Task Processing System API

Legal Information

Contents

Chapter 1 Overview

1.1 Introduction

This manual provides some open APIs designed on RESTful style for the third-party platform to connect to Task Processing System (TPS) and control Automatic Mobile Robots (AMRs) for warehouse scheduling. A typical application developed by these open APIs is also provided for reference.

The TPS is warehouse scheduling system used in the smart warehouse, it can generate and assign tasks to AMRs to carry the racks between workstations and storage sections according to task conditions. It can also monitor the task executing status to processing the exception situations in time, which makes sure the regular and accurate working of AMRs.

1.2 Update History

Summary of Changes in Version 3.1.4_May/2021

Version	Summary of Changes
Version 3.1.4_May/2021	1. Edited the request parameters of API ***getOutPod*** : added one parameter **agvTyp** (AMR type); deleted one parameter **sequence**.
	2. Extended the request parameters of API ***returnPod*** : added two parameters: **agvTyp** (AMR type) and **mapDataName** (specified inbound position).
	3. Added one API for closing or resuming the workstation: ***closeWorkstation*** .

Summary of Changes in Version 3.1_March/2020

Version	Summary of Changes
Version 3.1_March/2020	1. Edited the API absolute address to "/rcms/services/rest/hikTpsService/[apiName]".
	2. Added one URL of applying rack initialization task to AMR: ***initPod*** .
	3. Extended the request parameters of API ***getOutPod*** : added four parameters: **cacheStrategy** (strategy for rack outbound to buffer area), **taskTyp** (whether the task is CTU task), **podDir** (rack direction), and **podCode** (rack ID).

Version	Summary of Changes
	4. Extended the request parameters of API ***returnPod*** : added one parameter **podCode** (rack ID).
	5. Extended the request parameters of API ***getOutPodToWorkStationArea*** : added two parameters **podCode** (rack ID) and **podDir** (rack direction).

Summary of Changes in Version 2.5_July/2019

New document.

1.3 Notice

- When calling an API, make sure the reqCode (request IDs) in the request and response message are same.
- REST (REpresentational State Transfer) is a protocol design method which abstracts all information as the resources. The abstracted resources are marked by the uniform identifies, i.e., URI (Uniform Resource Identifiers) for simple and extendable management.

1.4 Suggestions to Third-Party Platform

Table 1-1 Suggestions to Third-Party Platform

Suggestion	Details
Develop the workstation client.	• Supports searching and executing work orders. • Outbound order: the rack arrival information should pop up on the pickup interface. • Inbound order: the rack arrival information should pop up for inbound scanning. • Counting order: the rack arrival information should pop up on the counting interface.
Recommend the bin for the inbound order.	Recommend the suitable rack for the inbound order according to the goods volume.
Recommend the area for the rack inbound.	Recommend the suitable area for the rack inbound.

Chapter 2 API Description

2.1 API Format

The APIs in this manual are all in URL format, which defines and provides a unique address for resources to access and implement different functions.

The detailed API format definition is shown below:

```
<protocol>://<address>[:port][abs_path]
```

protocol

Protocol type that designing APIs based on, in this manual, the protocol type is "http".

address

Domain name or IP address of network device.

port

Port No. of web server, the default port No. is 8181.

abs_path

An absolute address to define a resource, you can connect to and operate the resource via this address.

For TPS, the absolute address of its resources is "/rcms/services/rest/hikTpsService/[**apiName**]".

For the third-party platform, the absolute address of its resources is "/xxx/[**apiName**]".

Note

- The **apiName** is used to distinguish the resources and functions, such as "getOutPod", whose function is rack outbound.
- To simplify the description in this manual, we use **apiName** to replace the complete API format. For example, replace "http://10.11.12.13:80/rcms/services/rest/hikTpsService/getOutPod" by "getOutPod" (rack outbound); replace "http://10.11.12.13:80/xxx/notifyPodArr" by "notifyPodArr" (notify inbound status). So when you call APIs according to this manual, you must add other fields before the **apiName** to complete the API format.

2.2 Operation Method

To implement different functions of resources represented by each API, operation method is required. As the APIs in this manual is designed based on HTTP, the operation methods are same as that supported by HTTP.

Method	Description
POST	Create or add resources.
GET	Search or get resources.
PUT	Update or set resources.
DELETE	Delete resources.

Note

In this manual, only the POST operation method is available.

2.3 Message Format

During the development based on the open APIs, the request and response message for communication and interaction is in JSON format, and the fields in the message are named by lower camel case.

JSON format is a subset of JAVA script, which is a lightweight data format, and this format can be quickly parsed. See the example below.

```
{
  "code": "0",
  "data": "F01169C808C317111G",
  "message": "successful",
  "reqCode": "468513"
}
```

2.4 Others

Time Format

The time appeared in the interaction between device and system adopts ISO8601 format, that is, "YYYY-MM-DD hh:mm:ss". For example, 2019-06-01 08:30:00.

Error Processing

When calling the open APIs, if error occurs, the response message will directly return the error code, you can get the error description and reason according to the returned response message. See ***Status Code*** (Appendix. A) for detailed error codes and description.

Chapter 3 Security

3.1 Authentication

The authentication of the open API is based on token (**tokenCode**) transmitted during request and response. The token is a string generated by Hikrobot system and will be transmitted to the third-party platform for authentication when calling APIs.

Chapter 4 Business Process

This chapter introduces the basic business processes of rack outbound, rack inbound, ending tasks, and applying for available racks.

4.1 Rack Outbound

There are two rack outbound modes: carry rack to the target workstation, carry rack to a free workstation of the specified area.

Carry Rack to Specified Workstation

- When the third-party platform calls the API ***getOutPod*** , the command will be applied to an AMR via TPS, the AMR will carry the rack to the specified workstation, so the workstation ID (**wbCode**) is required.
- When the AMR arrives at the workstation, TPS will call the API ***notifyClient*** to notify the third-party platform.
- When multiple AMRs carry racks to the same workstation, only one AMR can be at the workstation and others should queue in the buffer area, that is only when the AMR in the workstation leaves, the next AMR can move to the workstation. So the parameter **liftStatus** of API ***getOutPod*** cannot be set to 2 (puts down the rack at the workstation and leaves), otherwise the AMR in the buffer area may move to the workstation and crash the rack at the workstation.
- If bins of multiple tasks are in the same direction of a same rack:
  TPS will send the rack status of all these tasks to the third-party platform when the rack arrives at the workstation, so pickup tasks of these bins can be performed together.
- If bins of multiple tasks are in different directions of a same rack:
  When pickup tasks in one direction are completed, the AMR will leave the workstation, rotate the rack to make the direction with tasks facing the workstation, and carry the rack to the workstation for performing other pickup tasks.
- When the pickup tasks of the rack are completed, the third-party platform calls the API ***returnPod*** , the rack inbound command will be applied to an AMR via TPS for carrying the rack back to the specified storage section.
- When the AMR arrives at the storage section, TPS will call the API ***notifyPodArr*** to notify the third-party platform.

Carry Rack to Free Workstation of Specified Area

- In this mode, the rack outbound tasks can be implemented at any workstation of the area, so the workstation ID (**wbCode**) is optional. The AMR carries the rack to the free workstation, puts down the rack, and comes into idle status.
- When the rack tasks are completed, the third-party platform will call API ***returnPod*** , the rack inbound command will be applied to an AMR via TPS for carrying the rack back to the specified storage section.
- When the AMR arrives at the storage section, TPS will call the API ***notifyPodArr*** to notify the third-party platform.

4.2 Rack Inbound

When the third-party platform calls the API ***returnPod*** for rack inbound, only the inbound strategy needs to be specified. TPS will manage all storage sections according to areas and arrange the suitable storage section for rack inbound.

4.3 End Tasks of Workstation

The third-party platform calls the API ***endAllTasksByTps*** for ending all tasks of the workstation. If the rack is not carried by an AMR, the task will be canceled directly and the AMR comes into idle status; if the rack is carried by an AMR, the task will be canceled and the AMR waits for commands in-place.

TPS calls the API ***podReturnArea*** for specifying the inbound strategy and then applies the rack inbound command to AMR.

Note

The rack inbound command is initiated by TPS without the third-party platform's calling.

When the AMR arrives at the storage section, TPS will call the API ***notifyPodArr*** to notify the third-party platform.

4.4 Apply for Available Racks

Inbound in loop requires that, there are always available racks on the workstation. When an AMR carries the rack away from the workstation, TPS will call the API ***applyForEmptyPod*** to apply for an available rack each time. When the application succeeded, an AMR will carry the available rack to the workstation.

Chapter 5 Typical Application

5.1 Rack Outbound and Inbound

In this application scene, the AMR carries racks out from the warehouse to a specified workstation for further process, and then the rack is carried back to the warehouse according to the configured task.

Steps

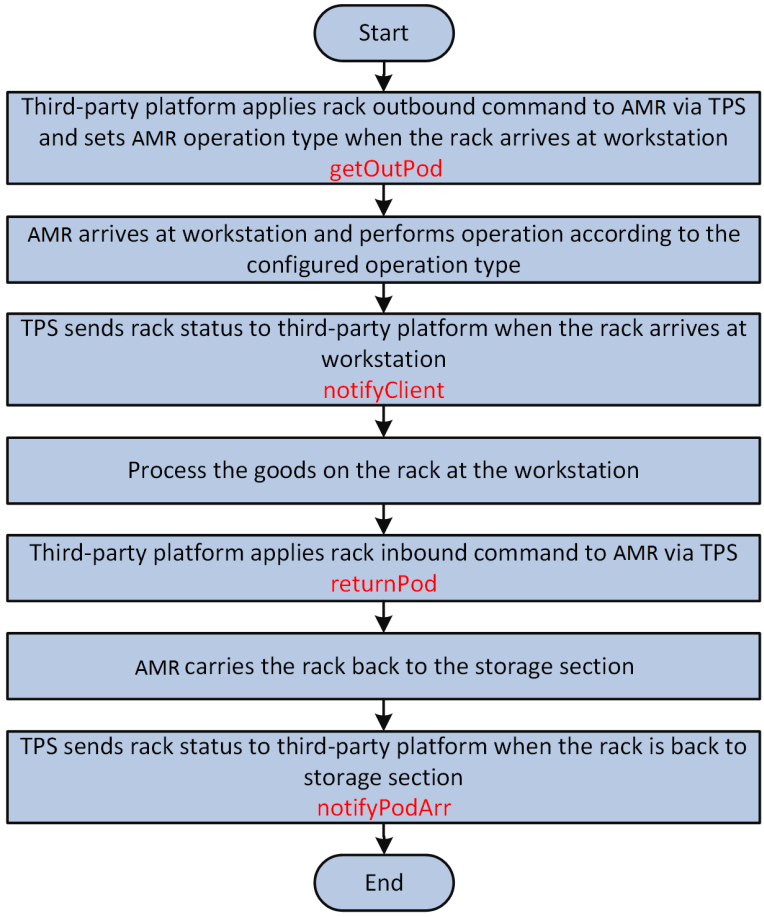

Figure 5-1 Flow of Rack Outbound and Inbound

1. The third-party platform calls ***getOutPod*** to apply rack outbound command to AMR via TPS and set AMR operation type (**liftStatus**) to "1", "2", or other value when the rack arrives at workstation.
2. The AMR arrives at workstation and performs one of the following operations according to the configured operation type.
   - **liftStatus** equals to "1": Puts down the rack at the workstation and waits nearby for inbound.
   - **liftStatus** equals to "2": Puts down the rack at the workstation and leaves.

- **liftStatus** is not configured or equals to other value: Carries with the rack at the workstation and waits for inbound.

3. The TPS calls ***notifyClient*** to send rack status to the third-party platform when the rack arrives at workstation.
4. Process the goods on the rack at the workstation.
5. The third-party platform calls ***returnPod*** to apply rack inbound command to AMR via TPS.

Note

- If the value of **liftStatus** is set to "2" when setting rack outbound parameters, the **taskTyp** of rack inbound parameters must be set to "4", which indicates that automatic inbound is enabled, and an optimal AMR will automatically go to the workstation and carry the rack back to the storage section.
- The values of **taskCode** in outbound and inbound parameters must be same.

6. AMR carries the rack back to the storage section.
7. The TPS calls ***notifyPodArr*** to send rack status to the third-party platform when the rack is back to storage section.

Chapter 6 API Reference

6.1 API List

List of Commonly Used APIs

Function	API Name	Provider
Apply rack outbound command to AMR via TPS.	***getOutPod***	TPS
Apply rack inbound command to AMR via TPS.	***returnPod***	TPS
End all tasks according to workstation ID.	***endAllTasksByTps***	TPS
TPS sends the rack status to the third-party platform under a specified condition.	***notifyClient***	Third-party Platform
Specify the rack inbound strategy according to material information of the rack.	***podReturnArea***	Third-party Platform

List of Optional APIs

Function	API Name	Provider
Apply rack initialization task to AMR.	***initPod***	TPS
Change storage section of rack.	***genMoveTaskByPod***	TPS
Search for storage section information according to rack ID.	***getBerthInfoByPodCode***	TPS
Carry rack to a specified storage section of a specified area.	***getOutPodToWorkStationArea***	TPS
Rotate the rack according to workstation ID and bin ID.	***rotatePodByTps***	TPS
Cancel task by task ID.	***endBinTasks***	TPS
Exchange storage sections for racks according to storage section ID or bin ID.		TPS
Unbind the rack and storage section by rack ID, and the rack becomes uninitialized.	***clearPodInfo***	TPS
Close or resume the workstation.	***closeWorkstation***	TPS

Function	API Name	Provider
Apply to the third-party platform for available racks.	***applyForEmptyPod***	Third-party Platform
TPS sends the rack status to the third-party platform when the rack is back to storage section.	***notifyPodArr***	Third-party Platform

6.2 API Provided by TPS

6.2.1 getOutPod

Implement rack outbound and set AMR operation type (**liftStatus**) when the rack arrives at the workstation. The TPS will apply the rack outbound command to an AMR.

API Definition

Table 6-1 POST http://[address][:port]/rcms/services/rest/hikTpsService/getOutPod

API Name	getOutPod
Function	Implement rack outbound and set AMR operation type (**liftStatus**) when the rack arrives at the workstation.
Protocol	REST
Provider	TPS
Caller	Third-party platform
Remarks	• If bins of multiple tasks are in the same direction of a same rack: TPS will send the rack status of all these tasks to the third-party platform when the rack arrives at the workstation, so pickup tasks of these bins can be performed together. • If bins of multiple tasks are in different directions of a same rack: When pickup tasks in one direction are completed, the AMR will leave the workstation, rotate the rack to make the direction with tasks facing the workstation, and carry the rack to the workstation for performing other pickup tasks. • One of **podCode** (**podDir**) and **binCode** should be configured, **podCode** and **podDir** are in priority. • The rack outbound tasks are executed according to task ID, so the **groupId** must be applied in order (from small to large).

Request	Parameter	Data Type	Max. Length	Req. or Opt.	Description
	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform
	clientCode	String	16	Opt.	Third-party platform ID, e.g., PDA, HCWMS, and it is provided by TPS for third-party platform
	tokenCode	String	64	Opt.	Token ID, it is provided by TPS for third-party platform
	cacheStrategy	String	32	Opt.	Strategy for rack outbound to buffer area
	taskTyp	String	2	Req.	Whether it is CTU task: "0"-non CTU task, "1"-CTU task
	data	List **Note** Up to 1000 fields are allowed in the list.			
	taskTyp	String	64	Req.	Task ID
	binCode	String	32	Opt.	Bin ID **Note** One of **podCode** (**podDir**) and **binCode** should be configured, **podCode** and **podDir** are in priority.
	podCode	String	16	Opt.	Rack ID **Note** One of **podCode** (**podDir**) and **binCode** should be configured, **podCode** and **podDir** are in priority.
	podDir	String	2	Opt.	Rack directions

					Note One of **podCode** (**podDir**) and **binCode** should be configured, **podCode** and **podDir** are in priority.
	agvTyp	String	16	Opt.	AMR type
	priority	String	3	Opt.	Task priority, range: [1,127], the larger the value, the higher the priority
	wbCode	String	32	Req.	Workstation ID
	groupId	String	8	Opt.	Task group No., which only consists of digits
	pickTime	String	8	Opt.	Pickup time, unit: second
	liftStatus	String	2	Opt.	AMR operation type when arrived at the workstation: "0"-AMR continues lifting the rack at the workstation and waits for inbound, "1"-AMR puts down the rack at the workstation and waits nearby for inbound, "2"-AMR puts down the rack at the workstation and it will be freed for executing a new task; by default, this node is not configured or its value is "0"
Response	code	String	6	Req.	Status code, see ***Status Code*** (Appendix B) for details.
	message	String	64	Req.	Returned status description, e.g., "successful"
	reqCode	String	64	Req.	Request ID, which is the same with that in corresponding request message
	data	String	2000	Opt.	Custom content to be returned
Sample	**Request**				

```
{
  "reqCode":"1541954B96B1112",
  "reqTime":"2018-08-08 10:30:00",
  "clientCode":"PDA",
  "tokenCode":"128654",
  "taskTyp": "0"
  "data":[{
    "taskCode":"000198",
    "binCode":"000020",
    "wbCode":"Workstation1",
    "priority":"3",
```

		<pre>"groupId":"23",
		"pickTime":"10",
		"liftStatus":"2"
		}]
		}</pre>

	Response	<pre>{
		"code": "0",
		"message": "successful",
		"reqCode": "1541954B96B1112"
		"data": "2131242341sdfs23"
		}</pre>

6.2.2 returnPod

Implement rack inbound. TPS will send the rack inbound command to an AMR.

API Definition

Table 6-2 POST http://[address][:port]/rcms/services/rest/hikTpsService/returnPod

API Name	returnPod				
Function	Implement rack inbound.				
Protocol	REST				
Provider	TPS				
Caller	Third-party platform				
Remarks	• When pickup tasks in one direction of the rack are completed, the rack inbound command can be applied. If there are other pickup tasks in the different direction of this rack, the AMR will leave the workstation, rotate the rack to make the direction with tasks facing the workstation, and carry the rack to the workstation for performing other pickup tasks. • One of **podCode** and **binCode** should be configured.				
Request	**Parameter**	**Data Type**	**Max. Length**	**Req. or Opt.**	**Description**
	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform

	clientCode	String	16	Opt.	Third-party platform ID, e.g., PDA, HCWMS, and it is provided by TPS for third-party platform
	tokenCode	String	64	Opt.	Token ID, it is provided by TPS for third-party platform
	taskCode	String	64	Req.	Task ID
	taskTyp	String	2	Opt.	Task type: "1"-initialize rack information for inbound (valid for first time inbound), "4"-automatic inbound (this value is valid only when **liftStatus** in getOutPod is "2"), other value or not configured-rack inbound (default, valid when **liftStatus**is "0", "1" or not configured).
	agvTyp	String	16	Opt.	AMR type
	mapDataName	String	32	Opt.	The specified return position
	returnPodStrategy	String	32	Req.	Inbound strategy ID, it specifies the area, which the rack should be carried back to
	binCode	String	32	Opt.	Storage bin ID, it is required when **taskTyp** is "1" or "4".
	podCode	String	16	Opt.	Rack ID
	wbCode	String	32	Opt.	Workstation ID, it is required when **taskTyp** is "1"
Response	code	String	6	Req.	Status code, see ***Status Code*** (Appendix B) for details.
	message	String	64	Req.	Returned status description, e.g., "successful".
	reqCode	String	64	Req.	Request ID, which is the same with that in corresponding request message
	data	String	2000	Opt.	Custom content to be returned
Sample	**Request**				

```
{
  "reqCode":"1541954B96B1112",
  "reqTime":"2018-08-08 10:30:00",
  "taskCode":"test169E0F39740116Q",
  "returnPodStrategy":"1211",
  "taskTyp":"1",
  "binCode":"00001",
```

		<pre>"podCode":"100001"
		"wbCode":"p02"
		}</pre>

	Response	<pre>"code": "0",
		"message": "successful",
		"reqCode": "1541954B96B1112",
		"data": "2131242341sdfs23"</pre>

6.2.3 initPod

Apply rack initialization task to AMR. AMR will detect the inbound rack No. and send to TPS.

API Definition

Table 6-3 POST http://[address][:port]/rcms/services/rest/hikTpsService/initPod

API Name	initPod				
Function	Apply rack initialization task to AMR. AMR will detect the inbound rack No. and send to TPS.				
Protocol	REST				
Provider	TPS				
Caller	Third-party platform				
Remarks	The rack initialization task can be completed by multiple AMRs.				
Request	**Parameter**	**Data Type**	**Max. Length**	**Req. or Opt.**	**Description**
	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform
	clientCode	String	16	Req.	Third-party platform ID, e.g., PDA, HCWMS, and it is provided by TPS for third-party platform
	tokenCode	String	64	Req.	Token ID, it is provided by TPS for third-party platform

	returnPodStrategy	String	32	Req.	Inbound strategy ID, it specifies the area, which the rack should be carried back to
	podNum	String	2	Req.	Number of racks to be initialized, range: [1,50]
	wbCode	String	32	Req.	Workstation ID
Response	code	String	6	Req.	Status code, see ***Status Code*** (Appendix B) for details.
	message	String	64	Req.	Returned status description, e.g., "successful".
	reqCode	String	64	Req.	Request ID, which is the same with that in corresponding request message
	data	String	2000	Opt.	Custom content to be returned
Sample	**Request**	`{ "reqCode":"468513", "reqTime":"2018-08-08 10:30:00", "clientCode":"PDA", "tokenCode":"128654", "returnPodStrategy":"1211", "podNum":"4", "wbCode":"p02" }`			
	Response	`{ "code": "0", "message": "successful", "reqCode": "1541954B96B1112", "data": "2131242341sdfs23" }`			

6.2.4 endAllTasksByTps

End all tasks according to workstation ID.

API Definition

Table 6-4 POST http://[address][:port]/rcms/services/rest/hikTpsService/endAllTasksByTps

API Name	endAllTasksByTps
Function	End all tasks according to workstation ID.
Protocol	REST
Provider	TPS
Caller	Third-party platform

Remarks	After calling the API, if the rack is not carried by an AMR, the task will be canceled directly and the AMR comes into idle status; if the rack is carried by an AMR, the task will be canceled and the AMR waits for commands in-place.				
Request	**Parameter**	**Data Type**	**Max. Length**	**Req. or Opt.**	**Description**
	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform
	clientCode	String	16	Opt.	Third-party platform ID, e.g., PDA, HCWMS, and it is provided by TPS for third-party platform
	tokenCode	String	64	Opt.	Token ID, it is provided by TPS for third-party platform
	taskTyp	String	2	Opt.	You can set it as "A" to end the workstation task.
	wbCode	String	32	Req.	Workstation ID
Response	code	String	6	Req.	Status code, see ***Status Code*** (Appendix B) for details.
	message	String	64	Req.	Returned status description, e.g., "successful"
	reqCode	String	64	Req.	Request ID, which is the same with that in corresponding request message
	data	String	2000	Opt.	Custom content to be returned
Sample	**Request**	`{ "reqCode":"1541954B96B1112", "reqTime":"2018-08-08 10:30:00", "clientCode":"PDA", "tokenCode":"128654", "wbCode":"Worksattion2" }`			
	Response	`{ "code": "0", "message": "successful", "reqCode": "1541954B96B1112" }`			

6.2.5 genMoveTaskByPod

Change storage section of rack.

API Definition

Table 6-5 POST http://[address][:port]/rcms/services/rest/hikTpsService/genMoveTaskByPod

API Name	genMoveTaskByPod				
Function	Change storage section of rack.				
Protocol	REST				
Provider	TPS				
Caller	Third-party platform				
Remarks	One of **areaTypCode**, **stgSecCode**, and **strategyCfgCode** must be configured. If all of them are configured, the **strategyCfgCode** is in priority. For the priority of **areaTypCode** and **stgSecCode**, it depends on the firstly transmitted parameter.				
Request	**Parameter**	**Data Type**	**Max. Length**	**Req. or Opt.**	**Description**
	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform
	clientCode	String	16	Opt.	Third-party platform ID, e.g., PDA, HCWMS; it is provided by TPS for third-party platform
	tokenCode	String	64	Opt.	Token ID, it is provided by TPS for third-party platform
	data	List			
	taskCode	String	64	Req.	Task ID
	areaTypCode	String	32	Opt.	Area ID
	stgSecCode	String	32	Opt.	Storage area ID
	strategyCfgCode	String	32	Req.	Strategy ID

	binCode	String	32	Req.	Bin ID
Response	code	String	6	Req.	Status code, see ***Status Code*** (Appendix B) for details.
	message	String	64	Req.	Returned status description, e.g., "successful".
	reqCode	String	64	Req.	Request ID, which is the same with that in corresponding request message
	data	String	2000	Opt.	Custom content to be returned
Sample	**Request**	`{ "reqCode":"1541954B96B1112", "reqTime":"2018-08-08 10:30:00", "clientCode":"PDA", "tokenCode":"128654", "data":[{ "taskCode":"000198", "areaTypCode":"000012", "stgSecCode":"000232", "strategyCfgCode":"2", "binCode":"0000010110202" }, { "taskCode":"000199", "areaTypCode":"000013", "stgSecCode":"000231", "strategyCfgCode":"2", "binCode":"0000010110203" }] }`			
	Response	`{ "code": "0", "message": "successful", "reqCode": "1541954B96B1112" }`			

6.2.6 getBerthInfoByPodCode

Search for storage section information according to rack ID.

API Definition

Table 6-6 POST http://[address][:port]/rcms/services/rest/hikTpsService/getBerthInfoByPodCode

API Name	getBerthInfoByPodCode				
Function	Search for storage section information according to rack ID.				
Protocol	REST				
Provider	TPS				
Caller	Third-party platform				
Request	**Parameter**	**Data Type**	**Max. Length**	**Req. or Opt.**	**Description**
	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform
	clientCode	String	16	Opt.	Third-party platform ID, e.g., PDA, HCWMS, and it is provided by TPS for third-party platform
	tokenCode	String	64	Opt.	Token ID, it is provided by TPS for third-party platform
	podCodes	List		Req.	Rack ID list
Response	code	String	6	Req.	Status code, see ***Status Code*** (Appendix B) for details.
	message	String	64	Req.	Returned status description, e.g., "successful".
	reqCode	String	64	Req.	Request ID, which is the same with that in corresponding request message
	data	List			
	mapDataCode	String			Storage section ID
	cooX	Float			X-coordinate of storage section
	cooY	Float			Y-coordinate of storage section

	status	Integer			Storage section status: "0"-enabled, "1"-disabled
	mapCode	String			Map ID
	podCode	String			Rack ID
	areaCode	String			Area ID
	stgSecCode	String			Storage section ID

Sample	**Request**	<pre>{
		"reqCode":"1541954B96B1112",
		"reqTime":"2018-08-08 10:30:00",
		"clientCode":"PDA",
		"tokenCode":"128654",
		"podCodes":["800001","900016"]
		}</pre>			

	Response	<pre>{
		"code":"0",
		"message": "successful",
		"reqCode": "1541954B96B1112",
		"data":[{
		"cooX":3000.0,
		"cooY":5000.0,
		"mapCode":"GC",
		"mapDataCode":"003000GC005000",
		"podCode":"800001",
		"status":1
		},
		{
		"cooX":4000.0,
		"cooY":5000.0,
		"mapCode":"GC",
		"mapDataCode":"003000GC005000",
		"podCode":"900016",
		"status":1
		}]
		}</pre>			

6.2.7 getOutPodToWorkStationArea

Carry rack to a workstation of the specified area.

API Definition

Table 6-7 POST http://[address][:port]/rcms/services/rest/hikTpsService/getOutPodToWorkStationArea

API Name	getOutPodToWorkStationArea				
Function	Carry rack to a workstation of the specified area.				
Protocol	REST				
Provider	TPS				
Caller	Third-party platform				
Request	**Parameter**	**Data Type**	**Max. Length**	**Req. or Opt.**	**Description**
	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform
	clientCode	String	16	Opt.	Third-party platform ID, e.g., PDA, HCWMS, and it is provided by TPS for third-party platform
	tokenCode	String	64	Opt.	Token ID, it is provided by TPS for third-party platform
	liftStatus	String	2	Opt.	AMR operation type when arrived at the workstation: "0"-AMR continues lifting the rack at the workstation and waits for inbound, "1"(default)-AMR puts down the rack at the workstation and waits nearby for inbound, "2"-AMR puts down the rack at the workstation and it will be freed for executing a new task
	data	List **Note** Up to 1000 fields are allowed in the list.			
	taskCode	String	64	Req.	Task ID

	binCode	String	32	Req.	Bin ID
	podCode	String	16	Req.	Rack ID
	podDir	String	2	Req.	Rack directions
	priority	String	3	Opt.	Task priority, range: [1,127], the larger the value, the higher the priority
	wbCode	String	32	Req.	ID of area, where workstations locate
Response	code	String	6	Req.	Status code, see ***Status Code*** (Appendix B) for details.
	message	String	64	Req.	Returned status description, e.g., "successful"
	reqCode	String	64	Req.	Request ID, which is the same with that in corresponding request message
	data	String	2000	Opt.	Custom content to be returned
Sample	**Request**	`{ "reqCode":"1541954B96B1112", "reqTime":"2018-08-08 10:30:00", "clientCode":"PDA", "tokenCode":"128654", "data":[{ "taskCode":"000198", "binCode":"0000010110101", "wbCode":"089765", "priority":"3" }] }`			
	Response	`{ "code": "0", "message": "successful", "reqCode": "1541954B96B1112" "data": "2131242341sdfs23" }`			

6.2.8 rotatePodByTps

Rotate the rack according to workstation ID and bin ID.

API Definition

Table 6-8 POST http://[address][:port]/rcms/services/rest/hikTpsService/rotatePodByTps

API Name	rotatePodByTps				
Function	Rotate the rack according to workstation ID and bin ID.				
Protocol	REST				
Provider	TPS				
Caller	Third-party platform				
Remarks	If both the **binCode** and **rotateDegree** are not configured, the AMR will rotate the rack for 180°; if both the **binCode** and **rotateDegree** are configured, the AMR will rotate the rack according to the bin ID, which indicates that **binCode** is in priority.				
Request	**Parameter**	**Data Type**	**Max. Length**	**Req. or Opt.**	**Description**
	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform
	clientCode	String	16	Opt.	Third-party platform ID, e.g., PDA, HCWMS, and it is provided by TPS for third-party platform
	tokenCode	String	64	Opt.	Token ID, it is provided by TPS for third-party platform
	wbCode	String	32	Opt.	Workstation ID
	binCode	String	32	Opt.	Storage bin ID
Response	code	String	6	Req.	Status code, see ***Status Code*** (Appendix B) for details.
	message	String	64	Req.	Returned status description, e.g., "successful".
	reqCode	String	64	Req.	Request ID, which is the same with that in corresponding request message
	data	String	2000	Opt.	Custom content to be returned
Sample	**Request**	`{ "reqCode":"1541954B96B1112",`			

		<pre>{
		  "wbCode":"p02",
		  "binCode":"00001"
		}</pre>

	Response	<pre>{
		  "code": "0",
		  "message": "successful",
		  "reqCode": "1541954B96B1112"
		}</pre>

6.2.9 endBinTasks

Cancel task by task ID.

API Definition

Table 6-9 POST http://[address][:port]/rcms/services/rest/hikTpsService/endBinTasks

API Name	endBinTasks				
Function	Cancel task by task ID.				
Protocol	REST				
Provider	TPS				
Caller	Third-party platform				
Request	**Parameter**	**Data Type**	**Max. Length**	**Req. or Opt.**	**Description**
	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform
	clientCode	String	16	Opt.	Third-party platform ID, e.g., PDA, HCWMS, and it is provided by TPS for third-party platform
	tokenCode	String	64	Opt.	Token ID, it is provided by TPS for third-party platform
	data	List			

		Note Up to 1000 fields are allowed in the list.			
	taskCode	String	64	Req.	Task ID
	binCode	String	32	Req.	Bin ID
Response	code	String	6	Req.	Status code, see ***Status Code*** (Appendix B) for details.
	message	String	64	Req.	Returned status description, e.g., "successful"
	reqCode	String	6	Req.	Request ID, which is the same with that in corresponding request message
	data	String	2000	Opt.	Custom content to be returned

Sample	**Request**	{
		"reqCode": "1541954B96B1112",
		"reqTime":"2018-08-08 10:30:00",
		"clientCode":"PDA",
		"tokenCode":"128654",
		"data":[{
		"taskCode":"000198",
		"binCode":"0000010110101"
		}]
		}			

	Response	{
		"code": "0",
		"message": "successful",
		"reqCode": "1541954B96B1112"
		"data": "2131242341sdfs23"
		}			

6.2.10 clearPodInfo

Unbind the rack and storage section by rack ID, and the rack becomes uninitialized.

API Definition

Table 6-10 POST http://[address][:port]/rcms/services/rest/hikTpsService/clearPodInfo

API Name	clearPodInfo
Function	Unbind the rack and storage section by rack ID, and the rack becomes uninitialized.
Protocol	REST

Provider	TPS				
Caller	Third-party platform				
Request	**Parameter**	**Data Type**	**Max. Length**	**Req. or Opt.**	**Description**
	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform
	clientCode	String	16	Opt.	Third-party platform ID, e.g., PDA, HCWMS, and it is provided by TPS for third-party platform
	tokenCode	String	64	Opt.	Token ID, it is provided by TPS for third-party platform
	podCode	String	16	Req.	Rack ID
Response	code	String	6	Req.	Status code, see ***Status Code*** (Appendix B) for details.
	message	String	64	Req.	Returned status description, e.g., "successful"
	reqCode	String	64	Req.	Request ID, which is the same with that in corresponding request message
	data	String	2000	Opt.	Custom content to be returned
Sample	**Request**	`{ "reqCode":"1541954B96B1112", "reqTime":"2018-08-08 10:30:00", "clientCode":"PDA", "tokenCode":"128654", "podCode":"000204" }`			
	Response	`{ "code": "0", "message": "successful", "reqCode": "1541954B96B1112" }`			

6.2.11 closeWorkstation

Close or resume the workstation. When a workstation is closed, the task in execution continues, queuing tasks suspend, and no task will be allocated to this workstation.

API Definition

Table 6-11 POST http://[address][:port]/rcms/services/rest/hikTpsService/closeWorkstation

API Name	closeWorkstation				
Function	Close or resume the workstation.				
Protocol	REST				
Provider	TPS				
Caller	Third-party platform				
Request	**Parameter**	**Data Type**	**Max. Length**	**Req. or Opt.**	**Description**
	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform
	clientCode	String	16	Opt.	Third-party platform ID, e.g., PDA, HCWMS, and it is provided by TPS for third-party platform
	tokenCode	String	64	Opt.	Token ID, it is provided by TPS for third-party platform
	wbCode	String	32	Req.	Workstation ID
	status	String	2	Req.	Close or resume the workstation: "1" (close), "0" (resume).
Response	code	String	6	Req.	Status code, see ***Status Code*** (Appendix B) for details.
	message	String	64	Req.	Returned status description, e.g., "successful".
	reqCode	String	64	Req.	Request ID, which is the same with that in corresponding request message
	data	String	2000	Opt.	Custom content to be returned

Sample	**Request**	{
		"reqCode":"1541954B96B1112",
		"reqTime":"2018-08-08 10:30:00",
		"wbCode":"p02",
		"status":"1"
		}

	Response	"code": "0", "message": "successful", "reqCode": "1541954B96B1112", "data": "2131242341sdfs23"

6.3 API Provided by Third-Party

6.3.1 notifyClient

TPS sends the rack status to the third-party platform under a specified condition.

API Definition

Table 6-12 POST http://[address][:port]/xxx/notifyClient

API Name	notifyClient				
Function	TPS sends the rack status to the third-party platform under a specified condition.				
Protocol	REST				
Provider	Third-party platform				
Caller	TPS				
Request	**Parameter**	**Data Type**	**Max. Length**	**Req. or Opt.**	**Description**
	method	String	64	Req.	API method name
	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform

	clientCode	String	16	Opt.	Third-party platform ID, e.g., PDA, HCWMS, and it is provided by TPS for third-party platform
	tokenCode	String	64	Opt.	Token ID, it is provided by TPS for third-party platform
	notifyTyp	String	2	Req.	Rack status to be notified: "1"-arrived at workstation, "2"-arrived at queue area
	wbCode	String	32	Req.	Workstation ID
	podCode	String	16	Opt.	Rack ID
	podTyp	String	16	Opt.	Rack type
	data	List			
	taskCode	String	64	Req.	Task ID
	binCode	String	16	Req.	Storage bin ID
Response	code	String	6	Req.	Status code, see ***Status Code*** (Appendix B) for details.
	message	String	64	Req.	Returned status description, e.g., "successful".
	reqCode	String	64	Req.	Request ID, which is the same with that in corresponding request message
	data	String	2000	Opt.	Custom content to be returned

Sample	**Request**	<pre>{
		"reqCode":"1541954B96B1112",
		"notifyTyp":"1",
		"wbCode":"p02",
		"data":[{
		"taskCode":"test169E0F39740116Q",
		"binCode":"00001"
		},{...}]
		}</pre>			

	Response	<pre>{
		"code": "0",
		"message": "successful",
		"reqCode": "1541954B96B1112"
		}</pre>			

6.3.2 podReturnArea

Specify the rack inbound strategy according to material information of the rack.

API Definition

Table 6-13 POST http://[address][:port]/xxx/podReturnArea

API Name	podReturnArea				
Function	Specify the rack inbound strategy according to material information of the rack.				
Protocol	REST				
Provider	Third-party platform				
Caller	TPS				
Request	**Parameter**	**Data Type**	**Max. Length**	**Req. or Opt.**	**Description**
	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform
	clientCode	String	16	Opt.	Third-party platform ID, e.g., PDA, HCWMS, and it is provided by TPS for third-party platform
	tokenCode	String	64	Opt.	Token ID, it is provided by TPS for third-party platform
	podCode	String	16	Req.	Rack ID
Response	code	String	6	Req.	Status code, see ***Status Code*** (Appendix B) for details.
	message	String	64	Req.	Returned status description, e.g., "successful".
	reqCode	String	64	Req.	Request ID, which is the same with that in corresponding request message
	data	String		Opt.	Inbound strategy ID
Sample	**Request**	`{ "reqCode":"1541954B96B1112", "podCode": "100001" }`			
	Response	`{ "code": "0",`			

		<pre>{
		  "message": "successful",
		  "data":"01",
		  "reqCode": "1541954B96B1112"
		}</pre>

6.3.3 applyForEmptyPod

Apply to the third-party platform for available racks.

API Definition

Table 6-14 POST http://[address][:port]/xxx/applyForEmptyPod

API Name	applyForEmptyPod				
Function	Apply to the third-party platform for available racks.				
Protocol	REST				
Provider	Third party platform				
Caller	TPS				
Request	**Parameter**	**Data Type**	**Max. Length**	**Req. or Opt.**	**Description**
	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform
	clientCode	String	16	Opt.	Third-party platform ID, e.g., PDA, HCWMS, and it is provided by TPS for third-party platform
	tokenCode	String	64	Opt.	Token ID, it is provided by TPS for third-party platform
	podTyp	String	32	Req.	Rack type
	podNum	String	32	Req.	Number of racks **Note** Now supports applying for only one rack each time.

	callTyp	String	32	Req.	Rack type to be applied for: "1"-loaded rack is allowed, "2"-loaded rack is not allowed
Response	code	String			Status code, see ***Status Code*** (Appendix B) for details.
	message	String			Returned status description, e.g., "successful"
	reqCode	String			Request ID, which is the same with that in corresponding request message
	data	String			Returned storage bin ID list

Sample	**Request**	{
		"reqCode":"1541954B96B1112",
		"podTyp":"1",
		"podNum":"4",
		"callTyp":"2"
		}			

	Response	{
		"code": "0",
		"message": "successful",
		"reqCode": "1541954B96B1112"
		}			

6.3.4 notifyPodArr

TPS sends the rack status to the third-party platform when the rack is back to storage section, and the third-party platform can unbind or bind rack and storage section.

API Definition

Table 6-15 POST http://[address][:port]/xxx/notifyPodArr

API Name	notifyPodArr				
Function	TPS sends the rack status to the third-party platform when the rack is back to storage section, and the third-party platform can unbind or bind rack and storage section.				
Protocol	REST				
Provider	Third-party platform				
Caller	TPS				
Request	**Parameter**	**Data Type**	**Max. Length**	**Req. or Opt.**	**Description**

	reqCode	String	32	Req.	Request ID, it is provided by third-party platform; if a same request is repeatedly submitted, the request ID must be same
	reqTime	String	20	Req.	Request timestamp, its format is "YYYY-MM-DD hh:mm:ss", and it is provided by third-party platform
	clientCode	String	16	Opt.	Third-party platform ID, e.g., PDA, HCWMS, and it is provided by TPS for third-party platform
	tokenCode	String	64	Opt.	Token ID, it is provided by TPS for third-party platform
	data	List			
	mapDataCode	String			Storage section ID
	cooX	Float			X-coordinate of storage section
	cooY	Float			Y-coordinate of storage section
	status	Integer			Storage section status: "0"-enabled, "1"-disabled
	mapCode	String			Map ID
	podCode	String			Rack ID
	areaCode	String			Area ID
	stgSecCode	String			Storage area ID
Response	code	String	6	Req.	Status code, see ***Status Code*** (Appendix B) for details.
	message	String	64	Req.	Returned status description, e.g., "successful".
	reqCode	String	64	Req.	Request ID, which is the same with that in corresponding request message
	data	String	2000	Opt.	Custom content to be returned
Sample	**Request**				

```
{
   "reqCode":"1541954B96B1112",
  "data":[{
    "cooX":10.0,
    "cooY":11.0,
    "status":1
```

		}] }

	Response	{
		"code": "0",
		"message": "successful",
		"reqCode": "1541954B96B1112"
		}

Chapter 7 General Messages

7.1 JSON_ResponseMsg

```
{
  "code": "",
/*required, string type, status code, value range: "0"-succeeded, "1"-failed
(incorrect parameter), "99"-unknown error*/
  "data": "",
/*required, string type, custom content to be returned, such as task ID*/
  "message": "",
/*required, string type, returned status description, e.g., "successful"*/
  "reqCode": ""
/*required, string type, request ID, which must be same with that in
corresponding request message*/
}
```

Appendix A. Storage Section, Rack, and Bin

In the warehouse, a storage section provides a location for rack to stay, and a rack is used to store the materials. Each rack contains one or multiple bins to classify the materials for storage.

Storage Section

The storage sections are marked on the warehouse map with different location codes or coordinates. You can locate the racks according to the storage section.

Rack

The rack can be classified into four types, refer to the figure below. The rack ID and direction are identified by the QR code under the rack.
The default rack ID contains 6 characters, and up to 16 characters are supported.

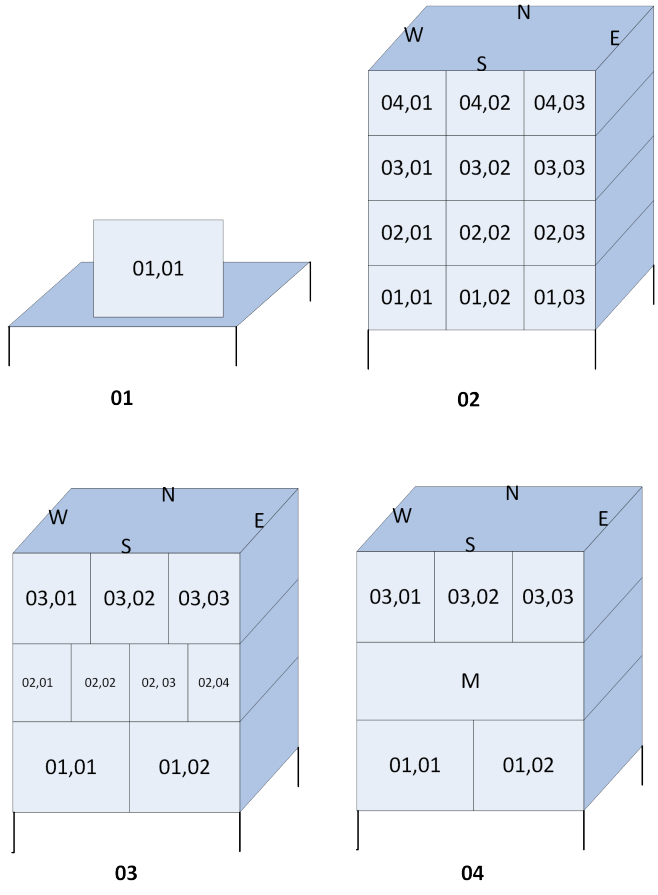

Figure A-1 Rack Types

Bin

The bin location is fixed when it is relative to the rack, so it can be numbered to quickly locate material location which is represented by ID, type, direction, floor No., and grid No. of rack.
The bin ID contains 13 characters, the meanings of the bits are shown as the below.

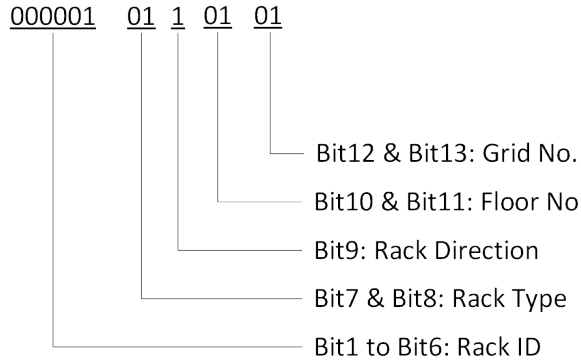

Figure A-2 Bin ID Description

For example, if the bin ID is "0000010110203", it indicates that the bin is on the grid No.3 and floor No.2 of eastward rack with type of 01 and ID of 000001.

Appendix B. Status Code

The status code returned in the response message are defined in the table below.

Status Code Description

code	Description
0	Succeeded.
1	Incorrect parameters.
6	No need to resend (the task of the same **reqCode** is not completed).
99	Unknown error, try again.
100	The task does not exist.


See Far, Go Further