”为“3.3.0”的字段参数。

2.4 协议

主机系统可以使用 TCP SOCKET、HTTP POST、WEBSOCKET 协议建立与 RMS 之间的连接。这三种协议同时只能使用一种。

2.4.1 SOCKET

当主机系统使用 TCP SOCKET 方法时，主机系统作为客户端，RMS 作为服务器。所有的请求消息、响应消息、回调消息、心跳消息都使用相同的 SOCKET 通道。 心跳消息详细介绍见 2.4.1.2。

RMS 使用***回车符加换行符("\r\n")***来分割数据包，也就是说主机系统发送给 RMS 的消息末尾需要使用分隔符"\r\n"，且消息体中不能包含回车换行符，所有 RMS 发送的消息末尾都会加上分隔符："\r\n"。

2.4.1.1 端口号

RMS 默认使用 8896 端口为 SOCKET 通信端口号。

2.4.1.2 长连接

对于 TCP SOCKET，作为服务器的 RMS 将主动检测 TCP 连接状态。如果发现客户端超过 30 秒没有向 RMS 发送任何消息，RMS 将主动断开与客户端的 TCP 连接。客户端需要判断当长时间未收到 RMS 发送的消息时也需要主动重连。

因此，当客户端长时间处于写空闲状态时，客户端需要通过当前 TCP 信道向 RMS 发送心跳消息。写空闲建议定义为 20 秒，换句话说，当客户端空闲超过 20 秒，它应该向 RMS 发送心跳消息。

RMS 不会对此心跳消息进行回复。同样的，RMS 会在写空闲超过 20 秒后向客户端发送心跳消息。

主机系统发送给 RMS 的心跳消息示例：

```
{
   "id": "clientid",
   "msgType": "com.geekplus.athena.api.msg.req.PingRequestMsg"
}
```

RMS 发送的心跳消息示例：

```
{
  "id": "clientid",
  "msgType": "com.geekplus.athena.api.msg.callback.PingRequestMsg"
}
```

2.4.2 HTTP

由于 HTTP 无法全双工通信，因此在 HTTP 协议下，RMS 和主机系统都需要为对方准备一个 HTTP 服务用于接收和处理消息。当主机系统发送请求消息时，主机系统充当客户端，RMS 作为服务器；当 RMS 发送回调消息时，主机系统充当服务器，RMS 作为客户端。

当使用 HTTP 协议时，主机系统必须使用 HTTP 请求中的 POST 方法访问 RMS，并且 HTTP 头中的 MIME 类型需要配置为“application/json”。同样，RMS 服务器使用“application/json”类型消息响应主机系统。响应消息在同一 HTTP 会话中返回给主机系统。

2.4.2.1 端口号

RMS 默认使用 8895 端口为 HTTP 的通信端口号。

2.4.2.2 长连接

当请求和响应通信完成后，HTTP 会话将不被保留，所以主机系统需要创建一个 HTTP 服务器来侦听请求消息并接受来自 RMS 的回调消息，同时给 RMS 回复回调响应消息。

2.4.3 **WEBSOCKET**

当使用 WEBSOCKET 协议时，主机系统作为客户端，RMS 作为服务器。所有的请求消息、响应消息、回调消息、心跳消息都使用相同的 WEBSOCKET 通道。

2.4.3.1 端口号

RMS 默认使用 8897 端口为 WEBSOCKET 通信的端口号。

2.4.3.2 长连接

由于 WEBSOCKET 协议本身有 keep live 的检测机制，因此业务系统无需发送心跳包。

3 任务与业务场景

RMS 通过任务使业务系统得以对 RMS 资源下发命令以实现场景需求。

业务场景指的是 RMS 处理的具体业务现场，是根据上游实际业务来确定的。RMS 支持的业务场景有：拣选、分拣、搬运等。

任务是在处理业务场景的具体执行单元，也就是说，不同场景的业务都是用 RMS 的特定类型任务去处理的。

对于上游业务系统来说，根据自身业务场景，通过接口给 RMS 系统发不同的任务，即可完成特定业务场景的业务需求。一种业务场景可包含多种任务，一种任务可存在于多个业务场景。

3.1 任务简介

任务是 RMS 系统处理业务的基本单元。RMS 系统也只会处理任务。上游业务系统需要将业务转化成若干个任务，按照实际业务需求，依次或者批量发给 RMS 系统，RMS 系统会处理一个个任务，从而完成上游业务的处理。

3.1.1 任务类型（TaskType）

不同的业务场景下，需要不同类型的任务来处理，目前 RMS 系统有如下几种任务类型。

任务类型	功能描述
GO_SOMEWHERE_TO_STAY	将指定的空闲机器人移动到指定位置。
DELIVER_SHELF_TO_STATION	该任务用于将货架送至标准工作站。 如果货架没有被任何机器人装载（顶起），RMS 将选择一个合适的机器人去取货架。如果货架被机器人装载，则将选择该机器人执行任务。 当货架被抬起后，机器人将货架送到指定工作站的排队区域，并排队到拣货位等待工人操作，直到工人完成操作，此过程中机器人不会放下货架。 工人操作完成后，主机系统应使用任务更新消息通知 RMS 将货架送回货架存储区（拣货完成后，无需其他操作）或将货架转面（如需要同一个货架的另一面）。 当货架被送回存储区并从机器人上卸载下来或机器人负载着该货架继续执行另一个任务时，当前任务将完成。 注意，一个任务仅支持一个工作站。如果想将货架送至多个工作站，则需要下发几个不同的任务。 特别注意，两个任务可以具有相同的货架编号（shelfCode）和工作站编号（stationId）。在这种情况下，只有机器人的下一个任务具有与当前任务相同的工作站编号或货架编号时，机器人才在在工人完成操作后继续将货架送到同一工作站。
DELIVER_SHELF	此任务用于将货架送至单点工作站或指定位置。 如果货架没有被任何机器人装载，RMS 将会选择一个合适的机器人去取货架。如果货架被机器人装载了，则将选择该机器人执行任务。 当货架被抬起之后，机器人将把货架送至指定工作站或指定位置，根据指令的不同，货架在指定位置保持顶升货架状态或是将货架卸载状态。详细说明见 3.4。
DELIVER_NEW_SHELF	此任务用于将货架送至单点工作站或指定位置，在拣选和搬运的业务场景中均可使用。与 DELIVER_SHELF 不同

任务类型	功能描述
	的是此任务不需要输入货架编号。只需要指定取货架的位置。机器人会自动扫描货架编码完成货架入场动作。当货架被抬起之后，机器人将把货架送至指定工作站或指定位置。此任务的前提是货架底部必须有二维码且现场机器人是具有向上扫码功能的。
GO_WORK	该任务类型适用于机器人一步一步执行一系列任务（所有这些任务在系统中只对应一个任务，任务 id 在第一次请求成功的时候会返回给上游），每次任务执行完之后，系统不会结束任务，而是等待上游系统继续发任务更新请求，直到任务更新请求通知系统结束任务，该任务才会在系统中置为完成状态。
CARRY_PACKAGE	该任务类型仅适用于辊筒机器人（有皮带或辊轮机构的用于包裹型货物的接收投递的机器人）。用于控制辊筒机器人的移动、到某个辊筒站点取包裹、去某个辊筒站点投递包裹。
DELIVER_PALLET	此任务表示叉车叉取托盘的任务，叉车去起点位置叉取托盘，然后到终点位置放下托盘
DELIVER_BOX	该任务仅适用于抱箱机器人（有抓取机构可将货架上面的货箱抓取并放置到自己身上）。该任务可以应用于以下多种场景： 调度机器人取一个货箱，并送到某个点或某个工作站，待工人完成拣货等相关操作后（也可以进行换箱、移除货箱等操作），再由机器人将此货箱送还。 调度一个空载的机器人到某点或某工作站，然后由工人放一个箱子上去，再将此箱子送回存储区。此操作会在 RMS 系统中添加新的货箱。 调度机器人取一个货箱，并将此货箱放置到指定的格子（即货位）上。
GO_WORK_ORDER_TO_PERSON	此任务类似于 GO_WORK，主机系统可下发多个目的点，RMS 会根据路径选择一个最优的点做为任务的终点。
MOVE_SHELF	内部使用，简单货架移动。
GO_MAINTAIN_TO_STAY	内部使用，去维保区维修。
GO_CHARGE_YOURSELF	内部使用，去充电。
GO_CIRCLE	内部使用，密集存储绕圈任务。
GO_TO_INSPECTION	内部使用，巡检地面及货架二维码信息。

3.1.2 任务指令（instruction）

任务指令指的是任务在执行过程中，需要 RMS 执行的业务逻辑上的操作。

注意，对于不同类型的任务，通常支持一种或几种任务指令，具体情况请参考后文对任务类型的详细介绍。

任务指令分类具体如下：

任务指令	具体业务含义
GO_FETCH	举起货架到某个地方，并保持举起状态
GO_TURN	机器人去某个转角度
GO_RETURN	运送货架到某个地方，并且放下改货架
CANCEL	取消任务
CANCEL_INSTRUCTION	取消动作，任务继续
GO_NEXT	机器人去某处，无其他动作，任务不结束
PRIORITY_SET	设置任务优先级
CLEAR_WAITPOINT	清除当前等待点
GO_RECEIVE	滚筒机器人收货
GO_DROP	滚筒机器人卸货
ALLOW_ENTER_ELEVATOR	允许进入电梯
ALLOW_LEAVE_ELEVATOR	允许离开电梯
UPDATE_PARCEL	分拣包裹任务更新
UPDATE_BOX	货箱信息的更新（仅适用于 DELIVER_BOX 任务类型更新指令）
FINISHED	结束任务

3.1.3 任务状态（taskStatus）

描述任务整个生命周期状态，状态分为 5 个，分别为：

任务状态	描述	是否发送回调
NEW	新任务。当 RMS 接收到任务时，它会缓存在机器人执行的任务队列中等待机器人执行，	否

任务状态	描述	是否发送回调
	此时这个任务状态为 NEW。	
ASSIGNED	已分配。当 RMS 将此任务分配了一个具体的机器人来进行执行时，则任务状态更改为 ASSIGNED。	否
EXECUTING	执行中。当机器人开始执行该任务，任务状态更改为 EXECUTING。	当任务状态为 EXECUTING 时，表示此任务正在执行中，RMS 将根据任务的执行阶段决定是否发送回调。
COMPLETED	已完成。当机器人完成该任务时，任务状态更改为 COMPLETED。	当任务从 EXECUTING 变更成 COMPLETED 时（同时也伴随着执行阶段的改变），这个时候 RMS 一定会主动发送回调消息给主机系统。
CANCELED	已取消。当一个任务被取消后，任务状态更改为 CANCELED。	任务状态更改为 CANCELED 时，这个时候 RMS 一定会主动发送回调消息给主机系统。

回调消息将在后文详细讲解。

注意，NEW，ASSIGNED，EXECUTING，COMPLETED 为任务正常执行流程，任务完成评判标准以 taskStatus 为 COMPLETED 为准。"taskStatus"回调具体过程信息以实际回调为准。

3.1.4 任务阶段（taskPhase）

描述任务执行的阶段，任务阶段描述的是机器人正在执行的任务的动作阶段。

任务阶段	描述
GO_FETCHING	正在取货架
SHELF_FETCHED	货架已经取到
GO_DELIVERING	正在搬运货架
QUEUING	正在排队
SHELF_ARRIVED	货架已经抵达目的地
GO_RETURN	货架归还中
SHELF_TURNING	货架旋转
MOVING	正在移动

任务阶段	描述
CHARGING	正在充电
ARRIVED_RECEIVE	滚筒机器人到达取货位，准备取货
ARRIVED_DROP	滚筒机器人到达卸货位，准备卸货
ARRIVED_WAIT_POINT	到达等待点
LEAVING_WAIT_POINT	离开等待点
RECEIVE_FINISH	滚筒机器人收货完成
DROP_FINISH	滚筒机器人卸货完成
BOX_FETCHED	取箱子完成
BOX_ARRIVED	送箱子到达
ARRIVED_ELEVATOR_ENTRY	到达电梯门口的等待点(电梯外)，等待进入电梯
ENTERED_ELEVATOR	已经进入电梯
LEAVED_ELEVATOR	离开电梯
ARRIVED	到达目的地
FETCHED	叉车取到托盘

3.1.5 任务生命周期中的消息

RMS 的一个任务在其生命周期中主要涉及到的消息交互有这些部分：任务请求消息及其响应、任务更新消息及其响应、任务回调消息及其响应。

主机系统向 RMS 发送任务请求消息，RMS 对请求做出响应（即向主机系统发送响应消息）。如果任务请求消息校验通过，RMS 会在响应消息的消息体中同时回复上游系统此任务的唯一标识（taskId）。

RMS 在收到正确的请求并做出响应后，会调度机器人执行任务。

此后，RMS 会根据唯一标识（taskId）向主机系统发送回调消息，主机系统也应做出响应回复 RMS，以此来完成一个完整的闭环操作。

主机系统若希望在任务未完成时对此任务进行更新，也可以使用唯一标识（taskId）来发送更新任务，同理主机系统应发送响应来回复 RMS。

下面将分别对其进行详细介绍。主机系统向 RMS 发送任务请求消息，RMS 对请求作出响应（即向主机系统发送响应消息）。

3.1.6 任务请求消息样式

任务请求消息的“msgType”必须为“RobotTaskRequestMsg”。

RMS 通过消息体中的“taskType”字段取值来区分不同的任务类型，不同的任务类型所需的消息体（“body”）内容也不一样。

请求消息样式如下：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskRequestMsg",
  "request": {
    "header": {
      "requestId": "9194E002AD324176974AE9438A011EE3",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "taskType": "GO_SOMEWHERE_TO_STAY",
      "field1": "xxx",
      "field2": "xxx"
    }
  }
}
```

3.1.7 任务的回调及其响应

当任务的执行状态或执行阶段发生改变，RMS 将发送一个任务回调消息（callback message）给主机系统，主机系统收到消息后需要回复一个响应消息去告知 RMS 已经收到该消息，至此一个回调消息的生命周期完成。如果没有在一定时间内（15 秒）内收到响应消息，RMS 可能会将此情况视为发送失败并重新发送回调消息。

回调消息实际上是 RMS 向主机系统的请求。

如前面章节所述，如果主机系统使用 HTTP 协议与 RMS 建立连接，则应该建立一个 HTTP 服务器来接收该消息，并且这是一个同步通信。

如果主机系统选择采用 TCP SOCKET、WEBSOCKET 协议，则充当服务器的 RMS 将在任务状态更改时直接将此消息发送给客户端。

3.1.7.1 回调消息样式

消息结构与其他消息无大差异，需注意此处的“msgType”的值一定为“RobotTaskCallbackMsg”。

回调消息样式如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "ca5ebc4251374e35a5f6afb2a52fd6dd",
         "version": "3.3.0"
      },
      "body": {
         "taskId": xxx,
         "taskType":"DELIVER_SHELF",
         "taskStatus":"EXECUTING",
         "taskPhase":"GO_DELIVERING",
         "field1": "xxx",
         "field2": "xxx"
      }
   }
}
```


3.1.7.2 产生任务回调的场景

在 RMS 中，有两个表示当前任务执行情况的字段“taskStatus”（任务状态）、“taskPhase”（执行阶段）。

当前任务在执行过程中完成某些特定动作后，例如：顶起货架、完成转面、进入或离开电梯等等，“taskPhase”（执行阶段）的值会发生改变。

而另一些特定动作如：机器人抵达目的地并放下货架，此时这两个字段的值都会发生改变。

每个任务在执行过程中均可能经历一个或几个执行阶段，而每种执行阶段也适用于多种或一种特定任务类型。

RMS 则会根据既定的业务逻辑将两个字段发生的改变以消息的形式发送给主机系统，这就是 RMS 的回调消息。

RMS 默认只会发送根据主机系统任务请求消息生成的任务产生的回调消息。

在实际环境中，RMS 会主动生成一些任务，例如：充电任务、去休息的任务；根据实际场景需要（如场地里面有自动门等），可配置是否发送这些任务回调消息给主机系统。

3.1.7.3 任务回调字段说明

任务回调消息的详细介绍会在后文介绍任务类型时进行描述。此处只介绍所有任务回调信息可能包含的字段信息。

任务回调消息可能包含的字段信息如下：

字段	类型	是否必要	描述	备注
taskId	Long	是	任务 id	
taskType	String	是	任务类型	
taskStatus	String	是	任务状态	
instruction	String	否	任务指令	
taskPhase	String	否	任务执行阶段	
robotId	Int	否	机器人 id	
stationId	Int	否	工作站 id	
chargerId	Int	否	充电站 id	
waitCellCode	String	否	当前暂停点的节点编号	
waitDir	Int	否	暂停点的机器人朝向	
nextCellCode	String	否	用于替代 waitDir，表示如果机器人的下一步是要去单元格	
dest	IPoint	否	终点索引	
destLocation	DPoint	否	终点坐标	
destCellCode	String	否	终点编号	
currentFloor	Int	否	当前楼层	
targetFloor	Int	否	目标楼层	
elevatorId	Int	否	电梯编号 id	
shelfCode	String	否	货架编码	
hostShelfCode	String	否	外部货架 code	
shelfSide	String	否	货架面	货架任务类型适用
parentTaskId	Long	否	任务 id	
destAreaId	Int	否	目标区域 ID	
destHostCode	String	否	外部终点 code	
hostStationCode	String	否	外部工作站 code	

字段	类型	是否必要	描述	备注
taskFailureCode	Int	否	任务失败 code	当任务失败时
taskFailure	TaskFailure	否	任务失败附加信息	当任务失败时
beltPos	Int	否	分拣机器人皮带编号	分拣场景适用
dropCellCode	String	否	投递点编码	
boxCode	String	否	货箱编号	货箱场景适用
latticeCode	String	否	格子编号	
liftId	Int	否	提升机编号	
startLatticeCode	String	否	取箱的货格位	
packageCount	Int	否	包裹数量	辊筒机器人等特殊场景适用
carryAmount	Int	否	此任务需要收取/投放的货物数量	
ext1	Int	否	透传字段 1，特殊业务场景下传给机器人使用	
ext2	String	否	透传字段 2，特殊业务场景下传给机器人使用	
startStationId	Int	否	起始工作站 id	
destList	List<DestList>	否	终点列表	订单到人场景适用，具体说明见 3.10.5.1 章节

3.1.7.4 通用任务回调说明

如果场地中存在自动门或是电梯等设备时，对于所有类型的任务，都有可能产生下表描述的任务阶段回调信息。主机系统可基于这些回调消息去控制自动门或是电梯。

任务阶段	说明
ARRIVED_WAIT_POINT	到达等待点，机器人会停下来等待主机系统清除等待点。
LEAVING_WAIT_POINT	离开等待点
ARRIVED_ELEVATOR_ENTRY	到达电梯门口的等待点(电梯外)，等待进入电梯

任务阶段	说明
ENTERED_ELEVATOR	已经进入电梯
LEAVED_ELEVATOR	已离开电梯

3.1.7.4.1 到达等待点（ARRIVED_WAIT_POINT）

消息示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
         "version": "3.3.0"
      },
      "body": {
         "taskId": xx,
         "taskType": "xxx",
         "taskStatus": "EXECUTING",
         "taskPhase": "ARRIVED_WAIT_POINT",
         "robotId": xxx,
         "waitCellCode": "xxxxx",
         "waitDir": xxxx,
         "nextCellCode": "xxxxx"
      }
   }
}
```


3.1.7.4.2 离开等待点（LEAVING_WAIT_POINT）

消息示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
         "version": "3.3.0"
      },
      "body": {
         "taskId": xx,
         "taskType": "xxx",
         "taskStatus": "EXECUTING",
         "taskPhase": " LEAVING_WAIT_POINT ",
         "robotId": xxx,
         "waitCellCode": "xxxxx"
      }
```

```
    }
}
```

3.1.7.4.3 到达电梯门口（ARRIVED_ELEVATOR_ENTRY）

消息示例如下：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWarehouseCode",
      "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
      "version": "3.3.0"
    },
    "body": {
    "taskId": xx,
    "taskType": "xxx",
      "taskStatus": "EXECUTING",
      "taskPhase": " ARRIVED_ELEVATOR_ENTRY ",
      "robotId": xxx,
      "currentFloor":1,
      "targetFloor": 2,
      "elevatorId":1
    }
  }
}
```

3.1.7.4.4 已进入电梯（ENTERED_ELEVATOR）

消息示例如下：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWarehouseCode",
      "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
      "version": "3.3.0"
    },
    "body": {
      "taskId": xx,
      "taskType": "xxx",
      "taskStatus": "EXECUTING",
      "taskPhase": "ENTERED_ELEVATOR",
      "robotId": xxx,
      "currentFloor": 1,
      "targetFloor": 2,
     "elevatorId": 1
    }
  }
}
```

3.1.7.4.5 离开电梯（LEAVED_ELEVATOR）

消息示例如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWarehouseCode",
      "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
      "version": "3.3.0"
    },
    "body": {
      "taskId": xx,
      "taskType": "xxx",
      "taskStatus": "EXECUTING",
      "taskPhase": "LEAVED_ELEVATOR",
      "robotId": xxx,
      "currentFloor": 2,
      "targetFloor": 2,
      "elevatorId": 1
    }
  }
}
```


3.1.8 任务更新及其响应

使用此消息类型，主机系统可以对尚未结束的任务内容进行更新，可以设置任务优先级、取消任务、清除暂停点，也可以对 DELIVER_SHELF_TO_STATION，DELIVER_SHELF、GO_WORK 等类型任务设置下一步执行指令。

主机系统向 RMS 发送任务更新请求（请求消息中必须包含“taskId”这个任务的唯一标识），RMS 对主机系统的请求做出响应。

当任务更新所产生的指令内容被 RMS 执行完毕后，RMS 同样会发送回调，主机系统也应对回调做出响应。

3.1.8.1 任务更新消息样式

任务更新消息结构与任务请求的消息结构相同，需要注意的是，此处“msgType”的值必须为“RobotTaskUpdateRequestMsg”。

区分更新指令的字段参数为“instruction”，且“taskId”不能为空。

请求消息样式如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskUpdateRequestMsg",
```

```
  "request": {
    "header": {
      "requestId": "411ccc219da94e7f8680dfdee39c43d5",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "taskId": xxx,
      "instruction": "xxx",
      "xxx": "xxx"
    }
  }
}
```

3.1.8.2 通用更新指令

通用更新指令是指在符合使用条件的情况下，所有类型的任务都可以配套使用。

指令	功能描述
PRIORITY_SET	设置已存在且尚未开始执行的任务的优先级。
CLEAR_WAITPOINT	清除暂停点。
ALLOW_ENTER_ELEVATOR	允许机器人进入电梯。
ALLOW_LEAVE_ELEVATOR	允许机器人离开电梯。

3.1.8.2.1 优先级设置（PRIORITY_SET）

优先级所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 PRIORITY_SET
taskId	long	必要	用来设置指定任务的优先级
priority	int	必要	优先级参数，0-99 区间值

请求消息示例：

```
{
 "id": "geekCode_geekWarehouseCode_001",
 "msgType": "RobotTaskUpdateRequestMsg",
 "request": {
```

```
 "header": {
  "requestId": "411ccc219da94e7f8680dfdee39c43d5",
  "clientCode": "geekCode",
  "warehouseCode": "geekWarehouseCode",
  "userId": "admin",
  "userKey": "123456",
  "language": "en_us",
  "version": "3.3.0"
 },
 "body": {
  "taskId": 12,
  "instruction": "PRIORITY_SET",
  "priority": 3
 }
}
}
```


响应消息示例：


```
{
 "id": "geekCode_geekWarehouseCode_001",
 "msgType": "RobotTaskUpdateResponseMsg",
 "response": {
  "header": {
   "responseId": "411ccc219da94e7f8680dfdee39c43d5",
   "code": 0,
   "msg": "Success"
  }
 }
}
```


3.1.8.2.2 清除暂停点（CLEAR_WAITPOINT）

清除暂停点所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	必须是 CLEAR_WAITPOINT
taskId	long	必要	当前进行的任务
waitCellCode	String	必要	清除指定的暂停点所在的单元格

请求消息示例：


```
{
 "id": "geekCode_geekWarehouseCode_001",
 "msgType": "RobotTaskUpdateRequestMsg",
 "request": {
  "header": {
   "requestId": "411ccc219da94e7f8680dfdee39c43d5",
   "clientCode": "geekCode",
   "warehouseCode": "geekWarehouseCode",
```

```
  "userId": "admin",
  "userKey": "123456",
  "language": "en_us",
  "version": "3.3.0"
 },
 "body": {
  "taskId": 12,
  "instruction": "CLEAR_WAITPOINT",
  "waitCellCode": "00350055"
 }
}
}
```


响应消息示例：


```
{
 "id": "geekCode_geekWarehouseCode_001",
 "msgType": "RobotTaskUpdateResponseMsg",
 "response": {
  "header": {
   "responseId": "411ccc219da94e7f8680dfdee39c43d5",
   "code": 0,
   "msg": "Success"
  }
 }
}
```


3.1.8.2.3 允许进入电梯（ALLOW_ENTER_ELEVATOR）

允许进入电梯所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	必须是 ALLOW_ENTER_ELEVATOR
taskId	long	必要	当前进行的任务 id

请求消息示例：


```
{
 "id": "geekCode_geekWarehouseCode_001",
 "msgType": "RobotTaskUpdateRequestMsg",
 "request": {
  "header": {
   "requestId": "c3defd52-6214-44a0-900c-7447ad9b27b0",
   "clientCode": "geekCode",
   "warehouseCode": "geekWarehouseCode",
   "userId": "admin",
   "userKey": "123456",
   "language": "en_us",
   "version": "3.3.0"
  },
  "body": {
```

```
    "taskId": 445,
     "instruction": "ALLOW_ENTER_ELEVATOR"
  }
 }
}
```


响应消息示例：


```
{
 "id": "geekCode_geekWarehouseCode_001",
 "msgType": "RobotTaskUpdateResponseMsg",
 "response": {
  "header": {
   "responseId": "c3defd52-6214-44a0-900c-7447ad9b27b0",
   "code": 0,
   "msg": "Success"
  }
 }
}
```


3.1.8.2.4 允许离开电梯（ALLOW_LEAVE_ELEVATOR）

允许离开电梯所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 ALLOW_LEAVE_ELEVATOR
taskId	long	必要	当前进行的任务 id

请求消息示例：


```
{
 "id": "geekCode_geekWarehouseCode_001",
 "msgType": "RobotTaskUpdateRequestMsg",
 "request": {
  "header": {
   "requestId": "0a207260-a0dc-40b0-9a92-79eab0513d87",
   "clientCode": "geekCode",
   "warehouseCode": "geekWarehouseCode",
   "userId": "admin",
   "userKey": "123456",
   "language": "en_us",
   "version": "3.3.0"
  },
  "body": {
     "taskId": 445,
      "instruction": "ALLOW_LEAVE_ELEVATOR"
  }
 }
}
```


响应消息示例：


```
{
```

```
"id": "geekCode_geekWarehouseCode_001",
"msgType": "RobotTaskUpdateResponseMsg",
"response": {
 "header": {
  "responseId": "0a207260-a0dc-40b0-9a92-79eab0513d87",
  "code": 0,
  "msg": "Success"
 }
}
}
```


3.1.8.3 专用更新指令

专用更新指令是指对某一些特定类型的任务（taskType）进行设置下一步的执行指令。此处不做详解介绍，在后文介绍每一种任务类型时详细介绍其更新指令。

3.2 GO_SOMEWHERE_TO_STAY

此任务类型适用于所有场景下的所有类型机器人，其作用是将指定的空闲机器人移动到指定位置。

3.2.1 场景说明

此任务类型适用于所有场景下的所有类型机器人。

3.2.2 可用指令

“GO_SOMEWHERE_TO_STAY”无可用指令。

3.2.3 任务下发

机器人到达指定地点后此任务直接结束。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskType	String	必要	任务类型。必须为“GO_SOMEWHERE_TO_STAY”	
robotId	Integer	必要	指定执行任务的机器人 ID	1000
dest	IPoint	三选一，至少填写一个值	终点坐标。三维坐标，即 z 轴，x 轴，y 轴。	{"z":1,"x":45,"y":42}
destCellCode	String		目的地单元格编码。	02350125

字段	类型	是否必要	描述	示例
destHostCode	String		主机系统对机器人目的点的描述。只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekDest01
priority	Integer	可选	该参数用来设置任务的优先级。值越大，表明在分配机器人执行任务期间该任务的优先级越高，priority 取值范围[0-99]	10

注意，若希望到某个点，若“dest”、“destCellCode”、“destHostCode”三个字段同时填写，则指代必须一致。

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskRequestMsg",
   "request": {
      "header": {
         "requestId": "9194E002AD324176974AE9438A011EE3",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskType": "GO_SOMEWHERE_TO_STAY",
         "robotId": 1000,
         "dest": {
            "z": 1,
            "x": 32,
            "y": 10
         }
      }
   }
}
```

此报文表示主机系统希望 RMS 调度“1000”号机器人去{"z":1,"x":32,"y":10}这个点。

RMS 的响应示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskResponseMsg",
   "response": {
      "header": {
         "responseId": "9194E002AD324176974AE9438A011EE3",
         "code": 0,
         "msg": "Success"
```

```
        },
        "body": {
          "taskId": 38
        }
      }
    }
```


3.2.4 任务更新

“GO_SOMEWHERE_TO_STAY”不支持任务更新。

3.2.5 任务回调

机器人在执行“GO_SOMEWHERE_TO_STAY”任务时，根据当前执行阶段的不同，RMS 会主动向主机系统发送回调消息，告知主机系统此任务目前的任务状态及执行阶段。

以下为“GO_SOMEWHERE_TO_STAY”会出现的任务执行阶段“taskPhase”。

taskPhase	发送时机
MOVING	机器人开始运动去某地
ARRIVED	机器人到达某地

以“MOVING”为例，RMS 主动发送的回调消息示例如下：


```
{
  "id": "*",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "DEFAULT",
      "requestId": "66f1d0b5b84141baa95c7c98b3ff0d5c",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 39,
      "taskType": "GO_SOMEWHERE_TO_STAY",
      "taskStatus": "EXECUTING",
      "taskPhase": "MOVING",
      "robotId": 1000,
      "dest": {
        "z": 1,
        "x": 6,
        "y": 7
      },
      "destLocation": {
        "z": 1,
        "x": 6.825,
        "y": 7.5
```

```
            },
            "destCellCode": "00650075"
        }
    }
}
```

主机系统应该给与 RMS 的响应如下：

```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskCallbackResponseMsg",
    "response": {
        "header": {
            "responseId": "66f1d0b5b84141baa95c7c98b3ff0d5c"
        }
    }
}
```

3.2.6 功能注意点

无。

3.3 GO_WORK

此任务用于指派机器人本体进行单纯移动，常用于机器人本体移动、复合型机械臂机器人等场景。

3.3.1 场景说明

3.3.1.1 场景概括

该任务为指派机器人本体去指定地点，无其他行为操作。

3.3.1.2 场景要点

需要注意任务为单纯的机器人移动，无法实现其他功能。目的地若为工作站，指向节点或工作站均能产生工作站排队功能。

3.3.2 可用指令

GO_WORK 支持的指令如下，具体用法会在下面的部分进行详细叙述。

行为	指令	用途
任务下发	GO_NEXT	指派机器人本体去某处

行为	指令	用途
任务更新	GO_NEXT	指派机器人本体去某处
任务更新	CANCEL	任务取消
任务更新	CANCEL_INSTRUCTION	动作取消

3.3.3 任务下发

在任务下发时可用指令“GO_NEXT”，下面将进行具体说明。任务下发后 RMS 会接收并生成一个任务，将任务号回复给主机系统。

3.3.3.1 GO_NEXT

当主机系统命令 RMS 指定机器人本体运动到目的地时使用此指令。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
任务主体字段	该部分为任务的主体信息字段			
taskType	String	必要	任务类型。为 GO_WORK	GO_WORK
instruction	String	必要	GO_NEXT	GO_NEXT
isContinue	Integer	可选	0，指令执行结束后任务结束。1，指令执行结束后任务不结束。为空视为 1，即 GO_NEXT 指令默认会继续任务。	1
机器人控制字段	该部分用于控制任务机器人。 单个、集合均填写时取并集，允许指定具体机器人执行任务，不指定时系统会调度最近的机器人。注：强制指定机器人的不可用时可能会使任务失败。			
robotId	Integer	可选	指定某个机器人执行任务	1
robotIds	list of Integer	可选	指定一个机器人集合去执行，系统会在其中选中一个距离最近符合条件的机器人去执行	[1,2,3]
robotAllocationStrategy	Integer	可选	0：优先，指定的机器人为优先指定，若机器人不可用会下发给其他机器人执行； 1：强制，必须为指定的机器人执行。	0

字段	类型	是否必要	描述	示例
			空默认为 0	
目的地控制字段	单个和集合如果同时传取并集。若为多个目的地，系统会根据目的地模式的远近选取最近的节点前往。若相同描述指代的目的地均填写了，则指代地点必须一致，否则报错。从 destCellCode 到 stationId 至少需要有一个值。			
destCellCode	String	可选	目的节点编码	1
destCellCodes	list of String		目的节点编码集合	["1","2","3"]
dest	IPoint		目的地索引坐标。即 z 轴，x 轴，y 轴。z 可省略,默认是 1(二维码区域专用,传索引值)	{"x":45,"y":42}, {"z":1,"x":45,"y":42}
dests	list of IPoint		节点索引集合，单个和集合同时传输时取并集	[{"z":1,"x":45,"y":42}, {"z":2,"x":45,"y":42}]
destHostCode	String		上位机系统关于当前任务的目的地坐标的描述	1
destHostCodes	list of String		上位机系统关于当前任务的目的地坐标的描述集合	[“1”,”2”,”3”]
stationId	Integer		工作站编号	2
hostStationCode	String		主机系统对工作站的编号的描述	2
其余控制字段	可根据需要进行选填			
priority	Integer	可选	该参数用来设置任务的优先级。值越大，表明在分配机器人执行任务期间该任务的优先级越高，priority 取值范围[0-99]	2

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskRequestMsg",
   "request": {
      "header": {
         "requestId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
```

```
        },
        "body": {
            "taskType": "GO_WORK",
            "instruction": "GO_NEXT",
            "destCellCode":"600045"
        }
    }
}
```

RMS 的响应示例如下：

```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskResponseMsg",
    "response": {
        "header": {
            "responseId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
            "code": 0,
            "msg": "Success"
        },
        "body": {
            "taskId": 30
        }
    }
}
```

3.3.4 任务更新

任务更新是指在任务 instruction 置为 READY 时进行的正常更新，使任务执行下一指令，同时也包括 CANCEL 任务取消和动作取消 CANCEL_INSTRUCTION。

1、任务取消为取消尝试，当任务即将完成时，任务正常结束而非被取消；

2、动作取消为取消当前任务正在执行的指令，将任务 instruction 重置为 READY。

RMS 对于主机系统的更新均会进行标准响应：

```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskUpdateResponseMsg",
    "response": {
        "header": {
            "responseId": "9beab88d-a407-4195-a1f8-ae3594c83924",
            "code": 0,
            "msg": "Success"
        }
     }
}
```

3.3.4.1 GO_NEXT

指派机器人本体去某处，以下情形可使用该指令：

1、机器人为本体状态，可使用该指令指派机器人去某处；

2、使用该指令时任务可以结束。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
任务主体信息	该部分将描述任务的主体信息部分			
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	GO_NEXT	GO_NEXT
isContinue	Integer	可选	0，指令执行结束任务结束。1，指令执行结束任务不结束。为空视为 1。 1、GO_NEXT 指令默认任务会继续请注意 2、若任务不继续，则目的地为空时会原地结束任务	1
目的地控制字段	单个和集合如果同时存在取并集。若为多个目的地，系统会根据目的地模式的远近选取最近的节点前往。若相同描述的目的地均填写了，则指代地点必须一致，否则报错。从 destCellCode 到 stationId 至少需要有一个值。			
destCellCode	String	可选	目的节点编码	1
destCellCodes	list of String		目的节点编码集合	["1","2","3"]
dest	IPoint		目的地索引坐标。即 z 轴，x 轴，y 轴。z 可省略,默认是 1(二维码区域专用,传索引值)	{"x":45,"y":42} {"z":1,"x":45,"y":42}
dests	list of IPoint		节点索引集合，单个和集合同时传输时取并集	[{"z":1,"x":45,"y":42}, {"z":2,"x":45,"y":42}]
destHostCode	String		上位机系统关于当前任务的目的地坐标的描述	1
destHostCodes	list of String		上位机系统关于当前任务的目的地坐标的描述集合	[“1”,”2”,”3”]
stationId	Integer		工作站编号	2
hostStationCode	String		主机系统对工作站的编号的描述	2

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
```

```
    "request": {
        "header": {
            "requestId": "9beab88d-a407-4195-a1f8-ae3594c83924",
            "clientCode": "geekCode",
            "warehouseCode": "geekWarehouseCode",
            "userId": "admin",
            "userKey": "123456",
            "language": "en_us",
            "version": "3.3.0"
        },
        "body": {
            "taskId": 30,
            "instruction": "GO_NEXT",
            "destCellCode": "655544"
        }
    }
}
```


3.3.4.2 CANCEL

当任务未结束或取消前，可使用 CANCEL 指令对任务进行取消，**任务取消是一种尝试**，任务在执行前为直接取消，执行中时：

1、任务即将完成则取消不起作用，任务正常完成；

2、若取消成功，系统会执行取消动作；

3、任务在举升货架、转面等不可打断的动作时不会处理取消，而会延后执行；

4、取消时不能被再次任务取消和动作取消，任务仅能被取消一次。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	用来取消指定任务的 id	12
instruction	String	必要	值必须是 CANCEL	CANCEL
cancelAction	String	可选	可选，0,2,3 以机器人实际状态为准进行处理，参照下方任务的处理模式： 参数 0（默认值）：默认处理 空载：原地释放 负载：若有老家位置，返回老家，卸载；若无，原地卸载 参数 2：原地处理 空载：原地释放 负载：原地卸载 参数 3：指定目的地处理 空载：原地释放	0

字段	类型	是否必要	描述	示例
			负载：到达目的地后停止，卸载，释放	
指定目的地描述	该部分用于任务取消时且取消动作为 3 指定目的地时使用，若选用取消动作 3 则下方至少填写一个，若均填写，则填写的值指代必须一致			
dest	IPoint	可选	终点索引坐标。三维坐标，即 z 轴，x 轴，y 轴	{"z":1,"x":45,"y":42}
destCellCode	String	可选	目的地单元格编码	202350125
destHostCode	String	可选	主机系统对机器人目的点的描述	geekDest

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "411ccc219da94e7f8680dfdee39c43d5",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 10000002,
         "instruction": "CANCEL"
      }
   }
}
```

3.3.4.3 CANCEL_INSTRUCTION

当任务在执行过程中时，可使用 CANCEL_INSTRUCTION 指令对任务当前动作进行中断：

1、任务在即将完成结束时（当前路径前进点为任务终点）取消动作不起作用，任务正常完成；

2、任务当前指令即将完成结束时（当前路径前进点为任务终点）取消动作不起作用，任务指令正常进入 READY 等待更新；

3、若取消成功，系统指令会进入 READY 等待更新；

4、任务在举升货架、转面等不可打断的动作时不会处理取消，而会延后执行；

5、任务在完成前或接收取消任务指令前可多次执行 CANCEL_INSTRUCTION 指令。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	用来取消指定任务的 id	12
instruction	String	必要	值必须是 CANCEL_INSTRUCTION	CANCEL_INSTRUCTION

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "411ccc219da94e7f8680dfdee39c43d5",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 10000002,
         "instruction": "CANCEL_INSTRUCTION"
      }
   }
}
```

3.3.5 任务回调

机器人在执行“GO_WORK”任务时，根据当前执行阶段的不同，RMS 会主动向主机系统发送回调消息，告知主机系统此任务目前的任务状态及执行阶段。

以下为“GO_WORK”可能出现的任务执行阶段“taskPhase”。

taskPhase	发送时机说明
MOVING	机器人本体运动去某地
ARRIVED	机器人本体运动到达某地

GO_WORK 任务类型，RMS 主动发送的回调消息字段如下：

字段	类型	描述	示例
taskId	long	当前进行的任务 id	30

字段	类型	描述	示例
taskType	String	产生该回调的任务类型	GO_WORK
taskStatus	String	EXECUTING 执行中 CANCELED 取消 COMPLETED 完成	EXECUTING
instruction	String	表示任务在执行哪个指令时发起的回调。 若为 CANCEL 表示任务在执行任务取消动作过程中； 若为 READY 表示任务已完成当前动作正在准备接受更新。	GO_NEXT
taskPhase	String	回调的阶段 严格按照当前正在执行的动作阶段进行回调，若正在执行 MOVING 被取消动作则仍然会回调 MOVING	MOVING
robotId	int	执行任务的机器人 ID	1
destLocation	DPoint	终点：绝对坐标	{"z": 1,"x": 29.5,"y": 10.5}
dest	IPoint	终点：索引坐标	{"z": 1,"x": 29,"y": 10}
destCellCode	String	终点：节点编码	102950105
destHostCode	String	终点：节点外部编码	546544
stationId	int	终点工作站 注：指令终点为工作站才会有此值	1
hostStationCode	String	终点工作站外部编码 注：指令终点为工作站才会有此值	station1

回调的报文示例如下，不同阶段信息的回调将附在任务回调信息之后。


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "8780c902ed714d10b82a2f20ac514a1c",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 37,
         "taskType": "GO_WORK",
         "taskStatus": "EXCUTING",
```

```
            "instruction": "GO_NEXT",
            "taskPhase": "MOVING",
            "robotId": 1000,
            "destLocation":{"z": 1,"x": 29.5,"y": 10.5},
            "dest":{"z": 1,"x": 29,"y": 10},
           "destCellCode":"102950105",
           "destHostCode":"45464",
           "stationId":"1"
        }
    }
}
```


对于 RMS 回调的所有消息，主机系统均需要进行响应。

主机系统应该给与 RMS 的响应如下：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskCallbackResponseMsg",
    "response": {
        "header": {
            "responseId": "8780c902ed714d10b82a2f20ac514a1c"
        }
    }
}
```


以下将给出 GO_WORK 任务不同阶段的回调消息示例。

3.3.5.1 MOVING

MOVING 阶段的回调消息示例如下：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskCallbackMsg",
    "request": {
        "header": {
            "warehouseCode": "geekWarehouseCode",
            "requestId": "8780c902ed714d10b82a2f20ac514a11",
            "version": "3.3.0"
        },
        "body": {
            "taskId": 37,
            "taskType": "GO_WORK",
            "taskStatus": "EXCUTING",
            "instruction": "GO_NEXT",
            "taskPhase": "MOVING",
            "robotId": 1000,
            "destLocation":{"z": 1,"x": 29.5,"y": 10.5},
            "dest":{"z": 1,"x": 29,"y": 10},
           "destCellCode":"102950105",
           "destHostCode":"45464",
           "stationId":"1"
```

```
        }
    }
}
```


3.3.5.2 ARRIVED

ARRIVED 阶段的回调消息示例如下：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskCallbackMsg",
    "request": {
        "header": {
            "warehouseCode": "geekWarehouseCode",
            "requestId": "8780c902ed714d10b82a2f20ac514a1c",
            "version": "3.3.0"
        },
        "body": {
            "taskId": 37,
            "taskType": "GO_WORK",
            "taskStatus": "EXCUTING",
            "instruction": "READY",
            "taskPhase": "ARRIVED",
            "robotId": 1000,
            "destLocation":{"z": 1,"x": 29.5,"y": 10.5},
            "dest":{"z": 1,"x": 29,"y": 10},
           "destCellCode":"102950105",
           "destHostCode":"45464",
           "stationId":"1"
        }
    }
}
```


3.3.5.3 CANCEL 相关回调

GO_WORK 任务被取消时：

1、未执行前会直接被取消，产生 taskStatus 为 CANCELED 的回调；

2、若执行中被正常取消，会在当前动作结束，开始执行取消动作时回调 taskStatus：EXECUTING，instruction：CANCEL，taskPhase 为具体的动作阶段，直到执行结束所有动作，会产生回调 taskStatus：CANCELED，此时表示任务正式结束；

3、若任务即将完成时被取消，任务不会执行取消，会正常产生 taskStatus 为 COMPLETED 的回调。

3.3.5.4 CANCEL_INSTRUCTION 相关回调

GO_WORK 被取消动作时：

1、未执行前、结束后不能被取消，接口报错；

2、若执行中被取消动作，当前动作结束后会产生一个 instruction：READY 回调，taskPhase 为当前正在执行的动作阶段而非正常结束的回调阶段，表示任务已完成动作取消准备好继续更新；

3、若动作即将完成前被取消动作，该取消可能执行失败，当前动作结束后正常产生阶段回调，instruction：READY，taskPhase 为正常动作完成的回调阶段。

3.3.6 功能注意点

GO_WORK 命令为指派机器人不断转移位置的一种指令，可用于所有类型机器人。

3.4 DELIVER_SHELF

此任务用于将货架送至单点工作站或指定位置。

3.4.1 场景说明

3.4.1.1 场景概括

该任务为典型的货架搬运任务，如果货架没有被任何机器人装载，RMS 将会选择一个合适的机器人去取货架。如果货架被机器人装载了，则将选择该机器人执行任务。当货架被抬起之后，机器人将把货架送至指定工作站或指定位置，根据指令的不同，货架在指定位置保持顶升货架状态或是将货架卸载状态。

3.4.1.2 场景要点

1、该任务类型额外具有一些控制参数用于不同场景，需要按需发送；

2、目的地若为工作站，指向节点或工作站均能产生工作站排队功能。

3.4.2 可用指令

DELIVER_SHELF 支持的指令如下，具体用法会在下面的部分进行详细叙述。

行为	指令	用途
任务下发	GO_FETCH	指派机器人搬运货架去某处后保持负载

行为	指令	用途
任务下发	GO_RETURN	指派机器人搬运货架去某处后卸载
任务下发	GO_NEXT	指派机器人本体去某处
任务更新	GO_FETCH	机器人搬运货架去某处后保持负载
任务更新	GO_RETURN	机器人搬运货架去某处后卸载
任务更新	GO_NEXT	指派机器人本体去某处
任务更新	CANCEL	任务取消
任务更新	CANCEL_INSTRUCTION	动作取消

3.4.3 任务下发

在任务下发时可用指令“GO_NEXT”、“GO_FETCH”、“GO_RETURN”，下面将进行具体说明。

任务下发后 RMS 会接收并生成一个任务，将任务号回调给对方。

RMS 的响应示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskResponseMsg",
   "response": {
      "header": {
         "responseId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
         "code": 0,
         "msg": "Success"
      },
      "body": {
         "taskId": 30
      }
   }
}
```

3.4.3.1 GO_FETCH

当主机系统命令 RMS 搬运指定货架到目的地保持负载时使用此指令。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
任务主体字段	该部分为任务的主体信息字段			
taskType	String	必要	任务类型。为 DELIVER_SHELF	DELIVER_SHELF
instruction	String	必要	GO_FETCH	GO_FETCH
isContinue	Int	可选	0，指令执行结束后任务结束。1，指令执行结束后任务不结束。 GO_FETCH 该处为空视为 1。当填写 0 时，任务将无法执行而异常	
机器人控制字段	该部分用于控制任务机器人。 单个、集合均填写时取并集，允许指定具体机器人执行任务，不指定时系统会调度最近的机器人。注：强制指定的机器人不可用时可能会使任务失败。			
robotId	Int	可选	指定某个机器人执行任务	1
robotIds	list of Int	可选	指定一个机器人集合去执行，系统会在其中选中一个距离最近符合条件的机器人去执行	[1,2,3]
robotAllocationStrategy	Int	可选	0：优先，指定的机器人为优先指定，若机器人不可用时会下发给其他机器人执行；1：强制，必须为指定的机器人执行。 空默认为 0	0
目的地控制字段	系统会按照以下优先级选择目的地，目的地 > 工作站，所有的值都会参与校验，所有的参数可同时传值。若多种目的地均填写了，则指代地点必须一致，否则报错。如未指定目的点，则目的点默认为货架老家位置，货架不存在老家位置则接口返回异常。			
destCellCode	String	可选	目的地：节点编码	1

字段	类型	是否必要	描述	示例
dest	IPoint	可选	目的地：索引坐标。即 z 轴，x 轴，y 轴。z 可省略,默认是 1(二维码区域专用，传索引值)	{"x":45,"y":42}, {"z":1,"x":45,"y":42}
destHostCode	String	可选	目的地：上位机系统关于当前任务目的地的描述	1
stationId	Int	可选	工作站：编号	2
hostStationCode	String	可选	工作站：主机系统对工作站的编号的描述	2
货架信息控制字段	该部分将描述货架相关的控制字段，shelfCode，hostShelfCode 至少需填写一个，若都填写，指代货架必须一致			
shelfCode	String	可选	货架编码。 此字段表示货架在 RMS 中的内部编码。	A000001
hostShelfCode	String	可选	货架在主机系统中的外部编码。 只有主机系统使用自有的货架编码并将编码同步到 RMS 才有效。	geekShelfCode0001
shelfScore	Int	可选	货架热度，取值范围[1-100]，默认为空。当该任务该执行时货架分数会被该值更新。不填写不更新。	1
neededSides	list of String	可选	货架面。标识工作站工人需要操作的面，可填值为 F、B、L、R（即前、后、左、右）。 指定多个面时系统会选则一个不需要转面的面到达工作站，可减少主机系统需要该货架多面时的转面频率。 系统计算逻辑为货架坐标与工作站的工作方向进行计算后获得货架实际需要转的角度生成转面任务。 不填写表示不转面。	["F","B"]
shelfTurnAngle	Int	可选	货架指定角度旋转，只能是 90 的倍数。例如：90,180,270,360 。这 4 个角度，neededSides 和 shelfTurnAngle 只能使用一个，两个不可同时使用。 该角度为旋转的相对角度。	90

字段	类型	是否必要	描述	示例
shelfTurnDirection	Int	可选	货架旋转方向: 0 代表顺时针，1 代表逆时针，默认 0。 该参数配合 shelfTurnAngle 使用。	0
其余控制字段	可根据需要进行选填			
priority	Int	可选	该参数用来设置任务的优先级。值越大，表明在分配机器人执行任务期间该任务的优先级越高，priority 取值范围[0-99]。	2

请求报文示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskRequestMsg",
   "request": {
      "header": {
         "requestId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskType": "DELIVER_SHELF",
         "shelfCode": "A00001",
         "instruction": "GO_FETCH",
         "destCellCode":"600045"
      }
   }
}
```


3.4.3.2 GO_RETURN

当主机系统命令 RMS 搬运指定货架到目的地进行卸载时使用此指令。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
任务主体字段	该部分为任务的主体信息字段			
taskType	String	必要	任务类型。为 DELIVER_SHELF	DELIVER_SHELF
instruction	String	必要	GO_RETURN	GO_RETURN
isContinue	Integer	可选	0，指令执行结束后任务结束。 1，指令执行结束后任务不结束。为空视为 0。	1
机器人控制字段	该部分用于控制任务机器人。 单个、集合均填写时取并集，允许指定具体机器人执行任务，不指定时系统会调度最近的机器人。注：强制指定且机器人不可用时可能会使任务失败。			
robotId	int	可选	指定某个机器人执行任务	1
robotIds	list of int	可选	指定一个机器人集合去执行，	[1,2,3]

字段	类型	是否必要	描述	示例
			系统会在其中选中一个距离最近符合条件的机器人去执行	
robotAllocationStrategy	int	可选	0：优先，指定的机器人为优先指定，若指定机器人不可用会下发给其他机器人执行 1：强制，必须为指定的机器人执行 空默认为 0	0
目的地控制字段	系统会按照以下优先级选择目的地，目的地 > 工作站，所有的值都会参与校验，所有的参数可同时传值。若多种目的地均填写了，则指代地点必须一致，否则报错。如未指定目的点，则目的点默认为货架老家位置，货架不存在老家位置则接口返回异常。			
destCellCode	String	可选	目的地：节点编码	1
dest	IPoint	可选	目的地:索引坐标。即 z 轴，x 轴，y 轴。z 可省略,默认是 1(二维码区域专用,传索引值)	{"x":45,"y":42}, {"z":1,"x":45,"y":42}
destHostCode	String	可选	目的地：上位机系统关于当前任务的目的地坐标的描述	1
stationId	Integer	可选	工作站：编号	2
hostStationCode	String	可选	工作站：主机系统对工作站的编号的描述	2
货架信息控制字段	该部分将描述货架相关的控制字段，shelfCode，hostShelfCode 至少需填写一个，若都填写，指代货架必须一致			
shelfCode	String	可选	货架编码。 此字段表示货架在 RMS 中的内部编码。	A000001
hostShelfCode	String	可选	货架在主机系统中的外部编码。只有主机系统使用自有的货架编码并将编码同步到 RMS 才有效。	geekShelfCode0001
shelfScore	int	可选	货架热度，取值范围[1-100]，默认为空。当该任务该执行时货架分数会被该值更新。不填写不更新。	1
neededSides	list of	可选	货架面。标识工作站工人需要	["F","B"]

字段	类型	是否必要	描述	示例
	String		操作的面，可填值为 F、B、L、R（即前、后、左、右）。指定多个面时系统会选则一个不需要转面的面到达工作站，可减少主机系统需要该货架多面时的转面频率 系统计算逻辑为货架坐标与工作站的工作方向进行计算后获得货架实际需要转的角度生成转面任务 不填写表示不转面。	
shelfTurnAngle	int	可选	货架指定角度旋转，只能是 90 的倍数。例如：90,180,270,360 。这 4 个角度，neededSides 和 shelfTurnAngle 只能使用一个，两个不可同时使用。该角度为旋转的相对角度。	90
shelfTurnDirection	int	可选	货架旋转方向: 0 代表顺时针，1 代表逆时针，默认 0。此参数配合 shelfTurnAngle 使用。	0
bindingStation	boolean	可选	是否和工作站进行绑定，绑定后机器人不可以执行其他工作站任务。仅会执行本工作站及其上货架的命令。绑定仅在任务结束时生效。	false
allowChangeShelfPlacement	boolean	可选	是否可以更改货架位置，将当前货架 placement 更新到当前任务终点。货架到达后会将目的地更新为新老家。	false
其余控制字段	可根据需要进行选填			
priority	Integer	可选	该参数用来设置任务的优先级。值越大，表明在分配机器人执行任务期间该任务的优先级越高，priority 取值范围[0-99]。	2

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskRequestMsg",
```

```
    "request": {
        "header": {
            "requestId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
            "clientCode": "geekCode",
            "warehouseCode": "geekWarehouseCode",
            "userId": "admin",
            "userKey": "123456",
            "language": "en_us",
            "version": "3.3.0"
        },
        "body": {
            "taskType": "DELIVER_SHELF",
            "shelfCode": "A00001",
            "instruction": "GO_RETURN",
            "destCellCode":"600045"
        }
    }
}
```

3.4.3.3 GO_NEXT

当主机系统命令 RMS 指定机器人本体运动到目的地时使用此指令。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
任务主体字段	该部分为任务的主体信息字段			
taskType	String	必要	任务类型。为 DELIVER_SHELF	DELIVER_SHELF
instruction	String	必要	GO_NEXT	GO_NEXT
isContinue	Integer	可选	0，指令执行结束后任务结束。1，指令执行结束后任务不结束。为空视为 1。 GO_NEXT 指令默认会继续任务请注意	1
机器人控制字段	该部分用于控制任务机器人。 单个、集合均填写时取并集，允许指定具体机器人执行任务，不指定时系统会调度最近的机器人。注：强制指定且机器人不可用时可能会使任务失败			
robotId	int	可选	指定某个机器人取执行任务	1
robotIds	list of int	可选	指定一个机器人集合去执行，系统会在其中选中一个距离最近符合条件的机器人去执行	[1,2,3]

字段	类型	是否必要	描述	示例
robotAllocationStrategy	int	可选	0：优先，指定的机器人为优先指定，若机器人不可用会下发给其他机器人执行； 1：强制，必须为指定的机器人执行。 空默认为 0。	0
目的地控制字段	系统会按照以下优先级选择目的地，目的地 > 工作站，所有的值都会参与校验，所有的参数可同时传值。若多种目的地均填写了，则指代地点必须一致，否则报错。必须指定一个目的点。			
destCellCode	String	可选	目的节点编码	1
dest	IPoint	可选	目的地索引坐标。即 z 轴，x 轴，y 轴。z 可省略,默认是 1(二维码区域专用,传索引值)	{"x":45,"y":42}, {"z":1,"x":45,"y":42}
destHostCode	String	可选	上位机系统关于当前任务的目的地坐标的描述	1
stationId	Integer	可选	工作站编号	2
hostStationCode	String	可选	主机系统对工作站的编号的描述	2
其余控制字段	可根据需要进行选填			
priority	Integer	可选	该参数用来设置任务的优先级。值越大，表明在分配机器人执行任务期间该任务的优先级越高，priority 取值范围[0-99]。	2

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskRequestMsg",
   "request": {
      "header": {
         "requestId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskType": "DELIVER_SHELF",
```

```
            "instruction": "GO_NEXT",
            "destCellCode":"600045"
        }
    }
}
```


3.4.4 任务更新

任务更新是指在任务 instruction 置为 READY 时进行的正常更新，使任务执行下一指令，同时也包括 CANCEL 任务取消和动作取消 CANCEL_INSTRUCTION。

1、任务取消为取消尝试，当任务即将完成时，任务会完成而非被取消；

2、动作取消为取消当前任务正在执行的指令，将任务 instruction 重置为 READY。

RMS 对于主机系统的更新均会进行标准响应：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskUpdateResponseMsg",
    "response": {
        "header": {
            "responseId": "9beab88d-a407-4195-a1f8-ae3594c83924",
            "code": 0,
            "msg": "Success"
        }
    }
}
```


3.4.4.1 GO_FETCH

指派机器人负载货架到某处，到达后负载货架。以下情形可使用该指令：

1、机器人为本体状态，可使用该指令去取货架，支持去取一个新货架；

2、机器人为负载状态，则可使用该指令进行负载移动。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	表示指派机器人搬运货架返回库区，必须为“GO_FETCH”	GO_FETCH
isContinue	Integer	可选	0：指令执行结束后任务结束。1：指令执行结束后任务不结束。为空视为 1；当填写 0 时，任务将无法执行而异常。	
目的地控制字段	系统会按照以下优先级选择目的地，目的地 > 工作站，所有的值都会参与校验，所有的参数可同时传值。若多种目的地均填写了，则指代地点必须一致，否则报错。如未			

字段	类型	是否必要	描述	示例
	指定目的点，则目的点默认为货架老家位置，货架不存在老家位置则接口返回异常。			
destCellCode	String	可选	目的地：节点编码	1
dest	IPoint	可选	目的地：索引坐标。即 z 轴，x 轴，y 轴。z 可省略,默认是 1(二维码区域专用,传索引值)	{"x":45,"y":42}, {"z":1,"x":45,"y":42}
destHostCode	String	可选	目的地：上位机系统关于当前任务的目的地坐标的描述	1
stationId	Integer	可选	工作站：编号	2
hostStationCode	String	可选	工作站：主机系统对工作站的编号的描述	2
货架信息控制字段	该部分将描述货架相关的控制字段，若都填写，指代货架必须一致。 若当前机器人已负载，则可不填写货架，但若填写则其指代必须与当前负载一致。			
shelfCode	String	可选	货架编码。 此字段表示货架在 RMS 中的内部编码。	A000001
hostShelfCode	String	可选	货架在主机系统中的外部编码。 只有主机系统使用自有的货架编码并将编码同步到 RMS 才有效。	geekShelfCode0001
shelfScore	int	可选	货架热度，取值范围[1-100]，默认为空。当该任务该执行时货架分数会被该值更新。不填写不更新。	1
neededSides	list of String	可选	货架面。标识工作站工人需要操作的面，可填值为 F、B、L、R（即前、后、左、右）。 指定多个面时系统会选则一个不需要转面的面到达工作站，可减少主机系统需要该货架多面时的转面频率 系统计算逻辑为货架坐标与工作站的工作方向进行计算后获得货架实际需要转的角度生成转面任务 不填写表示不转面。	["F","B"]
shelfTurnAngle	int	可选	货架指定角度旋转，只能是 90 的倍数。例如：90,180,270,360 。这 4 个角度，neededSides 和 shelfTurnAngle 只能使用一个，两个不可同时使用。 该角度为旋转的相对角度	90

字段	类型	是否必要	描述	示例
shelfTurnDirection	int	可选	货架旋转方向: 0 代表顺时针，1 代表逆时针，默认 0 该参数配合 shelfTurnAngle 使用	0

请求报文示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "9beab88d-a407-4195-a1f8-ae3594c83924",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 30,
         "instruction": "GO_FETCH",
         "destCellCode": "655544"
        }
   }
}
```


3.4.4.2 GO_RETURN

指派机器人负载货架到某处，到达后卸载货架。以下情形可使用该指令：

1、机器人为本体状态，可使用该指令去取货架到目的地后卸载，支持去取一个新货架；

2、机器人为负载状态，则可使用该指令进行移动到目的地后卸载货架；

3、使用该指令时任务可以结束。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
任务主体信息	该部分将描述任务的主体信息部分			
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	GO_RETURN	GO_RETURN
isContinue	Integer	可选	0，指令执行结束后任务结束。	1

字段	类型	是否必要	描述	示例
			1，指令执行结束后任务不结束。为空视为 0。 GO_RETURN 指令默认会结束任务请注意	
目的地控制字段	系统会按照以下优先级选择目的地，目的地 > 工作站，所有的值都会参与校验，所有的参数可同时传值。若多种目的地均填写了，则指代地点必须一致，否则报错。如未指定目的点，则目的点默认为货架老家位置，货架不存在老家位置则接口返回异常。			
destCellCode	String	可选	目的地：节点编码	1
dest	IPoint	可选	目的地：索引坐标。即 z 轴，x 轴，y 轴。z 可省略,默认是 1(二维码区域专用,传索引值)	{"x":45,"y":42}, {"z":1,"x":45,"y":42}
destHostCode	String	可选	目的地：上位机系统关于当前任务的目的地坐标的描述	1
stationId	Integer	可选	工作站：编号	2
hostStationCode	String	可选	工作站：主机系统对工作站的编号的描述	2
货架信息控制字段	该部分将描述货架相关的控制字段，若都填写，指代货架必须一致 若机器人当前在负载，则可不填写货架号，若填写则必须与当前负载一致 若机器人刚将一个货架进行了卸载，该指令允许指向一个新货架，原货架会被释放			
shelfCode	String	可选	货架编码。 此字段表示货架在 RMS 中的内部编码。	A000001
hostShelfCode	String	可选	货架在主机系统中的外部编码。只有主机系统使用自有的货架编码并将编码同步到 RMS 才有效。	geekShelfCode0001
shelfScore	int	可选	货架热度，取值范围[1-100]，默认为空。当该任务该执行时货架分数会被该值更新。不填写不更新。	1
neededSides	list of String	可选	货架面。标识工作站工人需要操作的面，可填值为 F、B、L、R（即前、后、左、右）。 指定多个面时系统会选则一个不	["F","B"]

字段	类型	是否必要	描述	示例
			需要转面的面到达工作站，可减少主机系统需要该货架多面时的转面频率 系统计算逻辑为货架坐标与工作站的工作方向进行计算后获得货架实际需要转的角度生成转面任务。 不填写表示不转面。	
shelfTurnAngle	int	可选	货架指定角度旋转，只能是 90 的倍数。例如：90,180,270,360 。这 4 个角度，neededSides 和 shelfTurnAngle 只能使用一个，两个不可同时使用。 该角度为旋转的相对角度。	90
shelfTurnDirection	int	可选	货架旋转方向: 0 代表顺时针，1 代表逆时针，默认 0。 此参数需配合 shelfTurnAngle 使用。	0
bindingStation	boolean	可选	是否和工作站进行绑定，绑定后机器人不可以执行其他工作站任务。仅会执行本工作站及其上货架的命令。绑定仅在任务结束时起效。	false
allowChangeShelfPlacement	boolean	可选	是否可以更改货架位置，将当前货架 placement 更新到当前任务终点 货架到达后会将目的地更新为新老家。	false

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "9beab88d-a407-4195-a1f8-ae3594c83924",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
```

```
        },
        "body": {
           "taskId": 30,
           "instruction": "GO_RETURN",
           "destCellCode": "655544"
          }
    }
}
```

3.4.4.3 GO_NEXT

指派机器人本体去某处，以下情形可使用该指令：

1、机器人为本体状态，可使用该指令指派机器人去某处；

2、机器人刚卸载一个货架且任务未结束，可使用该指令指派机器人去某处；

3、使用该指令时任务可以结束。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
任务主体信息	该部分将描述任务的主体信息部分			
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	GO_NEXT	GO_NEXT
isContinue	Integer	可选	0，指令执行结束后任务结束。1，指令执行结束后任务不结束。为空视为1。 GO_NEXT 指令默认任务会继续请注意。 若任务不继续，则目的地为空时会原地结束任务	1
目的地控制字段	系统会按照以下优先级选择目的地，目的地 > 工作站，所有的值都会参与校验，所有的参数可同时传值。若多种目的地均填写了，则指代地点必须一致，否则报错。必须指定一个目的点。			
destCellCode	String	可选	目的地：节点编码	1
dest	IPoint	可选	目的地：索引坐标。即 z 轴，x 轴，y 轴。z 可省略,默认是 1(二维码区域专用,传索引值)	{"x":45,"y":42}, {"z":1,"x":45,"y":42}
destHostCode	String	可选	目的地：上位机系统关于当前任务的目的地坐标的描述	1
stationId	Integer	可选	工作站：编号	2

字段	类型	是否必要	描述	示例
hostStationCode	String	可选	工作站：主机系统对工作站编号的描述	2

请求报文示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "9beab88d-a407-4195-a1f8-ae3594c83924",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 30,
         "instruction": "GO_NEXT",
         "destCellCode": "655544"
        }
   }
}
```


3.4.4.4 CANCEL

当任务未结束或取消前，可使用 CANCEL 指令对任务进行取消，*任务取消是一种尝试*，任务在执行前为直接取消，执行中时：

1、任务即将完成则取消不起作用，任务正常完成；

2、若取消成功，系统会执行取消动作；

3、任务在举升货架、转面等不可打断的动作时不会处理取消，而会延后执行；

4、取消时不能被再次任务取消和动作取消，任务仅能被取消一次。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	用来取消指定任务的 id	12
instruction	String	必要	值必须是 CANCEL	CANCEL
cancelAction	String	可选	可选，0,2,3 以机器人实际状态为准进行处理，参照下方任务的处理模式：	0

			参数 0（默认值）：默认处理 空载：原地释放 负载：若有老家位置，返回老家，卸载；若无，原地卸载 参数 2：原地处理 空载：原地释放 负载：原地卸载 参数 3：指定目的地处理 空载：原地释放 负载：到达目的地后停止，卸载释放	
指定目的地描述	该部分用于任务取消时且取消动作为 3 指定目的地时使用，若选用取消动作 3 则下方至少填写一个，若均填写，则指代必须一致			
dest	IPoint	可选	终点索引坐标。三维坐标，即 z 轴，x 轴，y 轴	{"z":1,"x":45,"y":42}
destCellCode	String	可选	目的地单元格编码	202350125
destHostCode	String	可选	主机系统对机器人目的点的描述	geekDest

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "411ccc219da94e7f8680dfdee39c43d5",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 10000002,
         "instruction": "CANCEL"
      }
   }
}
```

3.4.4.5 CANCEL_INSTRUCTION

当任务在执行过程中时，可使用 CANCEL_INSTRUCTION 指令对任务当前动作进行中断：

1、任务在即将完成结束时（当前路径前进点为任务终点）取消动作不起作用，任务正常完成；

2、任务当前指令即将完成结束时（当前路径前进点为任务终点）取消动作不起作用，任务指令正常进入 READY 等待更新；

3、若取消成功，系统指令会进入 READY 等待更新；

4、任务在举升货架、转面等不可打断的动作时不会处理取消，而会延后执行；

5、任务在完成前或接收取消任务指令前可多次执行 CANCEL_INSTRUCTION 指令。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	用来取消指定任务的 id	12
instruction	String	必要	值必须是 CANCEL_INSTRUCTION	CANCEL_INSTRUCTION

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "411ccc219da94e7f8680dfdee39c43d5",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 10000002,
         "instruction": "CANCEL_INSTRUCTION"
      }
   }
}
```

3.4.5 任务回调

机器人在执行“DELIVER_SHELF”任务时，根据当前执行阶段的不同，RMS 会主动向主机系统发送回调消息，告知主机系统此任务目前的任务状态及执行阶段。

以下为“DELIVER_SHELF”可能出现的任务执行阶段“taskPhase”。

taskPhase	发送时机
MOVING	机器人本体运动去某地
ARRIVED	机器人本体运动到达某地

taskPhase	发送时机
GO_FETCHING	机器人出发去取指定货架
SHELF_FETCHED	机器人取货架顶升完成
GO_DELIVERING	机器人举升货架出发到某处
SHELF_TURNING	机器人开始进行转面
GO_RETURN	机器人出发到某地卸载货架
SHELF_ARRIVED	机器人到达目的地

RMS 主动发送的回调消息字段如下：

字段	类型	描述	示例
通用任务信息	该部分为任务回调的主体部分，表明产生该回调的任务信息内容		
taskId	long	当前进行的任务 id	30
taskType	String	产生该回调的任务类型	DELIVER_SHELF
taskStatus	String	EXECUTING 执行中 CANCELED 取消 COMPLETED 完成	EXECUTING
instruction	String	表示任务在执行哪个指令时发起的回调。 若为 CANCEL 表示任务在执行任务取消动作过程中； 若为 READY 表示任务已完成当前动作正在准备接受更新；	GO_FETCH
taskPhase	String	回调的阶段。严格按照当前正在执行的动作阶段进行回调，若正在执行 MOVING 被取消动作则仍然会回调 MOVING	GO_FETCHING
robotId	int	执行任务的机器人 ID	1
通用任务信息：当前目的地描述	该部分将描述指令当前要到达的目的地		
destLocation	DPoint	终点：绝对坐标	{"z": 1,"x": 29.5,"y": 10.5}
dest	IPoint	终点：索引坐标	{"z": 1,"x": 29,"y": 10}
destCellCode	String	终点：节点编码	102950105

字段	类型	描述	示例
destHostCode	String	终点：节点外部编码	546544
stationId	int	终点工作站 注：指令终点为工作站才会有此值	1
hostStationCode	String	终点工作站外部编码 注：指令终点为工作站才会有此值	station1
专有任务信息	该部分将描述 DELIVER_SHELF 专有的任务回调信息，该任务为典型的货架任务，会额外回调货架相关信息。		
shelfCode	String	货架编码	1
hostShelfCode	String	货架在主机系统中的外部编码。 只有主机系统使用自有的货架编码并将编码同步到 RMS 才会展示。	geekShelfCode0001
shelfSide	String	货架的面	F

回调的报文示例如下，不同阶段信息的回调将附在任务回调信息之后。

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "8780c902ed714d10b82a2f20ac514a1c",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 37,
         "taskType": "DELIVER_SHELF",
         "taskStatus": "EXCUTING",
         "instruction": "GO_FETCH",
         "taskPhase": "GO_FETCHING",
         "robotId": 1000,
         "destLocation":{"z": 1,"x": 29.5,"y": 10.5},
         "dest":{"z": 1,"x": 29,"y": 10},
         "destCellCode":"102950105",
         "destHostCode":"45464",
         "stationId":"1",
         "shelfCode":"1",
         "hostShelfCode":"4556",
         "shelfSide":"F"
      }
   }
}
```

对于 RMS 回调的所有消息，主机系统均需要进行响应。

主机系统应该给与 RMS 的响应如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskCallbackResponseMsg",
  "response": {
    "header": {
      "responseId": "8780c902ed714d10b82a2f20ac514a1c"
    }
  }
}
```


以下将给出 DELIVEER_SHELF 任务不同阶段的回调消息示例。

3.4.5.1 MOVING

MOVING 阶段的回调消息示例如下：


```
{
  "id": "*",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWarehouseCode",
      "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 141,
      "taskType": "DELIVER_SHELF ",
      "taskStatus": "EXECUTING",
      "instruction": "GO_NEXT",
      "taskPhase": "MOVING",
      "robotId": 1293435,
      "dest": {
        "z": 1,
        "x": 138,
        "y": 67
      },
      "destLocation": {
        "z": 1,
        "x": 163.755,
        "y": 82.442
      },
      "destCellCode": "13850675"
    }
  }
}
```

3.4.5.2 ARRIVED

ARRIVED 阶段的回调消息示例如下：


```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 141,
         "taskType": "DELIVER_SHELF ",
         "taskStatus": "EXECUTING",
         "instruction": "READY",
         "taskPhase": "ARRIVED",
         "robotId": 1293435
      }
   }
}
```


3.4.5.3 GO_FETCHING

GO_FETCHING 阶段的回调消息示例如下：


```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 141,
         "taskType": "DELIVER_SHELF ",
         "taskStatus": "EXECUTING",
         "instruction": "GO_FETCH",
         "taskPhase": "GO_FETCHING",
         "robotId": 1293435,
         "stationId": 106,
         "shelfCode": "00000385",
         "shelfSide": "B",
         "dest": {
            "z": 1,
            "x": 138,
```

```
            "y": 67
          },
          "destLocation": {
            "z": 1,
            "x": 163.755,
            "y": 82.442
          },
          "destCellCode": "13850675"
      }
    }
}
```


3.4.5.4 SHELF_FETCHED

SHELF_FETCHED 阶段的回调消息示例如下：


```
{
  "id": "*",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
       "warehouseCode": "geekWarehouseCode",
      "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 141,
      "taskType": "DELIVER_SHELF ",
      "taskStatus": "EXECUTING",
      "instruction": "GO_FETCH",
      "taskPhase": "SHELF_FETCHED",
      "robotId": 1293435,
      "stationId": 106,
      "shelfCode": "00000385",
      "shelfSide": "B",
      "dest": {
        "z": 1,
        "x": 138,
        "y": 67
      },
      "destLocation": {
        "z": 1,
        "x": 163.755,
        "y": 82.442
      },
      "destCellCode": "13850675"
    }
  }
}
```

3.4.5.5 GO_DELIVERING

GO_DELIVERING 阶段的回调消息示例如下：


```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
          "warehouseCode": "geekWarehouseCode",
          "requestId": "7057982ded8c4203bcfda1084a9fea2c",
          "version": "3.3.0"
      },
      "body": {
         "taskId": 141,
         "taskType": "DELIVER_SHELF",
         "taskStatus": "EXECUTING",
         "instruction": "GO_FETCH",
         "taskPhase": "GO_DELIVERING",
         "robotId": 1293435,
         "stationId": 106,
         "shelfCode": "00000385",
         "shelfSide": "B",
         "dest": {
            "z": 1,
            "x": 162,
            "y": 79
         },
         "destLocation": {
            "z": 1,
            "x": 194.246,
            "y": 97.442
         },
         "destCellCode": "16250795"
      }
   }
}
```


3.4.5.6 SHELF_TURNING

SHELF_TURNING 阶段的回调消息示例如下：


```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "0e28d53e87a844a48dd80a735afa6182",
         "version": "3.3.0"
```

```
    },
    "body": {
      "taskId": 141,
      "taskType": "DELIVER_SHELF ",
      "taskStatus": "EXECUTING",
      "instruction": "GO_FETCH",
      "taskPhase": "SHELF_TURNING",
      "robotId": 1293435,
      "stationId": 106,
      "shelfCode": "00000385",
      "shelfSide": "B"
    }
  }
}
```


3.4.5.7 GO_RETURN

GO_RETURN 阶段的回调消息示例如下：


```
{
  "id": "*",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWarehouseCode",
      "requestId": "ac5a101f41534ccbbdbecc4a41f33e35",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 140,
      "taskType": "DELIVER_SHELF ",
      "taskStatus": "EXECUTING",
      "instruction": "GO_RETURN",
      "taskPhase": "GO_RETURN",
      "robotId": 1293459,
      "stationId": 105,
      "dest": { "z": 1, "x": 141, "y": 67 },
      "destLocation": {
        "z": 1,
        "x": 167.505,
        "y": 82.442
      },
      "destCellCode": "14150675",
      "shelfCode": "00000445",
      "shelfSide": "F"
    }
  }
}
```

3.4.5.8 SHELF_ARRIVED

任务执行过程中货架到达回调消息示例如下：


```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "e9b86b5f5e8342d7bfeef6cb0d0c49f5",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 140,
         "taskType": "DELIVER_SHELF ",
         "taskStatus": "EXECUTING",
         "instruction": "READY",
         "taskPhase": "SHELF_ARRIVED",
         "robotId": 1293459,
         "stationId": 105,
         "shelfCode": "00000445",
         "shelfSide": "F"
      }
   }
}
```


任务完成时货架到达回调消息示例如下：


```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "7d81ecccd09940b6a0a48e082e8cea53",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 140,
         "taskType": "DELIVER_SHELF ",
         "taskStatus": "COMPLETED",
         "instruction": "GO_RETURN",
         "taskPhase": "SHELF_ARRIVED",
         "robotId": 1293459,
         "stationId": 105,
         "dest": {
            "z": 1,
            "x": 141,
            "y": 67
         },
         "destLocation": {
```

```
            "z": 1,
            "x": 167.505,
            "y": 82.442
          },
          "destCellCode": "14150675",
          "shelfCode": "00000445",
          "shelfSide": "F"
        }
      }
    }
```


3.5.5.9 CANCEL 相关回调

DELIVER_SHELF 任务被取消时：

1、未执行前会直接被取消，产生 taskStatus 的 CANCELED 回调

2、若执行中被正常取消，会在当前动作结束，开始执行取消动作时回调 taskStatus：EXECUTING，instruction：CANCEL，taskPhase 为具体的动作阶段，直到执行结束所有动作，会产生回调 taskStatus：CANCELED，此时表示任务正式结束；

3、若任务即将完成时被取消，任务不会执行取消，会正常产生 taskStatus 为 COMPLETED 回调。

3.5.5.10 CANCEL_INSTRUCTION 相关回调

DELIVER_SHELF 被取消动作时：

1、未执行前、结束后不能被取消，接口提示不能取消；

2、若执行中被取消动作，当前动作结束后会产生一个 instruction：READY 回调，taskPhase 为当前正在执行的动作阶段而非正常结束的回调阶段，表示任务已完成动作取消准备好继续更新；

3、若动作即将完成前被取消动作，该取消可能执行失败，当前动作结束后正常产生阶段回调，instruction：READY，taskPhase 为正常动作完成的回调阶段。

3.4.6 功能注意点

无。

3.5 DELIVER_SHELF_TO_STATION

此任务类型用于指派机器人搬运货架至标准工作站。

3.5.1 场景说明

3.5.1.1 场景概括

该任务用于将货架送至**标准工作站**，通常应用于拣选类的业务场景。

如果货架没有被任何机器人装载，RMS 将选择一个合适的机器人去取货架。如果货架被机器人装载，则将选择该机器人执行任务。

当货架被抬起后，机器人将货架送到指定工作站的排队区域，并排队到拣货点等待工人操作，直到工人完成操作。在等待期间，机器人一直举着货架。

工人操作完成后，主机系统应更新任务通知 RMS 将货架送回货架存储区或将货架转面。

当货架被送回并从机器人上卸载下来或机器人负载着该货架继续执行另一个任务时，当前任务将完成。

注意，一个任务仅支持一个工作站。如果想将货架送至多个工作站，则需要执行几个不同的任务。

特别注意，两个任务可以具有相同的货架编号（*shelfCode*）和工作站编号（*stationId*）。在这种情况下，只有机器人的下一个任务具有与当前任务相同的工作站编号或货架编号时，机器人才在在工人完成操作后继续将货架送到同一工作站。

3.5.1.2 场景要点

当机器人在执行任务时，任务池中有相同货架相同工作站的任务，该任务将直接下发至该机器人的任务队列。

当机器人在离开工作站后收到了相同货架其他工作站的任务，将自动取消返回库区执行该任务。

3.5.2 可用指令

DELIVER_SHELF_TO_STATION 支持的指令如下，具体用法会在下面的部分进行详细叙述。

行为	指令	用途
任务下发	GO_FETCH	指派机器人搬运货架去往标准工作站
任务更新	GO_RETURN	指派机器人搬运货架返回库区卸载后任务结束
任务更新	GO_TURN	指派机器人搬运货架转面然后到达本工作站
任务更新	CANCEL	取消任务

3.5.3 任务下发

在任务下发时可用指令有“GO_FETCH”，下面将进行具体说明。

3.5.3.1 GO_FETCH

当主机系统命令 RMS 搬运指定货架到达一个标准工作站时使用此指令。
请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskType	String	必要	任务类型。必须为“DELIVER_SHELF_TO_STATION”	DELIVER_SHELF_TO_STATION
instruction	String	必要	表示搬运货架到工作站，必须为“GO_FETCH”	GO_FETCH
机器人控制字段	单个、集合均填写时取并集，允许指定具体机器人执行任务，不指定时系统会调度最近的机器人。注：强制指定且机器人不可用时可能会使任务失败			
robotId	int	可选	指定某个机器人取执行任务	1
robotIds	list of int	可选	指定一个机器人集合去执行，系统会在其中选中一个距离最近符合条件的机器人去执行	[1,2,3]
robotAllocationStrategy	int	可选	0：优先，指定的机器人为优先指定，若机器人不可用会下发给其他机器人执行 1：强制，必须为指定的机器人执行 空默认为 0	0
目的地控制字段	目的点字段必须有一个以上的值，若目的地两个字段均填写了，则指代地点必须一致，否则接口提示异常。该任务的目的地必须为标准工作站，不能为普通单点工位。			
stationId	int	二选一或都填写	工作站编号	2
hostStationCode	String		主机系统对工作站的编号的描述	2
货架信息控制字段	该部分将描述货架相关的控制字段，shelfCode，hostShelfCode 至少需填写一个，若都填写，指代货架必须一致			
shelfCode	String	可选	货架编码。 此字段表示货架在 RMS 中的内部编码。	A000001
hostShelfCode	String	可选	货架在主机系统中的外部编码。	geekShelfCode0001

字段	类型	是否必要	描述	示例
			只有主机系统使用自有的货架编码并将编码同步到 RMS 才有效。	
shelfScore	int	可选	货架热度，取值范围[1-100]，默认为空。当该任务该执行时货架分数会被该值更新。不填写不更新。	1
neededSides	list of String	可选	货架面。标识工作站工人需要操作的面，可填值为 F、B、L、R（即前、后、左、右）。 指定多个面时系统会选则一个不需要转面的面到达工作站，可减少主机系统需要该货架多面时的转面频率 系统计算逻辑为货架坐标与工作站的工作方向进行计算后获得货架实际需要转的角度生成转面任务 不填写表示不转面。	["F","B"]
其余信息控制字段	可根据需要进行选填			
priority	int	可选	该参数用来设置任务的优先级。值越大，表明在分配机器人执行任务期间该任务的优先级越高，priority 取值范围[0-99]	0

请求报文示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskRequestMsg",
   "request": {
      "header": {
         "requestId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskType": "DELIVER_SHELF_TO_STATION",
         "instruction": "GO_FETCH",
         "stationId": 1,
         "shelfCode": "A000001",
         "neededSides":["F"]
      }
   }
}
```


RMS 的响应示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskResponseMsg",
   "response": {
      "header": {
         "responseId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
         "code": 0,
         "msg": "Success"
      },
      "body": {
         "taskId": 30
      }
   }
}
```


3.5.4 任务更新

3.5.4.1 GO_RETURN

当机器人停留在标准工作站时可用该指令更新任务，系统将指派机器人返回库区放下货架结束任务。货架返回的为自己的老家位置（如开启动态调整则由 RMS 根据货架分数和单元格热度进行主动调整）。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	必须为“GO_RETURN”	GO_RETURN
shelfScore	int	可选	货架热度，取值范围[1-100]，默认为空。当该任务执行该指令时货架分数会被该值更新。不填写不更新。	1

请求报文示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "9beab88d-a407-4195-a1f8-ae3594c83924",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 30,
         "instruction": "GO_RETURN"
        }
   }
}
```


3.5.4.2 GO_TURN

当机器人停留在标准工作站时可用该指令更新任务，系统将指派机器人离开工作点然后不出工作站直接进入田字格（若田字格空闲）并执行转面命令转为需要的面。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	表示指派机器人进行转面然后回到工作站，必须为“GO_TURN”	GO_TURN
shelfScore	int	可选	货架热度，取值范围[1-100]，默认为空。当该任务执行该指令时货架分数会被该值更新。不填写	1

字段	类型	是否必要	描述	示例
			不更新。	
neededSides	list of String	必要	货架面。标识工作站工人需要操作的面，可填值为 F、B、L、R（即前、后、左、右）。 指定多个面时系统会选则一个不需要转面的面到达工作站，可减少主机系统需要该货架多面时的转面频率。 若填写了不需要转面的面，任务会直接回调指令完成。 系统计算逻辑为货架坐标与工作站的工作方向进行计算后获得货架实际需要转的角度生成转面任务	["F","B"]

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "9beab88d-a407-4195-a1f8-ae3594c83924",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 30,
         "instruction": "GO_TURN"
        }
   }
}
```

3.5.4.3 CANCEL

当任务未结束或取消前，可使用 CANCEL 指令对任务进行取消，执行前任务为直接取消，执行中时：

1、任务即将完成则取消不起作用，任务正常完成；

2、若取消成功，系统会执行取消动作；

3、DELIVER_SHELF_TO_STATION 命令在举升货架、转面等不可打断的动作时不会处理取消，而会延后执行；

4、取消时不能被再次任务取消和动作取消，任务仅能被取消一次。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
instruction	String	必要	值必须是 CANCEL	CANCEL
taskId	long	必要	用来取消指定任务的 id	12

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "411ccc219da94e7f8680dfdee39c43d5",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 10000002,
         "instruction": "CANCEL"
      }
   }
}
```

3.5.5 任务回调

机器人在执行“DELIVER_SHELF_TO_STATION”任务时，根据当前执行阶段的不同，RMS 会主动向主机系统发送回调消息，告知主机系统此任务目前的任务状态及执行阶段。注意示例消息并不包含所有的字段信息，具体说明见。

以下为“DELIVER_SHELF_TO_STATION”可能出现的任务执行阶段“taskPhase”。

taskPhase	发送时机
GO_FETCHING	机器人出发去取指定货架
SHELF_FETCHED	机器人取货架顶升完成
GO_DELIVERING	机器人举升货架出发到某处
QUEUING	机器人进入工作站排队区
SHELF_TURNING	机器人进入田字格开始转面，注意转面不一定发生
GO_RETURN	机器人离开工作点返回库区

taskPhase	发送时机
SHELF_ARRIVED	机器人到达目标点，如到达工作点、到达库区内均会产生

回调的报文示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "8780c902ed714d10b82a2f20ac514a1c",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 37,
         "taskType": "DELIVER_SHELF_TO_STATION",
         "taskStatus": "EXCUTING",
         "instruction": "GO_FETCH",
         "taskPhase": "GO_FETCHING",
         "robotId": 1000,
         "destLocation":{"z": 1,"x": 29.5,"y": 10.5},
         "dest":{"z": 1,"x": 29,"y": 10},
         "destCellCode":"102950105",
         "destHostCode":"45464",
         "stationId":"1",
         "shelfCode":"1",
         "hostShelfCode":"4556",
         "shelfSide":"F"
       }
   }
}
```


对于 RMS 回调的所有消息，主机系统均需要进行响应。

主机系统应该给与 RMS 的响应如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackResponseMsg",
   "response": {
      "header": {
         "responseId": "8780c902ed714d10b82a2f20ac514a1c"
      }
   }
}
```


下面将给出 DELIVER_SHELF_TO_STATION 任务不同阶段的回调消息示例。

3.5.5.1 GO_FETCHING

GO_FETCHING 阶段的回调消息示例如下：


```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 141,
         "taskType": "DELIVER_SHELF_TO_STATION",
         "taskStatus": "EXECUTING",
         "instruction": "GO_FETCH",
         "taskPhase": "GO_FETCHING",
         "robotId": 1293435,
         "stationId": 106,
         "shelfCode": "00000385",
         "shelfSide": "B",
         "dest": {
            "z": 1,
            "x": 138,
            "y": 67
         },
         "destLocation": {
            "z": 1,
            "x": 163.755,
            "y": 82.442
         },
         "destCellCode": "13850675"
      }
   }
}
```


3.5.5.2 SHELF_FETCHED

SHELF_FETCHED 阶段的回调消息示例如下：


```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
          "warehouseCode": "geekWarehouseCode",
         "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
         "version": "3.3.0"
```

```
    },
    "body": {
      "taskId": 141,
      "taskType": "DELIVER_SHELF_TO_STATION",
      "taskStatus": "EXECUTING",
      "instruction": "GO_FETCH",
      "taskPhase": "SHELF_FETCHED",
      "robotId": 1293435,
      "stationId": 106,
      "shelfCode": "00000385",
      "shelfSide": "B",
      "dest": {
        "z": 1,
        "x": 138,
        "y": 67
      },
      "destLocation": {
        "z": 1,
        "x": 163.755,
        "y": 82.442
      },
      "destCellCode": "13850675"
    }
  }
}
```


3.5.5.3 GO_DELIVERING

GO_DELIVERING 阶段的回调消息示例如下：


```
{
  "id": "*",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWarehouseCode",
      "requestId": "7057982ded8c4203bcfda1084a9fea2c",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 141,
      "taskType": "DELIVER_SHELF_TO_STATION",
      "taskStatus": "EXECUTING",
      "instruction": "GO_FETCH",
      "taskPhase": "GO_DELIVERING",
      "robotId": 1293435,
      "stationId": 106,
      "shelfCode": "00000385",
      "shelfSide": "B",
      "dest": {
```

```
            "z": 1,
            "x": 162,
            "y": 79
        },
        "destLocation": {
            "z": 1,
            "x": 194.246,
            "y": 97.442
        },
        "destCellCode": "16250795"
    }
  }
}
```


3.5.5.4 QUEUING

QUEUEING 阶段的回调消息示例如下：


```
{
  "id": "*",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWarehouseCode",
      "requestId": "9cc0db1b302245e780db4321ba9974de",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 141,
      "taskType": "DELIVER_SHELF_TO_STATION",
      "taskStatus": "EXECUTING",
      "instruction": "GO_FETCH",
      "taskPhase": "QUEUING",
      "robotId": 1293435,
      "stationId": 106,
      "shelfCode": "00000385",
      "shelfSide": "B",
      "dest": {
        "z": 1,
        "x": 161,
        "y": 80
      },
      "destLocation": {
        "z": 1,
        "x": 192.996,
        "y": 98.692
      },
      "destCellCode": "16150805"
    }
  }
```

```
}
```

3.5.5.5 SHELF_TURNING

SHELF_TURNING 阶段的回调消息示例如下：

```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "0e28d53e87a844a48dd80a735afa6182",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 141,
         "taskType": "DELIVER_SHELF_TO_STATION",
         "taskStatus": "EXECUTING",
         "instruction": "GO_FETCH",
         "taskPhase": "SHELF_TURNING",
         "robotId": 1293435,
         "stationId": 106,
         "shelfCode": "00000385",
         "shelfSide": "B"
      }
   }
}
```

3.5.5.6 GO_RETURN

GO_RETURN 阶段的回调消息示例如下：

```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "ac5a101f41534ccbbdbecc4a41f33e35",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 140,
         "taskType": "DELIVER_SHELF_TO_STATION",
         "taskStatus": "EXECUTING",
         "instruction": "GO_RETURN",
         "taskPhase": "GO_RETURN",
         "robotId": 1293459,
         "stationId": 105,
```

```
            "dest": {
               "z": 1,
               "x": 141,
               "y": 67
            },
            "destLocation": {
               "z": 1,
               "x": 167.505,
               "y": 82.442
            },
            "destCellCode": "14150675",
            "shelfCode": "00000445",
            "shelfSide": "F"
         }
      }
   }
```


3.5.5.7 SHELF_ARRIVED

任务执行过程中货架到达回调消息示例如下：


```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "e9b86b5f5e8342d7bfeef6cb0d0c49f5",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 140,
         "taskType": "DELIVER_SHELF_TO_STATION",
         "taskStatus": "EXECUTING",
         "instruction": "READY",
         "taskPhase": "SHELF_ARRIVED",
         "robotId": 1293459,
         "stationId": 105,
         "shelfCode": "00000445",
         "shelfSide": "F"
      }
   }
}
```


任务完成时货架到达回调消息示例如下：


```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
```

```
        "requestId": "7d81ecccd09940b6a0a48e082e8cea53",
        "version": "3.3.0"
      },
      "body": {
        "taskId": 140,
        "taskType": "DELIVER_SHELF_TO_STATION",
        "taskStatus": "COMPLETED",
        "instruction": "GO_RETURN",
        "taskPhase": "SHELF_ARRIVED",
        "robotId": 1293459,
        "stationId": 105,
        "dest": {
          "z": 1,
          "x": 141,
          "y": 67
        },
        "destLocation": {
          "z": 1,
          "x": 167.505,
          "y": 82.442
        },
        "destCellCode": "14150675",
        "shelfCode": "00000445",
        "shelfSide": "F"
      }
    }
}
```


3.5.5.8 CANCEL 相关回调

DELIVER_SHELF_TO_STATION 任务被取消时：

1、未执行前会直接被取消，产生 taskStatus 的 CANCELED 回调；

2、若执行中被正常取消，会在当前动作结束，开始执行取消动作时回调 taskStatus：EXECUTING，instruction：CANCEL，taskPhase 为具体的取消动作指令，直到执行结束所有动作，会回调 taskStatus：CANCELED。

3.5.6 功能注意点

DELIVER_SHELF_TO_STATION 任务仅允许指向标准工作站不允许指向单点工位。

3.6 DELIVER_NEW_SHELF

此任务类型用于指派机器人进行货架入场，并将此货架送至单点工作站或指定位置。

3.6.1 场景说明

此任务用于将货架入场，并送至单点工作站或指定位置，在拣选和搬运的业务场景中均可使用。

与 DELIVER_SHELF 不同的是此任务不需要输入货架编号。只需要指定取货架的位置。机器人会自动扫描货架编码完成货架入场动作。

当货架被抬起之后，机器人将把货架送至指定工作站或指定位置。

3.6.2 可用指令

DELIVER_NEW_SHELF 支持的指令如下，具体用法会在下面的部分进行详细叙述。

行为	指令	用途
任务下发	GO_FETCH	指派机器人搬运货架去某处，保持负载
任务下发	GO_RETURN	指派机器人搬运货架去某处，卸载
任务更新	GO_FETCH	机器人搬运货架去某处，保持负载
任务更新	GO_RETURN	机器人搬运货架去某处，卸载
任务更新	CANCEL	任务取消
任务更新	CANCEL_INSTRUCTION	动作取消

3.6.3 任务下发

在任务下发时可用指令“GO_FETCH”、“GO_RETURN”。任务下发后 RMS 会接收并生成一个任务，将任务号回复给主机系统。

3.6.3.1 GO_FETCH

当主机系统命令 RMS 搬运指定位置的货架到目的地保持负载时使用此指令，机器人到目的地后继续举着货架，此时该任务不会结束，RMS 将继续等待主机系统发送任务更新消息。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskType	String	必要	必须为 DELIVER_NEW_SHELF	DELIVER_NEW_SHELF
instruction	String	必要	GO_FETCH	GO_FETCH

字段	类型	是否必要	描述	示例
locationIndex	IPoint	三选一	新货架的入场位置	{"x":45,"y":42}
locationCellCode	String		新货架的入场位置	202350125
locationHostCode	String		新货架的入场位置。只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekCellCode
placementIndex	IPoint	三选一或全为空	新货架的库存位置	{"x":45,"y":42}
placementCellCode	String		新货架的库存位置	202350125
placementHostCode	String		新货架的库存位置。只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekCellCode
dest	point3D	可选	终点坐标。三维坐标，即 z 轴，x 轴，y 轴。	{"z":1,"x":45,"y":42}
destCellCode	String	可选	目的地单元格编码。	202350125
destHostCode	String	可选	主机系统对机器人目的点的描述。只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekDest
stationId	int	可选	工作站编号。此字段表示货架将要移动到的标准工作站在 RMS 中的内部 id。	1000
hostStationCode	String	可选	外部工作站编码。只有主机系统使用自有的工作站编码并将编码同步到 RMS 才有效。	geekStationCode1000
priority	Integer	可选	该参数用来设置任务的优先级。值越大，表明在分配机器人执行任务期间该任务的优先级越高，priority 取值范围[0-99]	2
neededSides	list of String	可选	货架面。标识工作站工人需要操作的面，可填值为 F、B、L、R（即前、后、左、右）。	[“F”]

请求消息：

```
{
 "id": "geekCode_geekWarehouseCode_001",
 "msgType": "RobotTaskRequestMsg",
```

```
"request": {
 "header": {
  "requestId": "9194E002AD324176974AE9438A011EE311",
  "clientCode": "geekCode",
  "warehouseCode": "geekWarehouseCode",
  "userId": "admin",
  "userKey": "123456",
  "language": "en_us",
  "version": "3.3.0"
 },
 "body": {
  "taskType": "DELIVER_NEW_SHELF",
  "instruction": "GO_FETCH",
  "locationCellCode": "01850075",
  "placementCellCode": "01850075",
  "stationId": 1
 }
}
}
```


响应报文：


```
{
 "id": "geekCode_geekWarehouseCode_001",
 "msgType": "RobotTaskResponseMsg",
 "response": {
  "header": {
   "responseId": "9194E002AD324176974AE9438A011EE311",
   "code": 0,
   "msg": "Success"
  },
  "body": {
   "taskId": 111
  }
 }
}
```


3.6.3.2 GO_RETURN

指派机器人负载货架到某处，到达后卸载货架，结束任务。请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskType	String	必要	必须为 DELIVER_NEW_SHELF	DELIVER_NEW_SHELF
instruction	String	必要	GO_RETURN	GO_RETURN
locationIndex	IPoint	三选一	新货架的入场位置	{"z":1,"x":45,"y":42}
locationCellCode	String		新货架的入场位置	202350125

字段	类型	是否必要	描述	示例
locationHostCode	String		新货架的入场位置。只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekCellCode
placementIndex	IPoint	三选一或全为空	新货架的库存位置	{"z":1,"x":45,"y":42}
placementCellCode	String		新货架的库存位置	202350125
placementHostCode	String		新货架的库存位置。只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekCellCode
dest	DPoint	可选	终点坐标。三维坐标，即 z 轴，x 轴，y 轴。	{"z":1,"x":45.3,"y":42.1}
destCellCode	String	可选	目的地单元格编码。	202350125
destHostCode	String	可选	主机系统对机器人目的点的描述。只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekDest
stationId	int	可选	工作站编号。此字段表示货架将要移动到的标准工作站在 RMS 中的内部 id。	1000
hostStationCode	String	可选	外部工作站编码。只有主机系统使用自有的工作站编码并将编码同步到 RMS 才有效。	geekStationCode1000
priority	Integer	可选	该参数用来设置任务的优先级。值越大，表明在分配机器人执行任务期间该任务的优先级越高，priority 取值范围[0-99]	2
neededSides	list of String	可选	货架面。标识工作站工人需要操作的面，可填值为 F、B、L、R（即前、后、左、右）。	[“F”]

请求报文示例如下：

```
{
 "id": "geekCode_geekWarehouseCode_001",
 "msgType": "RobotTaskRequestMsg",
 "request": {
  "header": {
```

```
  "requestId": "9194E002AD324176974AE9438A011EE322",
  "clientCode": "geekCode",
  "warehouseCode": "geekWarehouseCode",
  "userId": "admin",
  "userKey": "123456",
  "language": "en_us",
  "version": "3.3.0"
 },
 "body": {
  "taskType": "DELIVER_NEW_SHELF",
  "instruction": "GO_RETURN",
  "locationCellCode": "03550275",
  "placementCellCode": "03550275",
  "dest": {
   "z": 1,
   "x": 45,
   "y": 42
  }
 }
}
}
```


响应报文：


```
{
 "id": "geekCode_geekWarehouseCode_001",
 "msgType": "RobotTaskResponseMsg",
 "response": {
  "header": {
   "responseId": "1580964283215",
   "code": 0,
   "msg": "Success"
  },
  "body": {
   "taskId": 122
  }
 }
}
```


3.6.4 任务更新

3.6.4.1 GO_FETCH

当主机系统命令 RMS 搬运指定位置的货架到目的地保持负载时使用此指令。机器人到目的地后继续举着货架，此时该任务不会结束，RMS 将继续等待主机系统发送任务更新消息。

请求消息体的字段如下：

字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 GO_FETCH
taskId	long	必要	当前进行的任务 id
neededSides	list of String	可选	用来更新货架所需要呈现的面，neededSides 和 shelfTurnAngle 不可同时使用
allowChangeShelfPlacement	boolean	可选	是否允许改变货架的摆放位置，true 则将货架的 placement 更新到当前任务终点
stationId	int	工作站二选一，也可均不填写。工作站参数或坐标参数必须二选一，坐标参数优先级高于工作站	工作站 id
hostStationCode	String		外部工作站编码
dest	IPoint	目的地三个参数选一，也可以均不填写。工作站参数或坐标参数必须二选一，坐标参数优先级高于工作站	目的地
destCellCode	String		目的地单元格编码
destHostCode	String		外部编码。只有主机系统使用自有的位置编码并同步到 RMS 才有效。

请求消息示例：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
        "header": {
              "requestId": "411ccc219da94e7f8680dfdee39c43d5",
              "clientCode": "geekCode",
              "warehouseCode": "geekWarehouseCode",
              "userId": "admin",
              "userKey": "123456",
              "language": "en_us",
              "version": "3.3.0"
         },
        "body": {
              "taskId": 111,
             "instruction": "GO_FETCH",
             "stationId": 2
        }
    }
}
```

响应报文示例：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskResponseMsg",
      "response": {
          "header": {
              "responseId": "411ccc219da94e7f8680dfdee39c43d5",
              "code": 0,
              "msg": "Success"
           }
      }
}
```


3.6.4.2 GO_RETURN

指派机器人负载货架到某处，到达后卸载货架，结束任务。请求消息体的字段如下：

字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 GO_RETURN
taskId	long	必要	当前进行的任务 id
neededSides	list of String	可选	用来更新货架所需要呈现的面，neededSides 和 shelfTurnAngle 不可同时使用
allowChangeShelfPlacement	boolean	可选	是否允许改变货架的摆放位置，true 则将货架的 placement 更新到当前任务终点
stationId	int	工作站二选一，也可均不填写。但是工作站参数或坐标参数必须二选一，坐标参数优先级高于工作站	工作站 id
hostStationCode	String		外部工作站编码
dest	IPoint	目的地三个参数选一，也可以均不填写。但是工作站参数或坐标参数必须二选一，坐标参数优先级高于工作站	目的地
destCellCode	String		目的地单元格编码
destHostCode	String		外部编码。只有主机系统使用自有的位置编码并同步到 RMS 才有效。

请求消息：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskUpdateRequestMsg",
        "request": {
            "header": {
                "requestId": "411ccc219da94e7f8680dfdee39c43d5",
```

```
            "clientCode": "geekCode",
            "warehouseCode": "geekWarehouseCode",
            "userId": "admin",
            "userKey": "123456",
            "language": "en_us",
            "version": "3.3.0"
        },
        "body": {
            "taskId": 111,
            "instruction": "GO_RETURN",
            "stationId": 2
        }
    }
}
```


响应报文：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskResponseMsg",
    "response": {
        "header": {
            "responseId": "411ccc219da94e7f8680dfdee39c43d5",
            "code": 0,
            "msg": "Success"
        }
    }
}
```


3.6.4.3 CANCEL

当任务未结束或取消前，可使用 CANCEL 指令对任务进行取消，任务取消是一种尝试，任务在执行前为直接取消，执行中时：

1、任务即将完成则取消不起作用，任务正常完成；

2、若取消成功，系统会执行取消动作；

3、任务在举升货架、转面等不可打断的动作时不会处理取消，而会延后执行；

4、取消时不能被再次任务取消和动作取消，任务仅能被取消一次。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	用来取消指定任务的 id	12
instruction	String	必要	值必须是 CANCEL	CANCEL
cancelAction	String	可选	可选，0,2,3 以机器人实际状态为准进行处理，参照下方任务的处理模式：	0

字段	类型	是否必要	描述	示例
			参数 0（默认值）：默认处理 空载：原地释放 负载：若有老家位置，返回老家，卸载；若无，原地卸载 参数 2：原地处理 空载：原地释放 负载：原地卸载 参数 3：指定目的地处理 空载：原地释放 负载：到达目的地后停止，卸载释放	
指定目的地描述	该部分用于任务取消时且取消动作为 3 指定目的地时使用，若选用取消动作 3 则下方至少填写一个，若均填写，则该字段指代必须一致			
dest	IPoint	可选	终点索引坐标。三维坐标，即 z 轴，x 轴，y 轴	{"z":1,"x":45,"y":42}
destCellCode	String	可选	目的地单元格编码	202350125
destHostCode	String	可选	主机系统对机器人目的点的描述	geekDest

请求报文示例如下：

```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskUpdateRequestMsg",
    "request": {
        "header": {
            "requestId": "411ccc219da94e7f8680dfdee39c43d5",
            "clientCode": "geekCode",
            "warehouseCode": "geekWarehouseCode",
            "userId": "admin",
            "userKey": "123456",
            "version": "3.3.0"
        },
        "body": {
            "taskId": 10000002,
            "instruction": "CANCEL"
        }
    }
}
```

3.4.4.4 CANCEL_INSTRUCTION

当任务在执行过程中时，可使用 CANCEL_INSTRUCTION 指令对任务当前动作进行中断：

1、任务在即将完成结束时取消动作不起作用，任务正常完成；

2、任务当前指令即将完成结束时取消动作不起作用，任务指令正常进入 READY 等待更新；

3、若取消成功，系统指令会进入 READY 等待更新；

4、任务在举升货架、转面等不可打断的动作时不会处理取消，而会延后执行；

5、任务在完成前或接收取消任务指令前可多次执行 CANCEL_INSTRUCTION 指令。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	用来取消指定任务的 id	12
instruction	String	必要	值必须是 CANCEL_INSTRUCTION	CANCEL_INSTRUCTION

请求报文示例如下：

```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskUpdateRequestMsg",
    "request": {
        "header": {
            "requestId": "411ccc219da94e7f8680dfdee39c43d5",
            "clientCode": "geekCode",
            "warehouseCode": "geekWarehouseCode",
            "userId": "admin",
            "userKey": "123456",
            "version": "3.3.0"
        },
        "body": {
            "taskId": 10000002,
            "instruction": "CANCEL_INSTRUCTION"
        }
    }
}
```

3.6.5 任务回调

机器人在执行“DELIVER_NEW_SHELF”任务时，根据当前执行阶段的不同，RMS 会主动向主机系统发送回调消息，告知主机系统此任务目前的任务状态及执行阶段。

以下为“DELIVER_NEW_SHELF”可能出现的任务执行阶段“taskPhase”。

taskPhase	发送时机
GO_FETCHING	机器人出发去取指定货架
SHELF_FETCHED	机器人取货架顶升完成
GO_DELIVERING	机器人举升货架出发到某处

taskPhase	发送时机
SHELF_TURNING	机器人开始进行转面
GO_RETURN	机器人出发到某地卸载货架
SHELF_ARRIVED	机器人到达目的地

DELIVER_NEW_SHELF 任务的回调消息字段说明如下：

字段	类型	描述	示例
通用任务信息	该部分为任务回调的主体部分，表明产生该回调的任务信息内容		
taskId	long	当前进行的任务 id	30
taskType	String	产生该回调的任务类型	DELIVER_SHELF
taskStatus	String	EXECUTING 执行中 CANCELED 取消 COMPLETED 完成	EXECUTING
instruction	String	表示任务在执行哪个指令时发起的回调。 若为 CANCEL 表示任务在执行任务取消动作过程中； 若为 READY 表示任务已完成当前动作正在准备接受更新；	GO_FETCH
taskPhase	String	回调的阶段。严格按照当前正在执行的动作阶段进行回调	GO_FETCHING
robotId	int	执行任务的机器人 ID	1
通用任务信息：当前目的地描述	该部分将描述指令当前要到达的目的地		
destLocation	DPoint	终点：绝对坐标	{"z": 1,"x": 29.5,"y": 10.5}
dest	IPoint	终点：索引坐标	{"z": 1,"x": 29,"y": 10}
destCellCode	String	终点：节点编码	102950105
destHostCode	String	终点：节点外部编码	546544
stationId	int	终点工作站 注：指令终点为工作站才会有此值	1
hostStationCode	String	终点工作站外部编码 注：指令终点为工作站才会有此值	station1

字段	类型	描述	示例
专有任务信息	该部分将描述 DELIVER_NEW_SHELF 专有的任务回调信息，该任务为典型的货架任务，会额外回调货架相关信息。		
shelfCode	String	货架编码	1
hostShelfCode	String	货架在主机系统中的外部编码。 只有主机系统使用自有的货架编码并将编码同步到 RMS 才会展示。	geekShelfCode0001
shelfSide	String	货架的面	F

回调的报文示例如下，不同阶段信息的回调将附在任务回调信息之后。


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWarehouseCode",
      "requestId": "8780c902ed714d10b82a2f20ac514a1c",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 37,
      "taskType": "DELIVER_NEW_SHELF",
      "taskStatus": "EXCUTING",
      "instruction": "GO_FETCH",
      "taskPhase": "GO_FETCHING",
      "robotId": 1000,
      "destLocation":{"z": 1,"x": 29.5,"y": 10.5},
      "dest":{"z": 1,"x": 29,"y": 10},
      "destCellCode":"102950105",
      "destHostCode":"45464",
      "stationId":"1"
    }
  }
}
```


对于 RMS 回调的所有消息，主机系统均需要进行响应。

主机系统应该给与 RMS 的响应如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskCallbackResponseMsg",
  "response": {
    "header": {
      "responseId": "8780c902ed714d10b82a2f20ac514a1c"
    }
  }
}
```

以下将给出 DELIVEER_NEW_SHELF 任务不同阶段的回调消息示例。

3.6.5.1 GO_FETCHING

GO_FETCHING 阶段的回调消息示例如下：


```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 141,
         "taskType": "DELIVER_NEW_SHELF ",
         "taskStatus": "EXECUTING",
         "instruction": "GO_FETCH",
         "taskPhase": "GO_FETCHING",
         "robotId": 1293435,
         "stationId": 106,
         "dest": {
            "z": 1,
            "x": 138,
            "y": 67
         },
         "destLocation": {
            "z": 1,
            "x": 163.755,
            "y": 82.442
         },
         "destCellCode": "13850675"
      }
   }
}
```


3.6.5.2 SHELF_FETCHED

SHELF_FETCHED 阶段的回调消息示例如下：


```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
          "warehouseCode": "geekWarehouseCode",
         "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
         "version": "3.3.0"
```

```
        },
        "body": {
          "taskId": 141,
          "taskType": "DELIVER_NEW_SHELF ",
          "taskStatus": "EXECUTING",
          "instruction": "GO_FETCH",
          "taskPhase": "SHELF_FETCHED",
          "robotId": 1293435,
          "stationId": 106,
          "shelfCode": "00000385",
          "dest": {
            "z": 1,
            "x": 138,
            "y": 67
          },
          "destLocation": {
            "z": 1,
            "x": 163.755,
            "y": 82.442
          },
          "destCellCode": "13850675"
        }
    }
}
```


3.6.5.3 GO_DELIVERING

GO_DELIVERING 阶段的回调消息示例如下：


```
{
    "id": "*",
    "msgType": "RobotTaskCallbackMsg",
    "request": {
        "header": {
            "warehouseCode": "geekWarehouseCode",
            "requestId": "7057982ded8c4203bcfda1084a9fea2c",
            "version": "3.3.0"
        },
        "body": {
            "taskId": 141,
            "taskType": "DELIVER_NEW_SHELF",
            "taskStatus": "EXECUTING",
            "instruction": "GO_FETCH",
            "taskPhase": "GO_DELIVERING",
            "robotId": 1293435,
            "stationId": 106,
            "shelfCode": "00000385",
            "dest": {
                "z": 1,
                "x": 162,
```

```
            "y": 79
          },
          "destLocation": {
            "z": 1,
            "x": 194.246,
            "y": 97.442
          },
          "destCellCode": "16250795"
        }
    }
}
```


3.6.5.4 SHELF_TURNING

SHELF_TURNING 阶段的回调消息示例如下：


```
{
    "id": "*",
    "msgType": "RobotTaskCallbackMsg",
    "request": {
        "header": {
            "warehouseCode": "geekWarehouseCode",
            "requestId": "0e28d53e87a844a48dd80a735afa6182",
            "version": "3.3.0"
        },
        "body": {
            "taskId": 141,
            "taskType": "DELIVER_NEW_SHELF ",
            "taskStatus": "EXECUTING",
            "instruction": "GO_FETCH",
            "taskPhase": "SHELF_TURNING",
            "robotId": 1293435,
            "stationId": 106,
            "shelfCode": "00000385",
            "shelfSide": "B"
        }
    }
}
```


3.6.5.5 GO_RETURN

GO_RETURN 阶段的回调消息示例如下：


```
{
    "id": "*",
    "msgType": "RobotTaskCallbackMsg",
    "request": {
        "header": {
            "warehouseCode": "geekWarehouseCode",
            "requestId": "ac5a101f41534ccbbdbecc4a41f33e35",
```

```
            "version": "3.3.0"
        },
        "body": {
            "taskId": 140,
            "taskType": "DELIVER_NEW_SHELF ",
            "taskStatus": "EXECUTING",
            "instruction": "GO_RETURN",
            "taskPhase": "GO_RETURN",
            "robotId": 1293459,
            "stationId": 105,
            "dest": {
                "z": 1,
                "x": 141,
                "y": 67
            },
            "destLocation": {
                "z": 1,
                "x": 167.505,
                "y": 82.442
            },
            "destCellCode": "14150675",
            "shelfCode": "00000445",
            "shelfSide": "F"
        }
    }
}
```


3.6.5.6 SHELF_ARRIVED

SHELF_ARRIVED 会在货架到达指定位置时发生，不管是否放下。任务执行过程中货架到达回调消息示例如下：


```
{
    "id": "*",
    "msgType": "RobotTaskCallbackMsg",
    "request": {
        "header": {
            "warehouseCode": "geekWarehouseCode",
            "requestId": "e9b86b5f5e8342d7bfeef6cb0d0c49f5",
            "version": "3.3.0"
        },
        "body": {
            "taskId": 140,
            "taskType": "DELIVER_NEW_SHELF ",
            "taskStatus": "EXECUTING",
            "instruction": "READY",
            "taskPhase": "SHELF_ARRIVED",
            "robotId": 1293459,
            "stationId": 105,
            "shelfCode": "00000445"
```

```
        }
    }
}
```

任务完成时货架到达回调消息示例如下：

```
{
    "id": "*",
    "msgType": "RobotTaskCallbackMsg",
    "request": {
        "header": {
            "warehouseCode": "geekWarehouseCode",
            "requestId": "7d81ecccd09940b6a0a48e082e8cea53",
            "version": "3.3.0"
        },
        "body": {
            "taskId": 140,
            "taskType": "DELIVER_NEW_SHELF ",
            "taskStatus": "COMPLETED",
            "instruction": "GO_RETURN",
            "taskPhase": "SHELF_ARRIVED",
            "robotId": 1293459,
            "shelfCode": "00000445"
        }
    }
}
```

3.6.5.7 CANCEL 相关回调

DELIVER_NEW_SHELF 任务被取消时：

1、未执行前会直接被取消，产生 taskStatus 的 CANCELED 回调

2、若执行中被正常取消，会在当前动作结束，开始执行取消动作时回调 taskStatus：EXECUTING，instruction：CANCEL，taskPhase 为具体的动作阶段，直到执行结束所有动作，会产生回调 taskStatus：CANCELED，此时表示任务正式结束；

3、若任务即将完成时被取消，任务不会执行取消，会正常产生 taskStatus 为 COMPLETED 回调。

3.6.5.8 CANCEL_INSTRUCTION 相关回调

DELIVER_NEW_SHELF 被取消动作时：

1、未执行前、结束后不能被取消，接口提示不能取消；

2、若执行中被取消动作，当前动作结束后会产生一个 instruction：READY 回调，taskPhase 为当前正在执行的动作阶段而非正常结束的回调阶段，表示任务已完成动作取消准备好继续更新；

3、若动作即将完成前被取消动作，该取消可能执行失败，当前动作结束后正常产生阶段

回调，instruction：READY，taskPhase 为正常动作完成的回调阶段。

3.6.6 功能注意点

1、与 DELIVER_SHELF 不同的是此任务不需要输入货架编号。只需要指定取货架的位置。机器人会自动扫描货架编码完成货架入场动作。

2、机器人到达目的地后，若 40 秒扫不到货架码，则 RMS 会主动取消此任务。

3.7 CARRY_PACKAGE

此任务类型主要适用于一些车间与辊筒对接的工业场景，或一些特殊的点到点的收货、投货的搬运场景。

目前可执行此类任务的机器人均为辊筒机器人（不限定具体型号，凡是在机器人上位机构中加装了辊筒机构的机器人均可以称之为辊筒机器人），该任务类型用于辊筒机器人的移动、取包、投递控制。

目前支持的工作站类型为特殊的辊筒工作站，和标准工位或是单点工位不同，需要进行单独配置。

3.7.1 场景说明

3.7.1.1 去收货&去投货

调度机器人去某个辊筒工作站收取一定数量的货物，待机器人完成收货后，再调度机器人去某个辊筒工作站投递货物，货物投递完毕后任务结束。

3.7.1.2 多点收货&多点投货

调度机器人去某个辊筒工作站收取一定数量的货物，待机器人完成收货完毕后，再调度机器人去某个辊筒工作站投递货物。

若投递完毕后机器人身上还有货物，则可以调度此机器人去任意辊筒工作站继续投递货物，或者去任意辊筒工作站继续收取货物，直到机器身上货物投递完毕时结束此次任务。

3.7.1.3 去某点

调度一个机器人到某点或某工作站，然后由工人进行手动放货、取货。工人操作完毕后可以继续更新此任务，调度机器人去下一个目标任务地点。

可以使用取消指令或结束指令来结束此任务。

3.7.2 可用指令

“CARRY_PACKAGE”支持的指令如下，具体用法会在下面的部分进行详细叙述。

行为	指令	用途
任务下发	GO_RECEIVE	指派机器人去某辊筒工作站收货，等待指令更新
任务下发	GO_NEXT	指派机器人某处，等待指令更新
任务下发	GO_DROP	指派机器人去某辊筒工作站收货，然后去某个辊筒工作站卸货
任务更新	GO_RECEIVE	机器人去某辊筒工作站收货，等待指令更新
任务更新	GO_NEXT	指派机器人某处，等待指令更新
任务更新	GO_DROP	指派机器人去某辊筒工作站卸货
任务更新	FINISHED	结束任务并释放机器人

3.7.3 任务下发

3.7.3.1 GO_RECEIVE

当主机系统希望 RMS 调度机器人去某辊筒工作站收货时，可使用此指令。

表示机器人去指定工作站去收取一定数量的包裹，当完成相关动作后此任务并不会结束，等待主机系统进行任务更新。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskType	String	必要	任务类型。必须为“CARRY_PACKAGE”	CARRY_PACKAGE
instruction	String	必要	表示调度机器人去某辊筒工作站收货，必须为“GO_RECEIVE”	GO_RECEIVE
startStationId	Integer	二选一	去收包裹的辊筒工作站 id	1
hostStartStationCode	String		去收包裹的辊筒工作站在主机系统中的外部编码。 只有主机系统使用自有的工作站编码并将编码同步到 RMS 才有效。	geek_PL00001
packageCount	Integer	必要	包裹数量，表示接收货物的数量	2

字段	类型	是否必要	描述	示例
ext1	Integer	可选	通过 RMS 传递给机器人上位辊筒机构的参数	1
ext2	String	可选	通过 RMS 传递给机器人上位辊筒机构的参数	ss

请求报文示例如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskRequestMsg",
  "request": {
    "header": {
      "requestId": "9194E002AD324176974AE9438A011EE6",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "taskType": "CARRY_PACKAGE",
      "instruction": "GO_RECEIVE",
      "startStationId": 1,
      "packageCount": 2,
      "ext1": 1,
      "ext2": "1"
    }
  }
}
```


此报文表示主机系统希望 RMS 调度机器人去 1 号辊筒工作站收取 2 个货物。

RMS 的响应示例如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskResponseMsg",
  "response": {
    "header": {
      "responseId": "9194E002AD324176974AE9438A011EE6",
      "code": 0,
      "msg": "Success"
    },
    "body": {
      "taskId": 30
    }
  }
}
```

3.7.3.2 GO_DROP

当主机系统希望 RMS 调度机器人去某辊筒工作站收取一定数量的货物，在收货完毕后再调度机器人去某个辊筒工作站将身上所有货物进行投递，且在投递完成后任务结束，可使用此指令。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskType	String	必要	任务类型。必须为“CARRY_PACKAGE”	CARRY_PACKAGE
instruction	String	必要	表示调度机器人去某辊筒工作站收货，必须为“GO_DROP”	GO_DROP
startStationId	Integer	二选一	去收包裹的辊筒工作站 id	1
hostStartStationCode	String		去收包裹的辊筒工作站在主机系统中的外部编码。 只有主机系统使用自有的工作站编码并将编码同步到 RMS 才有效。	geek_PL00001
stationId	Integer	二选一	去投递包裹的辊筒工作站 id	2
hostStationCode	String		去投递包裹的辊筒工作站在主机系统中的外部编码。 只有主机系统使用自有的工作站编码并将编码同步到 RMS 才有效。	geek_PL00002
packageCount	Integer	必要	包裹数量，表示接收货物的数量	2
ext1	Integer	可选	通过 RMS 传递给机器人上位辊筒机构的参数	1
ext2	String	可选	通过 RMS 传递给机器人上位辊筒机构的参数	ss

请求报文示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskRequestMsg",
   "request": {
      "header": {
         "requestId": "9194E002AD324176974AE9438A011EE6",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
```

```
            "userKey": "123456",
            "language": "en_us",
            "version": "3.3.0"
        },
        "body": {
            "taskType": "CARRY_PACKAGE",
            "instruction": "GO_DROP",
            "startStationId": 1,
            "stationId": 2,
            "ext1": 1,
            "ext2": "1"
        }
    }
}
```


此报文表示主机系统希望 RMS 调度机器人去 1 号辊筒工作站收取 2 个货物，然后再去 2 号工作站将货物全部投递。

RMS 的响应示例如下：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskResponseMsg",
    "response": {
        "header": {
            "responseId": "9194E002AD324176974AE9438A011EE6",
            "code": 0,
            "msg": "Success"
        },
        "body": {
            "taskId": 30
        }
    }
}
```


3.7.3.3 GO_NEXT

当主机系统希望 RMS 调度机器人去某指定地点或工位时，可使用此指令。

机器人完成动作后，任务不结束并等待更新。可用于人工放货、取货。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskType	String	必要	任务类型。必须为“CARRY_PACKAGE”	CARRY_PACKAGE
instruction	String	必要	表示调度机器人去某辊筒工作站收货，必须为“GO_NEXT”	GO_NEXT
目的地描述	系统会按照以下顺序选择目的地, 目的地>工作站，这些值都会参与校验，若多种目的地格			

字段	类型	是否必要	描述	示例
	式均填写了，则指代地点必须一致，否则报错。目的地至少需要填写一个。			
dest	IPoint	可选	目的地：终点坐标索引（二维码适用）。	{"z":1,"x":45,"y":42}
destCellCode	String	可选	目的地：单元格编码	02350125
destHostCode	String	可选	目的地：主机系统对机器人目的点的描述。只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekDest01
stationId	Integer	可选	工作站：编码。 此字段表示工作站在 RMS 内部的编码。	1000
hostStationCode	String	可选	工作站：主机系统中的外部编码。 只有主机系统使用自有的工作站编码并将编码同步到 RMS 才有效。	geek_PL00002
ext1	Integer	可选	通过 RMS 传递给机器人上位辊筒机构的参数	1
ext2	String	可选	通过 RMS 传递给机器人上位辊筒机构的参数	ss

若目的地和工作站的信息都有填写，到目的地的优先级最高。

请求报文示例如下：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskRequestMsg",
  "request": {
    "header": {
      "requestId": "9194E002AD324176974AE9438A011EE6",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "taskType": "CARRY_PACKAGE",
      "instruction": "GO_NEXT",
      "dest": {
        "z": 1,
        "x": 45,
        "y": 42
      }
```

```
        }
    }
}
```

此报文表示主机系统希望 RMS 调度机器人去指定地点，然后等待任务更新。

RMS 的响应示例如下：

```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskResponseMsg",
    "response": {
        "header": {
            "responseId": "9194E002AD324176974AE9438A011EE6",
            "code": 0,
            "msg": "Success"
        },
        "body": {
            "taskId": 30
        }
    }
}
```

3.7.4 任务更新

任务更新时共有四个指令可以选用。

3.7.4.1 GO_RECEIVE

当主机系统希望 RMS 调度机器人继续去某辊筒工作站收货时，可使用此指令。

表示机器人去指定工作站去收取一定数量的包裹，当完成相关动作后此任务并不会结束，等待主机系统进行任务更新。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	表示调度机器人去某辊筒工作站收货，必须为“GO_RECEIVE”	GO_RECEIVE
startStationId	Integer	二选一	去收包裹的辊筒工作站 id	1
hostStartStationCode	String		去收包裹的辊筒工作站在主机系统中的外部编码。 只有主机系统使用自有的工作站编码并将编码同步到 RMS 才有效。	geek_PL00001
packageCount	Integer	必要	包裹数量，表示接收货物的数量	1

字段	类型	是否必要	描述	示例
ext1	Integer	可选	通过 RMS 传递给机器人上位辊筒机构的参数	1
ext2	String	可选	通过 RMS 传递给机器人上位辊筒机构的参数	ss

请求报文示例如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskUpdateRequestMsg",
  "request": {
    "header": {
      "requestId": "9194E002AD324176974AE9438A011EE6",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 30,
      "instruction": "GO_RECEIVE",
      "startStationId": 1,
      "packageCount": 1,
      "ext1": 1,
      "ext2": "1"
    }
  }
}
```


此报文表示主机系统更新了任务，并使 RMS 调度正在执行任务的机器人继续去 1 号辊筒工作站收取 1 个货物。

RMS 的响应示例如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskUpdateResponseMsg",
  "response": {
    "header": {
      "responseId": "9194E002AD324176974AE9438A011EE6",
      "code": 0,
      "msg": "Success"
    }
  }
}
```

3.7.4.2 GO_DROP

当主机系统希望 RMS 调度机器人去某辊筒工作站进行投递时（此时需保证机器人上还有货物可以投递），可使用此指令。

若机器人完成投递动作后无剩余货物，则此任务结束。否则将继续等待任务更新。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	表示调度机器人去某辊筒工作站收货，必须为“GO_DROP”	GO_DROP
stationId	Integer	二选一	去投递包裹的辊筒工作站 id	2
hostStationCode	String		去投递包裹的辊筒工作站在主机系统中的外部编码。 只有主机系统使用自有的工作站编码并将编码同步到 RMS 才有效。	geek_PL00002
packageCount	Integer	必要	包裹数量，表示接收货物的数量	2
ext1	Integer	可选	通过 RMS 传递给机器人上位辊筒机构的参数	1
ext2	String	可选	通过 RMS 传递给机器人上位辊筒机构的参数	ss

请求报文示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "9194E002AD324176974AE9438A011EE6",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 30,
         "instruction": "GO_DROP",
         "stationId": 1,
         "packageCount": 1,
         "ext1": 1,
```

```
            "ext2": "1"
        }
    }
}
```


此报文表示主机系统希望 RMS 调度机器人去 1 号辊筒工作站投递 1 个货物。

RMS 的响应示例如下：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskUpdateResponseMsg",
    "response": {
        "header": {
            "responseId": "9194E002AD324176974AE9438A011EE6",
            "code": 0,
            "msg": "Success"
        }
    }
}
```


3.7.4.3 GO_NEXT

当主机系统希望 RMS 调度机器人去某指定地点或工位时，可使用此指令。

机器人完成动作后，任务不结束并等待更新。可用于人工放货、取货。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	表示调度机器人去某辊筒工作站收货，必须为“GO_NEXT”	GO_NEXT
目的地描述	系系统会按照以下顺序选择目的地, 目的地>工作站，这些值都会参与校验，若多种目的地格式均填写了，则指代地点必须一致，否则报错。目的地至少需要填写一个。			
dest	IPoint	可选	目的地：终点坐标索引（二维码适用）。	{"z":1,"x":45,"y":42}
destCellCode	String	可选	目的地：单元格编码	02350125
destHostCode	String	可选	目的地：主机系统对机器人目的点的描述。只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekDest01
stationId	Integer	可选	工作站：编码。 此字段表示工作站在 RMS 内部的编码。	1000

字段	类型	是否必要	描述	示例
hostStationCode	String	可选	工作站：主机系统中的外部编码。 只有主机系统使用自有的工作站编码并将编码同步到 RMS 才有效。	geek_PL00002
ext1	Integer	可选	通过 RMS 传递给机器人上位辊筒机构的参数	1
ext2	String	可选	通过 RMS 传递给机器人上位辊筒机构的参数	ss

若目的地和工作站的信息都有填写，**到目的地的优先级最高。**

请求报文示例如下：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskUpdateRequestMsg",
  "request": {
    "header": {
      "requestId": "9194E002AD324176974AE9438A011EE6",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 30,
      "instruction": "GO_NEXT",
      "dest": {
        "z": 1,
        "x": 45,
        "y": 42
      }
    }
  }
}
```

此报文表示主机系统希望 RMS 调度机器人去指定地点，然后继续等待任务更新。

RMS 的响应示例如下：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskUpdateResponseMsg",
  "response": {
    "header": {
      "responseId": "9194E002AD324176974AE9438A011EE6",
```

```
            "code": 0,
            "msg": "Success"
        }
    }
}
```


3.7.4.4 FINISHED

当用户使用“GO_NEXT”指令移动机器人时，希望结束此任务并释放机器人，则可以使用此指令。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	表示调度机器人去某辊筒工作站收货，必须为“FINISHED”	FINISHED

请求报文示例如下：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskUpdateRequestMsg",
    "request": {
        "header": {
            "requestId": "9194E002AD324176974AE9438A011EE6",
            "clientCode": "geekCode",
            "warehouseCode": "geekWarehouseCode",
            "userId": "admin",
            "userKey": "123456",
            "language": "en_us",
            "version": "3.3.0"
        },
        "body": {
            "taskId": 30,
            "instruction": "FINISHED"
        }
    }
}
```


此报文表示主机系统希望 RMS 结束此任务，并释放机器人。

RMS 的响应示例如下：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskUpdateResponseMsg",
    "response": {
        "header": {
            "responseId": "9194E002AD324176974AE9438A011EE6",
            "code": 0,
```

```
            "msg": "Success"
        }
    }
}
```

3.7.5 任务回调

机器人在执行“CARRY_PACKAGE”任务时，根据当前执行阶段的不同，RMS 会主动向主机系统发送回调消息，告知主机系统此任务目前的任务状态及执行阶段。

以下为“CARRY_PACKAGE”可能出现的任务执行阶段“taskPhase”。

1、当主机系统给 RMS 下发“GO_RECEIVE”的指令时，无论此指令是任务下发还是任务更新，机器人在执行过程中会产生 3 个回调：MOVING、ARRIVED、RECEIVE_FINISH。

2、当主机系统给 RMS 下发“GO_DROP”的指令时，若是任务下发，机器人在执行过程中会产生 6 个回调：MOVING、ARRIVED、RECEIVE_FINISH、MOVING、ARRIVED、DROP_FINISH。若是任务更新，则机器人在执行过程中会产生 3 个回调：MOVING、ARRIVED、DROP_FINISH。

3、当主机系统给 RMS 下发“GO_NEXT”的指令时，无论此指令是任务下发还是任务更新，机器人在执行过程中会产生 2 个回调：MOVING、ARRIVED。

4、当主机系统给 RMS 下发“FINISHED”的指令时，RMS 会立即响应，但不会发送回调。

注意，任务是否结束仍然以“taskStatus”为准，当且仅当“taskStatus”为“COMPLETED”时，表示任务已结束。

taskPhase	发送时机
MOVING	机器人开始运动去某地
ARRIVED	机器人到达某地
RECEIVE_FINISH	机器人收货完成
DROP_FINISH	机器人投递完成

CARRY_PACKAGE 任务的回调消息字段说明如下：

字段	类型	描述	示例
taskId	long	当前进行的任务 id	30
taskType	String	产生该回调的任务类型	CARRY_PACKAGE
taskStatus	String	EXECUTING 执行中 CANCELED 取消 COMPLETED 完成	EXECUTING

字段	类型	描述	示例
instruction	String	表示任务在执行哪个指令时发起的回调	GO_NEXT
taskPhase	String	回调的阶段	MOVING
packageCount	int	包裹数量，表示接收/投递货物的数量	2
carryAmount	int	目前机器人身上的货物总数量	2
ext1	int	通过 RMS 传递给机器人上位辊筒机构的参数	1
ext2	String	通过 RMS 传递给机器人上位辊筒机构的参数	ss
robotId	int	执行任务的机器人 ID	1000
destLocation	DPoint	终点：绝对坐标	{"z": 1,"x": 29.5,"y": 10.5}
dest	IPoint	终点：索引坐标	{"z": 1,"x": 29,"y": 10}
destCellCode	String	终点：节点编码	102950105
destHostCode	String	终点：节点外部编码	546544
startStationId	int	收货工作站 注：指令终点为工作站才会有此值	1
stationId	int	投递工作站 注：指令终点为工作站才会有此值	1
hostStationCode	String	投递工作站外部编码 注：指令终点为工作站才会有此值	station1

以下将给出 CARRY_PACKAGE 任务不同阶段的回调消息示例。

3.7.5.1 MOVING

MOVING 阶段的回调消息示例如下：


```
{
   "id": "*",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
         "version": "3.3.0"
      },
```

```
    "body": {
      "taskId": 141,
      "taskType": "CARRY_PACKAGE",
      "taskStatus": "EXECUTING",
      "instruction": "GO_NEXT",
      "taskPhase": "MOVING",
      "robotId": 1293435,
      "dest": {
        "z": 1,
        "x": 138,
        "y": 67
      },
      "destLocation": {
        "z": 1,
        "x": 163.755,
        "y": 82.442
      },
      "destCellCode": "13850675"
    }
  }
}
```


3.7.5.2 ARRIVED

ARRIVED 阶段的回调消息示例如下：


```
{
  "id": "*",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWarehouseCode",
      "requestId": "d63a3b3cc13543d8aae52b1fc62abe60",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 141,
      "taskType": "CARRY_PACKAGE",
      "taskStatus": "EXECUTING",
      "instruction": "READY",
      "taskPhase": "ARRIVED",
      "robotId": 1293435
    }
  }
}
```


3.7.5.3 RECEIVE_FINISH

RECEIVE_FINISH 阶段的回调消息示例如下：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWareHouseCode",
      "requestId": "9194E002AD324176974AE9438A011EE6",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 30,
      "taskType": "CARRY_PACKAGE",
      "taskStatus": "EXECUTING",
      "taskPhase": "RECEIVE_FINISH",
      "robotId": 1000,
      "destLocation": {
        "z": 1,
        "x": 29.5,
        "y": 10.5
      },
      "dest": {
        "z": 1,
        "x": 29,
        "y": 10
      },
      "destCellCode": "10950105",
      "destHostCode": "45464",
      "startStationId": 1,
      "stationId": 2,
      "packageCount": 1,
      "carryAmount": 2,
      "ext1": 1,
      "ext2": "1"
    }
  }
}
```


3.7.5.4 DROP_FINISH

DROP_FINISH 阶段的回调消息示例如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWareHouseCode",
      "requestId": "9194E002AD324176974AE9438A011EE6",
      "version": "3.3.0"
    },
```

```
    "body": {
      "taskId": 30,
      "taskType": "CARRY_PACKAGE",
      "taskStatus": "EXECUTING",
      "taskPhase": "DROP_FINISH",
      "robotId": 1000,
      "destLocation": {
        "z": 1,
        "x": 29.5,
        "y": 10.5
      },
      "dest": {
        "z": 1,
        "x": 29,
        "y": 10
      },
      "destCellCode": "10950105",
      "destHostCode": "45464",
      "startStationId": 1,
      "stationId": 2,
      "packageCount": 1,
      "carryAmount": 2,
      "ext1": 1,
      "ext2": "1"
    }
  }
}
```


3.7.6 功能注意点

无。

3.8 DELIVER_PALLET

此任务是托盘的搬运任务，目前只有叉车可以执行此类任务。

3.8.1 场景说明

托盘任务，支持从一个位置叉取托盘，然后再将托盘送到另一个位置并放下。还支持从一个位置叉取托盘，送到某个位置不放下，等待上游去更新指令。

3.8.2 可用指令

DELIVER_PALLET 支持的指令如下，具体用法会在下面的部分进行详细叙述。

行为	指令	用途
任务下发	GO_FETCH	指派机器人搬运托盘去某处，保持负载
任务下发	GO_RETURN	指派机器人搬运托盘去某处，卸载
任务更新	GO_FETCH	机器人搬运托盘去某处，保持负载
任务更新	GO_RETURN	机器人搬运托盘去某处，卸载
任务更新	CANCEL	取消任务

3.8.3 任务下发

3.8.3.1 GO_FETCH

任务下发时指定 GO_FETCH 模式，将会从某个位置叉取托盘，然后送到某个位置，并等待上游的后续更新。

字段	类型	是否必要	描述	示例
taskType	String	必要	必须为 DELIVER_PALLET	DELIVER_PALLET
instruction	String	必要	机器人到目的地后继续举着托盘，此时该任务不会结束，RMS 将继续等待主机系统发送任务更新消息。	GO_FETCH
startPalletLatticeCode	String	可选	叉取托盘的托盘位编码。 此字段表示托盘位在 RMS 中的内部编码。	PL0000001
startPalletLatticeHostCode	String	可选	叉取托盘的托盘位在主机系统中的外部编码。 只有主机系统使用自有的托盘位编码并将编码同步到 RMS 才有效。	geek_PL00001
priority	Integer	可选	该参数用来设置任务的优先级。值越大，表明在分配机器人执行任务期间该任务的优先级越高，priority 取值范围[0-99]	2
startDest	DPoint	可选	叉取托盘的托盘位坐标。三	{"x":45,"y":42} {"z":1,"x":45,"y":42}

字段	类型	是否必要	描述	示例
			维坐标，即 z 轴，x 轴，y 轴。z 可省略,默认是 1(二维码区域专用,传索引值)	
startDestCellCode	String	可选	叉取托盘的托盘位的单元格编码。	202350125
startDestHostCode	String	可选	主机系统对机器人叉取托盘的托盘位的描述。 只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekDest
startLayer	Integer	可选	叉取托盘的托盘位的层数	2
dest	DPoint	可选	送托盘的目的地坐标。三维坐标，即 z 轴，x 轴，y 轴。z 可省略,默认是 1(二维码区域专用,传索引值)	{"x":45,"y":42} {"z":1,"x":45,"y":42}
destCellCode	String	可选	送托盘的目的地的单元格编码。	202350125
destHostCode	String	可选	主机系统对机器人送托盘的目的地的单元格描述。 只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekDest

使用 GO_FETCH 的情况下，起始点如果使用托盘位，“palletLatticeCode”、“palletLatticeHostCode”两个字段至少填写一个，若同时填写，两者表示的托盘位必须是一致的;

如果使用单元格加层数，“dest”、“destCellCode”、“destHostCode”三个字段至少填写一个，若同时填写，则三个对象表示的位置需要一致。且系统会校验该位置是否有托盘架。

如果同时传了托盘位和单元格加层数，则优先使用托盘位。

目的点只是机器人停留位置，使用单元格，“dest”、“destCellCode”、“destHostCode”三个字段至少填写一个，若同时填写，则单个对象表示的位置需要一致。

请求消息示例：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskRequestMsg",
   "request": {
      "header": {
         "requestId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
```

```
            "clientCode": "geekCode",
            "warehouseCode": "geekWarehouseCode",
            "userId": "admin",
            "userKey": "123456",
            "language": "en_us",
            "version": "3.3.0"
        },
        "body": {
            "taskType": "DELIVER_PALLET",
            "instruction": "GO_FETCH",
            "startDestCellCode":"157",
            "startLayer":1,
            "destCellCode":32
        }
    }
}
```


该消息表示，要从 157 号单元格的一层位置叉取托盘（如果该位置的一层没有对应一个托盘位，则报错），然后将托盘搬运到 32 号单元格处

RMS 响应信息如下


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskResponseMsg",
    "response": {
        "header": {
            "responseId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
            "code": 0,
            "msg": "Success"
        },
        "body": {
            "taskId": 30
        }
    }
}
```


3.8.3.2 GO_RETURN

任务下发时指定 GO_RETURN 模式，将会从某个位置叉取托盘，然后送到某个位置并放下托盘，然后结束任务。

字段	类型	是否必要	描述	示例
taskType	String	必要	必须为 DELIVER_PALLET	DELIVER_PALLET
instruction	String	可选	机器人到目的地后将托盘放下，此任务结束。	GO_RETURN

字段	类型	是否必要	描述	示例
startPalletLatticeCode	String	可选	叉取托盘的托盘位编码。 此字段表示托盘位在 RMS 中的内部编码。	PL0000001
startPalletLatticeHostCode	String	可选	叉取托盘的托盘位在主机系统中的外部编码。 只有主机系统使用自有的托盘位编码并将编码同步到 RMS 才有效。	geek_PL00001
destPalletLatticeCode	String	可选	放置托盘的托盘位编码。 此字段表示托盘位在 RMS 中的内部编码。	PL0000002
destPalletLatticeHostCode	String	可选	放置托盘的托盘位在主机系统中的外部编码。 只有主机系统使用自有的托盘位编码并将编码同步到 RMS 才有效。	geek_PL00002
priority	Integer	可选	该参数用来设置任务的优先级。值越大，表明在分配机器人执行任务期间该任务的优先级越高，priority 取值范围[0-99]	2
startDest	DPoint	可选	叉取托盘的托盘位坐标。三维坐标，即 z 轴，x 轴，y 轴。z 可省略,默认是 1(二维码区域专用,传索引值)	{"x":45,"y":42} {"z":1,"x":45,"y":42}
startDestCellCode	String	可选	叉取托盘的托盘位的单元格编码。	202350125
startDestHostCode	String	可选	主机系统对机器人叉取托盘的托盘位的描述。 只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekDest
startLayer	Integer	可选	叉取托盘的托盘位的层数	2
destCellCode	String	可选	放置托盘的托盘位的单元格编码。	202350125
destHostCode	String	可选	主机系统对机器人放置托盘的托盘位的描述。 只有主机系统使用自有的位置	geekDest

字段	类型	是否必要	描述	示例
			编码并同步到 RMS 才有效。	
layer	Integer	可选	放置托盘的托盘位的层数	2

请求消息示例：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskRequestMsg",
  "request": {
    "header": {
      "requestId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "taskType": "DELIVER_PALLET",
      "instruction": "GO_RETURN",
      "startPalletLatticeCode":"PL007",
      "destPalletLatticeCode":"PL008"
    }
  }
}
```


该消息表示，要从托盘位 PL007 的位置叉取托盘,然后送到 PL008 处放下托盘

RMS 响应消息示例：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskResponseMsg",
  "response": {
    "header": {
      "responseId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
      "code": 0,
      "msg": "Success"
    },
    "body": {
      "taskId": 30
    }
  }
}
```

3.8.4 任务更新

3.8.4.1 GO_FETCH

搬运托盘到目的地不放下托盘指令所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 GO_FETCH
taskId	long	必要	当前进行的任务 id
destCellCode	String	二选一	目的地单元格编码
destHostCode	String		外部坐标描述

请求消息示例：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "411ccc219da94e7f8680dfdee39c43d5",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 1199,
         "instruction": "GO_FETCH",
         "destCellCode":"01250285"
      }
   }
}
```

响应消息示例：

```
{
   "id":"geekCode_geekWarehouseCode_001",
   "msgType":"RobotTaskUpdateResponseMsg",
   "response":{
      "header":{
         "responseId":"411ccc219da94e7f8680dfdee39c43d5",
         "code":0,
         "msg":"Success"
      }
```

```
    }
}
```

3.8.4.2 GO_RETURN

搬运托盘到目的地放下托盘指令所需字段

字段	类型	是否必要	功能描述
instruction	String	必要	值必须 GO_RETURN
taskId	long	必要	当前进行的任务 id
destPalletLatticeCode	String	可选	目标托盘位编码
destPalletLatticeHostCode	String	可选	目标托盘位外部编码
layer	int	必要	放置托盘的托盘位的层数
destCellCode	String	二选一，也可以均不填写，与 layer 配合使用	目的地单元格编码
destHostCode	String		外部坐标描述

请求消息示例：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "411ccc219da94e7f8680dfdee39c43d5",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 1199,
         "instruction": "GO_RETURN",
         "destPalletLatticeCode":"PL003"
      }
   }
}
```

响应消息示例：

```
{
   "id":"geekCode_geekWarehouseCode_001",
   "msgType":"RobotTaskUpdateResponseMsg",
   "response":{
```

```
        "header":{
            "responseId":"411ccc219da94e7f8680dfdee39c43d5",
            "code":0,
            "msg":"Success"
        }
    }
}
```

3.8.4.3 CANCEL

取消搬运托盘任务			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 CANCEL
taskId	long	必要	当前进行的任务 id
cancelAction	int	必要	0 返回取托盘位置(叉车空载报错) 1 到达任务目的点，（空载报错） 2 原地停止（负载报错） 3 到达指定目的地（空载时，只需要传目的单元格，负载的情况下需要传单元格及层数(或者目标托盘位)）
destPalletLatticeCode	String	可选，二选一	目标托盘位编码
destPalletLatticeHostCode	String		目标托盘位外部编码
layer	int	层数，与目的点组合使用，找到相应的托盘位	层数，与目的点组合使用，找到相应的托盘位
destCellCode	String	二选一，也可以均不填写	目的地单元格编码
destHostCode	String		目的地外部描述

目标托盘位信息与目的地信息至少填写一个。

请求消息示例：

```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskUpdateRequestMsg",
    "request": {
        "header": {
            "requestId": "411ccc219da94e7f8680dfdee39c43d5",
            "clientCode": "geekCode",
            "warehouseCode": "geekWarehouseCode",
            "userId": "admin",
            "userKey": "123456",
            "language": "en_us",
```

```
            "version": "3.3.0"
        },
        "body": {
            "taskId": 1199,
            "instruction": "CANCEL",
            "cancelAction":3,
            "destCellCode":"01250285",
             "layer":3
        }
    }
}
```


响应消息示例：


```
{
    "id":"geekCode_geekWarehouseCode_001",
    "msgType":"RobotTaskUpdateResponseMsg",
    "response":{
        "header":{
            "responseId":"411ccc219da94e7f8680dfdee39c43d5",
            "code":0,
            "msg":"Success"
        }
    }
}
```


3.8.5 任务回调

机器人在执行“DELIVER_PALLET”任务时，根据当前执行阶段的不同，RMS 会主动向主机系统发送回调消息，告知主机系统此任务目前的任务状态及执行阶段。

以下为“DELIVER_PALLET”可能出现的任务执行阶段“taskPhase”。

taskPhase	发送时机
GO_FETCHING	机器人出发去取托盘
FETCHED	机器人取托盘完成
GO_DELIVERING	机器人送托盘出发到某处
GO_RETURN	机器人出发到某地放托盘
ARRIVED	机器人到达目的地
CANCELED	任务取消

以下将给出 DELIVER_PALLET 任务不同阶段的回调消息示例。

3.8.5.1 GO_FETCHING

GO_FETCHING 阶段的回调消息示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWareHouseCode",
         "requestId": "411ccc219da94e7f8680dfdee39c43d5",
         "version": "3.3.0"
      },
      "body": {
         "taskId":26,
         "taskType":"DELIVER_PALLET",
         "taskStatus":"EXECUTING",
         "taskPhase":"GO_FETCHING",
         "robotId":80012
         "destLocation":{
            "z":1,
            "x":34.373,
            "y":25.533
         },
         "destCellCode":"29",
         "startPalletLatticeHostCode":"PL001",
         "destPalletLatticeCode":"PL002"
      }
   }
}
```


3.8.5.2 FETCHED

FETCHED 阶段的回调消息示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWareHouseCode",
         "requestId": "411ccc219da94e7f8680dfdee39c43d5",
         "version": "3.3.0"
      },
      "body": {
         "taskId":26,
         "taskType":"DELIVER_PALLET",
         "taskStatus":"EXECUTING",
         "taskPhase":"FETCHED",
         "robotId":80012
```

```
            "destLocation":{
               "z":1,
               "x":34.373,
               "y":25.533
            },
            "destCellCode":"29",
            "startPalletLatticeHostCode":"PL001",
            "destPalletLatticeCode":"PL002"
         }
      }
   }
```


3.8.5.3 GO_DELIVERING

GO_DELIVERING 阶段的回调消息示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWareHouseCode",
         "requestId": "411ccc219da94e7f8680dfdee39c43d5",
         "version": "3.3.0"
      },
      "body": {
         "taskId":26,
         "taskType":"DELIVER_PALLET",
         "taskStatus":"EXECUTING",
         "taskPhase":"GO_DELIVERING",
         "robotId":80012
         "destLocation":{
            "z":1,
            "x":34.373,
            "y":25.533
         },
         "destCellCode":"29",
         "startPalletLatticeHostCode":"PL001"
      }
   }
}
```


3.8.5.4 ARRIVED

ARRIVED 阶段的回调消息示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
```

```
        "header": {
            "warehouseCode": "geekWareHouseCode",
            "requestId": "411ccc219da94e7f8680dfdee39c43d5",
            "version": "3.3.0"
        },
        "body": {
            "taskId":26,
            "taskType":"DELIVER_PALLET",
            "taskStatus":"EXECUTING",
            "taskPhase":"ARRIVED",
            "robotId":80012
            "destLocation":{
                "z":1,
                "x":34.373,
                "y":25.533
            },
            "destCellCode":"29",
            "startPalletLatticeHostCode":"PL001",
            "destPalletLatticeCode":"PL002"
        }
    }
}
```


3.8.5.5 GO_RETURN

GO_RETURN 阶段的回调消息示例如下：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskCallbackMsg",
    "request": {
        "header": {
            "warehouseCode": "geekWareHouseCode",
            "requestId": "411ccc219da94e7f8680dfdee39c43d5",
            "version": "3.3.0"
        },
        "body": {
            "taskId":26,
            "taskType":"DELIVER_PALLET",
            "taskStatus":"EXECUTING",
            "taskPhase":"GO_RETURN",
            "robotId":80012
            "destLocation":{
                "z":1,
                "x":34.373,
                "y":25.533
            },
            "destCellCode":"29",
            "startPalletLatticeHostCode":"PL001",
            "destPalletLatticeCode":"PL002"
```

```
        }
    }
}
```

3.8.5.6 CANCELED

任务取消后的回调消息示例如下：

```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskCallbackMsg",
    "request": {
        "header": {
            "warehouseCode": "geekWareHouseCode",
            "requestId": "411ccc219da94e7f8680dfdee39c43d5",
            "version": "3.3.0"
        },
        "body": {
            "taskId":26,
            "taskType":"DELIVER_PALLET",
            "taskStatus":"CANCELED",
            "robotId":80012
            "destLocation":{
                "z":1,
                "x":34.373,
                "y":25.533
            },
            "destCellCode":"29",
            "startPalletLatticeHostCode":"PL001",
            "destPalletLatticeCode":"PL002"
        }
    }
}
```

3.8.6 功能注意点

无。

3.8.7 场景示例

例如一个托盘任务要从 PL001 处去取一个托盘，然后先送到 32 号单元格处进行工作，然后，再把托盘放到 PL002 处。

1.任务下发内容

```
"body": {
    "taskType": "DELIVER_PALLET",
    "instruction": "GO_FETCH",
    "startPalletLatticeHostCode":"PL001",
    "destCellCode":32
```

}
2.开始取托盘，发送取托盘回调
"body": {
"taskId":26,
"taskType":"DELIVER_PALLET",
"taskStatus":"EXECUTING",
"taskPhase":"GO_FETCHING",
"robotId":80012
"destLocation":{
"z":1,
"x":34.373,
"y":25.533
},
"destCellCode":"12",
"startPalletLatticeHostCode":"PL001"
}
3.取到托盘，发送托盘已取到回调
"body": {
"taskId":26,
"taskType":"DELIVER_PALLET",
"taskStatus":"EXECUTING",
"taskPhase":"FETCHED",
"robotId":80012
"destLocation":{
"z":1,
"x":34.373,
"y":25.533
},
"destCellCode":"32",
"startPalletLatticeHostCode":"PL001"
}
4.送托盘，发送送托盘回调
"body": {
"taskId":26,
"taskType":"DELIVER_PALLET",
"taskStatus":"EXECUTING",
"taskPhase":"GO_DELIVERING",
"robotId":80012
"destLocation":{
"z":1,
"x":14.373,
"y":15.533
},
"destCellCode":"32",
"startPalletLatticeHostCode":"PL001"
}
5.托盘送达，发送托盘已送到回调
"body": {

```
    "taskId":26,
    "taskType":"DELIVER_PALLET",
    "taskStatus":"EXECUTING",
    "taskPhase":"ARRIVED",
    "robotId":80012
    "destLocation":{
        "z":1,
        "x":14.373,
        "y":15.533
    },
    "destCellCode":"32",
    "startPalletLatticeHostCode":"PL001"
}
```


然后等待上游更新.

6.上游更新把托盘放到 PL002 这个托盘位处


```
"body": {
    "taskId": 1199,
    "instruction": "GO_RETURN",
    "destPalletLatticeCode":"PL002"
}
```


7.开始还托盘，发送还托盘回调


```
"body": {
    "taskId":26,
    "taskType":"DELIVER_PALLET",
    "taskStatus":"EXECUTING",
    "taskPhase":"GO_RETURN",
    "robotId":80012
    "destLocation":{
        "z":1,
        "x":34.373,
        "y":35.533
    },
    "destCellCode":"55",//托盘要还的目的地
    "startPalletLatticeHostCode":"PL001",
    "destPalletLatticeCode":"PL002"
}
```


8.托盘还到目的地，发送已经送还的回调


```
"body": {
    "taskId":26,
    "taskType":"DELIVER_PALLET",
    "taskStatus":"COMPLETED",
    "taskPhase":"ARRIVED",
    "robotId":80012
    "destLocation":{
        "z":1,
        "x":34.373,
        "y":35.533
    },
```

```
    "destCellCode":"55",//托盘要还的目的地
    "startPalletLatticeHostCode":"PL001"，
    "destPalletLatticeCode":"PL002"
}
```

3.9 DELIVER_BOX

此任务类型适用于货箱的拣选或搬运场景，目前仅抱箱机器人（即 C200S 型机器人）可执行此类任务。

此任务类型较为灵活，根据情况选择请求字段进行组合，以实现业务需求。

目前支持的工作站类型仅为单点工位，非单点工位的工作站尚不支持。

3.9.1 场景说明

3.9.1.1 取箱&还箱

调度机器人取一个货箱，并送到某个点或某个工作站，待工人完成拣货等相关操作后，再由机器人将此货箱送还。

也可以调度机器人取一个货箱，并送到某个点或某个工作站，然后由工人将机器人身上的货箱置换为一个新的货箱，再将新的货箱送还。此操作会在 RMS 系统中移除原有的货箱，并添加新的货箱。

3.9.1.2 多点送货

调度机器人取一个货箱并送至某点/工作站后，可以继续调度这个机器人将货箱送至其他点/工作站。

或者调度一个空载的机器人到某点/工作站，然后继续调度此机器人去往其他点/工作站。

3.9.1.3 新货箱入库

调度一个空载的机器人到某点或某工作站，然后由工人放一个箱子上去，再将此箱子送回存储区。此操作会在 RMS 系统中添加新的货箱。

3.9.1.4 移除货箱

调度机器人取一个货箱，并送到某个点或某工作站，然后由工人将此货箱从机器人身上拿走，并结束此任务。此操作会在 RMS 系统中移除货箱。

3.9.1.5 货箱转移

调度机器人取一个货箱，并将此货箱放置到指定的格子（即货位）上。

3.9.2 可用指令

DELIVER_BOX 支持的指令如下，具体用法会在下面的部分进行详细叙述。

行为	指令	用途
任务下发	GO_FETCH	指派机器人搬运箱子去某处，保持负载
任务下发	GO_RETURN	指派机器人搬运箱子去某处，卸载
任务下发	GO_NEXT	指派机器人去某处，等待指令更新
任务更新	UPDATE_BOX	更新机器人上的箱子信息
任务更新	GO_FETCH	指派机器人搬运箱子去某处，保持负载
任务更新	GO_RETURN	指派机器人搬运箱子去某处，卸载
任务更新	GO_NEXT	指派机器人去某处，等待指令更新
任务更新	CANCEL	取消任务
任务更新	CANCEL_INSTRUCTION	取消动作

3.9.3 任务下发

3.9.3.1 GO_FETCH

当主机系统希望 RMS 调度机器人去取一个货箱并送至某点/某工作站，可使用此指令。机器人完成相关动作后此任务并不会结束，需要主机系统进行任务更新。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskType	String	必要	任务类型。必须为“DELIVER_BOX”	DELIVER_BOX
instruction	String	必要	表示取箱子至某地，或空载至某地，必须为“GO_FETCH”	GO_FETCH
boxCode	String	必要	货箱编号。	B0000001

字段	类型	是否必要	描述	示例
riseBoard	boolean	可选	到达目的地后是否升起拨指。 默认为 false，即到达目的地后不升起拨指。	true
deliverHeight	Integer	可选	到达目的地后抱叉停放高度。 单位：毫米 默认为 1115 毫米，即到达目的地后抱叉高度保持在 1115 毫米。	1115
目的地描述	目的地相关的值都会参与校验。若多种目的地格式均填写了，则指代地点必须一致，否则接口提示异常。目的地和工作站至少填写一个。			
dest	IPoint	可选	目的地：终点坐标。三维坐标，即 z 轴，x 轴，y 轴。	{"z":1,"x":45,"y":42}
destCellCode	String	可选	目的地：单元格编码	02350125
destHostCode	String	可选	目的地：主机系统对机器人目的点的描述。 只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekDest01
stationId	Integer	可选	工作站：送货箱去的工作站 id。 此字段表示工作站在 RMS 内部的 id。	1000
hostStationCode	String	可选	工作站：送货箱去的工作站 id 在主机系统中的外部编码。 只有主机系统使用自有的工作站编码并将编码同步到 RMS 才有效。	geek_PL00002

注意，若希望到某个目的地，“dest”、“destCellCode”、“destHostCode”三个字段至少填写一个，若同时填写，则三个值的表述必须一致。

若希望到某个工作站，“stationId”和“hostStationCode”两个字段至少填写一个，若同时填写，则两个值的表述必须一致。

若目的地和工作站的信息都有填写，则目的地的优先级最高。

请求报文示例如下：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskRequestMsg",
  "request": {
    "header": {
      "requestId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
```

```
            "userId": "admin",
            "userKey": "123456",
            "language": "en_us",
            "version": "3.3.0"
        },
        "body": {
            "taskType": "DELIVER_BOX",
            "instruction": "GO_FETCH",
            "boxCode":"B0000038",
            "stationId":1
        }
    }
}
```


此报文表示主机系统希望 RMS 调度机器人去取“B0000038”号货箱，然后把此货箱送至 1 号工作站。

RMS 的响应示例如下：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskResponseMsg",
    "response": {
        "header": {
            "responseId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
            "code": 0,
            "msg": "Success"
        },
        "body": {
            "taskId": 30
        }
    }
}
```


3.9.3.2 GO_RETURN

当主机系统希望 RMS 调度机器人取一个货箱，并将此货箱放置到指定的格子（即货位）上时，使用此指令。机器人完成相关动作后此任务直接结束。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskType	String	必要	必须为“DELIVER_BOX”	DELIVER_BOX
instruction	String	必要	表示取箱子并放至到指定格子，必须为“GO_RETURN”	GO_RETURN
boxCode	String	必要	货箱编号	B0000001
latticeCode	String	必要	表示送还货箱至指定的格子	L0000002

字段	类型	是否必要	描述	示例
destAsPlace	Boolean	可选	表示是否将此次任务指定的终点格子设置为货箱的老家格子。 true，将指定的终点格子设为货架的老家格子 false，不作设置 默认为 false（为空视为 false），即不会重设货箱的老家格子	true

请求报文示例如下：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskRequestMsg",
  "request": {
    "header": {
      "requestId": "a5a3eb48-1e10-4829-80cb-3b181d5339ad",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "taskType": "DELIVER_BOX",
      "instruction": "GO_RETURN",
      "boxCode":"B0000038",
      "latticeCode":"L0000002"
    }
  }
}
```

此报文表示主机系统希望 RMS 调度机器人去取“B0000038”号货箱，然后把此货箱送至 1 号工作站。

RMS 的响应示例如下：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskResponseMsg",
  "response": {
    "header": {
      "responseId": "a5a3eb48-1e10-4829-80cb-3b181d5339ad",
      "code": 0,
      "msg": "Success"
    },
    "body": {
      "taskId": 37
    }
  }
```

}

3.9.3.3 GO_NEXT

当主机系统希望 RMS 调度机器人空载来到某点/某工作站时，可使用此指令。

在发送指令时可以选择机器人完成相关动作后是否结束任务。若选择不结束任务，则需要主机系统进行任务更新。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskType	String	必要	任务类型。必须为“DELIVER_BOX”	DELIVER_BOX
instruction	String	必要	表示取箱子至某地，或空载至某地，必须为“GO_NEXT”	GO_NEXT
isContinue	Integer	可选	表示机器人执行完动作后此任务是否结束。 0，指令执行结束后任务结束。 1，指令执行结束后任务不结束。 默认为 1（为空视为 1），即认为机器人执行完动作后此任务不结束。	0
目的地描述	目的地相关的值都会参与校验。若多种目的地格式均填写了，则指代地点必须一致，否则接口提示异常。目的地和工作站至少填写一个。			
dest	IPoint	可选	目的地：终点坐标。三维坐标，即 z 轴，x 轴，y 轴。	{"z":1,"x":45,"y":42}
destCellCode	String	可选	目的地：单元格编码	02350125
destHostCode	String	可选	目的地：主机系统对机器人目的点的描述。 只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekDest01
stationId	Integer	可选	工作站：机器人空车去的工作站 id。 此字段表示工作站在 RMS 内部的 id。	1000
hostStationCode	String	可选	工作站：机器人空车去的工作站 id 在主机系统中的外部编码。 只有主机系统使用自有的工作站编码并将编码同步到 RMS 才有效。	geek_PL00002

注意，若希望到某个目的地，“dest”、“destCellCode”、“destHostCode”三个字段至少填写一个，若同时填写，则三个值的表述必须一致。

若希望到某个工作站，“stationId”和“hostStationCode”两个字段至少填写一个，若同时填

写，则两个值的表述必须一致。

若目的地和工作站的信息都有填写，到目的地的优先级最高。

请求报文示例如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskRequestMsg",
  "request": {
    "header": {
      "requestId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "taskType": "DELIVER_BOX",
      "instruction": "GO_NEXT",
      "stationId":1
    }
  }
}
```


此报文表示主机系统希望 RMS 调度机器人空车去 1 号工作站，且到达 1 号工作站后任务不结束，等待主机系统更新任务。

RMS 的响应示例如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskResponseMsg",
  "response": {
    "header": {
      "responseId": "9e055daa-a344-4d05-923a-e1c0b7141b96",
      "code": 0,
      "msg": "Success"
    },
    "body": {
      "taskId": 73
    }
  }
}
```


3.9.4 任务更新

3.9.4.1 UPDATE_BOX

当机器人已取到货箱并送至某地后，若希望将机器人身上的货箱置换为一个新的货箱并

添加到 RMS 系统中时，使用此指令。

当机器人已取到货箱并送至某地后，若希望将此货箱从机器人及 RMS 系统中移除时，使用此指令。

或者当 RMS 调度机器人空载到达某地后，若希望将一个新的货箱放到机器人身上并加入到 RMS 系统时，使用此指令。

此指令仅会对 RMS 系统中的资源信息进行操作及修改，不会调度机器人做任何动作，完成后将继续等待上游更新任务。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	表示取箱子并放至到指定格子，必须为“UPDATE_BOX”	UPDATE_BOX
boxOrder	String	必要	表示此次更新货箱信息的命令类型，有以下可选值。 REMOVE：移除货箱 ADD：新增货箱 EXCHANGE：交换货箱（即移除后再新增一个货箱）	EXCHANGE
boxCode	String	可选	根据"boxOrder"的取值不同，此处的 boxCode 会有不同含义 当“boxOrder”值为“EXCHANGE”时：必填，表示交换货箱后新加入的货箱的编号 当“boxOrder”值为“REMOVE”时：可选，若填写则表示将要被移除的货箱 当“boxOrder”值为“ADD”时：必填，表示新加入的货箱的编号	B0000001
placeLatticeCode	String	可选	当“boxOrder”值为“EXCHANGE”时：可选，表示交换货箱后新加入的货箱的库存位置。若不填则交换后货箱的 placeLattice 为被交换货箱的 placeLattice。 当“boxOrder”值为“ADD”时：可选，表示新加入的货箱的库存位置，若不填则新增货箱的 placeLattice 为 null。	L0000002

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
```

```
            "requestId": "9beab88d-a407-4195-a1f8-ae3594c83924",
            "clientCode": "geekCode",
            "warehouseCode": "geekWarehouseCode",
            "userId": "admin",
            "userKey": "123456",
            "language": "en_us",
            "version": "3.3.0"
        },
        "body": {
            "taskId": 30,
            "taskType": "DELIVER_BOX",
            "instruction": "UPDATE_BOX",
            "boxOrder": "EXCHANGE",
            "boxCode": "B00000xx",
            "placeLatticeCode": "L0000002"
        }
    }
}
```


此报文表示主机系统希望将机器人身上的货箱置换成编号为“B00000xx”的货箱，并为此货箱设置库存位置为“L0000002”。

RMS 的响应示例如下：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskUpdateResponseMsg",
    "response": {
        "header": {
            "responseId": "9beab88d-a407-4195-a1f8-ae3594c83924",
            "code": 0,
            "msg": "Success"
        }
    }
}
```


3.9.4.2 GO_RETURN

当机器人已取到货箱并送至某地后，若希望 RMS 调度此机器人将身上的货箱送还至库存区，使用此指令。

机器人完成相关动作后此任务直接结束。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	32
instruction	String	必要	表示取箱子并放至到指定格子，必须为“GO_RETURN”	GO_RETURN

字段	类型	是否必要	描述	示例
目的地描述字段	指定容器格与区域二选一			
latticeCode	String	可选	表示送还货箱至指定的格子	L0000002
destAreaId	Integer	可选	表示送还货箱至指定的逻辑区，RMS 会随机在指定的逻辑区中选择一个空闲的格子作为此次货箱送还的终点格位	1
destAsPlace	Boolean	可选	表示是否将此次任务指定的终点格子设置为货箱的老家格子。 true，将指定的终点格子设为货架的老家格子 false，不作设置 默认为 false（为空视为 false），即不会重设货箱的老家格子	true

注意，若“latticeCode”、“destAreaId”两个字段至少填写一个，若同时填写，则“latticeCode”的优先级最高。

若“latticeCode”、“destAreaId”都没有填写，则会检查货箱老家格子是否为空闲状态，若为空闲状态则将货箱送至货箱的老家格子。

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "629ef636-a97d-4043-a8e5-6547d6e6ca2c",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
       "taskId": 32,
         "instruction": "GO_RETURN"
      }
   }
}
```

此报文表示主机系统希望将机器人身上的货箱送还到其老家格子。

RMS 的响应示例如下，其中 body 体中包含货箱将要被送还的格子信息：

```
{
   "id": "geekCode_geekWarehouseCode_001",
```

```
"msgType": "RobotTaskUpdateResponseMsg",
"response": {
  "header": {
    "responseId": "629ef636-a97d-4043-a8e5-6547d6e6ca2c",
    "code": 0,
    "msg": "Success"
  },
  "body": {
    "LatticeLayer": 1,
    "latticeCode": "L0000002",
    "dest": {
      "z": 1,
      "x": 3,
      "y": 9
    },
    "taskId": 32,
    "boxCode": "B00000xx"
  }
}
}
```


3.9.4.3 GO_NEXT

当主机系统希望 RMS 调度机器人继续去往某点/某工作站时，可使用此指令。

在发送指令时可以选择机器人完成相关动作后是否结束任务。若选择不结束任务，则需要主机系统进行任务更新。

注意，此处若想在机器人完成相关动作后结束任务，则必须保证机器人身上已无货箱，否则校验无法通过。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	32
instruction	String	必要	表示取箱子至某地，或空载至某地，必须为“GO_NEXT”	GO_NEXT
isContinue	Integer	可选	表示机器人执行完动作后此任务是否结束。 0，指令执行结束后任务结束。 1，指令执行结束后任务不结束。 默认为 1（为空视为 1），即认为机器人执行完动作后此任务不结束。	0
目的地描述	目的地相关的值都会参与校验。若多种目的地格式均填写了，则指代地点必须一致，否			

字段	类型	是否必要	描述	示例
	则接口提示异常。目的地和工作站至少填写一个。			
dest	IPoint	可选	目的地：终点坐标。三维坐标，即 z 轴，x 轴，y 轴。	{"z":1,"x":45,"y":42}
destCellCode	String	可选	目的地：单元格编码	02350125
destHostCode	String	可选	目的地：主机系统对机器人目的点的描述。 只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekDest01
stationId	Integer	可选	工作站：机器人空车去的工作站 id。 此字段表示工作站在 RMS 内部的 id。	1000
hostStationCode	String	可选	工作站：机器人空车去的工作站 id 在主机系统中的外部编码。 只有主机系统使用自有的工作站编码并将编码同步到 RMS 才有效。	geek_PL00002

注意，若希望到某个点，“dest”、“destCellCode”、“destHostCode”三个字段至少填写一个，若同时填写，则三个值的表述必须一致。

若希望到某个工作站，“stationId”和“hostStationCode”两个字段至少填写一个，若同时填写，则两个值的表述必须一致。

若目的地和工作站的信息都有填写，到某个点的优先级最高。

请求报文示例如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskUpdateRequestMsg",
  "request": {
    "header": {
      "requestId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 78,
      "instruction": "GO_NEXT",
      "stationId": 1,
```

```
            "isContinue": 0
        }
    }
}
```


此报文表示主机系统希望 RMS 调度机器人空车去 1 号工作站，并且到达之后结束此任务。

RMS 的响应示例如下：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskUpdateResponseMsg",
    "response": {
        "header": {
            "responseId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
            "code": 0,
            "msg": "Success"
        }
    }
}
```


3.9.4.4 GO_FETCH

当主机系统希望 RMS 调度机器人去取一个货箱并送至某点/某工作站，可使用此指令。

机器人完成相关动作后此任务并不会结束，需要主机系统进行任务更新。

注意，此处若想发送任务更新时能够通过 RMS 的校验，则必须保证机器人身上无货箱，否则校验无法通过。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	32
instruction	String	必要	表示取箱子至某地，或空载至某地，必须为“GO_FETCH”	GO_FETCH
boxCode	String	必要	货箱编号	B0000001
riseBoard	boolean	可选	到达目的地后是否升起拨指。 默认为 false，即到达目的地后不升起拨指。	true
deliverHeight	Integer	可选	到达目的地后抱叉停放高度。 单位：毫米 默认为 1115 毫米，即到达目的地后抱叉高度保持在 1115 毫米	1115
目的地描述	目的地相关的值都会参与校验。若多种目的地格式均填写了，则指代地点必须一致，否则			

字段	类型	是否必要	描述	示例
	接口提示异常。目的地和工作站至少填写一个。			
dest	IPoint	可选	目的地：终点坐标。三维坐标，即 z 轴，x 轴，y 轴。	{"z":1,"x":45,"y":42}
destCellCode	String	可选	目的地：单元格编码	02350125
destHostCode	String	可选	目的地：主机系统对机器人目的点的描述。 只有主机系统使用自有的位置编码并同步到 RMS 才有效。	geekDest01
stationId	Integer	可选	工作站：送货箱去的工作站 id。 此字段表示工作站在 RMS 内部的 id。	1000
hostStationCode	String	可选	工作站：送货箱去的工作站 id 在主机系统中的外部编码。 只有主机系统使用自有的工作站编码并将编码同步到 RMS 才有效。	geek_PL00002

注意，若希望到某个目的地，“dest”、“destCellCode”、“destHostCode”三个字段至少填写一个，若同时填写，则三个值的表述必须一致。

若希望到某个工作站，“stationId”和“hostStationCode”两个字段至少填写一个，若同时填写，则两个值的表述必须一致。

若目的地和工作站的信息都有填写，目的地的优先级最高。

请求报文示例如下：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskUpdateRequestMsg",
  "request": {
    "header": {
      "requestId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 78,
      "instruction": "GO_FETCH",
       "boxCode": "B0000038",
      "stationId": 1
    }
```

```
    }
}
```

此报文表示主机系统希望 RMS 调度机器人取“B0000038”货箱去 1 号工作站，并且到达之后等待更新。

RMS 的响应示例如下：

```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskUpdateResponseMsg",
    "response": {
        "header": {
            "responseId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
            "code": 0,
            "msg": "Success"
        }
    }
}
```

3.9.4.5 CANCEL

当主机系统希望 RMS 取消机器人的当前任务时可以使用此指令。

请注意 RMS 的任务取消为一种取消尝试，当机器人已经即将完成任务时取消会有一定失败的可能，是否成功取消请按照 RMS 回调为准。

在收到任务取消指令后，RMS 首先尝试停止机器人，取消机器人的当前子任务并取消后续所有子任务，然后下发机器人的取消动作任务直至完成，回调上游系统任务取消，注意机器人任务取消直至执行取消动作结束中间还可能会发生其他回调信息。

当 RMS 任务即将完成时，机器人任务正常结束，回调上游任务结束 COMPLETED，此时任务取消指令实际上因为任务执行已接近结束导致没有执行。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	32
instruction	String	必要	必须为“CANCEL”	CANCEL
cancelAction	Integer	可选	可选值：0,2,3 以机器人实际状态为准进行处理，参照下方任务的处理模式： 0（默认值）：默认处理 空载：原地释放 负载：若货箱有老家货格且为空闲状态，将放回老家；否则无法通过校验 2：原地处理 空载：原地释放	0

字段	类型	是否必要	描述	示例
			负载：若货箱有老家货格且为空闲状态，将放回老家；否则无法通过校验 3：指定目的地处理 空载：原地释放 负载：若指定货格为空闲状态，将放置到指定格子；否则无法通过校验	
latticeCode	String	可选	格子编号，当且仅当 cancelAction=3 时需要填写	L0000001

请求报文示例如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskUpdateRequestMsg",
  "request": {
    "header": {
      "requestId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 153,
      "instruction": "CANCEL",
      "cancelAction": 3,
      "latticeCode": "L0000038"
    }
  }
}
```


此报文表示主机系统希望 RMS 取消 153 号任务并命令机器人将负载的货箱送到“L0000038”货格，并且在放下货箱后释放机器人。

RMS 的响应示例如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskUpdateResponseMsg",
  "response": {
    "header": {
      "responseId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
      "code": 0,
      "msg": "Success"
    }
  }
}
```

3.9.4.6 CANCEL_INSTRUCTION

RMS 的动作取消为中断机器人当前执行的动作，使任务转为可继续更新的状态。

如机器人已取到货箱正在送往工作站的途中，RMS 收到了上游发送的动作取消指令，此时 RMS 将会使机器人停下，并继续等待任务更新。

但若一个指令在路径末端即将完成，此时更新可能会失败，上游可以根据回调判断是否为正常结束、是否取消动作成功。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	32
instruction	String	必要	必须为“CANCEL_INSTRUCTION”	CANCEL_INSTRUCTION

请求报文示例如下：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskUpdateRequestMsg",
  "request": {
    "header": {
      "requestId": "8b3c2fb8-928c-4f1a-af15-2740913e43fa",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 135,
      "instruction": "CANCEL_INSTRUCTION"
    }
  }
}
```

RMS 的响应示例如下：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskUpdateResponseMsg",
  "response": {
    "header": {
      "responseId": "8b3c2fb8-928c-4f1a-af15-2740913e43fa",
      "code": 0,
      "msg": "Success"
    }
  }
```

}

3.9.5 任务回调

机器人在执行“DELIVER_BOX”任务时，根据当前执行阶段的不同，RMS 会主动向主机系统发送回调消息，告知主机系统此任务目前的任务状态及执行阶段。

以下为“DELIVER_BOX”可能出现的任务执行阶段“taskPhase”。

taskPhase	发送时机
GO_FETCHING	当机器人开始去取一个货箱时
BOX_FETCHED	表示机器人已经取到箱子
GO_DELIVERING	机器人开始去送一个箱子（且表示机器人到达终点时不会把箱子放下，任务不会结束）
GO_RETURN	机器人开始去送一个箱子（且表示机器人到达终点时会把箱子放下，任务会直接结束）
BOX_ARRIVED	机器人（携带箱子）已经到达终点。若指令动作需要机器人放下货箱，则此回调也表示货箱已经被送至目的地
MOVING	机器人开始运动去某地
ARRIVED	机器人到达某地

3.9.5.1 GO_FETCHING

GO_FETCHING 阶段的回调消息示例如下：


```
{
"id": "geekCode_geekWarehouseCode_001",
"msgType": "RobotTaskCallbackMsg",
"request": {
      "header": {
          "warehouseCode": "geekWarehouseCode",
          "requestId": "7b9ded8eb8ef4c41ad992fb66e95307d",
          "version": "3.3.0"
       },
      "body": {
           "taskId": 1688,
           "taskType": "DELIVER_BOX",
           "taskStatus": "EXECUTING",
           "instruction": "GO_FETCH",
           "taskPhase": "GO_FETCHING",
           "robotId": 50002,
           "stationId": 3,
```

```
            "dest": {
                "x": 50,
                "y": 42,
                "z": 1
            },
            "destLocation": {
                "x": 53.025,
                "y": 25.43,
                "z": 1
            },
            "destCellCode": "05050425",
            "currentFloor": 1,
            "boxCode": "X000803",
            "latticeCode": "L0000600"
        }
    }
}
```


3.9.5.2 BOX_FETCHED

BOX_FETCHED 阶段的回调消息示例如下：


```
{
"id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskCallbackMsg",
    "request": {
        "header": {
            "warehouseCode": "geekWarehouseCode",
            "requestId": "6b88624a7128405598ec709a85a54f86",
            "version": "3.3.0"
        },
        "body": {
            "taskId": 1688,
            "taskType": "DELIVER_BOX",
            "taskStatus": "EXECUTING",
            "instruction": "GO_FETCH",
            "taskPhase": "BOX_FETCHED",
            "robotId": 50002,
            "stationId": 3,
            "dest": {
                "x": 52,
                "y": 94,
                "z": 1
            },
            "destLocation": {
                "x": 55.125,
                "y": 58.68,
                "z": 1
            },
            "destCellCode": "05250945",
```

```
            "currentFloor": 1,
            "boxCode": "X000803",
            "latticeCode": "L0000600"
        }
    }
}
```


3.9.5.3 GO_DELIVERING

GO_DELIVERING 阶段的回调消息示例如下：


```
{
"id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskCallbackMsg",
    "request": {
        "header": {
            "warehouseCode": "geekWarehouseCode",
            "requestId": "1f79f8ce78cc4806b723dec721ecd096",
            "version": "3.3.0"
        },
        "body": {
            "taskId": 1688,
            "taskType": "DELIVER_BOX",
            "taskStatus": "EXECUTING",
            "instruction": "GO_FETCH",
            "taskPhase": "GO_DELIVERING",
            "robotId": 50002,
            "stationId": 3,
            "dest": {
                "x": 52,
                "y": 94,
                "z": 1
            },
            "destLocation": {
                "x": 55.125,
                "y": 58.68,
                "z": 1
            },
            "destCellCode": "05250945",
            "currentFloor": 1,
            "boxCode": "X000803",
            "latticeCode": "L0000600"
        }
    }
}
```


3.9.5.4 GO_RETURN

GO_RETURN 阶段的回调消息示例如下：

```
{
"id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWarehouseCode",
      "requestId": "d47f06437e71461ba5e0ff603954e625",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 1688,
      "taskType": "DELIVER_BOX",
      "taskStatus": "EXECUTING",
      "instruction": "GO_RETURN",
      "taskPhase": "GO_RETURN",
      "robotId": 50002,
      "stationId": 3,
      "dest": {
        "x": 50,
        "y": 42,
        "z": 1
      },
      "destLocation": {
        "x": 53.025,
        "y": 25.43,
        "z": 1
      },
      "destCellCode": "05050425",
      "currentFloor": 1,
      "boxCode": "X000803",
      "latticeCode": "L0000600"
    }
  }
}
```


3.9.5.5 BOX_ARRIVED

BOX_ARRIVED 阶段到达回调消息示例如下：


```
{
"id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWarehouseCode",
      "requestId": "ec543fa2b5ac4ccdb5473e1a6625e89c",
      "version": "3.3.0"
    },
    "body": {
      "taskId": 1688,
```

```
            "taskType": "DELIVER_BOX",
            "taskStatus": "EXECUTING",
            "instruction": "READY",
            "taskPhase": "BOX_ARRIVED",
            "robotId": 50002,
            "stationId": 3,
            "dest": {
               "x": 52,
               "y": 94,
               "z": 1
            },
            "destLocation": {
               "x": 55.125,
               "y": 58.68,
               "z": 1
            },
            "destCellCode": "05250945",
            "currentFloor": 1,
            "boxCode": "X000803",
            "latticeCode": "L0000600"
         }
      }
   }
```


3.9.5.6 MOVING

MOVING 阶段的回调消息示例如下：


```
{
"id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "86c67d77d772464a86740b6079283f33",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 1697,
         "taskType": "DELIVER_BOX",
         "taskStatus": "EXECUTING",
         "instruction": "GO_NEXT",
         "taskPhase": "MOVING",
         "robotId": 50002,
         "stationId": 3,
         "dest": {
            "x": 52,
            "y": 94,
            "z": 1
         },
```

```
            "destLocation": {
               "x": 55.125,
               "y": 58.68,
               "z": 1
            },
            "destCellCode": "05250945",
            "currentFloor": 1,
            "boxCode": "",
            "latticeCode": ""
         }
      }
   }
```


3.9.5.7 ARRIVED

ARRIVED 阶段的回调消息示例如下：


```
{
"id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "d4f64a6d67d848a59341cce17f2f53b9",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 1697,
         "taskType": "DELIVER_BOX",
         "taskStatus": "EXECUTING",
         "instruction": "READY",
         "taskPhase": "ARRIVED",
         "robotId": 50002,
         "stationId": 3,
         "dest": {
            "x": 52,
            "y": 94,
            "z": 1
         },
         "destLocation": {
            "x": 55.125,
            "y": 58.68,
            "z": 1
         },
         "destCellCode": "05250945",
         "currentFloor": 1,
         "boxCode": "",
         "latticeCode": ""
      }
   }
```

}

3.9.6 功能注意点

目前支持的工作站类型仅为单点工位，非单点工位的工作站尚不支持。

3.10 GO_WORK_ORDER_TO_PERSON

此任务类型用于纯机器人本体运动，一次下发多点，自主规划先后顺序进行作业的场景。常用于订单到人场景。

3.10.1 场景说明

3.10.1.1 场景概括

业务系统下发多个目的地后，RMS 自主规划所有点的顺序，然后通过每次更新以完成整个业务流程。

1、自主规划是因为 RMS 具有地图资源和机器人调配能力，可通过算法规划出高效率的运行线路；

2、提前规划点位顺序是为了方便上游能够点位顺序提前进行订单拣货。

3.10.1.2 首次多点规划

首次任务下发时下发多个任务终点，RMS 自行通过算法计算到达顺序。

3.10.1.3 单点作业结束去往下一点

在某点结束后，业务系统更新任务以指向下一任务点。

1、此时系统会进行路径重新计算顺序；

2、此时允许业务系统更换终点或指定终点。

3.10.2 可用指令

GO_WORK_ORDER_TO_PERSON 支持的指令如下，具体用法会在下面的部分进行详细叙述。

行为	指令	用途
任务下发	GO_NEXT	用于指定本次任务的所有目的地
任务更新	GO_NEXT	用于指定任务更新后的所有目的地

3.10.3 任务下发

在任务下发时可用指令有“GO_NEXT”，下面将进行具体说明。

3.10.3.1 GO_NEXT

当主机系统需要 RMS 调度机器人本身去往多个目的地，且由 RMS 调度决定先后顺序时，使用此指令，可控制机器人到达目的地后是否将任务结束，不结束时主机系统可继续更新目的地。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskType	String	必要	任务类型。必须为“GO_WORK_ORDER_TO_PERSON”	
instruction	String	必要	表示去往目的地，必须为“GO_NEXT”	GO_NEXT
目的地描述	所有的参数都会参与校验，单个和集合如果同时传取并集。若为多个目的地，系统会根据目的地模式生成目的地列表以逐步前往。若多种目的地格式均填写了，则指代地点必须一致，否则报错。目的地至少需要有一个值。			
destMode	String	可选	AUTO 为空默认此值，指目的地列表由 RMS 控制，通过算法决定先后顺序，且每次更新 FIXED 此模式表示由主机系统规定各目的地顺序，单目的地会默认添加至集合第一位该模式将贯穿整个任务全程	
destCellCode	String	可选	目的节点编码	1
destCellCodes	list of String	可选	目的节点编码集合	["1","2","3"]
dest	IPoint	可选	目的地索引坐标。即 z 轴，x 轴，y 轴。z 可省略,默认是 1(二维码区域专用,传索引值)	{"x":45,"y":42}，{"z":1,"x":45,"y":42}
dests	list of IPoint	可选	节点索引集合，单个和集合同时传输时取并集	[{"z":1,"x":45,"y":42},{"z":2,"x":45,"y":42}]

字段	类型	是否必要	描述	示例
destHostCode	String	可选	上位机系统关于当前任务的目的地坐标的描述	
destHostCodes	list of String	可选	上位机系统关于当前任务的目的地坐标的描述集合	
stationId	Integer	可选	工作站编号	
hostStationCode	String	可选	主机系统对工作站的编号的描述	
机器人控制字段	单个、集合均填写时取并集，允许指定具体机器人执行任务，不指定时系统会调度合适的机器人。			
robotId	Integer	可选	指定某个机器人去执行任务	
robotIds	list of Integer	可选	指定一个机器人集合去执行，系统会在其中选中一个距离最近符合条件的机器人去执行	
其余信息控制字段	可根据需要进行选填			
priority	Integer	可选	该参数用来设置任务的优先级。值越大，表明在分配机器人执行任务期间该任务的优先级越高，priority 取值范围[0-99]	
isContinue	Integer	可选	0，指令执行结束后任务结束。1，指令执行结束后任务不结束。为空视为 1。	

请求报文示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskRequestMsg",
   "request": {
      "header": {
         "requestId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskType": "GO_WORK_ORDER_TO_PERSON",
         "instruction": "GO_NEXT",
```

```
            "destCellCodes":["1","2","3"]
        }
    }
}
```

此报文表示主机系统希望 RMS 调度机器人去往 1,2,3 三个工作点且自行规划顺序。

RMS 的响应示例如下：

```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskResponseMsg",
    "response": {
        "header": {
            "responseId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
            "code": 0,
            "msg": "Success"
        },
        "body": {
            "taskId": 30
        }
    }
}
```

3.10.4 任务更新

3.10.4.1 GO_NEXT

当机器人到达某点后，主机系统进行了作业后，可使用该指令更新机器人去往下一个工作点。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	表示指派机器人去往剩余目的地，必须为“GO_NEXT”	GO_NEXT
目的地控制字段	目的地相关的值都会参与校验，单个和集合如果同时传取并集。若为多个目的地，系统会根据目的地模式生成目的地列表以此前往。 若目的地为空，系统按照已有的目的地列表顺序执行，若列表已空且任务不结束，系统响应：无目的地 若目的地不为空，将清空原有目的地列表更新为新的目的地列表。			
destCellCode	String	可选	目的节点编码	1
destCellCodes	list of String	可选	目的节点编码集合	["1","2","3"]

字段	类型	是否必要	描述	示例
dest	DPoint	可选	目的地索引坐标。即 z 轴，x 轴，y 轴。z 可省略,默认是 1(二维码区域专用,传索引值)	{"x":45,"y":42} {"z":1,"x":45,"y":42}
dests	list of DPoint	可选	节点索引集合，单个和集合同时传输时取并集	[{"z":1,"x":45,"y":42}, {"z":2,"x":45,"y":42}]
destHostCode	String	可选	上位机系统关于当前任务的目的地坐标的描述	12
destHostCodes	list of String	可选	上位机系统关于当前任务的目的地坐标的描述集合	[“1”,”2”,”3”]
stationId	Integer	可选	工作站编号	2
hostStationCode	String	可选	主机系统对工作站的编号的描述	2
其余信息控制字段	可根据需要进行选填			
isContinue	Integer	可选	0，指令执行结束后任务结束。 若系统目的地列表不为空，则到达下一个目的地后任务结束 若系统目的地列表已为空，则原地结束任务 1，指令执行结束后任务不结束。为空视为 1。	1

请求报文示例如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateRequestMsg",
   "request": {
      "header": {
         "requestId": "9beab88d-a407-4195-a1f8-ae3594c83924",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 30,
         "instruction": "GO_NEXT"
        }
   }
}
```

此报文表示主机系统希望机器人接着执行目的地列表中下一个就近目的地。

RMS 的响应示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateResponseMsg",
   "response": {
      "header": {
         "responseId": "9beab88d-a407-4195-a1f8-ae3594c83924",
         "code": 0,
         "msg": "Success"
      }
    }
}
```


3.10.5 任务回调

机器人在执行“GO_WORK_ORDER_TO_PERSON”任务时，根据当前执行阶段的不同，RMS 会主动向主机系统发送回调消息，告知主机系统此任务目前的任务状态及执行阶段。

以下为“GO_WORK_ORDER_TO_PERSON”可能出现的任务执行阶段“taskPhase”。

taskPhase	发送时机
MOVING	机器人出发去往目的地列表中的点
ARRIVED	机器人到达目的地列表中选定的点

3.10.5.1 MOVING

GO_WORK_ORDER_TO_PERSON 任务 MOVING 阶段的回调消息字段说明如下表。除 taskId 字段外都不是必填项。

字段	类型	描述	示例
taskId	long	当前进行的任务 id	30
taskType	String	GO_WORK_ORDER_TO_PERSON	
taskStatus	String	EXECUTING 执行中 CANCELED 取消（出发时不会出现） COMPLETED 完成（出发时不会出现）	EXECUTING
taskPhase	String	MOVING	MOVING
robotId	Integer	执行任务的机器人 ID	1

字段	类型	描述	示例
当前目的地部分	在此处将描述系统选定的当前目的地		
destLocation	DPoint	终点：绝对坐标，非必填	{"z": 1,"x": 29.5,"y":10.5}
dest	IPoint	终点：索引坐标，非必填	{"z": 1,"x": 29,"y": 10}
destCellCode	String	终点：节点编码，非必填	102950105
destHostCode	String	终点：节点外部编码，非必填	546544
stationId	Integer	终点工作站，非必填 注：任务指向工作站才会有此值	1
hostStationCode	String	终点工作站外部编码，非必填 注：任务指向工作站才会有此值	station1
目的地列表部分	在此处将描述系统中的目的地列表，只显示即将达到的目的地列表，当前目的地也在此列		
destList	list of DestList	目的地列表，非必填	

DestList 对象字段说明如下表：

字段	类型	说明	示例
order	Integer	顺序，表明该目的地的	1
destLocation	DPoint	终点：绝对坐标，非必填	{"z": 1,"x": 29.5,"y": 10.5}
dest	IPoint	终点：索引坐标，非必填	{"z": 1,"x": 29,"y": 10}
destCellCode	String	终点：节点编码，非必填	102950105
destHostCode	String	终点：节点外部编码，非必填	546544
stationId	Integer	终点工作站，非必填 注：任务指向工作站才会有此值	1
hostStationCode	String	终点工作站外部编码，非必填 注：任务指向工作站才会有此值	station1

回调的报文示例如下：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "RobotTaskCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWarehouseCode",
```

```
        "requestId": "8780c902ed714d10b82a2f20ac514a1c",
        "version": "3.3.0"
      },
      "body": {
        "taskId": 37,
        "taskType": "GO_WORK_ORDER_TO_PERSON",
        "taskStatus": "EXCUTING",
        "taskPhase": "MOVING",
        "robotId": 1000,
        "destLocation":{"z": 1,"x": 29.5,"y": 10.5},
        "dest":{"z": 1,"x": 29,"y": 10},
        "destCellCode":"102950105",
        "destHostCode":"45464",
        "stationId":"1",
       "hostStationCode":"station1",
       "destList":[
       {
            "order":1,
            "destLocation":{"z": 1,"x": 29.5,"y": 10.5},
            "dest":{"z": 1,"x": 29,"y": 10},
            "destCellCode":"102950105",
            "destHostCode":"45464",
            "stationId":"1",
           "hostStationCode":"station1"
       },
       {
            "order":2,
            "destLocation":{"z": 1,"x": 29.5,"y": 10.5},
            "dest":{"z": 1,"x": 29,"y": 10},
            "destCellCode":"102950105",
            "destHostCode":"45464",
            "stationId":"1",
            "hostStationCode":"station1"
       }
      ]
     }
   }
}
```


主机系统应该给与 RMS 的响应如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackResponseMsg",
   "response": {
      "header": {
         "responseId": "8780c902ed714d10b82a2f20ac514a1c"
      }
   }
}
```

3.10.5.2 ARRIVED

GO_WORK_ORDER_TO_PERSON 任务 ARRIVED 阶段的回调消息字段说明如下。除 taskId 字段外都不是必填项。

字段	类型	描述	示例
taskId	long	当前进行的任务 id	30
taskType	String	GO_WORK_ORDER_TO_PERSON	
taskStatus	String	EXECUTING 执行中 CANCELED 取消 COMPLETED 完成	EXECUTING
taskPhase	String	ARRIVED	ARRIVED
robotId	Integer	执行任务的机器人 ID	1
当前目的地部分	在此处将描述机器人到达的目的地		
destLocation	DPoint	终点：绝对坐标	{"z": 1,"x": 29.5,"y":10.5}
dest	IPoint	终点：索引坐标	{"z": 1,"x": 29,"y": 10}
destCellCode	String	终点：节点编码	102950105
destHostCode	String	终点：节点外部编码	546544
stationId	Integer	终点工作站 注：任务指向工作站才会有此值	1
hostStationCode	String	终点工作站外部编码 注：任务指向工作站才会有此值	station1

回调的报文示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "8780c902ed714d10b82a2f20ac514a1c",
         "version": "3.3.0"
      },
      "body": {
         "taskId": 37,
         "taskType": "GO_WORK_ORDER_TO_PERSON",
         "taskStatus": "EXCUTING",
         "taskPhase": "ARRIVED",
```

```
            "robotId": 1000,
            "destLocation":{"z": 1,"x": 29.5,"y": 10.5},
            "dest":{"z": 1,"x": 29,"y": 10},
            "destCellCode":"102950105",
            "destHostCode":"45464",
            "stationId":"1",
            "hostStationCode":"station1"
        }
    }
}
```


主机系统应该给与 RMS 的响应如下：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "RobotTaskCallbackResponseMsg",
    "response": {
        "header": {
            "responseId": "8780c902ed714d10b82a2f20ac514a1c"
        }
    }
}
```


3.10.6 功能注意点

目前支持的工作站类型仅为单点工位，非单点工位的工作站尚不支持。

3.10.7 场景示例

下发系统	接收系统	消息类型	任务执行阶段	任务状态	指令	要求字段	功能	响应
主机系统	RMS	RobotTaskRequestMsg	下发		GO_NEXT	多目的地 机器人（可指定）	下发任务至多目的地	任务号
RMS	主机系统	RobotTaskCallbackMsg	MOVING	EXCUTING		目的地排序	反馈上游规划的目的地 每次出发规划顺序并反馈	
RMS	主机系统	RobotTaskCallbackMsg	ARRIVED	EXCUTING		到达的目的地	反馈上游到达的目的地	
主机系统	RMS	RobotTaskUpdateRequestMsg			GO_NEXT	多目的地	可更换目的地	
循环出发到达直至更新至一个目的地结束任务								
主机系统	RMS	RobotTaskUpdateRequestMsg			GO_NEXT	isContinue=0	原地、指向目的地结束任务	
RMS	主机系统	RobotTaskCallbackMsg	ARRIVED	COMPLETED			结束任务	
取消情形								
主机系统	RMS	RobotTaskUpdateRequestMsg		EXCUTING	CANCEL	取消 取消动作	结束任务	
RMS	主机系统	RobotTaskCallbackMsg	ARRIVED	CANCELED			取消完成后的回调	

3.11 分拣自动模式相关任务

此任务在分拣场景中用于 RMS 自动发起任务，自动、快速进行收发包裹。该场景比较特殊，为两种任务复合完成该任务场景。

在该场景下，RMS 将定义一种数据类型 sortStation 如下：

自定义数据	含义	类型 说明	示例
sortStation	表示分拣的投递格口的描述，分为五个字段描述格口的控制属性	详见下述字段说明	{ "cellNo": "CHUTE_156", "cellCode": "05650605", "available": true, "allowDirs": [0, 1, 2, 3] }

sortStation 的各字段含义如下所示：

字段	类型	是否必要	描述	示例
cellNo	String	可选	上游对投递口的描述，仅透传使用，无逻辑	CHUTE_156
cellCode	String	必要	目的地必须为投递格口（障碍点），系统会寻找该点四周所有可停靠的点进行投递	454661
available	Bool	可选	目的地是否可用，不可用的点系统不会进行投递。默认值为 ture	true
allowDirs	list of int	可选	目的地可投递的方向，0 东向；1 西向；2 南向；3 北向，默认[0, 1, 2, 3]	[0, 1, 2, 3]
spare	Bool	可选	备用标识，默认 false	false

3.11.1 场景说明

3.11.1.1 场景概括

该场景下 RMS 会根据投递口、格口自主发起任务，场景流程如下：

1、RMS 发起任务将机器人调度往供包台进行排队

2、业务在机器人身上放置包裹后 RMS 提示主机系统包裹已就绪

3、主机系统更新任务投递至格口

自动执行分拣任务的要求为：地图配置有分拣排队区区域、排队区入口。系统检测到有该类配置即为开始自动生成分拣任务。投递口在系统中为一种障碍单元格，无特殊要求。场景任务分为排队和投递两大阶段。

3.11.1.2 自动发起任务排队

该场景下 RMS 会自动发起分拣任务，指派空闲机器人去往供包台等候供包。任务类型为 GO_STATION_TO_QUEUE。

3.11.1.3 指派机器人进行投递

在收到包裹后，业务系统进行任务更新，指派机器人进行投递，可选多个投递格口、备用投递格口、中转点。该阶段任务类型为 DELIVER_SINGLE_PARCEL。

3.11.2 可用指令

该场景下任务为自动执行，主机系统仅需更新指令。

任务类型	行为	指令	用途
GO_STATION_TO_QUEUE	任务下发（自动生成）	GO_SOMEWHERE	用于生成自动任务指派机器人去往供包台
DELIVER_SINGLE_PARCEL	任务下发（自动生成）		系统自动生成投递任务
DELIVER_SINGLE_PARCEL	任务更新	GO_NEXT	指派机器人先去往某处
DELIVER_SINGLE_PARCEL	任务更新	GO_DROP	指派机器人去投递包裹
DELIVER_SINGLE_PARCEL	任务更新	UPDATE_PARCEL	在机器人收到 GO_DROP 指令后在投递前随时可使用该指令更新机器人的目的地

3.11.3 任务下发

该场景下任务为自动生成，无需主机系统下发任务。分为两个阶段：

1、生成自动去往供包台排队的任务：GO_STATION_TO_QUEUE。

2、生成自动投递包裹的任务：DELIVER_SINGLE_PARCEL。

3.11.4 任务更新

该场景下去往供包台阶段不需要进行更新。在第二个投递阶段需要主机系统进行更新。

3.11.4.1 DELIVER_SINGLE_PARCEL 任务 GO_NEXT

当机器人在供包台收取到包裹后，主机系统作业结束后，可使用该指令更新机器人去往一个工作点，常用于在投递前进行缓存等作业。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	表示指派机器人去往目的地，必须为“GO_NEXT”	GO_NEXT
目的地控制字段	系统支持复数个目的地。若目的地为多个，系统将去往一个最近的点位。			
destCells	list of String	必要	目的地集合	
cellCode	String	必要	目的地节点编码 若为普通节点，视为一个普通目的地 若为一个障碍点（投递口），系统会寻找该点四周所有可停靠的节点纳入目标点	454661

请求报文示例如下：


```
{
 "id": "*",
 "msgType": "RobotTaskUpdateRequestMsg",
 "request": {
  "header": {
   "requestId": "9beab88d-a4w7-4195-a1f8-ae3594c83924",
   "clientCode": "geekCode",
   "warehouseCode": "geekWarehouseCode",
   "userId": "admin",
   "userKey": "111111",
   "version": "3.3.0",
   "language": "en_us"
  },
  "body": {
   "taskId": 242419,
   "instruction": "GO_NEXT",
   "destCells": [{
     "cellCode": "05550595"
    }, {
     "cellCode": "05550555"
    }
   ]
  }
 }
}
```

RMS 的响应示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateResponseMsg",
   "response": {
      "header": {
         "responseId": "9beab88d-a4w7-4195-a1f8-ae3594c83924",
         "code": 0,
         "msg": "Success"
      }
     }
}
```


3.11.4.2 DELIVER_SINGLE_PARCEL 任务 GO_DROP

当机器人在供包台收取到包裹后，主机系统作业结束后，可使用该指令更新机器人去往一个投递口进行投递包裹。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	表示指派机器人进行投递包裹，必须为“GO_DROP”	GO_DROP
parcelId	String	可选	用于透传主机信息，一般为供包台在主机系统内的名称，用于便利主机系统查询使用。	1254
正常投递口控制字段	系统支持多个目的地。 该部分为投递口，目的地必须为“格口”（主机系统中的格口在 RMS 里用的是障碍单元格表示）。若目的地为多个，RMS 将去往一个最近的可用点位。			
destCells	list of sortStation	可选	投递口集合	
备用投递口控制字段	系统支持复数个目的地。 该部分为投递口，目的地必须为“格口”（主机系统中的格口在 RMS 里用的是障碍单元格表示） 该目的地为多个，系统将去往一个最近的可用点位 仅有当投递口全部不可用时才会往备用投递口投递			
collectingCells	list of sortStation	可选	投递口集合	
中转点控制字段	系统支持复数个备用目的地。 该部分为中转点，必须为普通可达节点 该目的地为多个，系统将去往一个最近的可用点位			

字段	类型	是否必要	描述	示例
	仅投递口全部不可用、备用投递口全部不可用时才会去往中转点 当任一投递口、备用投递口可用时，系统自动去往投递。			
transferCells	list of sortStation	可选	中转点节点集合	

正常投递口、备用投递口和中转点信息至少要有一个。请求报文示例如下：


```
{
 "id": "*",
 "msgType": "RobotTaskUpdateRequestMsg",
 "request": {
  "header": {
   "requestId": "9beab88d-a4a7-4195-a1f8-ae3594c83924",
   "clientCode": "geekCode",
   "warehouseCode": "geekWarehouseCode",
   "userId": "admin",
   "userKey": "111111",
   "version": "3.3.0",
   "language": "en_us"
  },
  "body": {
   "taskId": 242447,
   "parcelId": "parcelId",
   "instruction": "GO_DROP",
   "destCells": [{
    "cellNo": "CHUTE_156",
    "cellCode": "05650605",
    "available": true,
    "allowDirs": [0, 1, 2, 3],
   }, {
    "cellNo": "CHUTE_016",
    "cellCode": "05250885",
    "available": true,
    "allowDirs": [0, 1, 2, 3],
   }
  ],
  "collectingCells": [{
    "cellNo": "CHUTE_069",
    "cellCode": "05650775",
    "available": true,
    "allowDirs": [0, 1, 2, 3],
   }, {
    "cellNo": "CHUTE_062",
    "cellCode": "04050775",
    "available": true,
    "allowDirs": [0, 1, 2, 3],
   }, {
    "cellNo": "CHUTE_055",
```

```
      "cellCode": "02250775",
      "available": true,
      "allowDirs": [0, 1, 2, 3],
     }
    ],
    "transferCells": [{
     "cellCode": "02250775"
     }
    ]
   }
  }
 }
```


RMS 的响应示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateResponseMsg",
   "response": {
      "header": {
         "responseId": "9beab88d-a4a7-4195-a1f8-ae3594c83924",
         "code": 0,
         "msg": "Success"
      }
     }
}
```


3.11.4.3 DELIVER_SINGLE_PARCEL 任务 UPDATE_PARCEL

当机器人在收到投递包裹命令后投递动作完成前，主机系统可使用该指令随时更新目的地，RMS 将随之更新任务路线及目标。

请求消息体的字段如下：

字段	类型	是否必要	描述	示例
taskId	long	必要	当前进行的任务 id	30
instruction	String	必要	表示更新任务信息，必须为“UPDATE_PARCEL”	UPDATE_PARCEL
parcelId	String	可选	用于透传主机信息，一般为供包台在主机系统内的名称，用于便利主机系统查询使用。	1254
正常投递口控制字段	系统支持复数个目的地。 该部分为投递口，目的地必须为“格口”（分拣系统中的格口在 RMS 里用的是障碍单元格表示） 若目的地为多个，系统将去往一个最近的可用点位			
destCells	list of	可选	投递口集合	

字段	类型	是否必要	描述	示例
	sortStation			
备用投递口控制字段	系统支持复数个目的地 该部分为投递口，目的地必须为“格口”（分拣系统中的格口在 RMS 里用的是障碍单元格表示）。 该目的地为多个时，系统将去往一个最近的可用点位。 仅有当投递口全部不可用时才会往备用投递口投递。			
collectingCells	list of sortStation	可选	投递口集合	
中转点控制字段	系统支持复数个备用目的地 该部分为中转点，必须为普通可达节点 该目的地为多个，系统将去往一个最近的可用点位 仅有当投递口全部不可用、备用投递口全部不可用时才会去往中转点 当任一投递口、备用投递口可用时，系统自动去往投递			
transferCells	list of sortStation	可选	中转点节点集合	

正常投递口、备用投递口和中转点信息至少要有一个。请求报文示例如下：


```
{
 "id": "*",
 "msgType": "RobotTaskUpdateRequestMsg",
 "request": {
  "header": {
   "requestId": "9beab88d-a4c7-4195-a1f8-ae3594c83924",
   "clientCode": "geekCode",
   "warehouseCode": "geekWarehouseCode",
   "userId": "admin",
   "userKey": "111111",
   "version": "3.3.0",
   "language": "en_us"
  },
  "body": {
   "taskId": 242447,
   "parcelId": "parcelId-1",
   "instruction": "UPDATE_PARCEL",
   "destCells": [{
     "cellNo": "CHUTE_156",
     "cellCode": "05650605",
     "available": true,
     "allowDirs": [0, 1, 2, 3],
    }, {
     "cellNo": "CHUTE_016",
     "cellCode": "05250885",
     "available": true,
     "allowDirs": [0, 1, 2, 3],
```

```
      }
     ],
     "collectingCells": [{
      "cellNo": "CHUTE_069",
      "cellCode": "05650775",
      "available": true,
      "allowDirs": [0, 1, 2, 3],
     }, {
      "cellNo": "CHUTE_062",
      "cellCode": "04050775",
      "available": true,
      "allowDirs": [0, 1, 2, 3],
     }, {
      "cellNo": "CHUTE_055",
      "cellCode": "02250775",
      "available": true,
      "allowDirs": [0, 1, 2, 3],
     }
     ],
     "transferCells": [{
      "cellCode": "02250775"
     }
     ]
    }
   }
}
```


RMS 的响应示例如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskUpdateResponseMsg",
   "response": {
      "header": {
         "responseId": "9beab88d-a4c7-4195-a1f8-ae3594c83924",
         "code": 0,
         "msg": "Success"
      }
    }
}
```


3.11.5 任务回调

系统在自主发起任务后会，会指派机器人去往排队区。其中排队阶段的自动任务 GO_STATION_TO_QUEUE 因策略不同会产生不同回调：

1、动态路径模式，系统会在机器人去往排队区入口的过程中动态更改路径，直接去往供包台，仅会出现一个任务。

2、非动态路径模式，系统会在机器人到达排队区入口后，再次生成一个去往供包台的任务，会出现两个任务。

机器人在执行该场景下的两个阶段两种任务时，根据当前执行阶段的不同，RMS 会主动向主机系统发送回调消息，告知主机系统此任务目前的任务状态及执行阶段。

以下为自动分拣场景下可能出现的任务执行阶段“taskPhase”。

该场景下会出现两个任务阶段两个任务类型：

任务类型	任务阶段	发送时机
GO_STATION_TO_QUEUE	MOVING	机器人出发去往排队区入口处/供包台
GO_STATION_TO_QUEUE	ARRIVED	机器人到达排队区入口处/供包台
DELIVER_SINGLE_PARCEL	FEEDING	供包台处机器人及其设备、传感器已准备完毕 无传感器模式，机器人准备完毕直接会回调 FEEDING 有传感器模式，必须为机器人准备完毕，且人工将包裹平推入机器人，传感器收到包裹后才会回调 FEEDING
DELIVER_SINGLE_PARCEL	MOVING	机器人出发去往某处 执行 GO_NEXT 指令
DELIVER_SINGLE_PARCEL	ARRIVED	机器人到达某处 执行 GO_NEXT 指令
DELIVER_SINGLE_PARCEL	DELIVERING	机器人开始去投递，另外若被更新了目的地，机器人路径更新后，也会进行此回调
DELIVER_SINGLE_PARCEL	DROP_FINISH	机器人投递结束

3.11.5.1 GO_STATION_TO_QUEUE 任务 MOVING

场景为自动任务指派机器人去往排队区/供包台。

回调消息字段说明如下：

字段	类型	描述	示例
taskId	long	当前进行的任务 id	30
taskType	String	仅有"GO_STATION_TO_QUEUE"	GO_STATION_TO_QUEUE
taskStatus	String	EXECUTING 执行中 CANCELED 取消（出发时不会出现） COMPLETED 完成（出发时不会出现）	EXECUTING
taskPhase	String	MOVING	MOVING
robotId	Integer	执行任务的机器人 ID	1

字段	类型	描述	示例
当前目的地部分	在此处将描述系统选定的当前目的地		
stationId	Integer	终点工作站 注：任务目的地为工作站才会有此值，两段任务去往排队区入口时不会出现	1
dest	IPoint	终点：索引坐标	{"z": 1,"x": 29,"y": 10}
destLocation	DPoint	终点：绝对坐标	{"z": 1,"x": 29.5,"y": 10.5}
destCellCode	String	终点：节点编码	10250105

回调的报文示例如下：


```
{
 "id":"*",
 "msgType":"RobotTaskCallbackMsg",
 "request":{
  "header":{
   "warehouseCode":"geekWarehouseCode",
   "requestId":"ccb13d2e70744516aeb67e2ef635d10d",
   "version":"3.3.0"
    },
  "body":{
   "taskId":242417,
   "taskType":"GO_STATION_TO_QUEUE",
   "taskStatus":"EXECUTING",
   "taskPhase":"MOVING",
   "robotId":1000,
   "stationId":1,
   "dest":{"z":1,"x":12,"y":90},
   "destLocation":{"z":1,"x":7.5,"y":54.3},
   "destCellCode":"01250905"
    }
   }
}
```


主机系统应该给与 RMS 的响应如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackResponseMsg",
   "response": {
      "header": {
         "responseId": "8780c902ed714d10b82a2f20ac514a1c"
      }
   }
}
```

3.11.5.2 GO_STATION_TO_QUEUE 任务 ARRIVED

场景为自动任务指派机器人到达了排队区入口/供包台。

回调消息字段说明如下：

字段	类型	描述	示例
taskId	long	当前进行的任务 id	30
taskType	String	仅有"GO_STATION_TO_QUEUE"	GO_STATION_TO_QUEUE
taskStatus	String	COMPLETED 完成	EXECUTING
taskPhase	String	ARRIVED	ARRIVED
robotId	Integer	执行任务的机器人 ID	1
当前目的地部分	在此处将描述系统选定的当前目的地		
stationId	Integer	终点工作站 注：任务目的地为工作站才会有此值，两段任务去往排队区入口时不会出现	1
dest	IPoint	终点：索引坐标	{"z": 1,"x": 29,"y": 10}
destLocation	DPoint	终点：绝对坐标	{"z": 1,"x": 29.5,"y": 10.5}
destCellCode	String	终点：节点编码	102950105

回调的报文示例如下：


```
{
 "id":"*",
 "msgType":"RobotTaskCallbackMsg",
 "request":{
  "header":{
   "warehouseCode":"DEFAULT",
   "requestId":"8bc97874192d4741a7c807d6d5536d2d",
   "version":"3.3.0"
    },
  "body":{
   "taskId":242417,
   "taskType":"GO_STATION_TO_QUEUE",
   "taskStatus":"COMPLETED",
   "taskPhase":"ARRIVED",
   "robotId":1000,
   "stationId":1,
   "dest":{"z":1,"x":12,"y":90},
   "destLocation":{"z":1,"x":7.5,"y":54.3},
   "destCellCode":"01250905"
```

```
    }
  }
}
```

主机系统应该给与 RMS 的响应如下：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackResponseMsg",
   "response": {
      "header": {
         "responseId": "8780c902ed714d10b82a2f20ac514a1c"
      }
   }
}
```

3.11.5.3 DELIVER_SINGLE_PARCEL 任务 FEEDING

场景为：

1、单皮带机器人准备完毕；

2、双皮带机器人平推收到任一包裹时。

回调消息字段说明如下：

字段	类型	描述	示例
taskId	long	当前进行的任务 id	30
taskType	String	"DELIVER_SINGLE_PARCEL"	DELIVER_SINGLE_PARCEL
taskStatus	String	EXCITING	EXECUTING
taskPhase	String	FEEDING，表示机器人及其附属机构已准备完毕	FEEDING
robotId	Integer	执行任务的机器人 ID	1
机器人准备完毕的位置字段	在此处将描述机器人准备完毕的所处位置描述		
stationId	Integer	所处供包台的工作站 ID	1
dest	IPoint	所处位置：索引坐标	{"z": 1,"x": 29,"y": 10}
destLocation	DPoint	所处位置：绝对坐标	{"z": 1,"x": 29.5,"y": 10.5}
destCellCode	String	所处位置：节点编码	102950105
beltPos	Integer	机器人皮带模式 1：用于双皮带机器人，1 号皮带已收包 2：用于双皮带机器人，2 号皮带已收包	4

字段	类型	描述	示例
		3：用于双皮带机器人，1,2 号皮带均已收包 4：一个占据两个皮带的大包裹，若为单皮带机器人，不需供包直接就为 4	

回调的报文示例如下：


```
{
 "id": "*",
 "msgType": "RobotTaskCallbackMsg",
 "request": {
  "header": {
    "warehouseCode": "DEFAULT",
   "requestId": "0388c6a71af1447a98140aa35edb171c",
   "version": "3.3.0"
  },
  "body": {
   "taskId": 242419,
   "taskType": "DELIVER_SINGLE_PARCEL",
   "taskStatus": "ASSIGNED",
   "taskPhase": "FEEDING",
   "robotId": 1000,
   "stationId": 1,
   "dest": {
    "z": 1,
    "x": 13,
    "y": 85
   },
   "destLocation": {
    "z": 1,
    "x": 8.1,
    "y": 51.3
   },
   "destCellCode": "01350855",
   "beltPos": 4
  }
 }
}
```


主机系统应该给与 RMS 的响应如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackResponseMsg",
   "response": {
      "header": {
         "responseId": "8780c902ed714d10b82a2f20ac514a1c"
      }
   }
}
```

3.11.5.4 DELIVER_SINGLE_PARCEL 任务 MOVING

场景为机器人收包完毕后，主机系统使用 GO_NEXT 更新任务使机器人先去往某处暂存，机器人开始出发时进行回调。

回调消息字段说明如下：

字段	类型	描述	示例
taskId	long	当前进行的任务 id	30
taskType	String	"DELIVER_SINGLE_PARCEL"	DELIVER_SINGLE_PARCEL
taskStatus	String	EXECUTING 执行中	EXECUTING
taskPhase	String	MOVING，表示机器人开始出发	MOVING
robotId	Integer	执行任务的机器人 ID	1
stationId	Integer	表示该任务是从哪个供包台接的任务	1
目的地字段	在此处将描述机器人目的地		
dest	IPoint	目的地：索引坐标	{"z": 1,"x": 29,"y": 10}
destLocation	DPoint	目的地：绝对坐标	{"z": 1,"x": 29.5,"y": 10.5}
destCellCode	String	目的地：节点编码	102950105
beltPos	Integer	机器人收包状态 1：用于双皮带机器人，1 号皮带已收包 2：用于双皮带机器人，2 号皮带已收包 3：用于双皮带机器人，1,2 号皮带均已收包 4：一个占据两个皮带的大包裹，若为单皮带机器人，不需供包直接就为 4	4

回调的报文示例如下：


```
{
 "id": "*",
 "msgType": "RobotTaskCallbackMsg",
 "request": {
  "header": {
   "warehouseCode": "DEFAULT",
   "requestId": "ebf0f3c9728d43e1b7530c131615e04d",
   "version": "3.3.0"
  },
  "body": {
   "taskId": 242419,
   "taskType": "DELIVER_SINGLE_PARCEL",
   "taskStatus": "EXECUTING",
```

```
  "taskPhase": "MOVING",
  "robotId": 1000,
  "stationId": 1,
  "dest": {
   "z": 1,
   "x": 55,
   "y": 55
  },
  "destLocation": {
   "z": 1,
   "x": 33.3,
   "y": 33.3
  },
  "destCellCode": "05550555",
  "beltPos": 4
 }
}
}
```


主机系统应该给与 RMS 的响应如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackResponseMsg",
   "response": {
      "header": {
         "responseId": "8780c902ed714d10b82a2f20ac514a1c"
      }
   }
}
```


3.11.5.5 DELIVER_SINGLE_PARCEL 任务 ARRIVED

场景为机器人收包完毕后，主机系统使用 GO_NEXT 更新任务使机器人先去往某处暂存，机器人到达目的地时进行回调。

回调消息字段说明如下：

字段	类型	描述	示例
taskId	long	当前进行的任务 id	30
taskType	String	"DELIVER_SINGLE_PARCEL"	DELIVER_SINGLE_PARCEL
taskStatus	String	EXECUTING 执行中	EXECUTING
taskPhase	String	ARRIVED，表示机器人到达	ARRIVED
robotId	Integer	执行任务的机器人 ID	1
stationId	Integer	表示该任务是从哪个供包台接的任务	1

字段	类型	描述	示例
机器人目的地字段	在此处将描述机器人目的地		
dest	IPoint	目的地：索引坐标	{"z": 1,"x": 29,"y": 10}
destLocation	DPoint	目的地：绝对坐标	{"z": 1,"x": 29.5,"y": 10.5}
destCellCode	String	目的地：节点编码	102950105
beltPos	Integer	机器人收包状态 1：用于双皮带机器人，1 号皮带已收包 2：用于双皮带机器人，2 号皮带已收包 3：用于双皮带机器人，1,2 号皮带均已收包 4：一个占据两个皮带的大包裹，若为单皮带机器人，不需供包直接就为 4	4

回调的报文示例如下：


```
{
 "id": "*",
 "msgType": "RobotTaskCallbackMsg",
 "request": {
  "header": {
   "warehouseCode": "DEFAULT",
   "requestId": "54727c1816ec44479f324fd59600bd00",
   "version": "3.3.0"
  },
  "body": {
   "taskId": 242419,
   "taskType": "DELIVER_SINGLE_PARCEL",
   "taskStatus": "EXECUTING",
   "taskPhase": "ARRIVED",
   "robotId": 1000,
   "stationId": 1,
   "dest": {
    "z": 1,
    "x": 55,
    "y": 55
   },
   "destLocation": {
    "z": 1,
    "x": 33.3,
    "y": 33.3
   },
   "destCellCode": "05550555",
   "beltPos": 4
  }
 }
}
```

主机系统应该给与 RMS 的响应如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackResponseMsg",
   "response": {
      "header": {
         "responseId": "8780c902ed714d10b82a2f20ac514a1c"
      }
   }
}
```


3.11.5.6 DELIVER_SINGLE_PARCEL 任务 DELIVERING

场景为机器人收包完毕后，主机系统使用 GO_DROP 更新任务使机器人去投递包裹，机器人开始出发时进行回调。此外，若被更新了目的地，当机器人路径更新成功后，也会回调此信息。

回调消息字段说明如下：

字段	类型	描述	示例
taskId	long	当前进行的任务 id	30
taskType	String	"DELIVER_SINGLE_PARCEL"	DELIVER_SINGLE_PARCEL
taskStatus	String	EXECUTING 执行中	EXECUTING
taskPhase	String	DELIVERING，表示机器人出发去投递	DELIVERING
robotId	Integer	执行任务的机器人 ID	1
stationId	Integer	表示该任务是从哪个供包台接的任务	1
机器人目的地字段	在此处将描述机器人目的地 若为投递口，则目的地为投递口旁选定的投递点 若为中转点，则目的地就是最终目的地		
dest	IPoint	目的地：索引坐标	{"z": 1,"x": 29,"y": 10}
destLocation	DPoint	目的地：绝对坐标	{"z": 1,"x": 29.5,"y": 10.5}
destCellCode	String	目的地：节点编码	102950105
beltPos	Integer	机器人收包状态 1：用于双皮带机器人，1 号皮带已收包 2：用于双皮带机器人，2 号皮带已收包 3：用于双皮带机器人，1,2 号皮带均已收包 4：一个占据两个皮带的大包裹，若为单皮	4

字段	类型	描述	示例
		带机器人，不需供包直接就为 4	
dropCellCode	String	机器人当前选定的投递口，若机器人去往中转点则此字段为空	05250885

回调的报文示例如下：


```
{
 "id": "*",
 "msgType": "RobotTaskCallbackMsg",
 "request": {
  "header": {
   "warehouseCode": "DEFAULT",
   "requestId": "0ab75a5117804ac3a8fe36209d9dd069",
   "version": "3.3.0"
  },
  "body": {
   "taskId": 242447,
   "taskType": "DELIVER_SINGLE_PARCEL",
   "taskStatus": "EXECUTING",
   "taskPhase": "DELIVERING",
   "robotId": 1000,
   "stationId": 1,
   "dest": {
    "z": 1,
    "x": 51,
    "y": 88
   },
   "destLocation": {
    "z": 1,
    "x": 30.9,
    "y": 53.1
   },
   "destCellCode": "05150885",
   "beltPos": 4,
   "dropCellCode": "05250885"
  }
 }
}
```


主机系统应该给与 RMS 的响应如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackResponseMsg",
   "response": {
      "header": {
         "responseId": "8780c902ed714d10b82a2f20ac514a1c"
      }
   }
```

}

3.11.5.7 DELIVER_SINGLE_PARCEL 任务 DROP_FINISH

场景为机器人收包完毕后，主机系统使用 GO_DROP 更新任务使机器人去投递包裹，机器人投递结束后进行回调。

回调消息字段说明如下：

字段	类型	描述	示例
taskId	long	当前进行的任务 id	30
taskType	String	"DELIVER_SINGLE_PARCEL"	DELIVER_SINGLE_PARCEL
taskStatus	String	COMPLETED 完成任务	COMPLETED
taskPhase	String	DROP_FINISH，表示机器人投递结束	DROP_FINISH
robotId	Integer	执行任务的机器人 ID	1
stationId	Integer	表示该任务是从哪个供包台接的任务	1
机器人目的地字段	在此处将描述机器人目的地 若为投递口，则目的地为投递口旁选定的投递点 若为中转点，则目的地就是最终目的地		
dest	IPoint	目的地：索引坐标	{"z": 1,"x": 29,"y": 10}
destLocation	DPoint	目的地：绝对坐标	{"z": 1,"x": 29.5,"y": 10.5}
destCellCode	String	目的地：节点编码	102950105
beltPos	Integer	机器人收包状态 1：用于双皮带机器人，1 号皮带已收包 2：用于双皮带机器人，2 号皮带已收包 3：用于双皮带机器人，1,2 号皮带均已收包 4：一个占据两个皮带的大包裹，若为单皮带机器人，不需供包直接就为 4	4
dropCellCode	String	机器人当前选定的投递口	05250885

回调的报文示例如下：


```
{
 "id": "*",
 "msgType": "RobotTaskCallbackMsg",
 "request": {
  "header": {
   "warehouseCode": "DEFAULT",
   "requestId": "7fc07ceb273a40c6bc2596977f506af8",
```

```
  "version": "3.3.0"
 },
 "body": {
  "taskId": 242447,
  "taskType": "DELIVER_SINGLE_PARCEL",
  "taskStatus": "COMPLETED",
  "taskPhase": "DROP_FINISH",
  "robotId": 1000,
  "stationId": 1,
  "dest": {
   "z": 1,
   "x": 51,
   "y": 88
  },
  "destLocation": {
   "z": 1,
   "x": 30.9,
   "y": 53.1
  },
  "destCellCode": "05150885",
  "beltPos": 4,
  "dropCellCode": "05250885"
 }
}
}
```


主机系统应该给与 RMS 的响应如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "RobotTaskCallbackResponseMsg",
   "response": {
      "header": {
         "responseId": "8780c902ed714d10b82a2f20ac514a1c"
      }
   }
}
```


3.11.6 功能注意点

该场景为分拣场景特有任务。

1、需要配合仓库控制接口中的 OPEN_STATION、CLOSE_STATION 命令来实时开启/关闭供包台。

2、供包台的可用与否，可用任务更新接口来实时更新。

3.11.7 场景示例

下发系统	接收系统	消息类型	任务类型	任务执行阶段	任务状态	指令	要求字段	功能	备注
RMS		自主生成	GO_STATION_TO_QUEUE			GO_SOMEWHERE		自主生成，去往排队区，若为动态路径，仅一次任务即可到达供包台，若非动态，会有两段任务到达供包台	
RMS	主机系统	RobotTaskCallbackMsg		MOVING	EXCUTING			反馈上游出发前往排队区	
RMS	主机系统	RobotTaskCallbackMsg		ARRIVED	EXCITING			反馈上游已经到达排队区	
RMS		自主生成	DELIVER_SINGLE_PARCEL					自主生成收发包任务	

下发系统	接收系统	消息类型	任务类型	任务执行阶段	任务状态	指令	要求字段	功能	备注
RMS	主机系统	RobotTaskCallbackMsg		FEEDING	EXCITING			机器人已准备供包	
主机系统	RMS	RobotTaskUpdateRequestMsg				GO_NEXT	目的地	指派机器人先到某处缓存	主机系统可根据场景选择是否进行该更新
RMS	主机系统	RobotTaskCallbackMsg		MOVING	EXCUTING			GO_NEXT更新后机器人出发	
RMS	主机系统	RobotTaskCallbackMsg		ARRIVED	EXCUTING			GO_NEXT更新后机器人到达目的地	
主机系统	RMS	RobotTaskUpdateRequestMsg				GO_DROP	格口	指派机器人投递	必要更新
RMS	主机系统	RobotTaskCallbackMsg		DELIVERING	EXECUTING			机器人出发去投递	若主机更新了目的地，机器人在更改路径后会再次

下发系统	接收系统	消息类型	任务类型	任务执行阶段	任务状态	指令	要求字段	功能	备注
									回调
RMS	主机系统	RobotTaskCallbackMsg		DROP_FINISH	COMPLETED			机器人投递结束	
主机系统	RMS	RobotTaskUpdateRequestMsg				UPDATE_PARCEL		可在GO_DROP后随时更新目的地	主机系统可根据需要决定是否更新

4 仓库资源更新请求

4.1 简介

使用此消息，主机系统可以更新 RMS 系统中的参数资源，例如系统参数和货架参数。

4.2 概述

4.2.1 传输方向

主机系统发送资源参数更新请求消息给 RMS，RMS 回复响应消息。

4.2.2 内容

指令类型	指令内容	描述
系统相关	SYSTEM	系统参数修改
货架相关	SHELF	货架参数修改

4.2.3 消息结构

需要注意的是，此处“msgType”的值必须为“ParameterInstructionRequestMsg”。

区分不同资源类型的字段参数为消息体中的“parameterType”，具体的仓库资源更新请求消息详见后文。

示例如下：

```
{
    "id": "postman_001",
    "msgType": "ParameterInstructionRequestMsg",
    "request": {
      "header": {
        "requestId": " c39af3cf-0599-49fd-9c9f-8ba9ce78e69c",
        "clientCode": "geekClient",
        "warehouseCode": "geekWarehouseCode",
        "userId": "admin",
        "userKey": "111111",
        "version": "3.3.0",
        "language": "en_us"
      },
      "body": {
        "parameterType": "xxx",
        "xxx": "xxx",
```

```
            "xxx":"xxx"
        }
    }
}
```

正确的请求消息在通过校验后，RMS 回复的响应如下：

```
{
    "id": "postman_001",
    "msgType": "ParameterInstructionResponseMsg",
    "response": {
        "header": {
            "responseId": "c39af3cf-0599-49fd-9c9f-8ba9ce78e69c",
            "code": 0,
            "msg": "Success"
        }
    }
}
```

4.3 请求类型

4.3.1 系统参数

暂无。

4.3.2 货架参数

该部分用于修改货架的相关参数。接口需要的参数如下：

字段	类型	是否必要	描述	示例
parameterType	String	必要	值必须是 SHELF，表示对货架资源进行更新	SHELF
指定单个元素部分	用于指定货架，shelfCode 和 hostShelfCode 二选一，若均填写，则指代必须一致			
shelfCode	String	可选	货架编码	A0001
hostShelfCode	String	可选	货架外部编码	S
可更新参数部分	该部分将描述可更新的参数列表			
参数—分数	用于更新货架分数			
shelfScore	String	可选	[1-100]货架分数，该值允许随时更新	1
货架—角度	用于更新货架角度，除非特别需要和较为特殊的场景，请仅使用 0,90,180,-90 四个值，请参考货架面和货架角度的对应关系：以 F 面朝地图 0 度方向标示为货架 0 度用以计算货架角度			

字段	类型	是否必要	描述	示例
shelfAngle	Float	可选	货架初始角度 例如：90。-180~+180	90
参数—位置	位置若要更新则需要三选一，若均填写，指代必须一致，货架位置允许随时更新，但必须与现场实际位置一致			
locationCellCode	String	可选	目的节点编码	08150635
locationIndex	IPoint	可选	目的地索引坐标。即 z 轴，x 轴，y 轴。z 可省略,默认是 1(二维码区域专用,传索引值)	{"x":45,"y":42} {"z":1,"x":45,"y":42}
locationHostCode	String	可选	上位机系统关于当前任务的目的地坐标的描述	geekHostLocation01
参数—老家	老家位置若要更新则需要三选一，若均填写，指代必须一致，货架老家位置允许随时更新。 老家位置在更新货架返回库区时且不指定位置时生效 若老家位置生效但老家位置被其他货架占用会报错 货架老家涉及货架的默认位置、位置自动调整等各个敏感功能，若主机系统未对该功能有深入了解且经过 RMS 技术人员协助请不要对该参数进行修改。			
placementIndex	IPoint	可选	货架老家位置的坐标	{“z”:1,"x":12,"y":15}
placementCellCode	String	可选	货架老家位置的二维码位置	08150635
placementHostCode	String	可选	货架老家位置外部编码	geekHostLocation01
参数—类别	货架类别 用于控制货架能在哪类单元格上通行 控制货架能被哪类型机器人所搬运			
shelfClassCode	String	可选	货架的类型	A
批量更新部分	该部分允许批量更新货架参数，若该部分和单独更新均填写了，则指代必须一致			
shelves	object	可选	格式为集合，填写和校验要求同单独指定货架	[{"shelfCode": "A0001","shelfScore": "20"},{"shelfCode": "A0002","shelfScore": "20"}]

更新货架 A0001 的分数消息示例如下：

```
{
   "id": "postman_001",
   "msgType": "ParameterInstructionRequestMsg",
   "request": {
      "header": {
         "requestId": "9bab88d-a407-4195-a1f8-ae3594c83924",
```

```
            "clientCode": "geekCode",
            "warehouseCode": "geekWarehouseCode",
            "userId": "admin",
            "userKey": "111111",
            "version": "3.3.0",
            "language": "en_us"
        },
        "body": {
            "parameterType": "SHELF",
            "shelfCode": "A0001",
            "shelfScore": "20"
        }
    }
}
```


4.3.3 工作站参数

暂无。

4.3.4 充电站参数

暂无。

5 仓库控制请求

5.1 简介

使用此消息类型，主机系统可以借由 RMS 来控制仓库运行，例如系统紧急停止和系统恢复、区域锁定和释放、添加或移除货架、添加或移除机器人、打开或关闭分拣场景的工作站。

例如，如果主机系统连接到紧急停止按钮时，当按下该按钮时，请求消息可发送至 RMS 以控制所有机器人停止运动。

5.2 概述

5.2.1 传输方向

主机系统发送控制请求消息给 RMS，RMS 回复响应消息。

5.2.2 指令

当前 RMS 支持的仓库控制指令如下表所示。

指令类型	指令	功能描述
系统相关	SYSTEM_STOP	系统急停
	SYSTEM_RECOVER	系统恢复
	FIRE_STOP	消防急停
任务相关	TASK_STOP	任务暂停
	TASK_RECOVER	任务恢复
货架相关	ENTER_SHELF	货架入场
	ADD_SHELF	添加货架
	REMOVE_SHELF	移除货架
货箱相关	ADD_BOX	添加货箱
	REMOVE_BOX	移除货箱
单元格相关	LOCK_CELL	锁定单元格

指令类型	指令	功能描述
	STOP_CELL	暂停单元格
	UNLOCK_CELL	解锁单元格
机器人相关	REMOVE_ROBOT	移除机器人
	RECOVER_ROBOT	恢复机器人
工作站相关	OPEN_STATION	打开工作站
	CLOSE_STATION	关闭工作站

5.2.3 消息结构

消息结构与任务请求的消息结构相同。

需要注意的是，此处“msgType”的值必须为“WarehouseInstructionRequestMsg”。

区分不同消息指令的字段参数为消息体中的“instruction”，具体的仓库控制请求消息详见后文。

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "WarehouseInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": "c39af3cf-0599-49fd-9c9f-8ba9ce78e69c",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "instruction": "xxx",
      "xxx": "xxx"
    }
  }
}
```

正确的请求消息在通过校验后，RMS 回复的响应如下：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "WarehouseInstructionResponseMsg",
  "response": {
    "header": {
      "responseId": "c39af3cf-0599-49fd-9c9f-8ba9ce78e69c",
      "code": 0,
```

```
        "msg": "Success"
      }
    }
}
```

5.2.4 字段含义

在仓库控制的请求消息中可能包含的字段如下表，具体的每一种仓库控制请求及响应消息的详细内容请参考后文。

字段	类型	功能描述	示例
instruction	String	仓库控制指令	SYSTEM_STOP
logicId	Integer	楼层 ID	1
robotId	Integer	机器人 ID	1000
robotIds	list of Integer	机器人 ID 集合	[1000,1001]
hostRobotCode	String	机器人外部编号	geekRobotCode
shelfCode	String	货架编码	A000001
shelfCodes	list of String	货架编码集	["A000001","A000002"]
hostShelfCode	String	外部货架编码	geekShelfCode1
hostShelfCodes	list of String	外部货架编码集	[“geekShelfCode1”, “geekShelfCode2”]
locationIndex	DPoint	货架入场时货架的位置	{"z": 1,"x": 29.5,"y": 10.5}
locationCellCode	String	货架入场时货架的位置 code	108550235
locationHostCode	String	货架入场时货架的位置的外部编码	geek00001
shelfScore	Integer	货架热度	10
shelfClassCode	String	货架类别	A
shelfAngle	float	货架初始角度	90.0
placementIndex	IPoint	货架的摆放位置（老家位置）	{"z": 1,"x": 29,"y": 10}
placementCellCode	String	货架的摆放位置 code（老家位置）	108550235
placementHostCode	String	货架的摆放位置外部编码（老	geek00001

字段	类型	功能描述	示例
		家位置）	
cellPoint	IPoint	单元格索引坐标	{"z": 1,"x": 29,"y": 10}
cellPoints	list of Ipoint	单元格索引坐标集	[{"z": 1,"x": 29,"y": 10}, {"z": 1,"x": 30,"y": 10}]
cellCode	String	单元格编码	108550235
cellCodes	list of String	单元格编码集	["108550235","108550245"]
hostCellCode	String	外部单元格编码	geekCode1
hostCellCodes	list of String	外部单元格编码集	["geekCode1","geekCode2"]
chargerId	Integer	充电站 ID	1
chargerIds	list of Integer	充电站 ID 集	["1","2"]
hostChargerCode	String	外部充电站 Code	geekChargerCode1
hostChargerCodes	list of String	外部充电站 Code 集	[“geekChargerCode1”, "geekChargerCode2"]
stationId	Integer	工作站 ID	1
stationIds	list of Integer	工作站 ID 集	["1","2"]
hostStationCode	String	外部工作站 Code	geekStationCode1
hostStationCodes	list of String	外部工作站 Code 集	[“geekStationCode1”, "geekStationCode2"]
mapArea	list of IPoint	锁格/解锁的单元格列表	[{"z": 1,"x": 29,"y": 10}, {"z": 1,"x": 30,"y": 10}]
warehouseId	Integer	仓库编号	1
shelfPoint	IPoint	货架的位置（实时位置）	{"z": 1,"x": 29,"y": 10}
shelfScoreStrategy	String	货架热度计算策略	
shelfMoveCount	Integer	货架交换数量	
boxCode	String	货箱编号	B0000001
latticeCode	String	格子编号	L0000001
placeLatticeCode	String	老家格子编号	L0000001

5.3 请求说明

主机系统可以发送仓库控制指令的请求消息给 RMS，当校验通过后 RMS 会立即予以响应，并执行相关操作。

5.3.1 系统相关

主机系统可以通过 API 接口发送仓库控制指令的请求消息给 RMS，管理员也可以通过前端地图操作页面中的“系统急停”和“系统恢复”按钮停止和恢复 RMS，其作用与主机系统发送仓库控制请求一致。

5.3.1.1 系统急停

当 RMS 收到“*SYSTEM_STOP*”指令后，RMS 系统立即停止调度机器人，默认策略为不再向机器人发送路径信息，所有的机器人在走到已收到的路径点后停止运动。若 RMS 中配置为在系统急停后通知机器人停止，则机器人会在安全的刹车距离内停止，而不是走到系统已下发的路径点。

主机系统可以按区域进行急停。若系统已经急停，该指令将回调系统已经急停。

字段	类型	是否必要	描述	示例
instruction	String	必要	值必须是 SYSTEM_STOP	SYSTEM_STOP
areaIds	list of int	可选	当使用区域急停时使用此字段指定区域，若为空，则为全局急停	[1,2,3,4]

请求消息示例：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "WarehouseInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": "c39af3cf-0599-49fd-9c9f-8ba9ce78e69d",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "instruction": "SYSTEM_STOP"
    }
  }
}
```

5.3.1.2 系统恢复

当 RMS 收到“*SYSTEM_RECOVER*”指令后，系统将会恢复，所有机器人将会恢复工作。

若当前系统处于全局急停状态，不能指定区域进行恢复。

字段	类型	是否必要	描述	示例
instruction	String	必要	值必须是 SYSTEM_RECOVER	SYSTEM_RECOVER
areaIds	list of int	可选	当使用区域恢复时使用此字段，为空表示全局系统恢复	[1,2,3,4]

请求消息示例：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "WarehouseInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": "c39af3cf-0599-49fd-9c9f-8ba9ce78e69e",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "instruction": "SYSTEM_RECOVER"
    }
  }
}
```


5.3.1.3 消防急停

基本等同于系统急停，但 RMS 会调度机器人离开预先设置好的消防通道单元格上。

字段	类型	是否必要	描述	示例
instruction	String	必要	值必须是 FIRE_STOP	FIRE_STOP
areaIds	list of int	可选	当使用区域急停时使用此字段指定区域，若为空，表示全局消防急停。	[1,2,3,4]

请求消息示例：


```
{
  "id": "geekCode_geekWarehouseCode_001",
```

```
  "msgType": "WarehouseInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": "c39af3cf-0599-49fd-9c9f-8ba9ce78e69f",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "instruction": "FIRE_STOP"
    }
  }
}
```


5.3.2 任务相关

主机系统可以通过 API 接口发送仓库控制指令的请求消息给 RMS，管理员也可以通过前端地图操作页面中的“任务暂停”和“任务恢复”按钮停止和恢复任务分配，其作用与主机系统发送仓库控制请求一致。

5.3.2.1 任务暂停

当 RMS 收到“*TASK_STOP*”指令后，系统任务立即停止分配，所有的新任务不再分配给机器人，且所有的机器人将在完成当前任务后停止运动。

任务暂停所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 TASK_STOP

请求消息示例：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "WarehouseInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": "c39af3cf-0599-49fd-9c9f-8ba9ce78e69r",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
```

```
        },
        "body": {
          "instruction": "TASK_STOP"
        }
      }
    }
```

5.3.2.2 任务恢复

当 RMS 收到“*TASK_RECOVER*”指令后，当系统任务恢复分配，RMS 会继续分配新任务给机器人。

任务恢复所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 TASK_RECOVER

请求消息示例：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "WarehouseInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": "a39af3cf-0599-49fd-9c9f-8ba9ce78e69c",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "instruction": "TASK_RECOVER"
    }
  }
}
```

5.3.3 货架相关

当配置一个仓库时，主机系统可以使用仓库控制请求告诉 RMS 货架的信息，并由 RMS 调度机器人完成此货架的相关操作。

每个指令的具体应用场景和使用方式将在后文分别介绍。

5.3.3.1 货架入场

当仓库中需要新添加一个货架时，可以使用“*ENTER_SHELF*”指令。

该指令需要主机系统在发送的仓库控制请求包含：货架编号信息（shelfCode）及货架最初的位置（如“location”，表示地图的入口、仓库的货架入场点等）。

RMS 在接收到此请求后，会调用机器人去货架的初始位置处扛起货架，并将此货架随机搬运至货架存储区的库位（“placement”）进行放置。

管理员可以通过前端仓库管理页面中的“货架入场”功能完成该操作。

货架入场所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 ENTER_SHELF
shelfCode	String	必要	货架编号，最短 2 位，最长不得超过 16 位
locationIndex	IPoint	三种参数必须选择一个	货架入场的实时位置 例如：{“z”:1,"x":12,"y":15}
locationCellCode	String		货架入场的二维码位置 例如： 08150635
locationHostCode	String		货架入场外部编码 例如：geekHostLocation01
placementIndex	IPoint	三种参数必须选择一个	货架老家位置的坐标 例如：{“z”:1,"x":12,"y":15}
placementCellCode	String		货架老家位置的二维码位置 例如： 08150635
placementHostCode	String		货架老家位置外部编码 例如：geekHostLocation01
shelfClassCode	String	可选	货架类别 例如：A
shelfAngle	Float	可选	货架初始角度 例如：90
shelfScore	Integer	可选	货架热度 例如：5
logicId	Integer	可选	逻辑区域 id 例如：2

请求消息示例：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "WarehouseInstructionRequestMsg",
   "request": {
      "header": {
         "requestId": "d39af3cf-0599-49fd-9c9f-8ba9ce78e69c",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
```

```
            "userKey": "123456",
            "language": "en_us",
            "version": "3.3.0"
        },
        "body": {
            "shelfCode": "001",
            "instruction": "ENTER_SHELF",
            "locationCellCode": "00850085",
           "placementCellCode": "08150635"
        }
    }
}
```


5.3.3.2 添加货架

当仓库中需要新添加一个货架时，也可以使用“*ADD_SHELF*”指令。

该指令需要主机系统在发送的仓库控制请求包含：货架编号信息（shelfCode）、货架最初的位置（如“locationCellCode”，表示地图的入口、仓库的货架入场点等）以及货架在存储区的摆放位置（如“placementCellCode”）。

RMS 在接收到此请求后，会在系统及数据库中记录此货架的信息，前端界面也会随即进行显示。

若用户想对此货架进行操作，则必须使用货架相关的任务请求。

字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 ADD_SHELF
shelfCode	String	必要	货架编号，最短 2 位，最长不得超过 16 位
locationIndex	IPoint	三种参数必须选择一个	货架入场的实时位置 例如：{“z”:1,"x":12,"y":15}
locationCellCode	String		货架入场的二维码位置 例如： 08150635
locationHostCode	String		货架入场外部编码 例如：geekHostLocation01
placementIndex	IPoint	三种参数可选择一个	货架老家位置的坐标 例如：{“z”:1,"x":12,"y":15}
placementCellCode	String		货架老家位置的二维码位置 例如：08150635
placementHostCode	String		货架老家位置外部编码 例如：geekHostLocation01

字段	类型	是否必要	功能描述
shelfClassCode	String	可选	货架类别 例如：A
shelfAngle	Float	可选	货架初始角度 例如：90
shelfScore	Integer	可选	货架热度 例如：5
logicId	Integer	可选	逻辑区域 id 例如：2

请求消息示例：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "WarehouseInstructionRequestMsg",
   "request": {
      "header": {
         "requestId": "g39af3cf-0599-49fd-9c9f-8ba9ce78e69c",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "shelfCode": "001",
         "instruction": "ADD_SHELF",
         "locationCellCode": "00850085",
         "placementCellCode": "08150635"
      }
   }
}
```


5.3.3.3 移除货架

当货架不再使用时，主机系统用“*REMOVE_SHELF*”指令告诉 RMS 从系统中移除该货架。当且仅当该货架无任何搬运任务时 RMS 才会将它移除，并在系统中删除此货架的信息及释放该货架占用的资源，例如单元格。货架移除后，RMS 默认该货架已不在仓库中。

字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 REMOVE_SHELF
shelfCode	String	二选一	货架编号，最短 2 位，最长不得超过 16 位
shelfCodes	list of String		货架编号的集合

请求消息示例：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "WarehouseInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": "w39af3cf-0599-49fd-9c9f-8ba9ce78e69c",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "shelfCode": "001",
      "instruction": "REMOVE_SHELF"
    }
  }
}
```


5.3.4 机器人相关

当机器人发生异常或不能很好的工作时，主机系统可以发送请求去移除该机器人，也可以在机器人被修好后发送请求去恢复该机器人。

当机器人被移除时，它将会从地图上被释放，且它的任务将会转移给其他机器人（不同的任务类型的任务转移逻辑不一致，具体任务转移逻辑见第八章），不再会有其他任务分配给它；它所占用的所有资源都将释放，RMS 将会认为该机器人已被人工移动到场地外。管理员也可以通过前端网页的地图操作页面来移除或恢复机器人。

5.3.4.1 移除机器人

移除机器人所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 REMOVE_ROBOT
robotId	String	必要	填写需要移除机器人的编号

请求消息示例：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "WarehouseInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": "d39af3cf-0599-49fd-9c9f-8ba9ce78e69c",
```

```
            "clientCode": "geekCode",
            "warehouseCode": "geekWarehouseCode",
            "userId": "admin",
            "userKey": "123456",
            "language": "en_us",
            "version": "3.3.0"
        },
        "body": {
            "instruction": "REMOVE_ROBOT",
            "robotId": 1000
        }
    }
}
```


5.3.4.2 恢复机器人

恢复机器人所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 RECOVER_ROBOT
robotId	String	必要	填写需要移除机器人的编号

请求消息示例：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "WarehouseInstructionRequestMsg",
    "request": {
        "header": {
            "requestId": "f39af3df-0599-49fd-9c9f-8ba9ce78e69c",
            "clientCode": "geekCode",
            "warehouseCode": "geekWarehouseCode",
            "userId": "admin",
            "userKey": "123456",
            "language": "en_us",
            "version": "3.3.0"
        },
        "body": {
            "instruction": "RECOVER_ROBOT",
            "robotId": 1000
        }
    }
}
```


5.3.5 工作站相关

主机系统可以通过工作站相关指令打开或关闭工作站，仅适用分拣场景。

5.3.5.1 打开工作站

当 RMS 收到“*OPEN_STATION*”指令后，对应供包台状态为可用，此时系统会自动生成工作站排队任务。

打开工作站所需字段

字段	类型	是否必要	功能描述	示例
instruction	String	必要	值必须是 OPEN_STATION	OPEN_STATION
stationId	int	必要	工作站号	1

请求消息示例：


```
{
  "id": "postman_001",
  "msgType": "WarehouseInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": "285ceeaa-b513-410b-a8e8-b9d73fed0979",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "version": "3.3.0",
      "language": "en_us"
    },
    "body": {
      "instruction": "OPEN_STATION",
      "stationId": 1
    }
  }
}
```


RMS 的响应消息示例如下：


```
{
    "id": "postman_001",
    "msgType": "WarehouseInstructionResponseMsg",
    "response": {
        "header": {
            "responseId": "285ceeaa-b513-410b-a8e8-b9d73fed0979",
            "code": 0,
            "msg": "Success"
        }
    }
}
```

5.3.5.2 关闭工作站

当 RMS 收到“*CLOSE_STATION*”指令后，系统会关闭指定的供包台，不再为其分配生成新的分拣任务。

打开工作站所需字段				
字段	类型	是否必要	功能描述	示例
instruction	String	必要	值必须是 CLOSE_STATION	
stationId	int	必要	工作站号	1

请求消息示例：

```
{
  "id": "postman_001",
  "msgType": "WarehouseInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": "285ceeaa-b513-410b-a8e8-b9d73fed0981",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "version": "3.3.0",
      "language": "en_us"
    },
    "body": {
      "instruction": "CLOSE_STATION",
      "stationId": 1
    }
  }
}
```

RMS 的响应消息示例如下：

```
{
    "id": "postman_001",
    "msgType": "WarehouseInstructionResponseMsg",
    "response": {
        "header": {
            "responseId": "285ceeaa-b513-410b-a8e8-b9d73fed0981",
            "code": 0,
            "msg": "Success"
        }
    }
}
```

5.3.6 单元格相关

当地图中的某些区域需要锁定，主机系统可以发送锁定单元格请求去锁定该区域。

当单元格锁定时，机器人不能进入被锁定区域。主机系统也可以发送解锁单元格请求去解锁地图单元格。

管理员也可以通过前端网页的地图操作页面来锁定或解锁地图单元格。

5.3.6.1 锁定单元格

请求中的单元格被锁定后，机器人将无法抵达该区域。但请求下发前正在行驶中且将要通过该区域机器人可以继续行驶穿过该区域。

锁定单元格所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 LOCK_CELL
cellPoint	IPoint	必须选择其中一种	单元格的坐标索引值 {"z": 1,"x": 29,"y": 10}
cellPoints	list of IPoint		单元格的坐标索引值集合 [{"z": 1,"x": 29,"y": 10},{"z": 1,"x": 30,"y": 10}]
cellCode	String		填写需要暂停单元格的二维码信息 "108550235"
cellCodes	list of String		填写需要暂停单元格的二维码列表 ["108550235","108550245"]
hostCellCode	String		外部单元格编码信息 "geekCode1"
hostCellCodes	String		外部单元格编码集 ["geekCode1","geekCode2"]
mapArea	list of IPoint		填写需要锁定的单元格数组坐标 [{"z": 1,"x": 29,"y": 10},{"z": 1,"x": 30,"y": 10}]

请求消息示例：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "WarehouseInstructionRequestMsg",
   "request": {
      "header": {
         "requestId": "285ceeaa-b513-410b-a8e8-b9d73fed0911",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
```

```
        "version": "3.3.0"
      },
      "body": {
        "instruction": "LOCK_CELL",
        "cellCode": "08550105"
      }
    }
}
```


5.3.6.2 暂停单元格

功能同锁定单元格，请求下发前正在行驶中且将要通过该区域机器人也将会停下。

暂停单元格所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 STOP_CELL
cellPoint	IPoint	必须选择其中一种	单元格的坐标索引值 {"z": 1,"x": 29,"y": 10}
cellPoints	list of IPoint		单元格的坐标索引值集合 [{"z": 1,"x": 29,"y": 10},{"z": 1,"x": 30,"y": 10}]
cellCode	String		填写需要暂停单元格的二维码信息 "108550235"
cellCodes	list of String		填写需要暂停单元格的二维码列表 ["108550235","108550245"]
hostCellCode	String		外部单元格编码信息 "geekCode1"
hostCellCodes	String		外部单元格编码集 ["geekCode1","geekCode2"]
mapArea	list of IPoint		填写需要锁定的单元格数组坐标 [{"z": 1,"x": 29,"y": 10},{"z": 1,"x": 30,"y": 10}]

请求消息示例：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "WarehouseInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": "285ceeaa-b523-410b-a8e8-b9d73fed0981",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
```

```
    "body": {
      "instruction": "STOP_CELL",
      "cellCode": "08550105"
    }
  }
}
```

5.3.6.3 解锁单元格

解除被锁定（LOCK_CELL）或暂停（STOP_CELL）的单元格，使其恢复至正常状态。

解锁单元格所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 UNLOCK_CELL
cellPoint	IPoint	必须选择其中一种	单元格的坐标索引值 {"z": 1,"x": 29,"y": 10}
cellPoints	list of IPoint		单元格的坐标索引值集合 [{"z": 1,"x": 29,"y": 10},{"z": 1,"x": 30,"y": 10}]
cellCode	String		填写需要暂停单元格的二维码信息 "108550235"
cellCodes	list of String		填写需要暂停单元格的二维码列表 ["108550235","108550245"]
hostCellCode	String		外部单元格编码信息 "geekCode1"
hostCellCodes	String		外部单元格编码集 ["geekCode1","geekCode2"]
mapArea	list of IPoint		填写需要锁定的单元格数组坐标 [{"z": 1,"x": 29,"y": 10},{"z": 1,"x": 30,"y": 10}]

请求消息示例：

```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "WarehouseInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": "285ceeaa-b513-411b-a8e8-b9d73fed0981",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "instruction": "UNLOCK_CELL",
```

```
            "cellCode": "08550105"
        }
    }
}
```

5.3.7 货箱相关

当仓库中需要新添加新的货箱、或者移除不再使用的货箱时，可以使用相关指令进行操作。

5.3.7.1 添加货箱

当仓库中需要新添加一个货箱时，也可以使用“*ADD_BOX*”指令。

该指令需要主机系统在发送的仓库控制请求包含：货箱的编号信息（boxCode）、货箱目前所在格子（latticeCode）以及货箱的老家位置（placeLatticeCode）。

RMS 在接收到此请求后，会在系统中记录此货箱的信息。

添加货箱所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 ADD_BOX
boxCode	String	必要	货箱编码 “B0000001”
latticeCode	String	必要	货箱当前所在格子的编码 “L0000001”
placeLatticeCode	String	必要	货箱老家格子的编码 “L0000002”

请求消息示例：

```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "WarehouseInstructionRequestMsg",
    "request": {
        "header": {
            "requestId": "285ceeaa-b113-410b-a8e8-b9d73fed0981",
            "clientCode": "geekCode",
            "warehouseCode": "geekWarehouseCode",
            "userId": "admin",
            "userKey": "123456",
            "language": "en_us",
            "version": "3.3.0"
        },
        "body": {
            "instruction": "ADD_BOX",
            "boxCode": "B0000001",
            "latticeCode":"L0000002",
```

```
            "placeLatticeCode": "L0000002"
        }
    }
}
```

5.3.7.2 移除货箱

当货箱不再使用时，主机系统用“*REMOVE_BOX*”指令告诉 RMS 从系统中移除该货箱。

当且仅当货箱无任务时 RMS 才会将它移除，并在系统中删除此货箱的相关信息及释放占用的资源。

移除货箱所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 REMOVE_BOX
boxCode	String	必要	货箱编码 “B0000001”

请求消息示例：

```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "WarehouseInstructionRequestMsg",
    "request": {
        "header": {
            "requestId": "285ceeaa-b513-41ww-a8e8-b9d73fed0981",
            "clientCode": "geekCode",
            "warehouseCode": "geekWarehouseCode",
            "userId": "admin",
            "userKey": "123456",
            "language": "en_us",
            "version": "3.3.0"
        },
        "body": {
            "instruction": "REMOVE_BOX",
            "boxCode": "B0000001"
        }
    }
}
```

6 查询请求

6.1 简介

使用此消息类型，主机系统可以查询 RMS 系统相关资源的信息及当前状态，例如任务信息、机器人信息、货架信息及工作站信息等。RMS 将会把主机系统查询的资源信息进行封装，并以响应的形式回复给主机系统。

主机系统可以依据查询到的资源信息进行自己的业务逻辑处理。

6.2 概述

6.2.1 传输方向

主机系统发送查询请求消息给 RMS，RMS 将查询结果通过响应消息返回给主机系统。

6.2.2 查询指令

当前 RMS 支持的查询内容如下表所示。

指令	说明
TASK	查询任务信息
SHELF	查询货架信息
CHARGER	查询充电站信息
STATION	查询工作站信息
ROBOT	查询机器人信息
CELL	查询地图单元格信息
LATTICE	查询格子信息（即货箱的货箱位）
BOX	查询货箱信息
DISTANCE	查询两点之间的距离

6.2.3 消息结构

消息结构与仓库控制请求的消息结构相同。需要注意的是，此处“msgType”的值必须

为“QueryInstructionRequestMsg”。

区分查询消息的字段参数为消息体中的“instruction”，具体的查询请求消息详见后文。

请求消息示例：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "QueryInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": "3cc22247-d338-4c93-8eb0-9b2e13566dd5",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "instruction": "TASK"
    }
  }
}
```


响应消息示例：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "QueryInstructionResponseMsg",
  "response": {
    "header": {
      "responseId": "3cc22247-d338-4c93-8eb0-9b2e13566dd5",
      "code": 0,
      "msg": "Success"
    },
    "body": {
      "tasks": [
        {
          "taskId": 7,
          "taskType": "DELIVER_BOX",
          "taskStatus": "EXECUTING",
          "robotId": 1000,
          "stationId": 1,
          "taskPhase": "BOX_FETCHED",
          "priority": 0,
          "destPoint": {
            "z": 1,
            "x": 6.825,
            "y": 5.5
          },
          "destIndex": {
            "z": 1,
```

```
                "x": 6,
                "y": 5
            },
            "destCellCode": "00650055"
        }
      ]
    }
  }
}
```


6.2.3 字段含义

在查询请求消息中可能包含的字段如下表，具体的每一种查询请求消息及响应消息的详细内容请参考后文。

字段	类型	功能描述	示例
instruction	String	查询指令	ROBOT
map	Integer	地图 ID	1
floor	Integer	楼层 ID	2
logicId	Integer	逻辑区域 ID	1
robotId	Integer	机器人 ID	1000
robotIds	list of Integer	机器人 ID 集合	[1000,1001]
robotStateCode	Integer	机器人状态值	0
robotIdleCode	Integer	机器人空闲状态值	0
robotErrorCode	Integer	机器人错误码	0
shelfCode	String	货架编码	A000001
shelfCodes	list of String	货架编码集	["A000001","A000002"]
hostShelfCode	String	外部货架编码	geekShelfCode1
hostShelfCodes	list of String	外部货架编码集	[“geekShelfCode1”, “geekShelfCode2”]
shelfLoadStatus	Integer	货架被装载状态集合	0
locationCellCodes	list of String	位置单元格编码集合	["cellCode1","cellCode2"]
cellCode	String	单元格编码	108550235
cellCodes	list of String	单元格编码集	["108550235","108550245"]

字段	类型	功能描述	示例
index	IPoint	单元格索引	{"z": 1,"x": 29,"y": 10}
indexes	list of IPoint	单元格索引集合	[{"z": 1,"x": 29,"y": 10},{"z": 1,"x": 30,"y": 10}]
hostCellCode	String	外部单元格编码	geekCode1
hostCellCodes	list of String	外部单元格编码集	["geekCode1","geekCode2"]
cellTypes	list of String	单元格类型集合	
cellStatus	list of String	单元格状态	
cellFunctions	list of String	单元格功能列表	
chargerId	Integer	充电站 ID	1
chargerIds	list of Integer	充电站 ID 集	["1","2"]
hostChargerCode	String	外部充电站 Code	geekChargerCode1
hostChargerCodes	list of String	外部充电站 Code 集	[“chargerCode1”,"chargerCode2"]
stationId	Integer	工作站 ID	1
stationIds	list of Integer	工作站 ID 集	["1","2"]
hostStationCode	String	外部工作站 Code	geekStationCode1
hostStationCodes	list of String	外部工作站 Code 集	[“stationCode1”,"stationCode2"]

6.3 查询类型

6.3.1 任务信息

任务信息查询只能查询正在执行中的任务，已完成的任务在此处不能查询。

6.3.1.1 所需字段

任务信息查询所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 TASK
queryType	String	非必要	查询信息返回模式，默认 simple simple：简要模式 detail：详细模式
taskId	long	任意选择一个，不填写查询全部	填写需要查询任务的任务号
taskIds	list of long		填写需要查询任务的集合
taskTypes	list of String	非必填，所有查询条件均为与关系	查询指定任务类型集合的任务 TASK_TYPES
taskStatus	list of String		查询指定任务状态集合的任务 TASK_STATUS
robotIds	int of String		查询指定机器人集合的任务 ROBOT_IDS
taskPhases	list of String		查询指定任务阶段集合的任务 TASK_PHASES
stationIds	int of String		查询指定工作站集合的任务 STATION_IDS
shelfCodes	list of String		查询指定货架集合的任务 SHELF_CODES

返回的任务信息属性说明如下表，注意只有 taskId 字段是必填，其余字段可能为空。

字段	类型	描述	仅详细模式
taskId	long	任务 ID	
taskType	String	任务类型	
taskStatus	String	任务状态	
robotId	int	机器人 ID	
stationId	Integer	工作站 ID	
chargerId	Integer	充电站 ID	
shelfCode	String	货架	
taskPhase	String	任务阶段	

字段	类型	描述	仅详细模式
priority	Integer	任务优先级	
destPoint	DPoint	子任务终点实际坐标	
destIndex	IPoint	子任务终点索引坐标	
destCellCode	String	子任务终点节点编码	
startPoint	DPoint	子任务起点实际坐标	是
startIndex	IPoint	子任务起点索引坐标	是
startCellCode	String	子任务起点节点编码	是
startHostCode	String	子任务起点节点外部编码	是
destHostCode	String	子任务终点节点外部编码	是
businessSequence	int	业务优先级[0, 100000]	是

6.3.1.2 消息示例

请求消息示例：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "QueryInstructionRequestMsg",
   "request": {
      "header": {
         "requestId": "9ef5dfdc-eca7-43ee-93b2-045a9e4dc8c4",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "instruction": "TASK",
         "queryType":"detail"
      }
   }
}
```

响应消息示例：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "QueryInstructionResponseMsg",
```

```
  "response": {
    "header": {
      "responseId": "9ef5dfdc-eca7-43ee-93b2-045a9e4dc8c4",
      "code": 0,
      "msg": "Success"
    },
    "body": {
      "tasks": [
        {
          "taskId": 7,
          "taskType": "DELIVER_BOX",
          "taskStatus": "EXECUTING",
          "robotId": 1000,
          "stationId": 1,
          "taskPhase": "BOX_ARRIVED",
          "priority": 0,
          "destPoint": {
            "z": 1,
            "x": 6.825,
            "y": 5.5
          },
          "destIndex": {
            "z": 1,
            "x": 6,
            "y": 5
          },
          "destCellCode": "00650055",
          "startPoint": {
            "z": 1,
            "x": 0.0,
            "y": 0.0
          },
          "businessSequence": 0
        }
      ]
    }
  }
}
```

6.3.2 货架信息

6.3.2.1 所需字段

查询货架信息所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 SHELF
queryType	String	非必要	查询信息返回模式，默认 simple。 simple：简要模式 detail：详细模式
shelfCode	String	任意选择填写，不填写查询全部	填写需要查询货架的货架号
shelfCodes	list of String		填写需要查询货架的货架号集合
hostShelfCode	String		填写需要查询货架的外部编号
hostShelfCodes	list of String		填写需要查询货架的外部编号集合
shelfLoadStatus	int	非必填，所有查询条件均为与关系	根据货架被机器人托举的装载状态查询，0 全部/默认，1 空载，2 负载 SHELF_LOAD_STATUS
locationCellCodes	list of String		查询指定点集合上的货架，CellCode 点上的所有货架 LOCATION_CELL_CODES

返回的货架信息属性说明如下表，注意只有 shelfCode 字段是必填，其余字段可能为空。

字段	类型	描述	仅详细模式
shelfCode	String	货架编码	
hostShelfCode	String	货架外部编码	
shelfStatus	String	货架状态, SUBMITTED（已提交在仓库入口处） IN_WAREHOUSE（正常在库） POS_CONFIRMED（位置确认）	
length	int	货架长度，单位 mm	
width	int	货架宽度，单位 mm	
angle	float	货架角度	
location	DPoint	货架位置，实际位置	

字段	类型	描述	仅详细模式
locationIndex	IPoint	位置索引	
locationCellCode	String	位置对应的单元格的编码	
locationHostCode	String	位置对应的单元格的外部编码	
placement	DPoint	货架老家位置	
placementIndex	IPoint	货架老家位置索引	
placementCellCode	String	货架老家单元格编码	
placementHostCode	String	货架老家单元格外部编码	
robotId	int	机器人 id，该货架在哪个机器人上	是
shelfHeat	int	货架热度	是
logicId	int	货架所在的逻辑区 id	是
warehouseCode	String	仓库编码	是

6.3.2.2 消息示例

请求消息示例：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "QueryInstructionRequestMsg",
   "request": {
      "header": {
         "requestId": "61289a2a-80fc-4fe3-a984-4b764dd294e6",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "instruction": "SHELF",
         "shelfCode": "001",
         "queryType": "simple"
      }
   }
}
```

响应消息示例：

```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "QueryInstructionResponseMsg",
    "response": {
        "header": {
            "responseId": "61289a2a-80fc-4fe3-a984-4b764dd294e6",
            "code": 0,
            "msg": "Success"
        },
        "body": {
            "shelves": [
                {
                    "shelfCode": "001",
                    "shelfStatus": "IN_WAREHOUSE",
                    "length": 880,
                    "width": 880,
                    "angle": 0.0,
                    "location": {
                        "z": 1,
                        "x": 9.975,
                        "y": 8.54
                    },
                    "locationIndex": {
                        "z": 1,
                        "x": 9,
                        "y": 8
                    },
                    "locationCellCode": "00950085",
                    "placement": {
                        "z": 1,
                        "x": 0.0,
                        "y": 0.0
                    },
                    "placementIndex": {
                        "z": 1,
                        "x": 0,
                        "y": 0
                    }
                }
            ]
        }
    }
}
```

6.3.3 充电站信息

6.3.3.1 所需字段

充电站信息查询所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 CHARGER
queryType	String	非必要	查询信息返回模式，默认 simple simple：简要模式 detail：详细模式
chargerId	Integer	任意选择填写，不填写查询全部	填写需要查询充电站的 Id
chargerIds	list of Integer		填写需要查询货架的货架号集合

返回结果

字段	类型	描述	仅详细模式
chargerId	int	充电站 id	
hostChargerCode	String	充电站外部编码	
status	String	充电站状态， VACANT（空闲）、 ALLOCATED（已分配）、 OCCUPIED_CHARGING（被机器人占用，正在给机器人充电）、 OCCUPIED_OFFPOWER（被机器人占用，未接通充电）	
openFlag	int	充电站开启状态，1:开启；0:关闭	
location	DPoint	充电站位置	
locationIndex	IPoint	位置索引	
locationCellCode	String	位置对应的单元格的编码	
locationHostCode	String	位置对应的单元格的外部编码	
robotId	int	机器人 id	是
channels	list of ChargerChannel	充电通道列表	是

对象 ChargerChannel 说明：

字段	类型	描述
id	int	通道编号
status	int	通道状态 0 表示正常，其他异常
voltage	float	充电电压
current	float	充电电流
temperature	int	充电温度

6.3.3.2 消息示例

请求消息示例：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "QueryInstructionRequestMsg",
   "request": {
      "header": {
         "requestId": "968ee236-aa8d-47d2-80ef-552a09437a99",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "instruction": "CHARGER",
         "chargerId": 1,
         "queryType": "detail"
      }
   }
}
```


响应消息示例：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "QueryInstructionResponseMsg",
   "response": {
      "header": {
         "responseId": "968ee236-aa8d-47d2-80ef-552a09437a99",
         "code": 0,
         "msg": "Success"
      },
      "body": {
         "chargers": [
```

```
            {
               "chargerId": 1,
               "status": "VACANT",
               "openFlag": 1,
               "location": {
                  "z": 1,
                  "x": 9.975,
                  "y": 7.5
               },
               "locationIndex": {
                  "z": 1,
                  "x": 9,
                  "y": 7
               },
               "locationCellCode": "00950075",
               "robotId": -1,
               "channels": [
                  {
                     "id": 0,
                     "status": -1,
                     "voltage": 0.0,
                     "current": 0.0,
                     "temperature": 0
                  },
                  {
                     "id": 0,
                     "status": -1,
                     "voltage": 0.0,
                     "current": 0.0,
                     "temperature": 0
                  }
               ]
            }
         ]
      }
   }
}
```


6.3.4 单元格信息

6.3.4.1 所需字段

字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 CELL
queryType	String	非必要	查询信息返回模式，默认 simple

字段	类型	是否必要	功能描述
			simple：简要模式 detail：详细模式
index	IPoint	任意选择填写，不填写查询全部	填写需要查询单元格的坐标索引
indexes	list of IPoint		填写需要查询单元格的坐标索引集合
cellCode	String		填写需要查询单元格的编号
cellCodes	list of String		填写需要查询单元格的编号集合
hostCellCode	String		填写需要查询单元格的外部编号
hostCellCodes	list of String		填写需要查询单元格的外部编号集合
cellTypes	list of String	非必填，所有查询条件均为与关系	填写单元格类型集合 单元格类型，NULL_CELL(空单元格)、SHELF_CELL（放置货架的单元格)、E2W_PATH_CELL（东向西过道的单元格）、W2E_PATH_CELL（西向东过道的单元格）、S2N_PATH_CELL（南向北过道的单元格）、N2S_PATH_CELL(北向南过道的单元格)、E2W_S2N_PATH_CELL（东向西过道与南向北过道交叉单元格）、E2W_N2S_PATH_CELL（东向西过道与北向南过道交叉单元格）、W2E_S2N_PATH_CELL（西向东过道与南向北过道交叉单元格）、W2E_N2S_PATH_CELL（西向东过道与北向南过道交叉单元格）、E2W_W2E_PATH_CELL（可向东西两个方向走的单元格）、N2S_S2N_PATH_CELL（可向南北两个方向走的单元格）、E2W_W2E_N2S_PATH_CELL(可向东西南三个方向走的单元格)、E2W_W2E_S2N_PATH_CELL（可向东西北三个方向走的单元格）、N2S_S2N_E2W_PATH_CELL（可向南北西三个方向走的单元格）、N2S_S2N_W2E_PATH_CELL（可向南北东三个方向走的单元格）、OMNI_DIR_CELL（可以向四个方向行走的单元格）、STATION_CELL（拣货站拣货的位置）、TURN_CELL（拣货站转向的位置，只有标准工位田字格使用）、QUEUE_CELL（拣货站排队路径）、CHARGER_CELL(充电桩位置)、CHARGER_PI_CELL（充电桩电源接口位置）、BLOCKED_CELL（阻塞的单元格）、ENTRY_CELL（入口）、EXIT_CELL（出口）、ELEVATOR_CELL(电梯)、DROP_CELL（投递点）、PALLET_RACK_CELL（托盘架）、BOX_RACK_CELL（固定货架）
cellStatus	list of String		填写单元格状态集合 VACANT（空位） ORDERED（被预订） ALLOCATED(已分配) OCCUPIED(被占用)

字段	类型	是否必要	功能描述
			OCCUPIED_BY_SHELF(被货架占用)
cellFunctions	list of String		填写单元格功能集合 BACKUP_CELL（后退功能）、BEEP_CELL（鸣笛功能）、WAIT_CELL（等待功能）、AVOID_CELL（避让功能）、SKIP_CELL（跳过功能）、REST_CELL（停靠功能）、TURN_CELL（转面功能）、 NOT_ALLOW_TURN_CELL（禁止转弯功能）、 FIRE_PASS（消防功能）、RECYCLEID（回收点功能）、RESTART_CELL（重启功能）、 STAY_BLOCKING_CELL（停留阻塞点）

返回的单元格信息属性说明如下表，注意只有 *cellCode* 字段是必填，其余字段可能为空。

字段	类型	描述	仅详细模式
cellCode	String	单元格编码	
location	DPoint	位置	
index	IPoint	位置索引	
hostCellCode	String	单元格外部编码	
cellType	String	单元格类型，NULL_CELL(空单元格)、 SHELF_CELL（放置货架的单元格)、 E2W_PATH_CELL（东向西过道的单元格）、 W2E_PATH_CELL（西向东过道的单元格）、 S2N_PATH_CELL（南向北过道的单元格）、 N2S_PATH_CELL(北向南过道的单元格)、 E2W_S2N_PATH_CELL（东向西过道与南向北过道交叉单元格）、E2W_N2S_PATH_CELL（东向西过道与北向南过道交叉单元格）、 W2E_S2N_PATH_CELL（西向东过道与南向北过道交叉单元格）、W2E_N2S_PATH_CELL（西向东过道与北向南过道交叉单元格）、 E2W_W2E_PATH_CELL（可向东西两个方向走的单元格）、N2S_S2N_PATH_CELL（可向南北两个方向走的单元格）、 E2W_W2E_N2S_PATH_CELL(可向东西南三个方向走的单元格)、E2W_W2E_S2N_PATH_CELL（可向东西北三个方向走的单元格）、 N2S_S2N_E2W_PATH_CELL（可向南北西三个方向走的单元格）、N2S_S2N_W2E_PATH_CELL（可向南北东三个方向走的单元格）、 OMNI_DIR_CELL（可以向四个方向行走的单元	

字段	类型	描述	仅详细模式
		格）、STATION_CELL（拣货站拣货的位置）、TURN_CELL（拣货站转向的位置，只有标准工位田字格使用）、QUEUE_CELL（拣货站排队路径）、CHARGER_CELL(充电桩位置)、CHARGER_PI_CELL（充电桩电源接口位置）、BLOCKED_CELL（阻塞的单元格）、ENTRY_CELL（入口）、EXIT_CELL（出口）、ELEVATOR_CELL(电梯)、DROP_CELL（投递点）、PALLET_RACK_CELL（托盘架）、BOX_RACK_CELL（固定货架）	
cellStatus	String	单元格状态， VACANT（空位） ORDERED（被预订） ALLOCATED(已分配) OCCUPIED(被占用) OCCUPIED_BY_SHELF(被货架占用)	
length	double	单元格长度	
width	double	单元格宽度	
floorId	int	楼层 id	是
warehouseCode	String	仓库编码	是
stationId	int	工作站 id	是
chargerId	int	充电站 id	是
allocatedRobotId	int	单元格分配的机器人 id	是
occupyRobotId	int	单元格占用的机器人 id	是
occupiedShelfCode	String	单元格占用的货架的编码	是
cellFunctions	list of String	单元格功能列表,BACKUP_CELL（后退功能）、BEEP_CELL（鸣笛功能）、WAIT_CELL（等待功能）、AVOID_CELL（避让功能）、SKIP_CELL（跳过功能）、REST_CELL（停靠功能）、TURN_CELL（转面功能）、NOT_ALLOW_TURN_CELL（禁止转弯功能）、FIRE_PASS（消防功能）、RECYCLEID（回收点功能）、RESTART_CELL（重启功能）、STAY_BLOCKING_CELL（停留阻塞点）	是

6.3.4.2 消息示例

请求消息示例：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "QueryInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": " c45ac33f-48be-4aba-ab14-1ed705ff8b72",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "instruction": "CELL",
      "index": {
        "z": 1,
        "x": 6,
        "y": 5
      },
      "queryType": "detail"
    }
  }
}
```


响应消息示例：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "QueryInstructionResponseMsg",
  "response": {
    "header": {
      "responseId": "c45ac33f-48be-4aba-ab14-1ed705ff8b72",
      "code": 0,
      "msg": "Success"
    },
    "body": {
      "cells": [
        {
          "cellCode": "00650055",
          "location": {
            "z": 1,
            "x": 6.825,
            "y": 5.5
          },
          "index": {
            "z": 1,
```

```
                "x": 6,
                "y": 5
            },
            "cellType": "STATION_CELL",
            "cellStatus": "VACANT",
            "length": 1.05,
            "width": 1.0,
            "floorId": 1,
            "warehouseCode": "DEFAULT",
            "stationId": 1
        }
    ]
  }
 }
}
```


6.3.5 工作站信息

6.3.5.1 所需字段

字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 STATION
queryType	String	非必要	查询信息返回模式，默认 simple simple：简要模式 detail：详细模式
stationId	Integer	任意选择填写，不填写查询全部	填写需要查询工作站的 Id
stationIds	list of Integer		填写需要查询工作站的 Id 集合
stationTypes	list of String	非必填，所有查询条件均为与关系	填写工作站类型集合 SINGLE_STATION //单点工作站 FEEDER_STATION //供包台 STANDARD_STATION //标准工作站
layouts	list of String		填写工作站布局集合（仅对标准工作站有效） UPRIGHT_ANTICLOCK //立式-逆时针 UPRIGHT_CLOCK //立式-顺时针 LAYDOWN_ANTICLOCK //卧式-逆时针 LAYDOWN_CLOCK //卧式-顺时针 TWOBYFOUR_CLOCK //两行四列顺时针 SORTING_LINE //垂直式分拣工作站 队头是工位 SINGLE_CELL //单点操作位置 FOURBYTWO_ANTICLOCK //四行两列-逆时针 d_shape //d 型排队工位 b_shape //b 型排队工位

字段	类型	是否必要	功能描述
			UNION_TWOCELL //两个单点工位组合，暂未实现 THREEBYTWO_CLOCK //三行两列-顺时针 TWOBYFOUR_ANTICLOCK //两行四列逆时针 FOURBYTWO_CLOCK //四行两列-逆时针 TWOBYTHREE_CLOCK // 两行三列顺时针 TWOBYTHREE_ANTICLOCK //两行三列逆时针 THREEBYTWO_ANTICLOCK //三行两列逆时针 FOURBYFOUR_CLOCK //4x4 顺时针 FOURBYFOUR_ANTICLOCK //4x4 逆时针

返回的工作站信息属性说明如下表，注意只有 *stationId* 字段是必填，其余字段可能为空。

字段	类型	功能描述
stationId	int	工作站 id
hostStationCode	String	工作站外部编码
stationType	String	工作站类型 SINGLE_STATION //单点工作站 FEEDER_STATION //供包台 STANDARD_STATION //标准工作站
layout	String	工作站布局 UPRIGHT_ANTICLOCK //立式-逆时针 UPRIGHT_CLOCK //立式-顺时针 LAYDOWN_ANTICLOCK //卧式-逆时针 LAYDOWN_CLOCK //卧式-顺时针 TWOBYFOUR_CLOCK //两行四列顺时针 SORTING_LINE //垂直式分拣工作站 队头是工位 SINGLE_CELL //单点操作位置 FOURBYTWO_ANTICLOCK //四行两列-逆时针 d_shape //d 型排队工位 b_shape //b 型排队工位 UNION_TWOCELL //两个单点工位组合，暂未实现 THREEBYTWO_CLOCK //三行两列-顺时针 TWOBYFOUR_ANTICLOCK //两行四列逆时针 FOURBYTWO_CLOCK //四行两列-逆时针 TWOBYTHREE_CLOCK // 两行三列顺时针 TWOBYTHREE_ANTICLOCK //两行三列逆时针 THREEBYTWO_ANTICLOCK //三行两列逆时针 FOURBYFOUR_CLOCK //4x4 顺时针 FOURBYFOUR_ANTICLOCK //4x4 逆时针
placeDir	String	工作站工作位置朝向 EAST // 工位在工作点东侧

字段	类型	功能描述
		SOUTH // 工位在工作点南侧 WEST // 工位在工作点西侧 NORTH // 工位在工作点北侧
queueRobotSize	int	排队的机器人数量
maxQueueRobotNum	int	工作站允许机器人最大排队数量
location	DPoint	位置
locationIndex	IPoint	位置索引
locationCellCode	String	位置对应的单元格的编码
stationStatus	String	分拣工作站状态，开启（OPEN）或是关闭（CLOSE）

6.3.5.2 消息示例

请求消息示例：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "QueryInstructionRequestMsg",
   "request": {
      "header": {
         "requestId": "136fa8db-7d02-47bd-bdeb-7d8b332784b4",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "instruction": "STATION",
         "stationId": 1,
         "queryType": "detail"
      }
   }
}
```


响应消息示例：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "QueryInstructionResponseMsg",
   "response": {
      "header": {
         "responseId": "136fa8db-7d02-47bd-bdeb-7d8b332784b4",
```

```
        "code": 0,
        "msg": "Success"
      },
      "body": {
        "stations": [
          {
            "stationId": 1,
            "stationType": "SINGLE_STATION",
            "layout": "SINGLE_CELL",
            "placeDir": "SOUTH",
            "queueRobotSize": 0,
            "maxQueueRobotNum": 1,
            "location": {
              "z": 1,
              "x": 6.825,
              "y": 5.5
            },
            "locationIndex": {
              "z": 1,
              "x": 6,
              "y": 5
            },
            "locationCellCode": "00650055",
            "stationStatus": "OPEN"
          }
        ]
      }
    }
}
```


6.3.6 机器人信息

6.3.6.1 所需字段

机器人信息查询所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 ROBOT
queryType	String	非必要	查询信息返回模式，默认 simple simple：简要模式 detail：详细模式
robotId	Int	任意选择填写，不填写查询全部	填写需要查询机器人的 Id
robotIds	list of Int		填写需要查询机器人的 Id 集合

机器人信息查询所需字段			
字段	类型	是否必要	功能描述
robotProducts	list of String	非必填，所有查询条件均为与关系	填写机器人产品集合
robotStateCode	int		根据机器人连接状态查询，0 全部/默认，1 正常（NORMAL、NEW_COMMER），2 连接异常（DISCONNECTED、CONNECTED_AGAIN）
robotIdleCode	int		根据机器人工作状态查询，0 全部/默认，1 工作中（机器人含任意任务且未被移除），2 空闲中
robotErrorCode	int		根据机器人异常状态查询，0 全部/默认，1 正常，2 异常

返回的机器人信息属性说明如下表，**注意只有 robotId 字段是必填**，其余字段可能为空。

字段	类型	描述	仅详细模式
robotId	int	机器人 id	
robotProduct	String	机器人产品型号，S10（分拣翻版机器人）、S10C（分拣单皮带机器人）、S20（分拣单皮带机器人）、S20T（分拣右翻板机器人）、S20C（分拣单皮带机器人）、S500C（分拣双皮带机器人）、P500L（货架举升机器人）、P800L（货架举升机器人）、P800TLP（带有固定托盘的举升机器人）、P500R（滚筒机器人）、P800R（滚筒机器人）、M100L（带举升功能的 SLAM 导航机器人）、M100R（带辊道功能的 SLAM 导航机器人）、M100D（带牵引功能的 SLAM 导航机器人）、M100N（无附加功能的 SLAM 导航机器人）、M500L（带举升功能的 SLAM 导航机器人）、M500R（带辊道功能的 SLAM 导航机器人）、M800L（带举升功能的 SLAM 导航机器人）、M800R（带辊道功能的 SLAM 导航机器人）、M1000L（带举升功能的 SLAM 导航机器人）、M1000R（带辊道功能的 SLAM 导航机器人）、F14L（1.4 吨举升式叉车高度 3 米）、F20T（2.0 吨地牛式叉车）、C200S（报箱机器人）、M100PR（辊筒型 M100, 平行辊筒）	
robotStatus	String	机器人连接状态，NEW_COMMER（新加入系统）、DISCONNECTED(链接断开)	

字段	类型	描述	仅详细模式
		、CONNECTED_AGAIN(再次连入系统)、NORMAL(工作正常)、SLEEPING(休眠中)、REMOVED_FROM_SYSTEM(从系统移除)	
errorCodes	String	机器人错误码，多个以逗号分割	
powerPercent	int	机器人电池电量	
angle	int	机器人角度	
location	DPoint	位置	
locationIndex	IPoint	位置索引	
locationCellCode	String	位置对应的单元格的编码	
locationHostCode	String	位置对应的单元格的外部编码	
robotPathMode	String	机器人路径模型 GO_DELIVERING, // 去送货架 GO_RETURN, // 归还货架 QUEUING, // 排队 GO_FETCHING, // 去取货架 GO_CHARGING, // 去充电 CHARGING, // 充电中 GO_REST, // 去固定位置 FIRMWARE_UPDATE, // 固件升级 IDLE, // 空闲状态 GO_RECEIVE, // 自动收货 GO_DROP, // 卸货 GO_FETCH_BOX, // 去取箱子 GO_DELIVER_BOX, // 去送箱子 GO_RETURN_BOX, // 去还箱子 MANUAL, //遥控模式 MAP_INIT,//地图初始化 INVALID, // 非法模式 PALLET_GO_FETCHING, // 去取托盘 GO_RETURN_PALLET, // 归还托盘 GO_DELIVER_PALLET; // 去送托盘	是
taskId	long	任务号	是
path	list of DPoint	路径	是
voltage	double	电池电压	是
current	double	电池电流	是

字段	类型	描述	仅详细模式
temperature	double	电池温度	是
ip	String	机器人 ip	是

6.3.6.2 消息示例

请求消息示例：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "QueryInstructionRequestMsg",
  "request": {
    "header": {
      "requestId": "6c545c40-a448-4539-b209-0d692b54c451",
      "clientCode": "geekCode",
      "warehouseCode": "geekWarehouseCode",
      "userId": "admin",
      "userKey": "123456",
      "language": "en_us",
      "version": "3.3.0"
    },
    "body": {
      "instruction": "ROBOT",
      "robotId": 1000,
      "queryType": "detail"
    }
  }
}
```


响应消息示例：


```
{
  "id": "geekCode_geekWarehouseCode_001",
  "msgType": "QueryInstructionResponseMsg",
  "response": {
    "header": {
      "responseId": "6c545c40-a448-4539-b209-0d692b54c451",
      "code": 0,
      "msg": "Success"
    },
    "body": {
      "robots": [
        {
          "robotId": 1000,
          "robotProduct": "C200S",
          "robotStatus": "CONNECTED_AGAIN",
          "errorCodes": "",
          "powerPercent": 55,
```

```
                "angle": 0.0,
                "location": {
                    "z": 1,
                    "x": 6.825,
                    "y": 7.5
                },
                "locationIndex": {
                    "z": 1,
                    "x": 6,
                    "y": 7
                },
                "locationCellCode": "00650075",
                "robotPathMode": "IDLE",
                "path": [],
                "voltage": 0.0,
                "current": 0.0,
                "temperature": 0.0,
                "ip": "127.0.0.1"
            }
        ]
    }
  }
}
```


6.3.7 格子信息

6.3.7.1 所需字段

格子信息查询所需字段			
字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 LATTICE
latticeCode	String	任意选择填写，不填写查询全部	填写需要查询格子的编号
latticeCodes	list of String		填写需要查询格子的编号集合
hostLatticeCode	String		填写需要查询格子的外部编号
hostLatticeCodes	list of String		填写需要查询格子的外部编号集合

返回的格子信息属性说明如下表，**注意只有 latticeCode 字段是必填**，其余字段可能为空。

字段	类型	功能描述
latticeCode	String	格子编号
rackCode	String	固定货架编号
logicId	int	逻辑区域 ID
latticeHeight	int	格子高度
latticeLayer	int	格子层数
latticeStatus	String	格子状态 VACANT //空位 ALLOCATED//已分配 OCCUPIED//被占用
location	DPoint	位置
locationIndex	IPoint	位置索引
locationCellCode	String	位置对应的单元格的编码
locationHostCode	String	位置对应的单元格的外部编码

6.3.7.2 消息示例

请求消息示例：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "QueryInstructionRequestMsg",
   "request": {
      "header": {
         "requestId": "208dad33-6d0f-4ae9-b8dc-b5eac98b5d05",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "instruction": "LATTICE",
         "latticeCode": "L0000001"
      }
   }
}
```

响应消息示例：

```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "QueryInstructionResponseMsg",
    "response": {
        "header": {
            "responseId": "208dad33-6d0f-4ae9-b8dc-b5eac98b5d05",
            "code": 0,
            "msg": "Success"
        },
        "body": {
            "lattices": [
                {
                    "latticeCode": "L0000001",
                    "rackCode": "R0000001",
                    "logicId": 1,
                    "latticeHeight": 215,
                    "latticeLayer": 1,
                    "latticeStatus": "OCCUPIED",
                    "location": {
                        "z": 1,
                        "x": 3.675,
                        "y": 9.37
                    },
                    "locationIndex": {
                        "z": 1,
                        "x": 3,
                        "y": 9
                    }
                }
            ]
        }
    }
}
```


6.3.8 货箱信息

6.3.8.1 所需字段

字段	类型	是否必要	功能描述
instruction	String	必要	值必须是 BOX
boxCode	String	任意选择填写，不填写查询全部	填写需要查询货箱的编号
boxCodes	list of String		填写需要查询货箱的编号集合
hostBoxCode	String		填写需要查询货箱的外部编号

字段	类型	是否必要	功能描述
hostBoxCodes	list of String		填写需要查询货箱的外部编号集合

返回的箱子信息属性说明如下表，**注意只有 boxCode 字段是必填**，其余字段可能为空。

字段	类型	描述
boxCode	String	货箱编号
currentLatticeCode	String	货箱当前所在格子编号
placeLatticeCode	String	货箱老家格子编号
currentRobotId	int	货箱当前所在机器人编号
boxStatus	String	货箱当前状态 VACANT//货箱被放在格子上且当前没有任务 ALLOCATED//已分配给某个机器人 LOADED//被机器人装载 ENTERED_LIFT_ENTRY//进入提升机入口 ARRIVED_LIFT_EXIT//到达提升机出口
location	DPoint	位置
locationIndex	IPoint	位置索引
locationCellCode	String	位置对应的单元格的编码
locationHostCode	String	位置对应的单元格的外部编码

6.3.8.2 消息示例

请求消息示例：

```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "QueryInstructionRequestMsg",
   "request": {
      "header": {
         "requestId": "dd080d4e-f6d0-4174-b8d5-326abdf60b9a",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
```

```
            "instruction": "BOX",
            "boxCode": "B0000001"
        }
    }
}
```


响应消息示例：


```
{
    "id": "geekCode_geekWarehouseCode_001",
    "msgType": "QueryInstructionResponseMsg",
    "response": {
        "header": {
            "responseId": "dd080d4e-f6d0-4174-b8d5-326abdf60b9a",
            "code": 0,
            "msg": "Success"
        },
        "body": {
            "boxes": [
                {
                    "boxCode": "B0000001",
                    "currentLatticeCode": "L0000001",
                    "placeLatticeCode": "L0000001",
                    "boxStatus": "VACANT",
                    "location": {
                        "z": 1,
                        "x": 3.675,
                        "y": 9.37
                    },
                    "locationIndex": {
                        "z": 1,
                        "x": 3,
                        "y": 9
                    }
                }
            ]
        }
    }
}
```


6.3.9 距离查询

该接口用于查询地图中任意两点间的系统规划路径距离。计算出来的路径长度与系统当前启用的路径规划算法有关，不是简单直线距离，是实际可通行的规划路径长度。

6.3.9.1 查询请求

距离查询的请求字段如下所示：

字段	类型	必填	功能描述	示例
instruction	String	必填	必须为 DISTANCE	DISTANCE
queryType	String	非必填	simple：简要模式（默认），简要模式将会返回简要信息 detail：详细模式，详细模式将会返回详细信息	simple
load	int	非必填	计算的路线为空载或负载路线 0 空载，为空默认为此值，1 负载	0
起点控制字段	任意选择一个，若多选，则对应位置表述必须相同。若为货架、机器人等移动要素，则计算为其当前节点或下一个即将到达的节点。			
startCellCode	String	多选一	节点编码	12121
startIndex	IPoint		节点索引坐标，z 不填写默认为 1	{"x":45,"y":42} {"z":1,"x":45,"y":42}
startHostCode	String		节点外部编码	122
startShelfCode	String		货架号，将以货架当前位置进行计算	555
startRobotId	int		机器人号，将以机器人当前位置进行计算	445
startStationId	int		工作站号，将以工作站当前位置进行计算，标准工位为以工作点进行计算	4
终点控制字段	任意选择一个，若多选，则对应点必须相同。若为货架、机器人等移动要素，则计算为其当前节点或下一个即将到达的节点（防止不在节点位置上）			
destCellCode	String	多选一	节点编码	12121
destIndex	IPoint		节点索引坐标，z 不填写默认为 1	{"x":45,"y":42} {"z":1,"x":45,"y":42}
destHostCode	String		节点外部编码	45
destShelfCode	String		货架号，将以货架当前位置进行计算	456
destRobotId	int		机器人号，将以机器人当前位置进行计算	778
destStationId	int		工作站号，将以工作站当前位置进行计算，标准工位为以工作点进行计算	4

示例报文如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "QueryInstructionRequestMsg",
   "request": {
      "header": {
         "requestId": "8b3c2fb8-928c-4f0a-af15-2740913e43fa",
         "clientCode": "geekCode",
         "warehouseCode": "geekWarehouseCode",
         "userId": "admin",
         "userKey": "123456",
         "language": "en_us",
         "version": "3.3.0"
      },
      "body": {
         "instruction": "DISTANCE",
          "startCellCode":"102950105",
          "destCellCode":"102950187"
      }
   }
}
```


6.3.9.2 响应消息

RMS 响应业务系统的信息如下：

字段	详细	类型	功能描述	示例
distance		float	距离，单位 m，精度小数点后三位。不可达时，返回 -1	45.545
cellNums	是	Integer	距离，节点个数。不可达时，返回-1	1

示例报文如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "QueryInstructionResponseMsg",
   "request": {
      "header": {
         "responseId":"411ccc219da94e7f8680dfdee39c43d5",
         "code": 0,
         "msg":"Success"
       },
      "body": {
         "distance":45.454,
         "cellNums":45
       }
    }
}
```

7 仓库事件通知回调

7.1 简介

当 RMS 发生一些关键事件时，会通过此消息通知主机系统。例如机器人的加入和移除等，主机系统可根据此信息更新自己系统内的相关信息。

7.2 概述

7.2.1 传输方向

RMS 主动发送事件消息通知主机系统。以 WarehouseCallbackMsg 消息类型作为区分。

7.2.2 事件信息

事件分类	事件类型	描述
系统相关/SYSTEM	TASK_STOP	任务暂停
	SYSTEM_STOP	系统急停
	FIRE_STOP	消防急停
	SYSTEM_RECOVER	系统恢复运行
	TASK_RECOVER	任务恢复
机器人相关/ROBOT	ROBOT_ADD	机器人加入系统
	ROBOT_REMOVED	机器人移除
	ROBOT_RECOVERED	机器人恢复
容器/CONTAINER	CONTAINER_LOCATION_UPDATED	容器位置更新

7.2.3 消息结构

消息结构与任务回调的消息结构相同。

需要注意的是，此处“msgType”的值必须为“WarehouseCallbackMsg”。

区分不同消息指令的字段参数为消息体中的“callbackCategory”，具体的仓库事件回调消息详见后文。**所有的仓库事件回调消息的 Body 体，只有事件分类（callbackCategory）及事件类型（callbackType）是必填项。**

消息示例如下：


```
{
 "id": "*",
 "msgType": "WarehouseCallbackMsg",
 "request": {
  "header": {
   "warehouseCode": "geekWarehouseCode",
   "requestId": "200eb05fc36943bfabb677cf72667b62",
   "version": "3.3.0"
  },
  "body": {
   "callbackCategory": "xxx",
   "callbackType": "xxx",
   "field1": "xxx"
  }
 }
}
```


该类型消息主机系统仅需进行响应即可，主机系统应该给与 RMS 的响应如下：


```
{
   "id": "geekCode_geekWarehouseCode_001",
   "msgType": "WarehouseCallbackResponseMsg",
   "response": {
      "header": {
         "responseId": "200eb05fc36943bfabb677cf72667b62"
      }
   }
}
```


7.3 通知回调类型

7.3.1 系统

7.3.1.1 任务暂停

当系统因各种原因触发任务暂停时会产生此回调。示例消息如下：


```
{
   "id": "*",
   "msgType": "WarehouseCallbackMsg",
   "request": {
      "header": {
         "warehouseCode": "geekWarehouseCode",
         "requestId": "200eb05fc36943bfabb677cf72667b62",
         "version": "3.3.0"
      },
      "body": {
```

```
        "callbackCategory": "SYSTEM",
        "callbackType": "TASK_STOP"
      }
    }
}
```

7.3.1.2 系统急停

当系统因各种原因触发系统急停时会产生此回调。

当全局进行急停时会同时急停所有区域，不再回调区域急停的信息。

当区域进行急停时会回调区域急停的信息。

字段	类型	描述	示例
callbackCategory	String	值必须是 SYSTEM	SYSTEM
callbackType	String	值为 SYSTEM_STOP	SYSTEM_STOP
source	String	指令来源： BROWSE：浏览器前端 INTERFACE：接口 EQUIPMENT：设备，如急停控制器等 DMP：DMP 系统	BROWSE
areaIds	list of int	当区域进行急停时会附加该信息。	[1,2,3,4]

示例消息如下：

```
{
  "id": "*",
  "msgType": "WarehouseCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWarehouseCode",
      "requestId": "12a5b7e18bb340289fe080a08db04482",
      "version": "3.3.0"
    },
    "body": {
      "callbackCategory": "SYSTEM",
      "source": "BROWSE",
      "callbackType": "SYSTEM_STOP"
    }
  }
}
```

7.3.1.3 消防急停

当系统因各种原因触发消防急停时会产生此回调。

当全局进行急停时会同时急停所有区域，不再回调区域急停的信息。

当区域进行急停时会回调区域急停的信息。

字段	类型	描述	示例
callbackCategory	String	值必须是 SYSTEM	SYSTEM
callbackType	String	值为 FIRE_STOP	FIRE_STOP
source	String	指令来源： BROWSE：浏览器前端 INTERFACE：接口 EQUIPMENT：设备，如急停控制器等 DMP：DMP 系统	BROWSE
areaIds	list of int	当区域进行急停时会附加该信息	[1,2,3,4]

示例消息如下：

```
{
  "id": "*",
  "msgType": "WarehouseCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "geekWarehouseCode",
      "requestId": "72533ccba57e4612880e92a6be60705d",
      "version": "3.3.0"
    },
    "body": {
      "callbackCategory": "SYSTEM",
      "callbackType": "FIRE_STOP",
      "source": "BROWSE"
    }
  }
}
```

7.3.1.4 系统运行

当系统因各种原因从系统急停恢复系统运行时会产生此回调。

字段	类型	描述	示例
callbackCategory	String	值必须是 SYSTEM	SYSTEM
callbackType	String	值为 SYSTEM_RECOVER	SYSTEM_RECOVER
source	String	指令来源： BROWSE：浏览器前端 INTERFACE：接口 EQUIPMENT：设备，如急停控制器等	BROWSE

字段	类型	描述	示例
		DMP：DMP 系统	
areaIds	list of int	当区域进行恢复时会附加该信息	[1,2,3,4]

示例消息如下：


```
{
 "id": "*",
 "msgType": "WarehouseCallbackMsg",
 "request": {
  "header": {
   "warehouseCode": "geekWarehouseCode",
   "requestId": "ed731009e6dd4b159119f493be28e6e9",
   "version": "3.3.0"
  },
  "body": {
   "callbackCategory": "SYSTEM",
   "callbackType": "SYSTEM_RECOVER",
   "source": "BROWSE"
  }
 }
}
```


7.3.1.5 任务运行

当系统因各种原因从任务暂停恢复任务运行时会产生此回调。


```
{
 "id": "*",
 "msgType": "WarehouseCallbackMsg",
 "request": {
  "header": {
   "warehouseCode": "geekWarehouseCode",
   "requestId": "c467b423403b49478dc7f33232722af1",
   "version": "3.3.0"
  },
  "body": {
   "callbackCategory": "SYSTEM",
   "callbackType": "TASK_RECOVER"
  }
 }
}
```

7.3.2 机器人

7.3.2.1 机器人注册

当机器人首次联入系统在系统中进行注册时产生此回调。
回调内容字段说明如下：

字段	类型	描述	示例
callbackCategory	String	值必须是 ROBOT	ROBOT
callbackType	String	值为 ROBOT_ADD	ROBOT_ADD
robotId	Int	机器人编号	1280563
location	DPoint	机器人位置信息	{"z": 1, "x": 7.875, "y": 8.54}
locationIndex	IPoint	机器人位置信息	{ "z": 1, "x": 7, "y": 8}
locationCellCode	String	机器人位置信息	00750085
robotAngle	Double	机器人角度	180.0
robotDirection	Int	机器人在地图上的朝向，0，1，2，3分别表示东南西北	0

消息示例如下：


```
{
 "id": "*",
 "msgType": "WarehouseCallbackMsg",
 "request": {
  "header": {
   "warehouseCode": "geekWarehouseCode",
   "requestId": "b538480241674247a8a14118b4b019bc",
   "version": "3.3.0"
  },
  "body": {
   "callbackCategory": "ROBOT",
   "callbackType": "ROBOT_ADD",
   "robotId": 1000,
   "location": {
    "z": 1,
    "x": 7.875,
    "y": 8.54
   },
   "locationIndex": {
    "z": 1,
    "x": 7,
```

```
   "y": 8
  },
  "locationCellCode": "00750085",
  "robotAngle": 180.0,
  "robotDirection": 2
 }
 }
}
```


7.3.2.2 机器人移除

当机器人被从系统中移除时会产生此回调。机器人移除后并不删除注册信息。

回调内容字段说明如下：

字段	类型	描述	示例
callbackCategory	String	值必须是 ROBOT	ROBOT
callbackType	String	值为 ROBOT_REMOVED	ROBOT_REMOVED
robotId	Int	机器人编号	1280563
location	DPoint	机器人位置信息	{"z": 1, "x": 7.875, "y": 8.54}
locationIndex	IPoint	机器人位置信息	{ "z": 1, "x": 7, "y": 8}
locationCellCode	String	机器人位置信息	00750085
robotAngle	Double	机器人角度	180.0
robotDirection	Int	机器人在地图上的朝向，0，1，2，3 分别表示东南西北	0

消息示例如下：


```
{
 "id": "*",
 "msgType": "WarehouseCallbackMsg",
 "request": {
  "header": {
   "warehouseCode": "geekWarehouseCode",
   "requestId": "38c3c9c73bc3486cb8047e3fcdb58968",
   "version": "3.3.0"
  },
  "body": {
   "callbackCategory": "ROBOT",
   "callbackType": "ROBOT_REMOVED",
   "robotId": 1000,
   "location": {
    "z": 1,
```

```
  "x": 6.825,
  "y": 8.54
 },
 "locationIndex": {
  "z": 1,
  "x": 6,
  "y": 8
 },
 "locationCellCode": "00650085",
 "robotAngle": 0.0,
 "robotDirection": 0
}
}
}
```


7.3.2.3 机器人恢复

当机器人在移除后又被加入到系统中时会产生此回调。常见于异常处理、维修等情况。

回调内容字段说明如下：

字段	类型	描述	示例
callbackCategory	String	值必须是 ROBOT	ROBOT
callbackType	String	值为 ROBOT_RECOVERED	ROBOT_RECOVERED
robotId	Int	机器人编号	1280563
location	DPoint	机器人位置信息	{"z": 1, "x": 7.875, "y": 8.54}
locationIndex	IPoint	机器人位置信息	{ "z": 1, "x": 7, "y": 8}
locationCellCode	String	机器人位置信息	00750085
robotAngle	Double	机器人角度	180.0
robotDirection	Int	机器人在地图上的朝向，0，1，2，3分别表示东南西北	0

消息示例如下：


```
{
 "id": "*",
 "msgType": "WarehouseCallbackMsg",
 "request": {
  "header": {
   "warehouseCode": "geekWarehouseCode",
   "requestId": "d6c6d5165f164858930bff2288c1d861",
   "version": "3.3.0"
  },
```

```
 "body": {
  "callbackCategory": "ROBOT",
  "callbackType": "ROBOT_RECOVERED",
  "robotId": 1000
 }
}
}
```


7.3.2.4 机器人事件

该部分用于透传机器人本体触发的一些事件，如满格检测、RFID 读取器、重量检测器等。

回调消息字段说明：

字段	类型	描述	示例
callbackCategory	String	回调事件分类 ROBOT	ROBOT
callbackType	String	回调事件类型 ROBOT_EVENT_FRAME	ROBOT_EVENT_FRAME
机器人相关信息	在此处描述机器人的相关信息		
robotId	int	机器人 ID	50004
product	String	机器人型号	M100L
pathMode	String	机器人工作模式	GO_DELIVERING
taskId	int	机器人任务 ID，若无任务则为空	45411
angle	double	机器人角度（-180~180）	0
index	IPoint3D	机器人位置，索引坐标	{"z": 1,"x": 16,"y": 6}
location	DPoint3D	机器人位置，绝对坐标	{"z": 1,"x": 16.491,"y": 8.496}
cellCode	String	机器人位置，单元格 code/节点 id	00150025
hostCellCode	String	机器人位置，单元格 code/节点 id,外部编码	geekplus123
frame	RobotEventFrame	该部分为一个消息体，描述机器人传输上的事件信息，由主机系统与机器人确定该参数对照	

RobotEventFrame 字段说明如下：

字段	类型	描述	示例
warehouseId	int	仓库 ID	geekplus01
createTime	long	事件发生时间	
eventVersion	short	事件格式版本	
eventGroup	short	事件分组	
eventType	short	事件类型	
eventSource	byte		
robotId	int	机器人 ID	
locationX	int	机器人 X 坐标	
locationY	int	机器人 Y 坐标	
orientation	short	机器人方向角	
lifterAngle	short	提升机构角度	
lifterHeight	short	机构高度位置	
currentTaskMode	short	当前任务模式	
currentTaskStage	short	当前任务阶段	
motionMode	short	运动模式	
batteryLevel	short	机器人电量	
ledState	short	指示灯状态码	
parameter1	int	参数 1，下同	
parameter2	int		
parameter3	int		
parameter4	int		
parameter5	int		
parameter6	int		
parameter7	int		
parameter8	int		

字段	类型	描述	示例
parameter9	int		
parameter10	int		
parameter11	int		
parameter12	int		
parameter13	int		
parameter14	int		
parameter15	int		
parameter16	int		
parameter17	int		
parameter18	int		
parameter19	int		
parameter20	int		

7.3.3 容器

在 RMS 中货架、箱子都是属于容器的一种。

7.3.3.1 容器位置更新

当系统内的容器位置被更新（**目前只支持货架位置更新**）时会产生此回调。注意容器位置更新并不是指容器在机器人搬运过程中的位置变化，而是基于 API 容器位置更新接口或是 RMS 前端做的位置更新操作所对应的事件。

回调内容字段说明如下：

字段	类型	描述	示例
callbackCategory	String	值必须是 CONTAINER	CONTAINER
callbackType	String	值为 CONTAINER_LOCATION_UPDATED	CONTAINER_LOCATION_UPDATED
容器位置更新信息	在此处描述容器的位置更新信息。仅当容器被更新时才会产生此回调，有信息出现变化时才会回调对应信息。		
containerModel	String	容器模型。	SHELF

字段	类型	描述	示例
		普通货架 SHELF 货箱架 RACK 货箱 BOX	
containerCode	String	容器编码，容器是货架则是货架编号，货箱则是货箱编号	A000009
fromIndex	IPoint	容器来源位置，索引坐标	{"z": 1, "x": 7, "y": 8}
fromLocation	DPoint	容器来源位置，绝对坐标	{"z": 1,"x": 16.491,"y": 8.496}
fromCellCode	String	容器来源位置，单元格 code/节点 id	00750085
fromHostCellCode	String	容器来源位置，单元格 code/节点 id，外部编码	D-0085
fromAngle	Float	容器来源角度，(-180,180]	180.0
fromGridType	String	来源容器格类型	预留
fromGridCode	String	来源容器格编码	预留
toIndex	IPoint	容器目标位置，索引坐标	{"z": 1, "x": 9, "y": 1}
toLocation	DPoint	容器目标位置，绝对坐标	{"z": 1,"x": 24.41,"y": 10.09}
toCellCode	String	容器目标位置，单元格 code/节点 id	00950095
toHostCellCode	String	容器目标位置，单元格 code/节点 id, 外部编码	D-0086
toAngle	Float	容器目标角度，(-180,180]	90
createTime	String	事件发生的时间，格式为 YYYY-MM-DD HH24:SS:MS	2020-02-21 12:54:21:23
toGridType	String	目标容器格类型	预留
toGridCode	String	目标容器格编码	预留

消息示例如下：


```
{
  "id": "*",
  "msgType": "WarehouseCallbackMsg",
  "request": {
    "header": {
      "warehouseCode": "GEEK",
      "requestId": "d6c6d5165f164858930bff2288c1d861",
      "version": "3.3.0"
```

```
        },
        "body": {
            "callbackCategory": "CONTAINER",
            "callbackType": "CONTAINER_LOCATION_UPDATED",
            "containerModel": "SHELF",
            "containerCode": "A0002",
            "fromIndex":{"z": 1,"x": 16,"y": 6},
            "fromLocation":{"z": 1,"x": 16.491,"y": 8.496},
            "fromCellCode":"05454235",
            "toIndex":{"z": 1,"x": 17,"y": 9},
            "toLocation":{"z": 1,"x": 18.491,"y": 19.496},
            "toCellCode":"05450425",
            "createTime":"2020-02-21 12:54:21:23"
        }
    }
}
```


7.3.4 充电站

暂无。

7.3.5 工作站

暂无。

8 任务转移逻辑

当 RMS 中正在执行任务的机器人被移除，针对不同类型的任务，RMS 的处理逻辑也是不一样的。具体说明见下表。

任务类型	机器人移除后处理逻辑
GO_SOMEWHERE_TO_STAY	当机器人执行任务过程中被移除，此任务将直接被取消，并且没有回调消息发送
GO_WORK	当机器人执行任务过程中被移除，任务会被取消
DELIVER_SHELF、DELIVER_SHELF_TO_STATION、DELIVER_NEW_SHELF	任务未被取消，且机器人未取到货架，RMS 会将该任务分配给其他机器人
	任务未被取消，且机器人在负载货架状态，需要更新一次该负载货架位置，之后 RMS 会将该任务重新分配给其他的机器人
	任务被取消，且取消时参数 cancelAction 值为 0，机器人为空载状态，则取消该任务；
	任务被取消，且取消时参数 cancelAction 值为 0，机器人为负载状态，负载货架无老家位置，则取消该任务；
	任务被取消，且取消时参数 cancelAction 值为 0，机器人为负载状态，负载货架有老家位置，则更新该负载货架位置后任务分配给其他的机器人
	任务被取消，且取消时参数 cancelAction 值为 2，则取消该任务
	任务被取消，且取消时参数 cancelAction 值为 3，机器人为空载状态，则取消该任务
	任务被取消，且取消时参数 cancelAction 值为 3，机器人为负载状态，则再更新该负载货架位置后任务分配给其他的机器人
CARRY_PACKAGE	当正在执行任务的机器人从 RMS 系统中被移除时，此机器人身上的“CARRY_PACKAGE”只有在满足以下条件时可以被转移到其他机器人上，其余状态下的任务都会被取消：

任务类型	机器人移除后处理逻辑
	1、此“CARRY_PACKAGE”任务尚未开始执行 2、此“CARRY_PACKAGE”任务已开始执行，但任务中的货物总量为 0（即机器人身上没有货物）
DELIVER_PALLET	如果被移除的机器人尚未取到托盘，则 RMS 会将此任务分配给其他机器人执行。 若机器人已取到托盘，任务会被取消。
DELIVER_BOX	如果被移除的机器人尚未取到货箱，则 RMS 会将此任务分配给其他机器人执行。 若机器人已取到货箱，任务会被取消，且货箱位置需要人为更新。
GO_WORK_ORDER_TO_PERSON	当机器人执行任务过程中被移除，任务会被取消
分拣相关任务	当机器人执行任务过程中被移除，任务会被取消

附录

附 1. API 响应代码

CODE	中文	英文	备注
0	成功	Success	
100	异常或其他未知错误	Exception or unknown error	
101	不支持该操作	Operation not supported	
110	RMS 正在初始化...	RMS is Initializing...	
121	操作失败	Operation failed	
200	无效的参数	Illegal argument	
201	资源已被占用	Illegal argument resource is in use	
202	未找到可用的机器人	Illegal argument no available robot	
203	机器人已被占用	Illegal argument robot is in use	
204	未找到可用的货架	Illegal argument no available shelf	
205	货架已被占用	Illegal argument shelf is in use	
206	未找到可用的单元格	Illegal argument no available cell	
207	单元格已被占用	Illegal argument cell is in use	

CODE	中文	英文	备注
208	未找到可用的工位	Illegal argument no available station	
209	工位已被占用	Illegal argument station is in use	
210	未找到可用的充电站	Illegal argument no available charging station	
211	充电站已被占用	Illegal argument charging station is in use	
212	参数不能为 null	Illegal argument parameter should not be null	
213	自定义路径错误	Illegal argument customized path error	
214	目标单元格已存在货架	Illegal argument cell is occupied by shelf	
215	资源申请超时	Illegal argument resource request timeout	
216	禁止跨层调用	Illegal argument invocation across layers is forbidden	
217	终点似乎不可达	Illegal argument endpoint is unreachable	
218	资源已经被禁用	Illegal argument resource is disabled	
219	无效的货架面	Illegal argument invalid need sides	
220	参数指代对象不匹配	argument data not match	
221	无效的暂停点	Illegal argument invalid wait cell code	
240	无效的请求报文	Illegal argument invalid request	
241	无效的 Header	Illegal argument invalid request header	
242	无效的 Body	Illegal argument invalid request body	

CODE	中文	英文	备注
243	无效的版本号	Illegal argument invalid request version	
244	无效的 request id	Illegal request id	
250	无效的凭证	Illegal argument invalid certificate	
251	无效的用户名	Illegal argument invalid username	
252	无效的密码	Illegal argument invalid password	
253	用户不存在	Illegal argument use not exists	
300	无效的指令. instruction:{}	Illegal argument invalid instruction. instruction:{}	
301	指令已经被废弃. instruction:{}	Illegal argument instruction has been deprecated. instruction:{}	
302	无效的任务类型. taskType:{}	Illegal task type. taskType:{}	
303	任务类型已被废弃. taskType:{}	The task type has been deprecated. taskType:{}	
400	记录不存在. recordId:{}	Record is not exists. recordId:{}	
401	记录已存在. recordId:{}	Record is already existed. recordId:{}	
402	任务不存在. taskId:{}	The task is not exists. taskId:{}	
403	任务已存在. taskId:{}	The task is already existed. taskId:{}	
1000	请求消息不能为空	Request message must not be null	
1001	非法的请求消息，消息解析失败	Illegal request message, parsing msg failed	
1002	MsgId 不能为空	MsgId must not be null	

CODE	中文	英文	备注
1003	MsgType 不能为空	MsgType must not be null	
1004	MsgType 不支持，msgType:{0}	MsgType cannot supported, msgType:{0}	
1005	Request 不能为空	Request must not be null	
1006	Header 不能为空	Header must not be null	
1007	Body 不能为空	Body must not be null	
1008	sign 不能为空	sign must not be null	
1009	无效的 sign	Illegal argument invalid request sign	
1110	ChannelId 不能为空	ChannelId must not be null	
1120	RequestId 不能为空	RequestId must not be null	
1130	ClientCode 不能为空	ClientCode must not be null	
1131	ClientCode 不存在，cilentCode:{0}	ClientCode not exists, cilentCode:{0}	
1140	WarehouseCode 不能为空	WarehouseCode must not be null	
1141	WarehouseCode 不存在，warehouseCode:{0}	WarehouseCode not exists, warehouseCode:{0}	
1150	UserId 不能为空	UserId must not be null	
1151	User 不存在，userId:{0}	UserId not exists, userId:{0}	
1152	User 已禁用，userId:{0}	UserId is deprecated, userId:{0}	
1153	Userkey 不能为空	Userkey must not be null	

CODE	中文	英文	备注
1154	用户认证失败	User authentication failed	
1160	Version 不能为空	Version must not be null	
1161	不支持的版本号，version:{0}	Version cannot supported, version:{0}	
1162	已废弃的版本号，version:{0}	Version is deprecated, version:{0}	
1170	语言不能为空	Language must not be null	
1171	不支持的语言，language:{0}	Language cannot supported, language:{0}	
1172	语言已废弃，language:{0}	Language is deprecated, language:{0}	
1173	语言不合法，language:{0}	Language is illegal, language:{0}	
1181	透传字段长度超出限制，length:{0}	Ext Overlength, length:{0}	
2000	任务类型不能为空	Task type must not be null	
2001	非法的任务类型，taskType:{0}	Illegal task type,taskType:{0}	
2002	任务类型已废弃，taskType:{0}	Task type is deprecated,taskType:{0}	
2003	不支持的任务类型，taskType:{0}	Task type cannot supported, taskType:{0}	
2010	任务指令不能为空	Task instruction must not be null	
2011	非法的指令类型，instruction:{0}	Illegal instruction type, instruction:{0}	
2012	指令类型已废弃，instruction:{0}	Instruction is deprecated, instruction:{0}	
2013	不支持的指令类型，instruction:{0}	Task instruction cannot supported, instruction:{0}	

CODE	中文	英文	备注
2014	货架还未就绪	Shelf not ready	
2015	当前任务状态不支持该指令，instruction:{0}	The instruction is not supported in the current task state,instruction:{0}	
2020	机器人任务编号不能为空	Robot task number must not be null	
2021	机器人任务不存在，taskId:{0}	Robot task not exists, taskId:{0}	
2022	多余的任务 ID，taskId:{0}	Task id is redundant, taskId:{0}	
2030	机器人编号不能为空	Robot id must not be null	
2031	机器人不存在，robotId:{0}	Robot not exists, robotId:{0}	
2032	传入的机器人 Id 不合法，robotId:{0}	Robot Id of the request is illegal, robotId:{0}	
2033	任务更新不允许传入 robotIds	robotIds is not allowed in TaskUpdate	
2040	货架编码不能为空	Shelf code must not be null	
2041	内外部货架编码不匹配，shelfCode:{0}，hostShelfCode:{1}	Shelf code does not match, shelfCode:{0}, hostShelfCode:{1}	
2042	无效的货架编码，shelfCode:{0}，hostShelfCode:{1}	Shelf code is invalid, shelfCode:{0}, hostShelfCode:{1}	
2043	货架编码不存在，shelfCode:{0}，hostShelfCode:{1}	Shelf code not exists, shelfCode:{0}, hostShelfCode:{1}	
2044	货架还有任务未完成	Shelf still has task to do	
2045	多余的货架编码，shelfCode:{0}	Shelf code is redundant,shelfCode:{0}	

CODE	中文	英文	备注
2046	当前任务的机器人与货架不匹配	The current task of the robot does not match the shelf	
2050	非法的货架面，neededSide:{0}	Illegal shelf side, neededSide:{0}	
2051	货架面不能为空	Shelf side must not be empty	
2052	货架旋转不能同时指定货架面和角度	Shelf rotation cannot specify shelf sides and angle at the same time	
2053	货架角度超出范围，shelfAngle:{0}	The shelf angle is out of range, shelfAngle: {0}	
2054	货架旋转方向非法	Shelf rotation direction illegal	
2055	shelfTurnAngle 不支持任务或指令：{0}	shelfTurnAngle nonsupport task or instruction:{0}	
2056	当前任务的工作站和单元格不能同时为空	The workstation and cell of the current task cannot be empty at the same time	
2057	货架无 Placement，不能执行此任务，shelfCode:{0}，hostShelfCode:{1}	Shelf do not have placement, can not execute thistask，shelfCode:{0}，hostShelfCode:{1}	
2060	工作站编号不能为空	Station id must not be null	
2061	内外部工作站不匹配，stationId:{0}，hostStationCode:{1}	Statiion code does not match, stationId:{0} , hostStationCode:{1}	
2062	无效的工作站编号，hostStationCode:{0}	Station id is invalid, hostStationCode:{0}	
2063	工作站不存在，stationId:{0}，hostStationCode:{1}	Station not exists, stationId:{0} , hostStationCode:{1}	
2064	多余的工作站编号，stationId:{0}	StationId is redundant, stationId:{0}	
2065	任务类型与工作站类型不匹配，stationId:{0}，	Station type can not pair tasktype，stationId:{0}，	

CODE	中文	英文	备注
	hostStationCode:{1}	hostStationCode:{1}	
2070	充电站编号不能为空	Charger id must not be null	
2080	终点区域编号不能为空	Destination area id must not be null	
2081	终点区域不存在，areaId:{0}	Destination area not exists, areaId:{0}	
2090	终点不能为空	Destination must not be null	
2091	内外部终点坐标不匹配，dest:{0}，destHostCode:{1}，destCellCode:{2}	Destination not pair, dest:{0}, destHostCode:{1}, destCellCode:{2}	
2092	终点坐标超出地图边界，dest:{0}，destHostCode:{1}，destCellCode:{2}，destAreaId:{3}	Destination out of map, dest:{0}, destHostCode:{1}, destCellCode:{2}, destAreaId:{3}	
2093	无效的终点二维码，destHostCode:{0}，destCellCode:{1}	Invalid destination cell code, destHostCode:{0}, destCellCode:{1}	
2094	无效的终点单元格，cell:{0}	Invalid destination cell, cell:{0}	
2095	移动的目标位置不合法	The destination cell is illegal	
2100	终点二维码编码不能为空	Destination cell code must not be null	
2101	终点二维码编码不存在，cellCode:{0}	Destination cell code not exists, cellCode:{0}	
2110	暂停点二维码编码不能为空	Wait cell code must not be null	
2111	暂停点二维码不存在，waitCell:{0}	Wait cell code not exists, waitCell:{0}	
2112	清除等待点失败	Clear wait cel failed	

CODE	中文	英文	备注
2120	货架分数超出范围，shelfScore:{0}	Shelf score is out of range, shelfScore:{0}	
2130	任务优先级超出范围，priority:{0}	Task priority is out of range, priority:{0}	
2131	业务序列号超出范围，businessSequence:{0}	Task business sequence is out of range, businessSequence:{0}	
2132	不结束任务的参数错误	The value of isContinue illegal	
2133	GO_FETCH 指令不可结束任务	GO_FETCH Instruction Unterminatable Task	
2140	无效的取消动作，cancelAction:{0}	illegal cancel action, cancelAction:{0}	
2141	取消任务失败	Cancel failed	
2142	不支持当前取消指令，因为任务不涉及货架，或机器人尚未取到货架	The cancelAction is not supported, because the task is not a shelf-task or the robot did not fetch the shelf	
2143	当前任务的货架无库存位置	The shelf of current task has no placement	
2144	重复取消	Cancel repeatedly	
2145	任务不支持取消	Task do not support cancel	
2146	任务已完成或已取消	Task is completed or canceled	
2147	取消操作不被允许	Cancel operation is not allowed	
2150	电梯 Id 不能为空	ElevatorId cannot be null	
2151	电梯不存在，ElevatorId:{0}	Elevator does not exist, ElevatorId:{0}	
2152	相关楼层不被电梯连通，{0}:{1}，{2}:{3}	The floors are not connected by elevator,{0}:{1},{2}:{3}	

CODE	中文	英文	备注
2160	机器人未到达电梯入口	Robot not arrived elevator entry	
2161	机器人不在电梯	Robot not in elevator	
2170	包裹数量不合法	The number of packages is not legal	
2171	目标工作站不是辊筒工作站，stationId:{0}, hostStationCode:{1}	The target station is not a roller station, stationId:{0},hostStationCode:{1}	
2180	货架的起始位置不能为空	The starting position of the shelf cannot be empty	
2181	无效的货架起始位置，locationIndex:{0}，locationCellCode:{1}，locationHosCode:{2}	Invalid shelf start location, locationIndex: {0}, locationCellCode: {1}, locationHosCode: {2}	
2182	托盘位编码不能为空	Pallet lattice code must not be null	
2183	无效的托盘位编码，palletLatticeCode:{0}，hostPalletLatticeCode:{1}	Pallet lattice code is invalid, palletLatticeCodee:{0}, hostPalletLatticeCode:{1}	
2184	托盘位不存在，palletLatticeCode:{0}，hostPalletLatticeCode:{1}，cellCode:{2}，layer:{3}	Pallet lattice not exists, palletLatticeCode:{0}, hostPalletLatticeCode:{1},cellCode:{2},layer:{3}	
2185	内外部托盘位编码不匹配，palletLatticeCode:{0}，hostPalletLatticeCode:{1}	Pallet lattice code does not match,palletLatticeCode:{0}, hostPalletLatticeCode:{1}	
2186	无效的托盘位高度，palletLatticeCode:{0}，hostPalletLatticeCode:{1}，height:{2}	Pallet lattice height is invalid, palletLatticeCode:{0}, hostPalletLatticeCode:{1}, height:{2}	
2187	起点不能为空	Start point must not be null	
2188	起点坐标超出地图边界，startDest:{0}，startDestHostCode:{1}，startDestCellCode:{2}	Start Point out ofmap，startDest:{0}，startDestHostCode:{1}，startDestCellCode:{2}	

CODE	中文	英文	备注
2190	皮带号为空	Belt pos is null	
2191	皮带号非法	Belt pos is illegal	
2200	任务未就绪	Task not ready	
2210	任务指定机器人分配策略非法	task specifies that the robot assignment strategy is illegal	
2211	找不到可以执行任务的机器人	there is no robot that can do this task	
2300	无效的箱子编号, boxCode:{0}, hostBoxCode:{1}	The box code is invalid, boxCode:{0}, hostBoxCode:{1}	
2301	箱子不存在, boxCode:{0}, hostBoxCode:{1}	The box is not exists, boxCode:{0}, hostBoxCode:{1}	
2302	内外部箱子不匹配, boxCode:{0}, hostBoxCode:{1}	The box code does not match, boxCode:{0}, hostBoxCode:{1}	
2303	箱子有任务未完成, boxCode:{0}	The box has tasks to be finish, boxCode:{0}	
2304	箱子编号已存在	The boxCode is already exists	
2305	箱子的目的地不能为空	The destination of the box cannot be null	
2306	箱子编号不能空	The boxCode cannot be empty	
2307	多余的货箱号, boxCode:{0}, hostBoxCode:{1}	The box code is redundant:{0}, hostBoxCode:{1}	
2310	格子编号不能为空	The lattice code must not be null	
2311	无效的格子编号, latticeCode:{0}, hostLatticeCode:{1}	The lattice code is invalid, latticeCode:{0}, hostLatticeCode:{1}	
2312	格子不存在, latticeCode:{0},hostLatticeCode:{1}	The lattice is not exists, latticeCode:{0}, hostLatticeCode:{1}	

CODE	中文	英文	备注
2313	内外部格子不匹配, latticeCode:{0}, hostLatticeCode:{1}	The lattice code does not match, latticeCode:{0}, hostLatticeCode:{1}	
2314	格子已被占用, lattice:{0}	The lattice has already been occupied, lattice:{0}	
2315	没有可用的格子	There is no available lattice	
2320	货箱的更新信息命令无效, boxOrder:{0}	The order of the box update information is illegal, boxOrder:{0}	
2321	不支持该更新命令, boxOrder:{0}	The order of the box update information is not supported, boxOrder:{0}	
2322	机器人上无货箱	There is no boxes on the robot	
2323	已有货箱在机器人上	There are already box on the robot	
3000	仓库指令不能为空	Warehouse instruction must not be null	
3001	非法的仓库指令，instruction:{0}	Illegal warehouse instruction, instruction:{0}	
3002	仓库指令已废弃，instruction:{0}	Warehouse instruction is deprecated, instruction:{0}	
3003	不支持的仓库指令，instruction:{0}	Warehouse instruction not supported, instruction:{0}	
3004	接口指令优先级低于当前急停优先级	The instruction priority is low then SYSTEM_STOP	
3010	机器人编号不能为空	Robot id must not be null	
3011	无效的机器人，robotId:{0}，hostRobotCode:{1}	Robot is invalid, robotId:{0}, hostRobotCode:{1}	
3012	内外部机器人编号不匹配，robotId:{0}，hostRobotCode:{1}	Robot id not pair, robotId:{0}, hostRobotCode:{1}	

CODE	中文	英文	备注
3013	机器人不存在，robotId:{0}，hostRobotCode:{1}	Robot not exists, robotId:{0}, hostRobotCode:{1}	
3014	该指令不支持机器人负载	This instruction does not support robot loads	
3020	货架编码不能为空	Shelf code must not be null	
3021	非法的货架编码，shelfCode:{0}	Illegal shelf code, shelfCode:{0}	
3022	内外部货架不匹配，{0}:{1}，{2}:{3}	Shelf code not pair, {0}:{1}, {2}:{3}	
3023	货架不存在，shelfCode:{0}	Shelf not exists, shelfCode:{0}	
3024	仓库货架入场中，shelfCode:{0}	Shelf entering, shelfCode:{0}	
3025	仓库货架已存在，shelfCode:{0}	Shelf already exists, shelfCode:{0}	
3030	货架入场点超出地图范围，{0}:{1}，{2}:{3}，{4}:{5}	Shelf entry point out of map, {0}:{1}，{2}:{3}，{4}:{5}	
3031	内外部货架入场点不匹配，{0}:{1}，{2}:{3}，{4}:{5}	Shelf entry not pair, {0}:{1}，{2}:{3}，{4}:{5}	
3032	此入场点已存在货架，{0}:{1}，{2}:{3}，{4}:{5}，占用货架:{6}	There is a shelf already exists at this entry, {0}:{1}, {2}:{3}, {4}:{5}, occupy shelf:{6}	
3033	入场点不能为空	Shelf entry must not be null	
3034	入场点单元格不合法，{0}:{1}，{2}:{3}，{4}:{5}	Shelf entry cell is illegal, {0}:{1}，{2}:{3}，{4}:{5}	
3035	货架的摆放位置超出地图范围，{0}:{1}，{2}:{3}，{4}:{5}	The placement of the shelf is out of the map, {0}: {1}, {2}: {3}, {4}: {5}	

CODE	中文	英文	备注
3036	货架的摆放位置已经被其他货架占用，{0}:{1}，{2}:{3}，{4}:{5}	The placement of the shelf has been occupied by other shelves, {0}: {1}, {2}: {3}, {4}: {5}	
3037	货架的摆放位置不能为空	The placement of the shelf cannot be empty	
3038	货架的摆放位置不匹配，{0}:{1}，{2}:{3}，{4}:{5}	The placement of the shelf does not match, {0}: {1}, {2}: {3}, {4}: {5}	
3039	货架摆放位置不合法，{0}:{1}，{2}:{3}，{4}:{5}	The placement of the shelf is illegal, {0}: {1}, {2}: {3}, {4}: {5}	
3040	仓库请求单元格编码不能为空	Request cell code must not be null	
3041	仓库请求无效的单元格编码，cellCode:{0}	Invalid cell code, cellCode:{0}	
3042	仓库请求内外部单元格不匹配，{0}:{1}，{2}:{3}	Cell Code not pair, {0}:{1}, {2}:{3}	
3043	仓库请求单元格不存在，cell:{0}	Cell code not exists, cell:{0}	
3050	仓库请求区域不存在，areaId:{0}	Request area not exists, areaId:{0}	
3051	此区域已满，areaId:{0}	This area is full, areaId:{0}	
3060	货架角度超出范围	shelf angle out of range	
3061	参数[shelfMoveCount]为空	The parameter [shelfMoveCount] is null	
3070	工作站 ID 不能为空	station ID cannot be empty	
3071	工作站不存在。stationId:{0}, hostStationCode:{1}	the station does not exist. stationId: {0}, hostStationCode: {1}	
3072	工作站非法。stationId:{0}, hostStationCode:{1}	the station is illegal. stationId: {0}, hostStationCode: {1}	

CODE	中文	英文	备注
3073	工作站不匹配。stationId:{0}, hostStationCode:{1}	the stations do not match. stationId: {0}, hostStationCode: {1}	
3080	当前系统急停控制器不可用，无法进行系统恢复！	System ScramController Not Available. Unable to Restore System!	
3081	系统必须全局运行中才能执行该指令	The system must be running globally to execute this instruction	
3082	系统当前在执行其他业务指令	The system is currently executing other business instructions	
3100	无效的箱子编号, boxCode:{0},hostBoxCode:{1}	The box code is invalid, boxCode:{0}, hostBoxCode:{1}	
3101	货箱已存在，boxCode:{0}	Box already exists, boxCode:{0}	
3102	无效的格子编号,latticeCode:{0}, hostLatticeCode:{1}	The lattice code is invalid, latticeCode:{0}, hostLatticeCode:{1}	
3103	格子不存在, latticeCode:{0},hostLatticeCode:{1}	The lattice is not exists, latticeCode:{0}, hostLatticeCode:{1}	
3104	货位格已经被占用，latticeCode:{0}	The lattice is already occupied, latticeCode:{0}	
3105	箱子有任务未完成, boxCode:{0}	The box has tasks to be finish, boxCode:{0}	
3106	箱子不存在, boxCode:{0}	The box is not exists, boxCode:{0}	
3107	货箱位置信息为空	The information of box position is null	
3108	箱子编号不能空	The boxCode cannot be empty	
3150	系统已经全局消防急停，不支持当前操作。	The system has already stopped in an fire emergency, not support current operation.	
3151	系统已经全局急停，不支持当前操作。	The system has already stopped in an emergency, not support current operation.	

CODE	中文	英文	备注
3152	区域：{0}已消防急停，不支持当前操作。	Area:{0} already stopped in an fire emergency, not support current operation.	
3153	区域：{0}已急停，不支持当前操作。	Area:{0} already stopped in an emergency, not support current operation.	
3200	巡检事务编码不能为空	Inspection transaction code cannot be empty	
3201	巡检区域不能为空，区域 1:{}, 区域 2:{}	The inspection area cannot be empty, area 1: {}, area 2: {}	
3202	巡检事务编码已存在	Inspection transaction code already exists	
4000	查询指令不能为空	Query instruction must not be null	
4001	非法的查询指令，instruction:{0}	Illegal query instruction, instruction:{0}	
4002	查询指令已过期，instruction:{0}	Query instruction is deprecated, instruction:{0}	
4003	不支持的查询指令，instruction:{0}	Unsupported query instruction, instruction:{0}	
4010	坐标不能为空。{0}	position cannot be empty.{0}	
4011	非法的坐标值。{0}	Illegal position value. {0}	
4012	坐标不匹配。{0}	the position value not pair.{0}	
4020	非法的参数值，{0}:{1}	Illegal parameter value, {0}:{1}	
4500	没有找到查询数据	Query data not found	
5000	参数类型不能为空。parameter:{0}	Parameter type cannot be empty. parameter:{0}	
5001	非法的参数类型。parameterType:{0}	Illegal parameter type. parameterType: {0}	

CODE	中文	英文	备注
5003	不支持的参数类型。parameterType:{0}	The type of parameter that is not supported. parameterType: {0}	
5010	参数名不能为空。parameter:{0}	Parameter name cannot be empty. parameter:{0}	
5011	无效的参数。parameter:{0}	Invalid parameter. parameter:{0}	
5012	已停用的参数。parameter:{0}	The parameter that has been deactivated. parameter:{0}	
5013	不支持的参数。parameter:{0}	Unsupported parameter. parameter:{0}	
5020	参数值不能为空。parameter:{0}	Parameter value cannot be empty. parameter:{0}	
5022	非法的参数值。parameter:{0}，value:{1}	Illegal parameter value. parameter:{0}, value:{1}	
5023	参数值超出范围。parameter:{0}，value:{1}	The parameter value is out of range. parameter:{0}, value:{1}	
5024	参数值不匹配。{0}，{1}	parameter value not matching. {0}, {1}	
5030	货架正在使用中	the shelf is in use.	
5041	货架位置不能为空	Shelf location must not be null	
5042	无效的货架位置。locationIndex:{0}, locationCellCode:{1}, locationHostCode:{2}	Invalid shelf location. locationIndex:{0}, locationCellCode:{1}, locationHostCode:{2}	
5043	非法的货架位置。locationIndex:{0}, locationCellCode:{1}, locationHostCode:{2}	Illegal shelf location. locationIndex:{0}, locationCellCode:{1}, locationHostCode:{2}	
5044	此坐标已经被其他货架占用。locationIndex:{0}, locationCellCode:{1}, locationHostCode:{2}, otherShelfCode:{3}	The location has been occupied by another shelf. locationIndex:{0}, locationCellCode:{1}, locationHostCode:{2}, anotherShelfCode:{3}	
5045	具有歧义的货架位置。locationIndex:{0},	Ambiguous shelf location. locationIndex:{0}, locationCellCode:{1},	

CODE	中文	英文	备注
	locationCellCode:{1}, locationHostCode:{2}	locationHostCode:{2}	
5050	货架摆放位置不能为空	Shelf placement must not be null	
5051	无效的摆放位置。placementIndex:{0}, placementCellCode:{1}, placementHostCellCode:{2}	Invalid placement. placementIndex:{0}, placementCellCode:{1}, placementHostCellCode:{2}	
5052	非法的摆放位置。placementIndex:{0}, placementCellCode:{1}, placementHostCellCode:{2}	Illegal placement. placementIndex:{0}, placementCellCode:{1}, placementHostCellCode:{2}	
5053	具有歧义的货架摆放位置。placementIndex:{0}, placementCellCode:{1}, placementHostCellCode:{2}	Ambiguous shelf placement. placementIndex:{0}, placementCellCode:{1}, placementHostCellCode:{2}	
5061	无效的货架角度。角度范围在[-180, 180]之间。	Invalid shelf angle. The valid range is [-180, 180].	
5062	无效的货架分数。分数范围在[1, 100]之间。	Invalid shelf score. The valid range is [1, 100].	
5070	工作站 id 不能为空	The workstation id cannot be empty	
5071	工作站不存在，id:{0}	The workstation does not exist, id: {0}	
5072	工作站状态不合法，status:{0}。可选值：[OPEN，CLOSE]	The workstation status is invalid, status: {0}. Optional values: [OPEN, CLOSE]	
5073	超出最大机器人排队上限，queueSize:{0}。取值范围：1-99	Exceeded the maximum robot queue limit, queueSize: {0}. Value range: 1-99	
5074	是否允许机器人在工作站停留的参数不合法，allowRobotToStay:{0}。可选值：[OPEN，CLOSE]	The parameter of whether to allow robot to stay at the workstation is illegal, allowRobotToStay: {0}. Optional values: [OPEN, CLOSE]	

CODE	中文	英文	备注
5075	在路径的倒数第几个节点开始排队的参数不合法，waitCellNum:{0}。取值范围：>=0	The parameter of waitCellNum is invalid, WaitCellNum: {0}. Value range:>= 0	
5076	皮带调整方向不合法，beltAdjustingDirection:{0}。可选值：[-1，1]	The belt adjustment direction is illegal, beltAdjustingDirection: {0}. Optional values: [-1, 1]	
5080	充电站 id 不能为空	The charging station id cannot be empty	
5081	充电站不存在，id:{0}	The charging station does not exist, id: {0}	
5082	充电站状态不合法，status:{0}。可选值：[OPEN，CLOSE]	The charging station status is illegal, status: {0}. Possible values: [OPEN, CLOSE]	
5083	机器人型号不合法	The robot product is illegal	
10100	机器人任务中，请作业结束后进行维修	Robot task, please repair after the end of the operation	
10102	机器人已返场	The robot has been recovered	
10103	机器人已返场，但未能找到休息点	The robot has recovered but failed to find a rest cell for it	
10104	维护区不存在	The maintenance area does not exist	
10105	已有机器人:{0}占用维修区，请等待	Robot :{0} occupied the maintenance area, please wait	
10106	加入机器人成功	Join robot success	
10107	机器人{0}不可用	Robot{0}not available	
10108	楼层已存在	Floors already exist	
10109	机器人还未准备好	The robot is not ready yet	

CODE	中文	英文	备注
10110	不能创建多个地图	Cannot create multiple maps	
10111	对象为空	Object is null	
10112	对象不合法，{0}：{1}	Object is illegal，{0}: {1}	
10113	对象不存在，{0}：{1}	Object not exists，{0}: {1}	
10114	对象已存在，{0}：{1}	Object already exists，{0}: {1}	