

# RCS接口对接说明

- 1. 文档说明
- 2. HTTP
  - 2.1 创建任务
  - 2.2 取消任务
  - 2.3 修改任务执行参数
  - 2.4 获取任务状态
  - 2.5 推送信息
  - 2.6 获取地图车辆状态
  - 2.7 获取车辆当前的任务
  - 2.8 暂停任务
  - 2.9 恢复任务
  - 2.10 批量创建任务
- 3 TCP
  - 3.1 调用说明
  - 3.2 报文结构
  - 3.3 请求示例
  - 3.4 响应示例
  - 3.5 响应结构
- 4 HTTP错误码映射表

版本变更记录

版本号	修订内容	修订日期
-----	------	------

V 1.0	首次制定	2023-02-05
V 1.1	增加TCP协议支持	2024-03-12
V 1.2	HTTP增加暂停/恢复任务接口	2024-08-20
V1.3	增加HTTP错误码映射表	2024-11-26
V1.4	WebSocket推送增补任务信息	2024-12-05
V1.5	任务下发增加是否回停车点参数	2025-05-05

# 1. 文档说明

本文档介绍RCS系统对外接口开放功能的调用说明，向用户开放调度任务操作的相关 API，API 支持 HTTP、TCP 两种协议方式进行数据交互。HTTP通信默认使用【50060】端口,TCP通信 默认使用【50061】

## 2. HTTP

### 2.1 创建任务

接口描述	外接系统将入库任务下发到RCS。
调用地址	<a href="http://IP:Port/Task/CreateTask">http://IP:Port/Task/CreateTask</a>

下发方	外接系统
下发报文示例：	<pre>{   "SysToken": "8d3bb2dc-2fc4-38cd-952a-44844b321945",   "ReceiveTaskID": "werwww4eeer",   "MapCode": "testMap",   "TaskCode": "task1",   "AgvGroupCode": "",   "AGVCode": "",   "IsGoParking": "",   "Variables": [     {       "Code": "StartPoint",       "Value": "PNT_30"     }   ] }</pre>
返回报文示例：	<pre>{   "Content": "任务唯一标识",   "Success": true,   "Code": "0" }</pre>
接收方	RCS
请求方式	基于RESTFUL规范的Post方法

#### 参数说明

字段名称	类型	是否必填	描述
------	----	------	----

SysToken	String	是	外接系统标识,需在RCS【外接系统配置】内维护
ReceiveTaskID	String	是	单号，外接系统中生成，唯一识别编号，任务完成后回调使用
MapCode	String	是	地图编码
TaskCode	String	是	任务模板编码
AgvGroupCode	String	否	车辆集群分组编码，标识调用指定集群内车辆执行任务，可设置为空
AGVCode	String	否	车辆编码，标识调用指定车辆执行任务，可设置为空
Priority	Int	否	任务优先级 1~10，默认为5
IsGoParking	bool?	否	任务完成后是否回停车点。  默认值：任务模版配置。  true：回停车点  false：不回停车点
Variables	String	是	任务参数数组，json格式字符串，格式参考示例
返回值：			

Success	Bool		是否成功
Content	String		【Success】为true时 【Content】为任务唯一标识码, 【Success】为false, 【Content】为失败原因。
Code	String		错误代码, "0"为成功

## 2.2 取消任务

接口描述	外接系统将停止任务下发到RCS。
接口地址	<a href="http://IP:Port/Task/StopAgvTask">http://IP:Port/Task/StopAgvTask</a>
下发方	外接系统
下发报文示例：	<pre>{   "ReceiveTaskID": "string",   "AgvCode": "string" }</pre>
返回报文示例：	<pre>{   "Content": "任务唯一标识",   "Success": true,   "Code": "0" }</pre>
接收方	RCS
请求方式	基于RESTFUL规范的Post方法

数据说明

字段名称	类型	是否必填	描述
ReceiveTaskID	String	否	单号，外接系统中生成，唯一识别编号
AgvCode	String	否	车辆编码
备注：ReceiveTaskID、AgvCode 只填一个即可，优先判定ReceiveTaskID，按任务单号取消任务，然后再按车辆编码。			
返回值：			
Success	Bool		是否成功
Content	String		【Success】为true， 【Content】为任务唯一标识码， 【Success】为false， 【Content】为失败原因。
Code	String		错误代码，"0"为成功

2.3 修改任务执行参数

接口描述	外接系统将任务运行参数修改下发到RCS。
调用地址	<a href="http://IP:Port/Task/ChangeTaskVariable">http://IP:Port/Task/ChangeTaskVariable</a>
下发方	外接系统

下发报文示例：	<pre>{   "ReceiveTaskID": "werwww4eeer",   "MapCode": "testMap",   "MakeTaskID": "task1",   "Variables": [     {       "Code": "StartPoint",       "Value": "PNT_30"     }   ] }</pre>
返回报文示例：	<pre>{   "Content": "",   "Success": true,   "Code": "0" }</pre>
接收方	RCS
请求方式	基于RESTFUL规范的Post方法

### 数据说明

字段名称	类型	是否必填	描述
ReceiveTaskID	String	是	单号，外接系统中生成，唯一识别编号，任务完成后回调使用
MapCode	String	是	地图编码

MakeTaskID	String	否	调度系统生成任务唯一标识
Variables	String	是	任务参数数组，json格式字符串，格式参考示例
返回值：			
Success	Bool		是否成功
Content	String		【Success】为false时 【Content】为失败原因。
Code	String		错误代码，"0"为成功

## 2.4 获取任务状态

接口描述	外接系统将任务运行参数修改下发到RCS。
调用地址	<a href="http://IP:Port/Task/GetTaskState">http://IP:Port/Task/GetTaskState</a>
下发方	外接系统
下发报文示例：	<pre>{   "id": "string" }</pre>
返回报文示例：	1
接收方	RCS
请求方式	基于RESTFUL规范的Post方法



字段名称	类型	是否必填	描述
Id	String	是	单号，外接系统中生成，唯一识别编号
返回值：			int
[Description("接口报错")] Error = -2, [Description("未找到")] NoFound = -1, [Description("未开始")] NoStart = 0, [Description("准备执行")] Ready = 1, [Description("执行中")] Runing = 2, [Description("终止执行")] Canceled = 4, [Description("暂停")] Pause = 8, [Description("执行完成")] Finish = 32,			

## 2.5 推送信息

任务中向第三方系统推送信息，根据任务模板配置推送相应车辆数据和任务执行数据。

接口描述	RCS推送到外接系统。
调用地址	第三方系统配置接口路径

下发方	RCS
推送报文示例：	<pre>{   "ReceiveTaskID": "taskID001",   "MakeTaskID": "taskID001",   "MachineCode": "taskID001",   "Name": "agv01",   "QrCode": "QR00000001",   "XCoordinate": 50.115,   "YCoordinate": 55.113, }</pre>
响应报文示例：	<pre>{   "Success": true,   "ErrorCode": 0,   "Msg": "请求成功！" }</pre>
接收方	第三方系统
请求方式	基于RESTFUL规范的Post方法

#### 数据说明

字段名称	类型	描述
ReceiveTaskID	String	单号，外接系统中生成，唯一识别编号，任务完成后回调使用 <b>默认推送</b>
MakeTaskID	String	调度系统生成任务唯一标识 <b>默认推送</b>

TaskState	String	任务状态 ， 默认推送 Ready ,//准备执行 Runing ,//执行中 Canceled,//终止执行 Pause,//暂停 Finish,//执行完成
MachineCode	String	设备码
Name	String	设备名称
QrCode	String	点识别码
XCoordinate	double	x坐标
YCoordinate	double	y坐标
AgvAngle	double	车身角度
GoodsQrCode	String	货物识别码
AgvPower	Int	电量
BufferNum	double	指令缓冲数
Height	double	举升高度
ErrorCode	String	错误码
ErrorInfo	String	错误信息

AgvState	String	<p>运行状态：</p> <p>Standby ,//待机</p> <p>Free ,/// 空闲</p> <p>NormalRunning,/// 正常运行</p> <p>Error ,/// 故障</p> <p>Charge,/// 充电</p> <p>RequestOnline, /// 请求上线</p> <p>RequestOffline /// 请求下线</p>
ScheduleState	String	<p>调度状态:</p> <p>AgvStatusFree, //空闲</p> <p>AgvStatusCharging, //充电中</p> <p>AgvStatusScheduling, //调度中</p> <p>AgvStatusFreeScheduling, //空闲任务调度中</p> <p>AgvStatusWaring, //报警</p>
OnlineStatus	String	<p>在线状态:</p> <p>RequestOnline , /// 请求上线</p> <p>Online ,/// 在线</p> <p>OffLine ,/// 下线</p> <p>LostIn /// 失联</p>

其他自定义信息	String	Json结构，与上述属性同级，属性名对应指令里配置的Key，属性值对应Value，Value可配置为固定值和关联变量，使用固定值时可配置常量输出。  <b>注意：自定义key值不能与以上字段名称重复，</b>
响应数据：		
Code	Int	返回代码：0成功 1失败
Success	Bool	
ErrorCode	Int	接口调用错误码
Msg	String	返回接口调用描述

## 2.6 获取地图车辆状态

通过WebSocket方式连接，地图运行后，可获取RCS推送的车辆信息

### 1. WebSocket连接地址示例：

ws://127.0.0.1:50051/AgvState?MapCode=45r3r

127.0.0.1:50051为服务器地址，MapCode为地图编码

### 1. 返回数据格式说明：

```
{
  "DataList": [
    {
      "MachineCode": "LA22071004",
      "Name": "LA22071004",
      "QrCode": "",

```

```
"XCoordinate": -100.0,
"YCoordinate": -100.0,
"AgvAngle": 0.0,
"GoodsQrCode": "",
"AgvPower": 0,
"BufferNum": 0.0,
"Height": 0.0,
"SchduleState": "0",
"AgvState": "Free",
"OnlineStatus": 2,
"ErrorCode": "",
"ErrorInfo": "",
"AgvType": 1,
"ComAddress": "127.0.0.1",
"StoreyId": "",
"IsCanDispatcher": true,
"PathList": [],
"Speed": 0.0,
"WarnInfo": "",
"TrayRotationAngle": 0.0,
},
{
  "MachineCode": "LA22071003",
  "Name": "LA22071003",
  "QrCode": "RA0160000011000",
  "XCoordinate": 16.0,
  "YCoordinate": 11.0,
  "AgvAngle": 0.0,
```

```

    "GoodsQrCode": "",
    "AgvPower": 92,
    "BufferNum": 0.0,
    "Height": 0.0,
    "SchduleState": "AgvStatusFree",
    "AgvState": "Free",
    "OnlineStatus": 1,
    "ErrorCode": "",
    "ErrorInfo": "",
    "AgvType": 1,
    "IsCanDispatcher": true,
    "Speed": 0.0,
    "WarnInfo": "",
    "TrayRotationAngle": 0.0,    }
]
}

```

#### 数据说明

字段名称	类型	描述
MachineCode	String	设备码,车辆唯一标识
Name	String	车辆显示名称
QrCode	String	当前二维码
XCoordinate	double	车辆坐标X
YCoordinate	double	车辆坐标Y

AgvAngle	double	车身角
GoodsQrCode	String	货物识别码
AgvPower	Int	电量
BufferNum	int	指令缓冲数
Height	double	举升高度
AgvState	String	运行状态 :  Standby ,//待机  Free ,/// 空闲  NormalRunning,/// 正常运行  Error ,/// 故障  Charge,/// 充电  RequestOnline, /// 请求上线  RequestOffline /// 请求下线
SchduleState	String	调度状态:  AgvStatusFree, //空闲  AgvStatusCharging, //充电中  AgvStatusScheduling, //调度中  AgvStatusFreeScheduling, //空闲任务调度中  AgvStatusWaring, //报警
OnlineStatus	Int	在线状态:  RequestOnline=0 , /// 请求上线  Online=1 ,/// 在线  OffLine=2 ,/// 下线  LostIn=3 /// 失联
ErrorCode	String	错误码列表, 参考 <a href="#">错误映射表</a>
ErrorInfo	String	错误信息列表



AgvType	Int	车辆类型  ECRBT_Camel600 = 1, // 潜伏式顶升600KG  ECRBT_Camel1000 = 2, // 潜伏式顶升1000KG
IsCanDispatcher	Bool	调度状态：  true // 开启调度  false // 停止调度, 停止调度的车辆无法执行任务
Speed	double	车辆速度 (m/s)
WarnInfo	String	告警信息，参考 <a href="#">告警信息映射表</a>
TrayRotationAngle	double	托盘角度
IsLoadGood	Bool	是否载货
NetworkState	Int	网络状态：  Disconnected=0, /// 断开  Connected=1, /// 已连接  TimeOut=2, /// 超时  Error=3 /// 连接错误
TaskState	String	任务状态：  Ready, // 准备执行  Runing, // 执行中  Canceled, // 终止执行  Pause, // 暂停  Finish, // 执行完成
TaskType	String	任务类型：  "充电任务"  "搬运任务"  "空闲任务"
TaskCode	String	单号

TaskName	String	任务模版名
----------	--------	-------

错误映射表

错误码	错误信息
0x01	二维码丢失
0x02	位置超差
0x03	角度超差
0x04	二维码无效
0x05	安全距离不足
0x06	上线点位置无效
0X11	前防撞条触发
0X12	后防撞条触发
0X13	左电机故障
0X14	右电机故障
0X15	雷达触发
0X16	举升电机故障
0X17	旋转电机故障
0X18	路径堵塞
0X19	丢码
0X20	停止点丢码
0X21	路径错误
0X22	位置偏差
0X23	充电错误

0X24	结束充电错误
0X25	Flash读取错误
0X26	左轮运动错误
0X27	右轮运动错误
0X28	雷达3号防区触发

告警映射表

告警码	告警信息
0x00	货物码错误
0x01	货物码丢失
0x02	货物偏移过大FDM
0x03	二维码未识别

## 2.7 获取车辆当前的任务

接口描述	外接系统获取车辆执行任务编码
接口地址	<a href="http://IP:Port/Task/GetTaskByAgvCode">http://IP:Port/Task/GetTaskByAgvCode</a>
下发方	外接系统
下发报文示例：	{ "id": "string" }
返回报文示例：	""

接收方	RCS
请求方式	基于RESTFUL规范的Post方法

#### 数据说明

字段名称	类型	是否必填	描述
Id	String	否	车辆编码
返回值：			
	String		RCS生成的任务编码

## 2.8 暂停任务

接口描述	外接系统暂停AGV当前执行任务
接口地址	<a href="http://IP:Port/Task/ChangeTaskStateByTask">http://IP:Port/Task/ChangeTaskStateByTask</a>
下发方	外接系统
下发报文示例：	<pre>{   "MapCode": "string",   "TaskCode": "string", }</pre>
返回报文示例：	<pre>{   "Content": "",   "Success": true,   "Code": "0" }</pre>

接收方	RCS
请求方式	基于RESTFUL规范的Post方法

#### 数据说明

字段名称	类型	是否必填	描述
MapCode	String	是	地图编码
TaskCode	String	是	第三方系统发送的任务编码
返回值：			
Success	Bool		
Content	String		【Success】为false时 【Content】为失败原因。
Code	String		错误代码，"0"为成功

## 2.9 恢复任务

接口描述	外接系统通过任务编码恢复AGV当前执行任务
接口地址	<a href="http://IP:Port/Task/RecoverAgvTaskByTask">http://IP:Port/Task/RecoverAgvTaskByTask</a>
下发方	外接系统
下发报文示例：	<pre>{   "MapCode": "string",   "TaskCode": "string", }</pre>

返回报文示例：	<pre>{   "Content": "",   "Success": true,   "Code": "0" }</pre>
接收方	RCS
请求方式	基于RESTFUL规范的Post方法

## 数据说明

字段名称	类型	是否必填	描述
MapCode	String	是	地图编码
TaskCode	String	是	第三方系统发送的任务编码
返回值：			
Success	Bool		是否成功
Content	String		【Success】为false时 【Content】为失败原因。
Code	String		错误代码，"0"为成功

## 2.10 批量创建任务

接口描述	外接系统将批量任务通过列表下发到RCS。
调用地址	<a href="http://IP:Port/Task/CreateTaskList">http://IP:Port/Task/CreateTaskList</a>
下发方	外接系统

下发报文示例：

```
[
  {
    "SysToken": "WMS",
    "ReceiveTaskID": "2024111400001",
    "MapCode": "Test",
    "TaskCode": "test",
    "AgvGroupCode": "",
    "AGVCode": "",
    "Priority": 2,
    "Variables": [
      {
        "Code": "StartPoint",
        "Value": "PNT_1_31"
      }
    ]
  },
  {
    "SysToken": "WMS",
    "ReceiveTaskID": "2024111400002",
    "MapCode": "Test",
    "TaskCode": "test",
    "AgvGroupCode": "",
    "AGVCode": "",
    "Priority": 10,
    "Variables": [
      {
        "Code": "StartPoint",
        "Value": "PNT_1_33"
      }
    ]
  }
]
```

	<pre>    }   ] } ]</pre>
返回报文示例：	<pre>{   "DataList": [     {       "Content": "1856878057181155328",       "ReceiveCode": "2024111400001",       "Success": true,       "Code": "0"     },     {       "Content": "1856878057206321152",       "ReceiveCode": "2024111400002",       "Success": true,       "Code": "0"     }   ] }</pre>
接收方	RCS
请求方式	基于RESTFUL规范的Post方法

参数说明

字段名称	类型	是否必填	描述
SysToken	String	是	外接系统标识,需在 RCS 【外接系统配置】 内维护



ReceiveTaskID	String	是	单号，外接系统中生成，唯一识别编号，任务完成后回调使用
MapCode	String	是	地图编码
TaskCode	String	是	任务模板编码
AgvGroupCode	String	否	车辆集群分组编码，标识调用指定集群内车辆执行任务，可设置为空
AGVCode	String	否	车辆编码，标识调用指定车辆执行任务，可设置为空
Priority	Int	否	任务优先级 1~10，默认为5
Variables	String	是	任务参数数组，json格式字符串，格式参考示例
返回值：			
Success	Bool		是否成功
ReceiveCode	String		下发任务的单号
Content	String		<b>【Success】</b> 为true时 <b>【Content】</b> 为当前任务唯一标识码， <b>【Success】</b> 为false， <b>【Content】</b> 为失败原因。
Code	String		错误代码，"0"为成功

# 3 TCP

## 3.1 调用说明

RCS作为服务器接受客户端的请求并向客户端作出响应。API 请求由头部和数据区组成，头部是定长的，用于标识数据包基本信息以及数据区的长度及类型等，数据区为序列化后的 JSON 数据，根据头部中的类型信息进行反序列化可以得到相应的 JSON 对象。API 的响应也由头部和数据区组成，头部为根据请求得到的定长字节(响应头部与对应请求头部的关系见下文)，数据区也为序列化后的 JSON 数据。

## 3.2 报文结构

名称	内容	长度 (byte)	描述
报文同步头	0x5A	1 (uint8)	报文同步头, 用于确定报文头部的开始
协议版本	1	1 (uint8)	协议的主版本号, 目前均填 0x01
序号		2 (uint16)	请求及响应的序号 (0 ~ 65535) , 请求包与响应包的这个字段是相同的, API 使用者自行填入序号。RCS对每个请求的响应都使用这个序号。
数据区长度		4(uint32)	数据区长度, 即 JSON 序列化数据的长度
任务类型 (编号)		2(uint16)	标识报文的类型, 即 API 的编号, 具体见 下表

内部使用区域		6(uint8[6])	程序内部使用（不必关注该区域的内容，其内容可能不为 0 且可能会发生变化）
数据区		取决于头部中的数据区长度	JSON 序列化的数据内容

如上表所示, 报文头部的长度为  $1 + 1 + 2 + 4 + 2 + 6 = 16$  字节, 请求与响应的报文头部长度是相同的。

1. 如果某个报文类型没有数据区, 那么数据区长度的内容为 0x00000000。
2. 报文类型(编号)必须是下文 API 编号中的一种, 机器人若收到未定义的报文类型将会响应一个错误。
3. 若 报文头部 格式错误无法解析, 机器人将 断开连接 , 不做任何响应。
4. 若数据区解析错误也将会响应一个错误。

注意:

1. 每个报文头部正确的请求都会有一个响应, 编号用于将请求与响应对应起来, 一般情况下, 响应的编号为请求编号加 10000 (0x2710)
2. 机器人任何情况下都不会主动发送数据
3. 机器人若收到不属于某个端口的报文类型, 将会响应错误
4. 保留区域的 6 个字节必须填充不可省略, 用 0x000000000000 填充即可

任务类型说明

告警码	告警信息
0x00 0x01	创建任务
0x00 0x02	停止任务
0x00 0x03	修改变量
0x00 0x04	获取任务状态
0x00 0x05	获取车辆当前任务
0x00 0x06	恢复任务

### 3.3 请求示例

以下文中的 任务取消 （0x00 0x02 ）请求为例

请求数据区结构如下：

字段名	类型	描述	可缺省
ReceiveTaskID	String	外界系统任务编号	否
AgvCode	String	Agv编码	否

任务取消请求, 参数为 ReceiveTaskID、AgvCode , 标准 JSON 对象如下（[详细说明参考HTTP对接](#)）：

```
{
  "ReceiveTaskID": "40b6d226-cf75-4342-bd4a-1212eb21fee9",
  "AgvCode": "Agv01"
}
```

在报文中, 需要将上面的 JSON 对象序列化并塞入报文的数据区, 并将序列化后的长度填入报文头部的数据区长度中。

序列化后的数据为 { "ReceiveTaskID": "40b6d226-cf75-4342-bd4a-1212eb21fee9",  
"AgvCode": "Agv01"}，转化为十六进制为 7b 22 52 65 63 65 69 76 65 54 61 73 6b 49 44 22 3a  
22 34 30 62 36 64 32 32 36 2d 63 66 37 35 2d 34 33 34 32 2d 62 64 34 61 2d 31 32 31 32 65  
62 32 31 66 65 65 39 22 2c 22 41 67 76 43 6f 64 65 22 3a 22 41 67 76 30 31 22 7d，长度为  
74(0x4a) 个字节，因此头部为 5A 01 00 01 00 00 00 4a 00 02 00 00 00 00 00 00，将头部与数据区  
拼接在一起得到一个请求如下(共 90 字节):

```
5A 01 00 01 00 00 00 4a 00 02 00 00 00 00 00 00
7b 22 52 65 63 65 69 76 65 54 61 73 6b 49 44 22 3a 22 34 30 62 36 64 32 32 36 2d 63 66 37
35 2d 34 33 34 32 2d 62 64 34 61 2d 31 32 31 32 65 62 32 31 66 65 65 39 22 2c 22 41 67 76
43 6f 64 65 22 3a 22 41 67 76 30 31 22 7d
```

### 3.4 响应示例

以下文中的 任务取消（0x00 0x02）号响应为例

响应数据区结构如下：

字段名	类型	描述	可缺省
data	number	查询数据	是
msg	number	备注说明	是
retcode	number	API 错误码	否

每一次返回的数据区 JSON Object 中 Key 的顺序是不保证的，因为 JSON 并没有规定这一点。

RCS要返回的 JSON 对象如下：

```
{
"retcode":-1,
"msg":"StopAgvTask Result :\" map is not running\\",
"data":null
}
```

序列化后的数据为 {"retcode":-1,"msg":"StopAgvTask Result :\" map is not running\\", "data":null}, 转化为十六进制为 7b 22 72 65 74 63 6f 64 65 22 3a 2d 31 2c 22 6d 73 67 22 3a 22 53 74 6f 70 41 67 76 54 61 73 6b 20 52 65 73 75 6c 74 20 3a 5c 22 20 6d 61 70 20 69 73 20 6e 6f 74 20 72 75 6e 6e 69 6e 67 5c 22 22 2c 22 64 61 74 61 22 3a 6e 75 6c 6c 7d

长度为 78(0x4e) 字节，因此响应的头部为 5A 01 00 01 00 00 00 3C 2A FC 00 00 00 00 00 00 拼接后数据如下(共 94 字节)：

```
5A 01 00 01 00 00 00 3C 2A FC 00 00 00 00 00 00

7b 22 72 65 74 63 6f 64 65 22 3a 2d 31 2c 22 6d 73 67 22 3a 22 53 74 6f 70 41 67 76 54 61
73 6b 20 52 65 73 75 6c 74 20 3a 5c 22 20 6d 61 70 20 69 73 20 6e 6f 74 20 72 75 6e 6e 69
6e 67 5c 22 22 2c 22 64 61 74 61 22 3a 6e 75 6c 6c 7d
```

### 3.5 响应结构

API 响应数据区通常包含如下字段：

字段名	类型	描述	可缺省
-----	----	----	-----

data	number	查询数据	是
msg	number	备注说明	是
retcode	number	API 错误码	否

API 错误码为响应数据区 JSON 对象包含的错误码（retcode），用于指示对请求执行不成功或请求出错等错误。若 ret\_code 为 0 或缺省，说明没有错误，若不为 0，说明发生了错误，错误详情查看msg。

没有错误时，msg 也可能会缺省，即会出现数据区为空的情况。

data为API返回的查询数据。

## 4 HTTP错误码映射表

错误码	异常信息
4000	异常报错
4001	地图为空
4002	地图未运行
4003	任务下发唯一ID重复
4004	任务编码未找到
4005	AGV群组未找到
4006	未找到AGV匹配群组
4007	指定AGV不匹配任务模板车型配置
4008	未找到任务模板配置的车型群组
4009	超过任务集群配置的最大任务数量
4010	任务变量未赋值
4011	任务变量转换失败
4012	未在地图找到当前传入点位信息

4013	超过任务模板配置的最大任务数量
4014	任务编码且AGV编号为空
4015	未找到指定任务