

润木机器人调度系统与任务系统通讯接口文档

目录

润木机器人调度系统与任务系统通讯接口文档	1
前言	2
历史版本	3

1 MRSE 调度系统提供接口	4
1.1 说明.....	4
2.1 向调度系统插入订单(常用等级: *****)	7
2.2 取消订单(常用等级: ****)	14
2.3 释放订单占用的 AGV(常用等级: *)	15
2.4 确认订单	16
2.5 确认 AGV	17
2.6 查询订单状态	18
2.7 查询订单状况	18
2.8 查询 AGV 的任务状态	19
2.9 查询 AGV 的状况	20
2.10 查询 AGV 的信息	23
2.11 查询所有 AGV 的状况	25
2.12 查询所有 AGV 的信息	26
2.13 请求为 AGV 充电	28
2.14 请求为 AGV 结束充电	29
2.15 暂停 AGV	30
2.16 恢复 AGV	30
2.17 查询 AGV 的路线(剩余路线)	31
2.18 查询调度系统中的拓扑地图(点位地图)	33
3.1 调度系统推送订单开始执行的信号给外部系统	34
3.2 调度系统推送订单完成的信号给外部系统	35
3.3 调度系统推送订单被取消的信号给外部系统	35
3.4 调度系统推送子订单完成的信号给外部系统	36
3.5 调度系统推送请求确认信号	37

前言

本文档为润木机器人调度系统执行标准订单的通讯接口文档,上层订单生成系统可以通过该文档的接口对给调度系统下发订单,来间接调度 AGV 去执行订单的目的,上层订单生成系统一般为 MES, WCS, WMS 等。

注意: 本协议版权归润木机器人(深圳)有限公司所有,未经允许,不得使用、复制或传播。

历史版本

版本号	时间	作者	修改说明
V1.0.0	2024.5.17	YangYang	初始版本
V1.0.1	2024.12.12	YangYang	增加了查询拓扑地图，查询 AGV 路线， 暂停/恢复 AGV 的接口

1 MRSE 调度系统提供接口

1.1 说明

本协议采用标准的 Http 协议，json 格式，其中：
调度系统 ip（简称 MRSE_IP）：**xxx.xxx.xxx.xxx**；
端口号（简称 PORT）：默认 **5566**；
统一使用 json 格式，采用 utf-8 编码
本文档里所有关于距离，坐标，长度等字段的单位都是“m”，线速度是“m/s”
为了接口的统一，本 http 协议的 URL 全部遵循一个 base url,base url 为 <http://IP:PORT/MRSE/REST>，
全部的指令均采用 Post 方式，body 部分均包裹一个通用的外壳，外壳是一个大的 json 对象（注意本协议很多地方采用了嵌套 json），body 分为 Post body 和 Reply body，Post body 格式如下：

调度系统从任务系统接收指令 body					
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	指令编号	uuid	string	交互的唯一序列，如果有重复，系统默认是同一条指令，防止一个指令重复调用	否
	时间戳	timeStamp	string	报文的生成时间，为“yyyy-MM-dd hh:mm:ss”的字符串格式，可为空，但最好填上“yyyy-MM-dd hh:mm:ss”	是
	协议版本号	version	string	使用的协议文档版本号，格式为“xx.xx.xx”	否
	数据	data	json	真正携带的数据，该字段是个 json 对象，如果不需要携带任何数据，该字段可以为空	是
应答	指令编号	uuid	string	必须与 Post 时收到的 uuid 相同	否

	响应码	code	int	通过响应码来判断指令是执行成功还是失败，如果为 0 则代码成功，否则为失败，如果成功，且回复里携带了数据，则查询 result 字段来获取真正的数据，如果失败，则查询 errMsg 字段来查询具体的错误信息，通常响应码 2 指 json 格式方面的错误	否
	时间戳	timeStamp	string	报文的生成时间，为“yyyy-MM-dd hh:mm:ss”的字符串格式，可为空，但最好填上	是
	回复数据	result	json	回复里真正携带的数据，该字段是个 json 对象或 json 数组，如果不需要携带任何数据或者回复码 code 不为 0，该字段可以为空	是
	错误信息	errMsg	string	携带的具体的错误信息，如果回复码是 0，则该字段为空，如果不为 0，则必须包含此字段来反馈错误信息	是
示例	请求	{ "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "version": "1.0.0", "data": { } }			
	应答	成功	{ "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0, "result": { } }		
		失败	{ "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 1, "errMsg": "the agv1 is not existed" }		

调度系统向订单来源系统推送指令 body					
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	指令编号	uuid	string	交互的唯一序列，如果有重复，系统默认认为是同一条指令，防止一个指令重复调用	否
	时间戳	timeStamp	string	报文的生成时间，为“yyyy-MM-dd hh:mm:ss”的字符串格式，可为空，但最好填上“yyyy-MM-dd hh:mm:ss”	是
	协议版本号	version	string	使用的协议文档版本号，格式为“xx.xx.xx”	否
	数据	data	json	真正携带的数据，该字段是个 json 对象或者 json 数组，如果不需要携带任何数据，该字段可以为空	是
应答	指令编号	uuid	string	必须与 Post 时收到的 uuid 相同	否
	响应码	code	int	通过响应码来判断指令是执行成功还是失败，如果为 0 则代码成功，否则为失败，如果成功，且回复里携带了数据，则查询 result 字段来获取真正的数据，如果失败，则查询 errMsg 字段来查询具体的错误信息	否
	时间戳	timeStamp	string	报文的生成时间，为“yyyy-MM-dd hh:mm:ss”的字符串格式，可为空，但最好填上	是
	回复数据	result	json	回复里真正携带的数据，该字段是个 json 对象或 json 数组，如果不需要携带任何数据或者回复码 code 不为 0，该字段可以为空	是
	错误信息	errMsg	string	携带的具体的错误信息，如果回复码是 0，则该字段为空，如果不为 0，则必须包含此字段来反馈错误信息	是
示例	请求	{ "uuid": "1234", "destination": "WMS", "timeStamp": "2023-10-21 10:59:58",			

		<pre>"version": "1.0.0", "data": { } }</pre>	
	应答	成功	<pre>{ "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0, "result": { } }</pre>
		失败	<pre>{ "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 1, "errMsg": "" }</pre>

2.1 向调度系统插入订单(常用等级: *****)

接口名	InsertOrder				
功能说明	<p>向调度系统请求 AGV 去搬运货物是以订单(Order)的形式, 订单是调度系统的派单的最小调度单元, 但是订单可以由多个子订单(SubOrder)组成, 任务系统调用该接口把任务以订单的形式插入到调度系统的订单队列中, 从而申请 AGV 去执行相应的搬运任务, 订单可以指定 AGV, 也可以不指定 AGV, 如果指定了 AGV, 则调度系统会安排该指定的 AGV 去执行该订单, 且如果该 AGV 满足执行订单的条件, 会立刻将该订单派送给该 AGV 执行, 否则该订单在订单队列中排队, 等待该 AGV 满足执行条件再执行, 如果该订单未被指定 AGV, 调度系统会自行分配某台满足条件的空闲 AGV 去执行, 如果能找到满足条件的空闲 AGV, 会立刻将该订单派发给该 AGV 执行, 否则该订单在订单队列中等待, 直到找到满足条件空闲 AGV (通常是指有某台 AGV 完成了旧订单, 进入了空闲状态)</p>				
提供方	MRSE 调度系统				
调用方	任务系统				
请求参数	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
	订单编号	order::id	string	唯一地标识订单, 任务系统通过该编号来查询和追踪该订单的执行状况和生命周期, 该字段可为空, 如果为空, 则由调度系统自行生成一个唯一	是

				的编号	
	订单名称	order::name	string	订单的描述性的名字,可重复	是
	调整子订单列表的功能使能	order:: allowAdjustSubOrderSequence	bool	如果设为 true, 表示允许调度系统可以自行优化该订单中的子订单序列的执行顺序,此时不一定会严格按照插入时的子订单的顺序来执行,比如去点 2 举升->点 3 下降,可能最后调度系统会调整为去点 3 下降->点 2 举升, 该字段可为空,默认为空, 在为空的情况下,默认值为 false, 通常都应该设为 false	是
	在该订单完成后是否释放所占用的 AGV 的功能使能	order:: releaseAgvAfterOrderCompleted	bool	一般用于订单完成,而流程没有完成, 如果设为 true,说明在该订单被执行完后会释放其占用的 AGV, 此时调度系统会为该 AGV 派发下一条订单或者去充电, 否则 AGV 会一直被该订单占用直到其收到手动释放指令或者收到下一条” releaseAgvAfterOrderCompleted” 为 true 的订单, 并且订单完成, 此时表明流程完成,在释放订单前可以接收并执行后续的剩余订单,剩余订单是指 id 和本订单相同, 并且 orderInsertionMode 为 3 的订单, 该字段可为空, 默认为空, 在为空的情况下,默认值为 true, 释放指令请参见	是
	优先级	order::data::priority	int	必须为正数,值越大优先级越高,当订单队列中有多条未执行的订单排队时, 根据优先级来插队	否
	指定的 AGV	order::data::agvId	int	为该订单指定特定的	是

				AGV，只有该 AGV 才能执行该订单，当字段值为 0 时表示不指定 AGV，该字段可为空，在为空的情况下，默认值为 0，表示不指定 AGV，交由调度系统来决定由哪台 AGV 来执行	
	可选的 AGV 列表	order::data::optionalAgvs	数组	表示该订单只能交由这些 AGV 中的任意一台执行，缩小了调度系统选择 AGV 的范围，该字段可为空，默认为空，该字段只能在未指定 AGV 的情况下才有效，也就是 order::data::agvId 为空或者值为 0 的情况下才有效	是
	排除的 AGV 列表	order::data::excludeAgvs	数组	表示该订单禁止交由这些 AGV 执行，缩小了调度系统选择 AGV 的范围，该字段可为空，默认为空，该字段只能在未指定 AGV 和没有可选的 AGV 列表的情况下才有效	是
	子订单的编号	order::data::subOrder::id	string	标识该子订单，通过该编号来查询和追踪该子订单的执行状况和生命周期，该编号在该订单中必须唯一，但是允许与其他订单的子订单编号相同，所以查询一个子订单必须通过订单编号+子订单编号组合来查询	否
	开始执行子订单的确认信号	order::data::subOrder::requireStartConfirm	bool	用来告诉调度系统，执行该子订单前需要先收到外部的确认信号才能执行，否则一直等待，该字段可为空，默认为空，在为空的情况下的默认值是 false，表示不需要确认信号	是
	结束子订单的完成确认	order::data::subOrder::requireEndConfirm	bool	用来告诉调度系统，该子订单被 AGV 完成后需要	是

	信号			先收到外部的确认信号才算真正完成,从而才能继续执行下一条子订单,该字段可为空,默认为空,在为空的情况下的默认值是 false,表示不需要确认信号	
	目标节点编号	order::data::subOrder::data::vertex	int	表示 AGV 要移动到达的目标节点号	否
	目标站点位置	order::data::subOrder::data::station	int	表示 AGV 要移动到达的目标站点号(注意目标节点号字段和目标站点号字段只能二选一,如果两个字段同时存在,目标节点号优先),每个站点一定是个节点,但反过来不一定,站点号会对应一个节点号,通常一个站点代表着一个工作站	否
	在目标点上的动作类型	order::data::subOrder::data::action::type	string	表示在目标点的动作命令类型,类型与 AGV 类型有关,常用的动作命令是举升,下降,该字段也可以为空,如果为空,说明 AGV 只是移动	是
	在目标点上的动作值	order::data::subOrder::data::action::value		表示目标点的动作参数值,具体的值类型是可变的,根据动作的类型以及 AGV 的类型来确定,比如如果是带机械手的协作 AGV, 值类型为字符串,描述机械手该如何动作,如果是叉车,这里值类型是 int,表示叉车应该举升到多高的高度,该字段也可以为空,比如对于潜伏举升 AGV 的举升和下降动作,则不需要该字段,因为用不上参数值	是
	开始执行动作的确认信号	order::data::subOrder::data::action::requireStartConfirm	bool	用来告诉调度系统,执行该动作前需要先收到外部的确认信号才能执行,否则一直等待,该字段可为空,默认为空,在为空	

				的情况下的默认值是 false，表示不需要确认信号	
	动作完成的确认信号	order::data::subOrder::data::action::requireEndConfirm	bool	用来告诉调度系统,该动作被 AGV 完成后需要先收到外部的确认信号才算真正完成,从而才能继续执行下一条子订单,类似收到放行信号才能走,该字段可为空,默认为空,在为空的情况下的默认值是 false，表示不需要确认信号	
	强制将订单插入到订单队列的最前面	forceIntoFrontOfOrderQueue	bool	如果为 true，该订单会强制插队,直接被安排到订单队列的最前面,而不管优先级大小,一旦有可用的空闲 AGV,该订单会被调度系统最优先分配给 AGV (注意: 如果之后又有其他其它订单调用本接口强制插入到订单队列的最前面,本订单的最前面位置会被抢占),该字段可为空,默认为空, 为空时默认值为 false	是
	订单插入模式	orderInsertionMode	int	0 New 新订单, 1 Append 将订单添加到老订单的后面, 2 Cover 将订单覆盖老订单中未执行的剩余子订单, 3 Remain 作为老订单的剩余订单 (New 模式插入的订单的 id 必须是新的且唯一,其他模式插入的订单的 id 在调度系统中必须已存在, 依赖一条老订单,表示他们是同一个流程, 其中 Remain 模式只有在该订单占用的 AGV 在等待释放的条件下才有效), 该字段可为空, 在为空的情况下默认为	是

				New 模式	
	端口号	port	int	如果需要调度系统主动将该订单的完成信号或取消信号等推送给该任务系统,则需要设置该字段,调度系统会按照订单从哪个主机 ip 地址来,就推送回哪台主机 ip 去,该字段表明调度系统推送的端口号,字段可为空,如果为空,表明不需要推送	是
应答	订单编号	orderId	string	该订单的编号	否
	订单状态	orderState	int	该订单的执行状态,状态的枚举值说明请参考下表中的查询订单状态指令,该字段可为空,如果为空,则说明该订单还未被执行,只是插入到订单队列中排队	是
	订单派发给的 AGV 编号	agvId	int	执行该订单的 AGV 编号,如果订单还没有被派发给任何 AGV 去执行,该字段为空	是
	请求	Post: http://IP:PORT/MRSE/REST/InsertOrder 订单 1: 派 AGV1 去节点 20 举升, 然后去节点 25 下降 ("StageLift"和"StageFall"分别是潜伏举升型 AGV 的举升命令和下降命令的专有字符串), 并且在订单完成后将信号用 http 推送给端口为 8800 的系统 <pre> { "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "order": { "data": { "agvId": 1, "subOrders": [{ "data": { "action": { "type": "StageLift" }, "vertex": 20 } }] } } } } </pre>			

		<pre> "id": "1" }, { "data": { "action": { "type": "StageFall" }, "vertex": 25 }, "id": "2" }], "priority": 1 }, "id": "1", "name": "go", "releaseAgvAfterOrderCompleted": true } "orderInsertionMode": 0, "port": 8800, } }</pre> <p>订单 2: 派 AGV2 移动到节点 20, 并且在订单完成后将信号用 http 推送给端口为 8800 的系统</p> <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "order": { "data": { "agvId": 2, "subOrders": [{ "data": { "vertex": 20 }, "id": "1" }], "priority": 1 }, "id": "2", "name": "go to station1", } } }</pre>
--	--	--

		<pre> "releaseAgvAfterOrderCompleted": true } "orderInsertionMode": 0, "port": 8800, } } </pre>
示例	应答	<pre> { "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0, "result": { "orderId": "1", "orderState": 1, "agvId": 1, } } </pre>

2.2 取消订单(常用等级: ****)

接口名	CancelOrder				
功能说明	取消指定的订单，如果订单还未执行，还在订单队列中排队，则直接将该订单从订单队列中删除，如果订单正在被相应的 AGV 执行，则终止执行，如果订单正在执行，且包含着"releaseAgvAfterOrderCompleted"且为 false，则 AGV 被订单占用了，即使取消了该订单，AGV 仍然会被占用，此时订单被取消后，AGV 依然不会执行下条订单或者去充电，如果想要 AGV 执行下条订单或者去充电，需要调用 2.3 的"释放订单占用的 AGV"接口				
提供方	MRSE 调度系统				
调用方	任务系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	订单编号	orderId	string	要取消的订单编号	否
	子订单编号	subOrderId	string	要取消的子订单编号，如果该字段为空，则取消整条订单，否则只取消该订单下的孩子订单，默认为空，表示取消整条订单	是
	是否软取消	soft	bool	取消分为两种取消，一种是硬取消，一种是软取消，如果为 true，表示软取消，则提示执行该订单的 AGV 不是直接停下来，而是会先去处理某些事，比如先去卸货架，具体要做的事是由 AGV 车型以及调度的业务逻辑决定的，此处只是做个提示，该字段可为空，默认为空，在为空的情况下值为	是

				false, 硬取消	
应答					
示例	请求	Post: http://IP:PORT/MRSE/REST/CancelOrder 取消订单 1 <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "orderId": "1", "soft": false } }</pre> 取消订单 1 下的子订单 100 <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "orderId": "1", "subOrderId": "100", "soft": false } }</pre>			
	应答	<pre>{ "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 1, "errMsg": "the order1 is not existed" }</pre>			

2.3 释放订单占用的 AGV(常用等级：*)

接口名	ReleaseAgv				
功能说明	释放订单占用的 AGV, 释放后 AGV 相当于流程执行完了, 可以执行下条订单或者去充电了, 如果 AGV 此时正在执行订单, 会直接取消订单, 注意释放和取消订单的区别, 前者是取消流程, 后者只是取消订单, 流程>订单, 取消流程包括了取消订单				
提供方	MRSE 调度系统				
调用方	任务系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	AGV 编号	agvId	int	要释放的 AGV 编号	是
	订单编号	orderId	string	通过 orderId 来查找 AGV 编号, agvId 和 orderId 必须有一个非空, 如果两个都非空, 则优先	是

				使用 orderId 字段	
应答					
示例	请求	Post: http://IP:PORT/MRSE/REST/ReleaseAgv <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "orderId": "1" } }</pre>			
	应答	<pre>{ "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0 }</pre>			

2.4 确认订单

接口名	ConfirmOrder				
功能说明	如果子订单或动作中有 requireStartConfirm 或 requireEndConfirm 字段, 那负责执行此订单的 AGV 在执行到这些步骤时, 会暂停下来, 直到收到外部系统 (包括任务系统, 下同) 的确认信号后才会继续执行订单的下一步, 任务系统可以通过该接口向调度系统下发订单的确认信号, 确认信号外部系统也可以提前下发给调度系统, 如果提前下发, 那之后 AGV 执行到这些步骤时会直接跳过, 执行下一步				
提供方	MRSE 调度系统				
调用方	任务系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	订单编号	orderId	string	要确认的订单编号	否
	子订单编号	subOrderId	string	要确认的订单的子订单编号	否
	确认类型	confirmType	int	确认类型, 1 等待子订单的 startConfirm, 2 等待子订单的 endConfirm, 3 等待动作的 startConfirm, 4 等待动作的 endConfirm	否
应答					
示例	请求	Post: http://IP:PORT/MRSE/REST/ConfirmOrder <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "orderId": "1", "subOrderId": "1", "confirmType": 1 } }</pre>			
	应答				

		<pre> } } </pre>
	应答	<pre> { "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0, } </pre>

2.5 确认 AGV

接口名	ConfirmAgv				
功能说明	<p>如果子订单或动作中有 requireStartConfirm 或 requireEndConfirm 字段, 那负责执行此订单的 AGV 在执行到这些步骤时, 会暂停下来, 直到收到外部系统 (包括任务系统, 下同) 的确认信号后才会继续执行订单的下一步, 任务系统可以通过该接口向调度系统下发 AGV 的确认信号, 此接口与 "ConfirmOrder" 接口类似, 区别在于前者是通过订单来确认, 后者是通过 AGV 来确认, 并且不能提前下发确认信号, 只有 AGV 当前正在等待确认, "ConfirmAgv" 才能生效</p>				
提供方	MRSE 调度系统				
调用方	任务系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	AGV 编号	agvId	int	要确认的 AGV	否
	确认类型	confirmType	int	确认类型, 1 等待子订单的 startConfirm, 2 等待子订单的 endConfirm, 3 等待动作的 startConfirm, 4 等待动作的 endConfirm, 该字段可为空, 如果为空, 则调度系统不比对 confirmType 是否与 AGV 正在等待的确认信号相同	是
应答					
示例	请求	Post: http://IP:PORT/MRSE/REST/ConfirmAgv <pre> { "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "agvId": 1, "confirmType": 1 } } </pre>			
	应答	<pre> { "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0, } </pre>			

2.6 查询订单状态

接口名	QueryOrderState				
功能说明	查询订单状态，也可以查询订单的子订单状态				
提供方	MRSE 调度系统				
调用方	任务系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	订单编号	orderId	string	要查询的订单编号	否
	子订单编号	subOrderId	string	要查询的子订单编号，该字段可为空，如果为空，则说明查询的是整个订单的状态，否则查询订单的子订单的状态	是
应答	状态	state	int	0 未开始，1 正在执行，2 正在取消，3 已完成，4 已取消	否
示例	请求	Post: http://IP:PORT/MRSE/REST/QueryOrderState <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "orderId": "1" } }</pre>			
	应答	<pre>{ "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0, "result": { "state": 1 } }</pre>			

2.7 查询订单状况

接口名	QueryOrderStatus				
功能说明	查询订单状况				
提供方	MRSE 调度系统				
调用方	任务系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	订单编号	orderId	string		否
应答	订单状态	state	int	0 未开始，1 正在执行，2 正在取消中，3 已完成，4 已取消	否

	负责执行的 AGV 编号	agvId	int	表明订单由哪台 AGV 执行，如果订单还没被任何 AGV 执行，该字段为空	是
	子订单编号	subOrderId	string	正在执行的子订单编号，如果订单还没开始执行，该字段为空	是
	确认类型	confirmType	int	正在等待的确认类型，0 不等待任何确认，1 等待子订单的 startConfirm，2 等待子订单的 endConfirm，3 等待动作的 startConfirm，4 等待动作的 endConfirm，该字段可未空，默认为空，在为空的情况下默认值为 0，表明不等待任何确认	是
	错误	error	string	订单执行过程中发生的错误信息，如果未发生任何错误，则为空	是
示例	请求	Post: http://IP:PORT/MRSE/REST/QueryOrderStatus <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "orderId" : "1" } }</pre>			
	应答	<pre>{ "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0, "result": { "state": 1, "agvId": 1, "subOrderId": "1" } }</pre>			

2.8 查询 AGV 的任务状态

接口名	QueryAgvState				
功能说明	查询 AGV 的任务状态（-1 未知状态，0 正在初始化中状态，1 空闲状态，2 执行任务状态，3 等待被释放的状态，4 前往充电状态，5 充电对接状态，6 正在充电中状态，7 结束充电对接状态）				
提供方	MRSE 调度系统				
调用方	外部系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	AGV 编号	agvId	int		否
应答	AGV 状态	state	int	-1 未知状态，0 正在初始化中状态，1 空闲状态，2 执行任务状态，3 等待被释放的状态，4 前往充电状态，5 充电对接状态，6 正在充电中状态，7 结束充电对接状态	否
示例	请求	Post: http://IP:PORT/MRSE/REST/QueryAgvState <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "agvId" : 1 } }</pre>			
	应答	<pre>{ "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0, "result": { "state": 2 } }</pre>			

2.9 查询 AGV 的状况

接口名	QueryAgvStatus				
功能说明	查询 AGV 的状况，状况主要是包括了 AGV 的运行时的动态信息				
提供方	MRSE 调度系统				
调用方	任务系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	AGV 编号	agvId	int		否
	对于是默认值的字	fieldWithDefaultValueAreSetToEmpty	bool	对于是默认值的字段返回时设置为空，主要	是

	段设置为空			是为了减小应答包的体积大小，节省宽带方便传输，该字段可为空，在为空的情况下，默认值为 true	
应答	任务状态	state	int	参照 QueryAgvState 接口	否
	确认类型	confirmType	int	等待的确认信号类型，一共有 4 种确认类型，分别是 1 子订单的 requireStartConfirm, 2 子订单的 requireEndConfirm, 3 动作的 requireStartConfirm, 4 动作的 requireEndConfirm	是
	当前点	currentVertex	int	AGV 的当前点，为 0 时表示 AGV 还不知道在哪个点上	否
	位姿 x	x	float	AGV 当前的 x 坐标（单位：m），如果调度系统还未获取到 Agv 的当前位姿，或者 AGV 是磁导航 AGV，则该字段为空	是
	位姿 y	y	float	AGV 当前的 y 坐标（单位：m），如果调度系统还未获取到 Agv 的当前位姿，或者 AGV 是磁导航 AGV，则该字段为空	是
	位姿 theta	theta	float	AGV 当前的角度（单位：弧度），如果调度系统还未获取到 Agv 的当前位姿，或者 AGV 是磁导航 AGV，则该字段为空	是
	位置信度	confidence	float	描述了当前 slam 位置的准确度，如果调度系统还未获取到 Agv 的当前位姿，或者 AGV 是磁导航 AGV，则该字段为空	是

	电量	battery	int	AGV 当前的剩余电量 (0%~100%)	否
	当前订单	orderId	string	AGV 当前正在执行的订单编号, 如果当前未执行任何订单, 此字段为空	是
	当前子订单	subOrderId	string	AGV 当前正在执行的订单的子订单编号, 如果当前未执行任何订单, 此字段和 orderId 字段一起为空	是
	速度	speed	float	AGV 当前的行驶速度 (单位: m)	否
	网络连接状况	networkConnected	bool	如果 AGV 当前与调度系统的通讯连接正常, 则值为 true, 否则为 false	否
	是否被暂停	paused	bool	AGV 是否被暂停, 注意这里的暂停是指 PauseAgv 接口的暂停	否
	是否被交管	trafficControl	bool	AGV 当前是否被交管了	是
	AGV 载台升降状态	liftState	int	0 静止, 1 举升中, 2 下降中 (该字段为可选字段, 只有潜伏举升型 AGV 才有此字段)	是
	AGV 载台升降位置	stagePosition	int	0 未知, 1 在原点, 2 在上面, 3 在下面, 4 在上极限位, 5 在下极限位该字段为可选字段, 只有潜伏举升型 AGV 才有此字段)	是
	请求	Post: http://IP:PORT/MRSE/REST/QueryAgvStatus <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "agvId": 1, "fieldWithDefaultValueAreSetToEmpty": false } }</pre>			

示例		<pre>{ "code": 0, "result": { "status": { "battery": 100, "confidence": 1.0, "currentVertex": 106, "subOrderId": "a4214b01-7523-413d-a8e0-b9f95284af60", "networkConnected": true, "orderId": "1", "paused": false, "performingAction": false, "speed": 0.0, "state": 1, "theta": 5.832533402379764, "trafficControl": false, "x": -28.250629, "y": 14.056995, "confirmType": 0 } }, "timeStamp": "2023-10-24 16:50:47", "uuid": "1234" }</pre>
	应答	

s

2.10 查询 AGV 的信息

接口名	QueryAgvInfo				
功能说明	查询 AGV 的信息，信息包括 Status，同时还包括了一些 AGV 的属性				
提供方	MRSE 调度系统				
调用方	任务系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	AGV 编号	agvId	int		否
	对于是默认值的字段设置为空	fieldWithDefaultValueAreSetToEmpty	bool	对于是默认值的字段返回时设置为空，主要是为了减小应答包的体积大小，节省宽带方便传输，该字段可为空，在为空的情况下，默认值为 true	是
应答	AGV 状况	status	AgvStatus	AGV 状况，参考 QueryAgvStatus 接口	否
	AGV 的 ip 地	ipAddress	string		否

	址				
	AGV 名字	name	string		否
	AGV 类型	type	string		否
示例	请求	Post: http://IP:PORT/MRSE/REST/QueryAgvInfo { "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "agvId": 1, "fieldWithDefaultValueAreSetToEmpty": false } }			
	应答	{ "code": 0, "result": { "agvInfo": { "ipAddress": "0.0.0.1", "name": "", "status": { "battery": 100, "confidence": 1.0, "currentVertex": 60, "subOrderId": "47f48d2d-bd62-40b5-88bf-9de88c89f814", "networkConnected": true, "orderId": "58", "paused": false, "performingAction": false, "speed": 0.0, "ssttate": 1, "theta": 0.735888525514476, "trafficControl": false, "x": -2.550629, "y": -2.843005, "confirmType": 0 }, "type": "" } }, "timeStamp": "2023-10-24 17:11:57", "uuid": "1234" }			

2.11 查询所有 AGV 的状况

接口名	QueryAllAgvsStatus				
功能说明	查询所有 AGV 的状况				
提供方	MRSE 调度系统				
调用方	外部系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	对于是默认值的字段设置为空	fieldWithDefaultValueAreSetToEmpty	bool	对于是默认值的字段返回时设置为空，主要是为了减小应答包的体积大小，节省带宽方便传输，该字段可为空，在为空的情况下，默认值为 true	是
应答	AgvStatus 数组		array	所有 AGV 状况的数组	否
示例	请求	Post: http://IP:PORT/MRSE/REST/QueryAllAgvsStatus <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "fieldWithDefaultValueAreSetToEmpty": false } }</pre>			
	应答	<pre>{ "code": 0, "result": [{ "agvId": 1, "status": { "battery": 100, "confidence": 1.0, "currentVertex": 115, "subOrderId": "{612f7e7b-b138-4baf-b6eb-0959ac201e90}", "networkConnected": true, "orderId": "4", </pre>			

		<pre> "paused": false, "performingAction": false, "speed": 0.0, "taskState": 1, "theta": 1.533207713764385, "trafficControl": false, "x": -10.89280653441715, "y": 7.005877121215736 } }, { "agvId": 2, "status": { "battery": 100, "confidence": 1.0, "currentVertex": 34, "subOrderId": "1fb2571a-3712-41d2-b2d3-14bca1e070c8", "networkConnected": true, "orderId": "3", "paused": false, "performingAction": false, "speed": 0.0, "taskState": 1, "theta": 4.186041095118258, "trafficControl": false, "x": -25.209119716735266, "y": -16.219175845798173 } }], "timestamp": "2023-10-24 21:05:25", "uuid": "1234" }</pre>
--	--	--

2.12 查询所有 AGV 的信息

接口名	QueryAllAgvsInfo				
功能说明	查询 AGV 的信息，信息包括 Status，同时还包括了一些 AGV 的属性				
提供方	MRSE 调度系统				
调用方	外部系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	AGV 编号	agvId	int		否
	对于是默认	fieldWithDefaultValueAreSetToEmpty	bool	对于是默认值的字段	是

	值的字段设置为空			返回时设置为空, 主要是为了减小应答包的体积大小, 节省宽带方便传输, 该字段可为空, 在为空的情况下, 默认值为 true	
应答	AgvInfo 数组		array	所有 Agv 信息的数组	否
示例	请求	Post: http://IP:PORT/MRSE/REST/QueryAllAgvsInfo { "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "fieldWithDefaultValueAreSetToEmpty": false } }			
	应答	{ "code": 0, "result": [{ "agvId": 1, "agvInfo": { "ipAddress": "0.0.0.1", "name": "", "status": { "battery": 100, "confidence": 1.0, "currentVertex": 71, "subOrderId": "288b032c-1424-4d93-b63e-d6c75370b605", "networkConnected": true, "orderId": "1", "paused": false, "performingAction": false, "speed": 0.0, "taskState": 1, "theta": 0.9, "trafficControl": false, "x": -33.350629, "y": 10.556995 } }, "type": "" }] }			

		<pre> }, { "agvId": 2, "agvInfo": { "ipAddress": "0.0.0.2", "name": "", "status": { "battery": 100, "confidence": 1.0, "currentVertex": 2, "subOrderId": "{f7e8c1e6-f331-4a78-9de6-0994d515eb42}", "networkConnected": true, "orderId": "2", "paused": false, "performingAction": false, "speed": 0.0, "taskState": 1, "theta": 0.0, "trafficControl": false, "x": -34.200629, "y": 6.556995 }, "type": "" } }], "timestamp": "2023-10-24 21:16:22", "uuid": "1234" }</pre>
--	--	--

2.13 请求为 AGV 充电

接口名	RequestGoCharging				
功能说明	请求为 AGV 充电，充电是由调度系统和任务系统共同决策的，调用此接口后，AGV 不一定会立即去充电，如果它当前有正在执行的订单的话，它会等到当前订单执行完了再去充电				
提供方	MRSE 调度系统				
调用方	任务系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	AGV 编号	agvId	int	要去充电的 AGV	否
	充电桩编号	chargerId	int	为 AGV 指定充电桩充电，如果值为 0，则说明不指定充电桩，交由调度系统自身去选择充电桩，此字段可为空，在为空的	是

				情况下默认值为 0, 也就是不指定充电桩	
应答	立马去充电	goChargingNow	bool	如果为 true, 则说明 AGV 立马去充电了, 否则正在等待 AGV 的当前订单完成	否
	充电桩编号	chargerId	int	分配给 AGV 的充电桩的编号, 如果 AGV 没有立马去充电, 则此字段的值为 0	否
示例	请求	Post: http://IP:PORT/MRSE/REST/RequestGoCharging <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "agvId": 1, "chargerId": 0 } }</pre>			
	应答	<pre>{ "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0, "result": { "goChargingNow": true, "chargerId": 1 } }</pre>			

2.14 请求为 AGV 结束充电

接口名	RequestEndCharging				
功能说明	请求为 AGV 结束充电				
提供方	MRSE 调度系统				
调用方	任务系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	AGV 编号	agvId	int	要结束充电的 AGV	否
应答					
示例	请求	Post: http://IP:PORT/MRSE/REST/RequestEndCharging <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "agvId": 1 } }</pre>			
	应答				

		<pre> }</pre>
	应答	<pre> { "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0 }</pre>

2.15 暂停 AGV

接口名	PauseAgv				
功能说明	暂停 AGV				
提供方	MRSE 调度系统				
调用方	任务系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	AGV 编号	agvId	int		否
应答					
示例	请求	Post: http://IP:PORT/MRSE/REST/PauseAgv <pre> { "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "agvId" : 1 } }</pre>			
	应答	<pre> { "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0 }</pre>			

2.16 恢复 AGV

接口名	ResumeAgv				
功能说明	恢复 AGV				
提供方	MRSE 调度系统				
调用方	任务系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	AGV 编号	agvId	int		否
应答					
示例	请求	Post: http://IP:PORT/MRSE/REST/ResumeAgv <pre> { "uuid": "1234",</pre>			

		<pre> "source": "WMS", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "agvId" : 1 } </pre>
	应答	<pre> { "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0 } </pre>

2.17 查询 AGV 的路线（剩余路线）

接口名	QueryAgvPath				
功能说明	查询 AGV 的剩余路径				
提供方	MRSE 调度系统				
调用方	任务系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	AGV 编号	agvId	int		否
应答	距离	distance	double	路线的长度，也就是距离	否
	WayPoint 数组	waypoints	数组	实际的路线，两个 WayPoint 之间表示一个路段	否
	节点编号	vertex	int		否
	节点 X 坐标	x	double		否
	节点 Y 坐标	y	double		否
	路段的行驶指令	edgeMotionInstruction	int	是个枚举值，0 停止，1 前进，2 后退，3 左侧移，4 右侧移，字段可为空，为空时表示默认值 1 前进	是
	行驶速度	speed	double	给 AGV 在这段路上的建议速度，实际速度不一定是这个	否
示例	请求	Post: http://IP:PORT/MRSE/REST/QueryAgvPath <pre> { "uuid": "1234", "source": "WMS", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "agvId" : 1 } } </pre>			
	应答	<pre> { </pre>			

		<pre>"code": 0, "result": { "distance": 10.2, "waypoints": [{ "edgeMotionInstruction": 2, "speed": 0.5, "vertex": 72, "x": -33.350629, "y": 6.556995 }, { "speed": 0.5, "vertex": 73, "x": -33.350629, "y": 2.706995 }, { "speed": 0.5, "vertex": 74, "x": -33.350629, "y": -1.743005 }, { "speed": 0.5, "vertex": 75, "x": -33.350629, "y": -6.043005 }, { "speed": 0.0, "vertex": 76, "x": -33.350629, "y": -9.793005 }] }, "timeStamp": "2024-12-12 10:23:52", "uuid": "1234" }</pre>
--	--	--

2.18 查询调度系统中的拓扑地图（点位地图）

接口名	QueryTopologicalMap				
功能说明	查询调度系统中的拓扑地图（由节点数组和边数组组成）				
提供方	任务系统				
调用方	MRSE 调度系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数					
应答	节点编号	id	int	节点编号	否
	节点 X 坐标	x	double	节点的 x 坐标	否
	节点 Y 坐标	y	double	节点的 y 坐标	否
	边的起始节点	fromId	int	边的起始节点 id	否
	边的结束节点	toId	int	边的结束节点 id	否
示例	请求	Post: http://IP:PORT/MRSE/REST/QueryTopologicalMap <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58" }</pre>			
	应答	<pre>{ "code": 0, "result": { "edges": [{ "fromId": 1, "toId": 2 }, { "fromId": 2, "toId": 1 }, { "fromId": 3, "toId": 2 }, { "fromId": 3, "toId": 5 }], "vertices": [</pre>			

		<pre>{ "id": 1, "x": -36.450629, "y": 10.556995 }, { "id": 2, "x": -36.450629, "y": 6.556995 }, { "id": 3, "x": -36.450629, "y": 2.756995 }, { "id": 4, "x": -36.450629, "y": -1.693005 }], "timeStamp": "2024-12-12 09:07:19", "uuid": "1234" }</pre>
--	--	---

3.1 调度系统推送订单开始执行的信号给外部系统

接口名	OrderStartExecuting				
功能说明	当订单开始被 AGV 执行后，将信号推送给外部系统				
提供方	任务系统				
调用方	MRSE 调度系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	订单编号	orderId	string		否
	AGV 编号	agvId	int	负责执行此订单的 AGV 编号	否
应答					
示例	请求	Post: http://IP:PORT/MRSE/REST/OrderStartExecuting { "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "orderId": "1", "agvId": 1 } }			

		<pre>} }</pre>
	应答	<pre>{ "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0 }</pre>

3.2 调度系统推送订单完成的信号给外部系统

接口名	OrderCompleted				
功能说明	当订单被完成后，主动推送信号给外部系统				
提供方	任务系统				
调用方	MRSE 调度系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	订单编号	orderId	string		否
	AGV 编号	agvId	int	负责执行此订单的 AGV 编号	否
应答					
示例	请求	Post: http://IP:PORT/MRSE/REST/OrderCompleted <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "orderId": "1", "agvId": 1 } }</pre>			
	应答	<pre>{ "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0 }</pre>			

3.3 调度系统推送订单被取消的信号给外部系统

接口名	OrderCancelled				
功能说明	当订单被取消后，主动推送信号给外部系统				
提供方	外部系统				
调用方	MRSE 调度系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	订单编号	orderId	string	该子订单所属的订单编号	否
	子订单编号	subOrderId	string	被取消时正在执行的子订单的编号	否

	AGV 编号	agvId	int	负责执行此订单的 AGV 编号， 如果订单是在还未开始执行前 就被取消了，则编号为 0	否
应答					
示例	请求	Post: http://IP:PORT/MRSE/REST/OrderCancelled <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "orderId": "1", "subOrderId": "1", "agvId": 1 } }</pre>			
	应答	<pre>{ "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0 }</pre>			

3.4 调度系统推送子订单完成的信号给外部系统

接口名	SubOrderCompleted				
功能说明	当订单的某条子订单完成后，将信号推送给外部系统				
提供方	任务系统				
调用方	MRSE 调度系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	订单编号	orderId	string		否
	子订单编号	subOrderId	string		否
	AGV 编号	agvId	int	负责执行此订单的 AGV 编号	否
应答					
示例	请求	Post: http://IP:PORT/MRSE/REST/SubOrderCompleted <pre>{ "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "orderId": "1", "subOrderId": "1", "agvId": 1 } }</pre>			
	应答	<pre>{</pre>			

		<pre> "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0 } </pre>
--	--	--

3.5 调度系统推送请求确认信号

接口名	RequestConfirm				
功能说明	调度系统向任务系统申请获得确认信号				
提供方	任务系统				
调用方	MRSE 调度系统				
	输入参数中文名	输入参数英文名	输入参数格式	含义	是否可空
请求参数	订单编号	orderId	string		否
	子订单编号	subOrderId	string		否
	确认类型	confirmType	int	一共有 4 种确认类型, 分别是 1 子订单的 requireStartConfirm, 2 子订单的 requireEndConfirm, 3 动作的 requireStartConfirm, 4 动作的 requireEndConfirm	否
应答	确认	confirm	bool	如果为 true, 则表示外部系统同意 AGV 继续执行下一步, 否则 AGV 继续等待, 直到外部系统主动给调度系统发送确认信号, 任务系统可通过调用 ComfirmOrder 或者 ConfirmAgv 接口来主动给调度系统发送确认信号	否
示例	请求	Post: http://IP:PORT/MRSE/REST/RequestConfirm <pre> { "uuid": "1234", "version": "1.0.0", "timeStamp": "2023-10-21 10:59:58", "data": { "orderId": "1", "subOrderId": "1", "confirmType": 1 } } </pre>			
	应答	<pre> { "uuid": "1234", "timeStamp": "2023-10-21 10:59:58", "code": 0, "result": { "confirm": true } } </pre>			

		<div>}</div> <div>}</div>
--	--	---------------------------