Keenon Cloud API Development Document V2.2.0

1 Update History

Version	Updated	function points
v1.2.0	2022-05-19	Add webhook configuration
v1.3.0	2022-09-09	Add overseas access domain name
v1.4.0	2022-11-23	Increase the hotel cabin door control interface
v1.5.0	2023-1-16	Increase the hotel's remote dispatch interface
v1.6.0	2023-04-15	Increase the hotel machine delivery task callback document
v1.7.0	2023-08-22	1、Add map, point, machine real-time location interface 2、Correct documentation errors
V1.8.0	2024-01-05	1. Added the service channel for real-time status callback of remote and local food delivery tasks 2. Added interface for querying remote and local task status
V1.9.0	2024-12-11	1. Added query for cleaning robot schedule list 2. Added query for cleaning robot task records 3. Added query for robot status 4. Added query for robot cleaning area list

V2.0.0	2025-01-07	1. Added robot battery level query 2. Added remote command for cleaning robot to return to charging 3. Added remote command for cleaning robot to end task 4. Added remote command for cleaning robot to pause task
V2.1.0	2025-02-11	1. Added query for robot cleaning mode 2. Added query for robot return point 3. Added remote command for issuing temporary tasks 4. Added remote creation of scheduled tasks 5. Added remote modification of scheduled tasks 6. Added remote deletion, enabling, and disabling of scheduled tasks
V2.2.0	2025-02-20	1. Added query for business type list of scenarios 2. Added business type query field to the scenario ID query for point list 3. Added business type query to the call interface 4. Added map point information query v2

2 Access Notice

2.1 Some common business scenarios and corresponding interface

Scenario 1: Query robot task records

1. Obtain access token API: first exchange access token through ClientID and client secret ;
2. Get store list API: get storeId through access token ;
3. Obtain the food delivery/hotel task record API under the store according to the store: obtain the task record of the robot through the storeId and timestamp;

Scenario 2: Randomly call a robot in the store

1. Obtain the scene list API according to the store: obtain the scene sceneCode list through the storeId ;
2. Query point list API: get point uuid list through sceneCode ;
3. initiate remote call service through point uuid , pointId , storeId and sceneCode ;
4. If there are many different types of robots under the store, they can also be filtered by robotType ;

Scenario 3: Designate a robot for remote call control

1. Get the robot list API according to the store: get the robot list under the store through storeId ;
2. Query robot status API: Get the scene information currently used by the robot and the current callable status through the robotId ;
3. Query point list API: get point uuid list through sceneCode ;
4. Initiate remote call service API: initiate remote call service through point uuid , pointId , storeId , sceneCode and specified robotId ;

Precautions

The robot APP system has been continuously iteratively upgraded, and with each upgrade, more interface capabilities have been continuously opened for customers to use. The ensuing problem is that the open interfaces used by customers may not work

properly when running on low versions. In order to ensure the normal operation of the robot application, please pay attention to the APP version of the target machine during development, and select the applicable open interface according to the APP version. For details, see the interface compatibility description of each interface.
Frequently Asked Questions and Answers
Q1: Why is it possible to call the query robot status, but it fails to call the calling interface? A1: The remote call capability currently provided is only applicable to deployed robots. When this happens, you can consult the after-sales engineer to see if the robot deployment has been completed. Q2: Why is the scene information I queried through the robot status inconsistent with what is displayed on the robot app? A2: Generally, this kind of situation is caused by the instability of the robot network, and the scene is switched locally but failed to be reported to the cloud in time. You can pull the application scene again through the app. Q3: Why does the call status I query through the robot status always show as false? A3: If the robot call status is unavailable, you can check it according to the following steps:

1. Check whether the robot network is normal;
2. Check whether the remote call setting on the robot app is turned on;
3. Check if the robot is currently idle;
4. Check whether the current version of the robot meets the compatibility requirements;

2.2 Interface calling rules

1. Domestic Keenon production environment domain name

api_domain: https://console.peanut.keenonrobot.com

2. The domain name of the EU Keenon production environment

api_domain: https://es.robotkeenon.com

3. The domain name of Japan's Keenon production environment

api_domain: https://cloud.robotkeenon.com

Common noun description

parameter	illustrate
client_id	The unique Id of the client identification number in the open platform, which is generated internally by Keenon , does not support custom registration for the time being, and all resources of the open platform customer are mounted under ClientID
client_secret	key corresponding to the ClientID is generated synchronously by the open platform when the ClientID is generated. It needs to be kept by the customer properly. It does not support password retrieval. If you forget it, you need to contact Keenon to reset the password
access_token	When the client requests the open platform interface, it needs to exchange the client_id and client_secret for the access_token in advance , and the open platform interface can be accessed normally within the timeliness of the access_token . When the access_token becomes invalid, it can be refreshed again
robotId	A unique identifier for the robot's identity.
storeId	The unique ID of a store can be simply understood as the unique ID of an offline physical store on the cloud, and it is also the entrance of many resources
sceneCode	The unique Id of the scene, the concept of the scene is mainly used in the point configuration of the robot, and it is mainly a concept derived to solve the different configurations of points under a store.

General interface call description

- Request encoding: All request bodies must use UTF-8 encoding, and all responses will also use the same encoding;
- Request authentication: Keenon open interface only obtains Token-related interfaces without Token signature, and all other interfaces must carry token

authentication. The format is to add Authorization: bearer space access_token in the HTTP request header

- Interfaces or fields marked with a ~~strikethrough~~ in the following documents indicate that the interface or field is no longer maintained and will be deleted in subsequent version updates. It is not recommended to continue to use it. In order to ensure the normal invocation of services, generally such interfaces or fields will normally maintain 2-3 version iterations, and will be notified in advance if the interface is deprecated.

Suggestions for task class query interface

- The open platform provides customer-bound robot-related task records, but the customer's calls are subject to risk control restrictions , and frequent access will cause the IP or client_id to be disabled.
- It is recommended to use the callback mechanism to obtain the latest task status for calling tasks. If active query is selected, it is recommended that the query interval be around 3s
- The recommended query interval for robot local tasks is about 1H.

request return code

code	msg	meaning
610000	Request successful	successful request
610001	Wrong user name or password	Wrong username or password (corresponding to client_id and client_secret)
610401	Token verification failed	token is invalid or expired
610403	Insufficient operation permission	No permission to access interface or resource
610500	Server exception	An exception occurred during access
610601	Request parameter exception	Request parameter exception
617000	Request IP has been restricted	Request IP to trigger risk control

3 API interface list

3.1 Unified description of return body

Except for the token acquisition interface on the open platform, the other interfaces use the following Json format to return:
{"code": Integer, //The status code returned by the request, see the enumeration description of the request return code "msg": "String", // Request return message, see request return code enumeration description "data": Object/List<Object> //The specific data format is subject to the description of each interface. This field is only returned when the code is 610000}

3.2 Common interface (all business prerequisites)

This part of the interface is used to obtain the basic token, storeId , and robotId . For all business scenarios, you must first call this part of the interface to obtain basic authorization information. This part of the interface has no APP compatibility restrictions.

Get access_token

interface description

- Obtain the access token through this interface when the first request is initiated, and each subsequent valid request will extend the validity period of the token
- Repeated requests to this interface cannot reset the validity period of the token
- By default, the time limit for obtaining the token for the first time is 7200s, but it is not recommended to write it into the configuration or code as a constant, and it is best to dynamically parse it from the request result each time

request settings
Path: / api /open/ oauth /token
Method: POST
Headers

parameter name	parameter values	Required	example	Remark
Content-Type	application/x-www-form- urlencoded	yes		

Body

parameter name	Parameter Values	Required	example	Remark
client_id	String	yes		
client_secret	String	yes		
grant_type	String	yes	client_credentials	The parameter is a fixed value client_credentials

return data

name	type	Default Value	Remark	Additional Information
access_token	String		authorization token	
token_type	String	bearer	Token form, here is a fixed value bearer	
expires_in	Integer		The remaining valid time of the token is in seconds	

scope	String	all	Valid domain, this field is temporarily invalid	

Example

{
" access_token ": "009a8332-634e-4886-9c99-324caac55166",
" token_type ": "bearer",
" expires_in ": 6616,
"scope": "all"
}

Get store list

interface description

- This interface will return a list of all stores associated with the client_id
- If you need permission for a new store, please contact customer service staff

request settings
Path: / api /open/data/v1/store/list
Method: GET
Headers

parameter name	parameter value	Required	example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

return data

name	type	Default Value	Remark	Additional Information
code	Integer			
msg	String			
data	Object[ ]			
├─storeId	String		Store Id	
├─storeName	String		store name	
├─customerName	String		customer name	
├─brandName	String		brand name	
├─address	String		store address	
├─country	String		nation	
├─province	String		Province	

Example

```
{
"code": 610000,
"msg": "Request successful",

"data": [
{
" storeId ": "S00000001",
" storeName ": "No.1 store",
" customerName ": "Zhang San",
" brandName ": "Zhang San Group",
"address": "Shanghai Oriental Pearl Tower of Pudong New Area",
"country": "China",
"province": "Shanghai"
},
{
" storeId ": "S00000002",
" storeName ": "No. 2 store",
" customerName ": "Zhang San" ,
" brandName ": "Zhangsan Group",
"address": "Pearl of the East, Pudong New Area, Shanghai",
"country": "China",
"province": "Shanghai"
}
]
}
```

Obtain the robot list under the store through the store Id

request settings
Path: / api /open/data/v1/store/robot/list
Method: GET
Headers

parameter name	parameter value	Required	example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

parameter name	Parameter Values	Required	example	Remark

storeId	String	yes	S00000001	Store Id

return data

name	type	Default Value	Remark	Additional Information
code	Integer			
msg	String			
data	Object[]			
├─robotId	String		Robot Unique ID	
├─robotName	String		robot name	
├─onlineStatus	Integer		Online status 0: offline 1: online	
├─power	Integer		battery status	
├─robotModel	String		Robot model	This field is only returned by T-series robots above APP1.8.0
├─appVersion	String		Robot app version	
├─city	String		city	
├─onlineType	Integer		Online status: 2: wifi3: 3G4: 4g5: unknown	

Example

```
{
"code": 610000,
"msg": "Request successful",
"data": [
{
" robotId ": "AA:BB:CC:DD:EE:FF",
" robotName ": "Test Robot",
" onlineStatus ": 1,
"power": 66,
" robotModel ": " T2",
" appVersion ": "v1.4.4rc4-0-g0bc7c46",
"city": "No.49 Donglu Road, Pudong New Area, Shanghai, near Building 30, Jinling City",
" onlineType ": 2
}
]
}
```

3.3 Robot task recording interface

- This part of the interface is used to obtain the task record of the robot. Sub task query is currently not supported (that is, when a robot runs a parent task that includes multiple segmented sub tasks, the details of the sub tasks cannot be queried)

Obtain the food delivery task record under the store according to the store

request settings
Path: / api /open/data/v1/store/task/food/list
Method: GET
Headers

parameter name	parameter value	Required	example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

parameter name	Parameter Values	Required	example	Remark
storeId	String	yes		Store Id
page	Integer	no		page number, the default value is 1
size	Integer	no		Page size, the default is 20, the maximum value is 100
startTime	Long	no		Timestamp in milliseconds
endTime	Long	no		Timestamp in milliseconds

return data

name	type	Default Value	Remark	Additional Information
code	Integer			
msg	String			
data	Object[]			
├─total	Integer		total data	
├─list	Object[]			
├─├─robotId	String		Robot Unique ID	
├─├─startTime	String		Starting time	
├─├─endTime	String		End Time	
├─├─backTime	String		return time	
├─├─taskMileage	Integer		Mission running mileage, in meters	

├─├─storeId	String		Store Id	
├─├─taskMode	Integer		task type	0-Meal delivery 2-Direct delivery 3-Multi-order meal delivery 4-Snack
├─├─taskStatus	Integer		task completion	1: completed - 1: not completed

Interface Compatibility :

Applicable models	T series
Minimum APP version	Food delivery v1.4.0

Obtain the hotel task records under the store according to the store

request settings
Path: / api /open/data/v1/store/task/hotel/list
Method: GET
Headers

parameter name	parameter value	Required	example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

parameter name	Parameter Values	Required	example	Remark

storeId	String	yes		Store Id
page	Integer	no		page number, the default value is 1
size	Integer	no		Page size, the default is 20, the maximum value is 100
startTime	Long	no		Timestamp in milliseconds
endTime	Long	no		Timestamp in milliseconds

return data

name	type	Default Value	Remark	Additional Information
code	Integer			
msg	String			
data	Object[ ]			
├─total	Integer		total data	
├─list	Object[ ]			
├─├─robotId	String		Robot Unique ID	
├─├─startTime	String		Starting time	
├─├─endTime	String		End Time	
├─├─totalMileage	Double		Mission running mileage, in meters	
├─├─storeId	String		Store Id	

├─├─bizType	Integer		task type	0: room delivery; 1: general delivery; 2: dispatch; 3: takeaway; 4: return to origin; 5: container

Interface Compatibility :

Applicable models	W series
Minimum APP version	Hotel v1.4.0

3.4 Remote call service interface

This part of the interface is used to realize the remote call function of the robot. That is, the remote control robot reaches a specified position.

Obtain the scene list according to the store

request settings
Path: / api /open/scene/v1/info/list
Method: GET
Headers

parameter name	parameter value	Required	example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

parameter name	Parameter Values	Required	example	Remark
storeId	String	yes		Store Id

return data

name	type	Default Value	Remark	Additional Information
code	Integer			
msg	String			
data	Object[ ]			
├─sceneCode	String		Scene Unique ID	
├─sceneName	String		scene name	

Example

```
{
"code": 610000,
"msg": "success",
"data": {

"list": [
{
" sceneCode ": "dhads6",
" sceneName ": "Default scene"
}
]
}
```

}

Interface Compatibility:

Applicable models	T series	W series
Minimum APP version	Food delivery v1.8.0	Hotel v1.6.0

Query the point list according to the scene Id

Request setting

Path: / api /open/scene/v1/target/list

Method: GET

Headers

parameter name	parameter value	Required	example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

parameter name	Parameter Values	Required	example	Remark
sceneCode	String	yes		scene ID

return data

name	type	Default Value	Remark	Additional Information
code	Integer			

msg	String			
data	Object[ ]			
├─pointName	String		point name	
├─area	String		area name	
├─uuid	String		The unique identifier of the point	
├─pointId	String		Point Id	

Example

```
{
"code": 610000,
"msg": "success",
"data": {

"list": [
{
" pointName ": "Table No. 1",
"area": "Area A",
" uuid ": " abcdefg ",
" pointId ": "12"
}
]
}
}
```

Remarks: It is recommended to get it once a day

Interface Compatibility:

Applicable models	T series	W series

Minimum APP version	Food delivery v1.8.0	Hotel v1.6.0

Query the status of the robot

Request to set

Path: /api/open/scene/v1/robot/status

Method: GET

Headers

parameter name	parameter value	Required	example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

parameter name	Parameter Values	Required	example	Remark
robotId	String	yes		Robot Unique ID

return data

name	type	Default Value	Remark	Additional Information
code	Integer			
msg	String			
data	Object[ ]			
├─robotId	String		Robot Id	

├─onlineStatus	Boolean		online	
├─robotName	String		robot name	
├─canBeCalled	Boolean		Is it possible to call	
├─chargeStatus	Integer		charging	1: charging -1: discharging
├─power	Integer		electricity	
├─sceneCode	String		Current use scene identification	
├─sceneName	String		current scene name	

Example

```
{
"code": 610000,
"msg": "success",
"data": {

"list": [
{
" robotId ": "EE:DD:FF:GG:HH",
" onlineStatus ": true,
" robotName ": "food delivery robot",
" canBeCalled ": false,
" chargeStatus ": 1,
"power": 87,
" sceneCode ": "1gvEDF",
" sceneName ": "Test scene"
}
]
}
}
```

Interface compatibility:

Applicable models	T series	W series
Minimum APP version	Food delivery v1.8.0	Hotel v1.6.0

Initiate a remote call service

Request setting

Path: /api/open/scene/v3/robot/call/task

Method: POST

Headers

parameter name	parameter value	Required	example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Body

parameter name	Parameter Values	Required	example	Remark
uuid	String	yes		Point unique identifier Id
pointId	String	yes		Point Id
storeId	String	yes		Store Id

sceneCode	String	no		Scenario Id, passing in this value can improve the efficiency and success rate of calls
robotType	String	no		Robot type, if not filled, the default is food delivery food: food delivery robot hotel: hotel robot
robotId	String	no		Robot Unique ID
queuingTime	Integer	no		Queue scheduling time, in seconds

return data

name	type	Default Value	Remark	Additional Information
code	Integer			
msg	String			
data	object			
├─waitQueuing	Integer		queuing queue	
├─taskNo	String		TaskId	

Example

```
{
"code": 610000,
"msg": "success",
"data":{
"taskNo":"29b41cc5354f4e46b36a4abf1630b79e",
"waitQueuing":1
}
}
```

Initiates a remote call service for multi-point delivery

Note that to use this interface, you must be upgraded with a fixed version of the app or higher.

Request setting

Path: /api/open/scene/v4/robot/call/task

Method: POST

Headers

parameter name	parameter value	Required	example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Body

parameter name	parameter value	Required	example	Remark
uuid	String	Yes		Point Unique Identifier Id
pointId	String	Yes		
storeId	String	Yes		
sceneCode	String	Yes		

robotType	String	No		Robot type, default is food delivery. food: food delivery robot. hotel:hotel robot
robotId	String	Yes		
taskRequest	json	Yes		Multi-task information
-type	String	Yes		Task type: normalCallTask, fixed value. app Normal delivery call task, required field
-dishDetectionSwitch	Boolean	No		If or not support plate detection, default false
-extendedInfo	String	No		Extended fields, e.g., order information for third parties
-points	object[]	Yes		Multi-point distribution data
--uuid	String	Yes		Point Unique Identifier Id
--pointId	String	Yes		
--pointName	String	Yes		
--menus	object[]	Yes		Correspondence between plate layers and items
---sourceMenuId	String	No		Item and tray relationship id, 3rd-party system generation

---name	String	No		Name of the item, not required
---pictureUrl	String	No		Item image path, not required
---quantity	Integer	No		Number of items, not required
---price	Double	No		Price of item, not required
---layer	Integer	Yes		Meal layers, starting at 0. For example, 0: the first layer, which must have

Example

```
{
"uuid": "1swdwfeh45T545",
"pointId": "1",
"storeId": "S00802958",
"sceneCode": "yrewgh",
"robotType": "food",
"robotId": "0C:C6:55:4B:28:BD",
"taskRequest": {
"type": "normalCallTask",
"dishDetectionSwitch": false
"extendedInfo":""
"points": [
{
"uuid": "1",
"pointId": "1",
"pointName": "8",
"menus": [
{
```

```
"sourceMenuId": "11111",
"name": "",
"pictureUrl": "http:xxxx",
"quantity": 1,
"price": 12.3,
"layer": 0
},
{
"sourceMenuId": "11111",
"name": "",
"pictureUrl": "http:xxxx",
"quantity": 1,
"price": 12.3,
"layer": 1
}
]
},{
"uuid": "2",
"pointId": "2",
"pointName": "2",
"menus": [
{
"sourceMenuId": "11111",
"name": "",
"pictureUrl": "http:xxxx",
"quantity": 1,
"price": 12.3,
"layer": 0
}
]
}
]
}
```

}

return data

name	type	Default Value	Remark	Additional Information
code	Integer			
msg	String			
data	Object			
├─waitQueuing	Integer		queue	Designated robot call return 1
├─taskNo	String		task Id	

Example

```
{
"code": 610000,
"msg": "
success",
"data":{
"taskNo":"29b41cc5354f4e46b36a4abf1630b79e",
"waitQueuing":1
}
}
```

Error codes returned by creating a multipoint remote task

Error Code	Description
610403	Insufficient operation privileges
610401	token validation failure
610601	Request parameter exception
614920	No robot available
610609	Interface calls too frequent

Remote Call Service for Multi-Point Delivery

Note: To use this interface, the app must be upgraded to a fixed version or higher.

Request Settings

- Path：/api/open/scene/v4/robot/call/task
- Method： POST

Headers

Parameter Name	Parameter Value	Required	Example	Remarks
Authorization	bearer access_token	Yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Body

Parameter Name	Data Type	Required	Example	Remarks
uuid	String	Yes		Point Unique Identifier Id
pointId	String	Yes		Point Id
storeId	String	Yes		Store Id
sceneCode	String	Yes		Scene Id
robotType	String	No		Robot type, no fill default is food delivery. food:food delivery robot. hotel:hotel robot
robotId	String	Yes		Robot Sn
taskRequest	json	Yes		Multi-point task information
caseType	String	No		Business type of the point in the hybrid resource package
-type	String	Yes		Task type: normalCallTask, fixed value.
-dishDetectionSwitch	Boolean	No		app normalCallTask, fixed value. app normalDeliveryCallTask, required field
-extendedInfo	String	No		Whether to support plate detection, default false

Parameter Name	Data Type	Required	Example	Remarks
-points	object[]	Yes		Extended fields, such as, order information of the third party
--uuid	String	Yes		Multi-point Delivery Data
--pointId	String	Yes		Unique Identifier Id of the point
--pointName	String	Yes		Id of the point
--menus	object[]	Yes		Name of the point
---sourceMenuId	String	No		Corresponding relationship between plate layer and item
---name	String	No		Relationship between the item and the plate layer id, generated by the third-party system
---pictureUrl	String	No		Name of the item, can't be passed if there is no such scenario
---quantity	Integer	No		Picture Path of the item, can't be passed if there is no such scenario
---price	Double	No		Quantity of the item, can't be passed if there is no such scenario Price of the item, can't be

Parameter Name	Data Type	Required	Example	Remarks
				passed without this scenario
---layer	Integer	Yes		Meal level, start from 0, e.g., 0: the first level, must be passed.

Request Body Example

```
{
  "uuid": "1swdwfeh45T545",
  "pointId": "1",
  "storeId": "S00802958",
  "sceneCode": "yrewgh",
  "robotType": "food",
  "robotId": "0C:C6:55:4B:28:BD",
  "taskRequest": {
    "type": "normalCallTask",
    "dishDetectionSwitch": false,
    "extendedInfo": "",
    "points": [
      {
        "uuid": "1",
        "pointId": "1",
        "pointName": "8",
        "menus": [
          {
            "sourceMenuId": "11111",
            "name": "",
            "pictureUrl": "http:xxxx",
            "quantity": 1,
            "price": 12.3,
            "layer": 0
          },
          {
            "sourceMenuId": "11111",
            "name": "",
            "pictureUrl": "http:xxxx",
            "quantity": 1,
            "price": 12.3,
            "layer": 1
          }
        ]
      },
      {
        "uuid": "2",
        "pointId": "2",
        "pointName": "2",
        "menus": [
          {
            "sourceMenuId": "11111",
            "name": "",
            "pictureUrl": "http:xxxx",
            "quantity": 1,
            "price": 12.3,
            "layer": 0
          }
        ]
      }
    ]
  }
}
```

Response Data

Name	Type	Default	Remarks	Other Info
code	Integer			
msg	String			
data	Object			
├─waitQueuing	Integer		Queue position	(returns 1 if assigned to robot)
├─taskNo	String		Task ID	

Example

```
{
  "code": 610000,
  "msg": "success",
  "data": {
    "taskNo": "29b41cc5354f4e46b36a4abf1630b79e",
    "waitQueuing": 1
  }
}
```

Initiate the task of returning the robot to the origin

Request setting
Path: / api /open/scene/v2/robot/call/back/task
Method: POST

Headers

parameter name	parameter value	Required	example	Remark

Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

parameter name	Parameter Values	Required	example	Remark
storeId	String	yes		Store Id
robotId	String	no		Robot Unique ID

return data

name	type	Default Value	Remark	Additional Information
code	Integer			
msg	String			
data	object			
├─waitQueuing	Integer		queuing queue	
├─taskNo	String		TaskId	

Example

```
{
"code": 610000,
"msg": "success",
"data":{
"taskNo":"29b41cc5354f4e46b36a4abf1630b79e",
"waitQueuing":1
}
}
```

Interface compatibility:

Applicable models	T series
Minimum APP version	Food delivery v1.8.0

Actively query task status

Request to set
Path: / api /open/scene/v1/robot/call/task
Method: GET

Headers

parameter name	parameter value	Required	example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

parameter name	Parameter Values	Required	example	Remark
taskNo	String	yes		TaskId

return data

name	type	Default Value	Remark	Additional Information
code	Integer			
msg	String			
data	object			
├─taskNo	String		TaskId	
├─taskStatus	Integer		task status	0 - task failed 1 - task queued 2 - calling 3 - in progress 4 - task completed 5 - task canceled 6 - target point reached 7 - waiting

├─errorCode	Integer		robot error code	300, "The remote call function is not enabled on the robot side" 301, "The robot side is being set up" 302, "The robot side is executing a task and cannot be called" 303, "The robot side refuses to execute the task" 304, "There is no corresponding task on the robot side , unable to operate" 305, "The robot cannot recognize the target point" 400, "The robot manually canceled the task" 401, "The corresponding task record was not found on the robot" 402, "An unknown exception occurred on the robot and the task ended" 403, "The robot The terminal judged that the task timed out" 500, "The server did not find an available robot" 501, "The task has not been updated for a long time, and the task has ended in the background" 502, "The robot is offline, causing the task to not be updated for a long time" 503, "The robot has not registered to call the service"
├─remark	String		task notes	
├─deviceId	String		User ID	return client_id

├─taskData	object		task current data	
├─├─remainingDistance	Double		Remaining mileage in meters	
├─├─robotId	String		Robot Unique ID	
├─├─waitQueuing	Integer		current queue order	
├─├─lastUpdateTime	String		last update time	

Interface Compatibility:

Applicable models	T series	W series
Minimum APP version	Food delivery v1.8.0	Hotel v1.6.0

Cancel call task

Request to set

Path: / api /open/scene/v1/robot/call/task

Method: DELETE

Headers

parameter name	parameter value	Required	example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

parameter name	Parameter Values	Required	example	Remark
taskNo	String	yes		TaskId

return data

name	type	Default Value	Remark	Additional Information
code	Integer			
msg	String			

Interface Compatibility:

Applicable models	T series	W series

Minimum APP version	Food delivery v1.8.0	Hotel v1.6.0

Actively Acquiring the Status of Remote and local Meal Delivery Tasks

Request Settings
Path：/api/open/scene/v1/robot/task/info
Method： POST
Headers

Parameter Name	Parameter Value	Required	Example	Remarks
Authorization	bearer access_token	Yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

Parameter Name	Parameter Value	Required	Example	Remarks
taskNo	String	Yes		Task ID for Local or Remote Tasks id

return data

parameter name	Parameter Value	Required	Example	Remarks

bizType	String	Yes		RobotTaskState:Real-Time Task Status Type
clientId	String	Yes		
data	Object	Yes		
-robotSn	String	Yes		Robot SN
-taskNo	String	Yes		Task Number
-taskType	String	Yes		Task Type 0: Non-task, 1: Meal Delivery, 2: Multi-point, 3: Remote Task, 4: Remote Tray Return Task, 5: Snack, 6: Loop Tray Return, 7: To Dishwashing Area, 8: Welcoming, 9: Greeting, 10: Return to Base, 11: Charging, 12: Charging and Return Task, 13: Direct-to-Point
-taskState	String	Yes		Main Task Status 0: Failed, 1: Queuing, 2: Calling, 3: In Task, 4: Completed, 5: Task Cancelled
-taskNoType	Integer	Yes		Task Source 1: Remote Task, 2: Local Task

-errorCode	Integer	Yes		Error Codes (reported when the main task status is failure): 300: "Remote call function not activated on the robot"301: "Robot is currently being configured"302: "Robot is executing a task and cannot be called"303: "Robot refuses to execute the task"304: "No corresponding task on the robot, cannot operate"305: "Robot unable to recognize the target point"400: "Task manually cancelled by the robot"401: "Robot did not find a corresponding task record"402: "Robot encountered an unknown exception leading to task termination"403: "Robot determined the task to be overdue"500: "Server did not find an available robot"
-subTaskInfoList	Object[]	Yes		Sub-task Status Information
--uuid	String	Yes		Current Sub-task Location uuid
--pointId	String	Yes		Current Sub-task Location ID
--pointName	String	Yes		Current Sub-task Location Name
--type	Integer	Yes		Current Sub-task Location Type 1: Target Point, 2: Dishwashing Area, 3: Serving Area, 4: Replacement Point
--taskState	String	Yes		Current Sub-task Status 1: Waiting, 2: In Progress, 3: Paused, 4: Arrived at Destination, 5: Early Meal Collection, 6: Cancelled Task, 7: Task Anomaly

--taskDistance	String	Yes		Current Sub-task Distance (in meters)

Return data example

```
{
    "robotSn": "54:EF:33:CA:E4:FF",
    "taskNo": "Zdpbzv5LehGMAZ5h",
    "taskNoTypa": 1,
    "taskState": 3,
    "taskType": 1,
    "errorCode": 400,
    "extendedInfo": "",
    "subTaskInfoList": [
        {
            "uuid": "e23rt5ty",
            "pointId": 2,
            "pointName": "桌号 1",
            "type": 1,
            "taskState": 1,
            "taskDistance": "12.3"
        },
        {
            "uuid": "y2ftt5er",
            "pointId": 3,
            "pointName": "桌号 3",
            "type": 2,
            "taskState": 1,
            "taskDistance": "12.3"
        }
    ]
}
```

3.5 Robot map, point, real-time location interface

Get map data interface

Request to set
Path: api/open/custom/robot/map
Method: GET

Headers

parameter name	parameter value	Required	example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

parameter name	Parameter Type	Required	example	Remark
sceneCode	String	yes		Scene code
buildingInfo	String	no		Building information
floorInfo	String	yes		Floor information

return data

name	type	Default Value	example	Remark
code	Integer			
msg	String			

data	Object			
content	byte[]		Binary stream	

Example

```
{
"code": 610000,
  "msg": "suceess",
  "data": {
    content:[]
  }
}
```

Get map point information interface

Request to set
Path: /api/open/custom/robot/map/position
Method: GET

Headers

parameter name	**parameter value**	**Required**	**example**	**Remark**
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

parameter name	Parameter Type	Required	example	Remark
sceneCode	String	yes		Scene code
buildingInfo	String	no		Building information
floorInfo	String	yes		Floor information

return data

name	type	Default Value	example	Remark
code	Integer			
msg	String			
data	Object			
├─targetList	Object[]		Point information	
├─├─id	int		id	
├─├─targetId	int		Point id	
├─├─name	String		Point name	
├─├─type	String		Point type	
├─├─floor	String		floor	
├─├─phone	String		phone	
├─├─mapMd5	String		Point md5	

├─├─elevatorId	int		Elevator id	
├─├─positionX	Double		x-coordinate	
├─├─positionY	Double		y-coordinate	
├─├─positionZ	Double		z-coordinate	
├─├─orientationX	Double		x direction	
├─├─orientationY	Double		y direction	
├─├─orientationZ	Double		z direction	
├─├─orientationW	Double		w direction	
├─├─floorInfo	String		Floor information	
├─├─buildingInfo	String		Building information	

Example

```
{
    "code": 610000,
    "msg": "suceess",
    "data": {
        "targetList": [
            {
                "id": 343,
                "targetId": 11,
                "name": "",
                "type": "",
                "floor": 1,
                "phone": "13433",
                "mapMd5": "dfjdlkfajl",
                "elevatorId": 4343,
                "positionX": 2980.13,
                "positionY": 2124.43,
                "positionZ": 3.4343,
                "orientationX": 1.343,
                "orientationY": 3.434,
                "orientationZ": 5.434343,
                "orientationW": 3.4343556,
                "floorInfo": "",
                "buildingInfo": ""

            }
        ]
    }
}
```

PS: support C-series robot

Get map point information V2

Request setting

- Path：/api/open/custom/robot/v2/map/position
- Method： GET

Headers

parameter name	parameter value	Required	example	Remark
Authorization	bearer access_token	Yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

parameter name	Parameter Type	Required	example	Remark
sceneCode	String	Yes		Scenario code
buildingInfo	String	No		Building information
floorInfo	String	Yes		Floor information
caseType	String	Yes		Robot business type

Return Data

name	type	Default Value	example	Remark
code	Integer			
msg	String			
data	Object			
├─targetList	Object[]		Point information	
├─├─id	int		id	
├─├─targetId	int		point id	
├─├─name	String		point name	
├─├─type	String		point type	
├─├─floor	String		floor	
├─├─phone	String		phone	
├─├─mapMd5	String		point md5	
├─├─elevatorId	int		elevator id	
├─├─positionX	Double		x-coordinate	
├─├─positionY	Double		y-coordinate	
├─├─positionZ	Double		z-coordinate	
├─├─orientationX	Double		x-direction	
├─├─orientationY	Double		y-direction	
├─├─orientationZ	Double		z-direction	
├─├─orientationW	Double		w-direction	
├─├─floorInfo	String		floor information	

├─├─buildingInfo	String		building information	

Return Data Example

```
{
    "code": 610000,
    "msg": "suceess",
    "data": {
        "targetList": [
            {
                "id": 343,
                "targetId": 11,
                "name": "",
                "type": "",
                "floor": 1,
                "phone": "13433",
                "mapMd5": "dfjdlkfajl",
                "elevatorId": 4343,
                "positionX": 2980.13232,
                "positionY": 2124.43434,
                "positionZ": 3.4343,
                "orientationX": 1.343,
                "orientationY": 3.434,
                "orientationZ": 5.434343,
                "orientationW": 3.4343556,
                "floorInfo": "",
                "buildingInfo": ""
            }
        ]
    }
}
```

PS: Support C-series robot

Get the real-time location of the machine

Request to set
Path: /api/open/custom/robot/location
Method: GET

Headers

parameter name	parameter value	Required	example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

parameter name	Parameter Type	Required	example	Remark
robotSn	String	yes		robot sn

return data

name	type	Default Value	example	Remark
code	Integer			
msg	String			
data	Object			
├─building	String		Building information	
├─floor	String		Floor information	
├─coordinate	String		coordinate	

├─takeElevatorStatus	int		Riding condition	0：not use 1:　use

Example

```
{
    "msg": "Request successful",
    "code": 610000,
    "data": {
        "building": "",
        "floor": 1,
        "coordinate": "-8.22,12.07,-1.53",
        "takeElevatorStatus": 0
    }
}
```

Query the Battery Level of Robot

Request Setting

- Path：/api/open/custom/robot/battery/level
- Method：GET

Headers

parameter name	parameter value	Required	example	Remark

Authorization	bearer access_token	Yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

parameter name	Parameter Type	Required	example	Remark
robotSn	String	Yes		Robot sn

Return Data

name	type	Default Value	example	Remark
code	Integer			
msg	String			
data	Object			
├─batteryLevel	int		Battery level	

Return Data Example

```
{
    "msg": "Request successful",
    "code": 610000,
    "data": {
        "batteryLevel": 65
     }
}
```

3.6 Hotel robot interface

This part of the interface is used to call the relevant capabilities of the hotel robot.

Query the current position of the robot

Request to set
Path: / api /open/custom/robot/position
Method: GET
Param

name	Location	type	required	illustrate
storeId	query	string	true	Store Id
robotSn	query	string	true	Robot SN
Authorization	header	string	true	token, such as : bearer 7e5b85a4-68fe-46bc-8d48-fe7a13bbad0a

return example

```
{
"code": 610000,
"status": true,
"msg": "success",
"timestamp": 33623270873,
"data": {
"building": " building ",
"floor": " floor ",
" coordinate ": "2,20.0,3" ,

" takeElevatorStatus ": 0// Elevator status
}
}
```

return result

status code	status code meaning	illustrate	data model
200	OK	success	Inline

return data structure

status code 200

name	type	required	constraint	illustrate
» code	integer	true	none	Business status code, 610000-success
» status	boolean	true	none	Status, true-success, false-failure
» msg	string	true	none	prompt information

name	type	required	constraint	illustrate
» timestamp	number	true	none	timestamp
» data	object	true	none	return data
»» building	String[]	true	none	Buildings, multiple buildings are separated by ","
»» floor	string	true	none	floor
»» coordinate	string	true	none	position coordinates, (x, y, angle)
»» takeElevatorStatus	string	true	none	Elevator status

Interface Compatibility:

Applicable models	W series
Minimum APP version	Hotel v1.11.0

Synchronize robot current position

The background will send a command to the app to request to report the current position of the robot, and calling this interface will not immediately return the current position data of the robot.

Request setting
Path: / api /open/custom/robot/position
Method: POST

Param

name	Location	type	required	illustrate
storeId	query	string	true	Store Id
robotSn	query	string	true	Robot SN
Authorization	header	string	true	token, such as : bearer 7e5b85a4-68fe-46bc-8d48-fe7a13bbad0a

return example

```
{
"code": 610000,
"status": true,
"msg": "success",
"timestamp": 33623270873,
"data": true
}
```

return result

status code	status code meaning	illustrate	data model
200	OK	success	Inline

return data structure

status code 200

name	type	required	constraint	illustrate
» code	integer	true	none	Business status code, 610000-success
» status	boolean	true	none	Status, true-success, false-failure
» msg	string	true	none	prompt information
» timestamp	number	true	none	timestamp
» data	boolean	true	none	return data, true-success, false-failure

Interface Compatibility:

Applicable models	W series
Minimum APP version	Hotel v1.11.0

Query the status of the robot cabin

Request setting
Path: / api /open/custom/robot/cabin
Method: GET

Param

name	Location	type	required	illustrate
storeId	query	string	true	Store Id
robotSn	query	string	true	Robot SN
Authorization	header	string	true	token, such as : bearer 7e5b85a4-68fe-46bc-8d48-fe7a13bbad0a

return example

```
{
"code": 610000,
"status": true,
"msg": "success",
"timestamp": 33623270873,
"data": {

" cabinType ": 1, // cabin type
"cabins": [
{
"cabin": 1, // cabin number
" cabinDoorStatus ": " cabin door status ", // 1 - closed , 2 - open , 3 - closing ,
4 - opening, 5 - exception
}
]

}
}
```

return result

status code	status code meaning	illustrate	data model
200	OK	success	Inline

return data structure

status code 200

name	type	required	constraint	illustrate
» code	integer	true	none	Business status code, 610000-success
» status	boolean	true	none	Status, true-success, false-failure
» msg	string	true	none	prompt information
» timestamp	number	true	none	timestamp
» data	object	true	none	return data
»» cabinType	string	true	none	Cabin
»» cabins	[object]	true	none	cabin
»»» cabin	string	true	none	cabin number
»»» cabinDoorStatus	boolean	true	none	hatch status

Interface Compatibility:

Applicable models	W series
Minimum APP version	Hotel v1.11.0

Open/close robot hatch

The robot is not allowed to control the hatch during the upgrade, but can control the hatch at other times, but the hatch will be automatically closed immediately if it is opened during the movement;

To control the door switch, the correct input parameters must be combined with the cabin type. For example, if the robot is in a double cabin type , it can only accept 0 (upper cabin) and 2 (lower cabin) for hatch control. However, in the case of four cabins , 0, 1, 2, and 3 can be accepted for cabin door control.

This interface has a call frequency limit, and the same robot can only be called once within 10s;

Request setting
Path: / api /open/custom/robot/cabin/door
Method: POST
Param

name	Location	type	required	illustrate
storeId	the body	string	true	Store Id
robotSn	the body	string	true	Robot SN
cabin	the body	string	true	cabin number
ctrlType	the body	string	true	Door opening/closing, 0-closed, 1-opened

name	Location	type	required	illustrate
Authorization	header	string	true	token, such as : bearer 7e5b85a4-68fe-46bc-8d48-fe7a13bbad0a

return example

{
"code": 610000,
"status": true,
"msg": "success",
"timestamp": 33623270873,
"data": true
 	
}

return result

status code	status code meaning	illustrate	data model
200	OK	success	Inline

return data structure

status code 200

name	type	required	constraint	illustrate
» code	integer	true	none	Business status code, 610000-success
» status	boolean	true	none	Status, true-success, false-failure
» msg	string	true	none	prompt information
» timestamp	number	true	none	timestamp
» data	boolean	true	none	return data, success-true, failure-false

Interface Compatibility:

Applicable models	W series
Minimum APP version	Hotel v1.11.0

PUT switch outbound call switch configuration

After it is turned off, the robot will no longer perform text messages, internal calls, and external text messages/voice calls in the remote call and remote dispatch mode.

request settings
Path : /custom/robot/call/config/{ storeId }/{ robotSn }
Method : PUT

Param :

name	Location	type	required	illustrate
storeId	path	string	yes	Store Id
robotSn	path	string	yes	Robot SN
Authorization	header	string	yes	bearer token value

return example

```
{
 "code" : 610000 ,
 "msg" : "success" ,
 "data" : true
}
```

return result

status code	status code meaning	illustrate	data model
200	OK	success	Inline

return data structure

status code 200

name	type	required	constraint	Chinese name	illustrate
» code	integer	true	none		none
» msg	string	true	none		none
» data	boolean	true	none	whether to open	true-on, false-off, default true

Interface Compatibility:

Applicable models	W series
Minimum APP version	Hotel v1.12.0

GET Get outbound switch configuration

request settings
Path : /custom/robot/call/config/{ storeId }/{ robotSn }
Method : GET

Param :

name	Location	type	required	illustrate
storeId	path	string	yes	Store Id
robotSn	path	string	yes	Robot SN
Authorization	header	string	yes	bearer token value

return example

```
{
 "code" : 610000 ,
 "msg" : "success" ,
 "data" : true
}
```

return result

status code	status code meaning	illustrate	data model
200	OK	success	Inline

return data structure
status code 200

name	type	required	constraint	Chinese name	illustrate
» code	integer	true	none		none
» msg	string	true	none		none
» data	boolean	true	none	whether to open	true-on, false-off, default true

Interface Compatibility:

Applicable models	W series
Minimum APP version	Hotel v1.12.0

POST Create an order (dispatch)

When placing an order through this interface, the robot will first go to the drop-off point to wait for the user to put the object, and then go to the drop-off point to carry out the delivery task. When placing and picking up the object, enter the corresponding four-digit password (if any).

request settings
Path : / api /open/custom/order/create
Method : POST

Param :

name	Location	type	required	Chinese name	illustrate
Authorization	header	string	yes		bearer token value
the body	the body	object	no		none

name	Location	type	required	Chinese name	illustrate
» storeId	the body	string	yes	Store ID	none
» robotSn	the body	string	no	Robot SN	none
» mobile	the body	string	no	Phone number	No area code required
» endPoint	the body	string	yes	Delivery point Id	none
» endPointName	the body	string	yes	Delivery point name	none
» putCode	the body	string	no	put object code	
» takeCode	the body	string	no	Pick up code	
» startPoint	the body	string	no	Drop point Id	Do not fill in the default origin
» isShopOrder	the body	boolean	no	Is it a container order	true - is a container order, container orders are not currently supported

name	Location	type	required	Chinese name	illustrate
» remark	the body	string	no	Remark	none
» orderName	the body	string	no	order name	none
» goods	the body	[object]	no	commodity	Non-container orders can not be transmitted
»» merchandiseId	the body	integer	yes	Product ID	none
»» merchandiseNumber	the body	integer	yes	Number of Products	none

return example

```
{
 "code": 610000,
 "status": true,
 "msg": "success",
 "timestamp": 33623270873,
 "data": {
   "orderId": "16014290363287945",
   "orderQueue": [
     "16014290363287941",
     "16014290363287942"
  ],
   "orderStatus": "ORDER_PAY_SUCCESS",
   "orderStatusDesc": "to be delivered"
}}
```

return result

status code	status code meaning	illustrate	data model
200	OK	success	Inline

return data structure

name	type	required	constraint	illustrate
» code	integer	true	none	Business status code, 610000-success
» status	boolean	true	none	Status, true-success, false-failure
» msg	string	true	none	prompt information
» timestamp	number	true	none	timestamp
» data	object	true	none	return data
»» orderId	string	true	none	orderId
»» orderStatus	string	true	none	Order Status
»» orderStatusDesc	string	true	none	Order Status Description
»» orderQueue	[string]	true	none	List of queued orders (returns only the order number)

Interface Compatibility:

Applicable models	W series
Minimum APP version	Hotel v1.12.0

3.7 Hospital robot interface

Robot List Data Query Interface

Request Settings

- Path: /api/open/service/v1/data/robot/runtime/dashboard
- Method: POST

Headers

Parameter Name	Parameter Value	Required	Example	Remarks
Authorization	bearer access_token	Yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Response Data Example

```
{
 "status": true,
 "msg": "Success",
 "code": 610000,
 "timestamp": 1684227308220,
 "data": {
  "totalNum": 100,
  "todayReceiveTask": 15,
  "statistics": [{
   "robotModel": "X201",
   "numOfRobotModel": 123,
   "idle": 0,
   "busy": 1,
   "error": 3,
   "offline": 3,
   "robotList": [{
    "robotName": "",
    "battery": "",
    "position": "",
    "callStatus": "",
    "isOnline": ""
   }]
  }]
 }
}
```

Field Descriptions

Name	Type	Default	Remarks	Other Info
todayReceiveTask	int		Today's received task count	
totalNum	int		Total number of robots	
robotModel	String		Robot model	
numOfRobotModel	int		Quantity per model	
Idle	int		Idle count	
busy	int		Busy count	
error	int		Error count	
offline	int		Offline count	
robotList	array		Robot list	
robotName	String		Robot name	

Name	Type	Default	Remarks	Other Info
battery	int		Battery level	
position	String		Robot location	
callStatus	int		Call status: 0: Disabled 1: Available 2: Temporarily unavailable	
isOnline	String		Online status: 0: Offline 1: Online	

Task Template List Query Interface

Request Settings

- Path: `/api/open/service/v1/task/template/list`
- Method: `POST`

Headers

Parameter Name	Parameter Value	Required	Example	Remarks
Authorization	bearer access_token	Yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Body Parameters

Name	Location	Type	Required	Description
page	body	int	No	Page number (default: 1)
size	body	int	No	Items per page (default: 10)
templateName	body	string	No	Fuzzy match
templateId	body	int	No	Template ID

Response Data Example

```
{
 "status": true,
 "msg": "Success",
 "code": 610000,
 "timestamp": 1684227308220,
 "data": [{
  "taskTemplateVO": {
   "id": 229,
   "tenantId": "266809909706753",
   "templateName": "Specimen Delivery 333",
   "icon": "postoperativeInstrument",
   "taskGroupId": 208,
   "taskDescription": "Ward nurse initiates, robot collects specimens
from nurse station and delivers to lab",
   "taskType": "0",
   "accessUsers": null,
   "accessRobots": ["60:FB:00:96:94:65", "60:FB:00:91:2F:34"],
   "visibleRange": ["Test Group 2"],
   "optionalPoints": "",
   "isUsed": 1,
   "createUser": "285760498696193",
   "updateUser": "284491795136513",
   "createTime": 1683770823000,
   "updateTime": 1683770823000,
   "isDel": 0,
   "groupName": "Disabled"
  }
  }]
}
```

Field Descriptions

Name	Type	Default	Remarks	Other Info
id	int		Task template ID	
templateName	String		Task name	
icon	String		Icon	
taskGroupName	String		Task group	
taskDescription	String		Task description	
accessUsers	String		Authorized users	
accessRobots	String		Authorized robots (SNs comma-separated)	
visibleRange	String		Visible scope (user groups comma-separated)	
optionalPoints	String		Optional points	
isUsed	boolean		Published status: false: Disabled true: Enabled	
createUser	String		Creator	
updateUser	String		Updater	
createTime	String		Creation time	

Name	Type	Default	Remarks	Other Info
updateTime	String		Update time	
taskType	String		Task type: 0: Call 1: Loading/Unloading 2: Disinfection 3: Charging 4: Return	

Initiate Call Request Interface

Request Settings

- Path: /api/open/service/v1/task
- Method: POST

Headers

Parameter Name	Parameter Value	Required	Example	Remarks
Authorization	bearer access_token	Yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Body Parameters

Name	Location	Type	Required	Description
targetPoints	body	array	Yes	Target point information
id	body	int	Yes	Point ID
name	body	string	Yes	Point name
group	body	string	Yes	Point group
urgency	body	int	Yes	Urgency level: 0: Normal 1: Urgent
taskDescription	body	string	No	Task description
taskType	body	int	Yes	Task type: 0: Call 1: Loading/Unloading 2: Disinfection 3: Charging 4: Return
excutorType	body	int	Yes	Execution method: 0: Immediate 1: Scheduled
taskMode	body	string	Yes	Task mode: 0: SN-specified 1: Smart recommendation
robotName	body	string	Yes	Robot name
robotSn	body	string	Yes	Robot serial number

Name	Location	Type	Required	Description
attachMsg	body	json string	Yes	Additional info: {"orderInfo":[{"orderNo":"S2222222","deliveryPointId":"1"}]}
orderinfo	body	jsonArray	Yes	Additional order info
orderNo	body	string	Yes	
deliveryPointId	body	string	Yes	
orderNum	body	int	Yes	
orderName	body	string	Yes	
orderDetail	body	string	Yes	

Request Example

```
{
  "targetPoints": [{
    "id": "1",
    "name": "2",
    "group": "Group Name"
  }],
  "urgency": 0,
  "taskDescription": "Task description",
  "taskType": 0,
  "excutorType": 0,
  "taskTemplateId": "",
  "taskMode": "0",
  "robotSn": "50:13:95:B3:B7:FA"
}
```

Sub-Task Record Query Interface

Request Settings

- Path: /api/open/service/v1/subTask/list
- Method: POST

Headers

Parameter Name	Parameter Value	Required	Example	Remarks
Authorization	bearer access_token	Yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Body Parameters

Name	Location	Type	Required	Description
orderNo	body	string	Yes	Order number

Response Example

```
{
 "code": 610000,
 "status": true,
 "msg": "success",
 "timestamp": 33623270873,
 "data": {
  "id": 1,
  "taskId": 2,
  "taskUuid": "12312",
  "departId": 1,
  "departName": "",
  "startPower": 1,
  "startTime": 123213123123,
  "targetId": 1,
  "target": "123123",
  "endPower": 111,
  "endTime": 1231232,
  "mileage": 0.0,
  "duration": 123213123123,
  "waitTime": 123213123123,
  "status": 11,
  "taskType": 121,
  "doorId": 1,
  "reason": 1,
  "remark": "",
  "orderNo": ""
 }
}
```

Field Descriptions

Name	Type	Default	Remarks	Other Info
taskId	int		Sub-task ID	
taskUuid	String		Parent task UUID	
departId	int		Departure point code	
departName	String		Departure point name	
startPower	int		Initial battery level	

Name	Type	Default	Remarks	Other Info
startTime	long		Start time	
targetId	int		Destination code	
target	String		Destination name	
endPower	int		Arrival battery level	
endTime	long		Arrival time	
mileage	double		Delivery distance (km)	
duration	long		Delivery duration	
waitTime	long		Waiting time at destination	
status	int		Completion status: -1: Created 0: Completed 1: Failed 2: Canceled 3: Passive cancel 4: Delivery started 5: Arrived	
taskType	int		Task type: 1: Delivery 2: Return 3: Charging 11: Call 17: Disinfection 18: Loading/Unloading	
doorId	int		Compartment ID	
reason	int		Completion/failure reason: 101: No pickup	

Name	Type	Default	Remarks	Other Info
			102: Delivery timeout 103: Delivery timeout 104: App crash	
remark	String		Additional notes	
orderNo	String		Order number	

3.8 Cleaning robot interface

Query Cleaning Robot Scheduling List Interface

Path:/api/open/ custom/clean/strategy/list

Method: GET

Headers

Parameter name	Parameter Value	Is it necessary	Example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

name	type	Required	illustrate
storeId	Strring	yes	Store ID
robot Sn	Strring	no	Robot SN
cleanName	String	no	Task Name
currentPage	Integer	no	Page number, the default value is 1
pageSize	Integer	no	Page size, default is 20, maximum value is 100

Return data example

```
{
    "code": 610000,
"msg": "success",
    "data": {
        "count":5,
        "currentPage": 1,
     "pageSize": 20,
        "entities": [{
            "storeId": "C00000102",
            "robotSn": "34:7D:E4:98:A1:79",
            "id": "2d66b0d92c0b423fb8c6ce963eb86ef4",
      "cleanName": "任务 2",
"startTime": "11:31",
"endTime": "12:31",
"timeCleanType": 1,
"cycleType": "1",
"cleanTimes": 2,
"cleanModel": "吸尘",
"floorName": "1",
"areaName": "Area 343",

"c leanSpend ": 144,

" cleanArea ": 8

            }]

      }

}
```

name	type	default value	Remark	Additional Information
storeId	String		Store ID	
robotSn	String		Robot SN	
id	String		Task ID	
cleanName	String		Task Name	
startTime	String		Start time	Format hh:mm
endTime	String		End time	Format hh:mm
timeCleanType	Integer		Scheduled task type	1 quantitative 2 regular
cycleType	String		Frequency	0 once only, 1 recurring, 2 daily
cleanTimes	Integer		Cleaning frequency	
cleanModel	String		Cleaning Mode	
floorName	String		floor	

areaName	String		Cleaning area	
c leanSpend	Integer		Estimated cleaning time	Unit: Seconds
cleanArea	Integer		Cleaning area	

Query cleaning robot task record interface

Path:/api/open/ custom/clean/log/list

Method: GET

Headers

Parameter name	**Parameter Value**	**Is it necessary**	**Example**	**Remark**
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

name	type	Required	illustrate
storeId	Strring	yes	Store ID
robot Sn	Strring	no	Robot SN
currentPage	Integer	no	Page number, the default value is 1
pageSize	Integer	no	Page size, default is 20, maximum value is 100

Return data example

```
{
      "code": 610000,
"msg": "success",
      "data": {
            "count":5,
            "currentPage": 1,
       "pageSize": 20,
            "entities": [{
                  "uuid": "3f0df4f8b41949d28933478b132edc78",
                  "storeId": "C00000102",
                  "robotSn": "34:7D:E4:98:A1:79",
                  "strategyType": 16,
         "strategyName": "任务 2",
"targetMapName": "2131",
"mapNameList": ["区域 40461"],
"cleanArea": 5.34,
"cleanTiming": 230,
"cleanAreaTiming": 56,
" clean Efficiency": 56.2,
" startTime ": "2024-12-10 19:40:00",

"endTime": "2024-12-10 19:50:00",

"appVersion": "v3.5.3test2-0-g8556afe",

"rosVersion": "slam-ros-v2.8.0-c30testpathuploader1-0-g43ea2e8"        }]

      }

}
```

name	type	default value	Remark	Additional Information
uuid	String		Main task ID	
storeId	String		Store ID	
robotSn	String		Robot SN	
strategyType	Integer		Task Type	16 Scheduled tasks 17 Quantitative tasks 18 Immediate tasks
strategyName	String		Task Name	
targetMapName	String		Map Name	
mapNameList	Array		Cleaning area	
cleanArea	Double		Cleaning area	

clean efficiency	Double		Cleaning efficiency	
cleanAreaTiming	Double		Total working hours	Unit: Seconds
cleanTiming	Double		Cleaning time	Unit: Seconds
startTime	String		Start time	Format yyyy-mm-dd hh:mm:ss
endTime	String		End time	Format yyyy-mm-dd hh:mm:ss
appVersion	String		App Version	
rosVersion	String		ros version	

Cleaning robot status query interface

Path:/api/open/ custom/clean/robot/status

Method: GET

Headers

Parameter name	Parameter Value	Is it necessary	Example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

name	type	Required	illustrate
robot Sn	Strring	yes	Robot SN

Return data example

```
{
      "code": 610000,
"msg": "success",
      "data": {
"childState": {
"lifting": false,
        "navigating": false
    },
    "globalState": {
        "faulting": false,
        "locationSuc": true,
        "lock": true,
        "rosConnect": true,
        "scheduling": false,
        "scram": false,
        "upgrading": false
    },
    "hardwareState": {
        "armrests": 1,
        "bilgeTankState": -1,

        "cleanWaterTank": -1,

        "dustBag": 1,

        "rollingBrushPushRod": 0

},

"mainState": 1,

"subState": 11,

"robotSn": "34:7D:E4:98:A3:A1"

}

      }

}
```

name	type	Remark	Additional Information
mainState	Integer	Main Status	See mainState enumeration
subState	Integer	Secondary status	See subState enumeration
robotSn	String	Robot SN	
armrests	Integer	Armrest status	0 Not in place 1 In place
bilgeTankState	Integer	Wastewater tank status	-1 No hardware 0 Empty 1 Middle 2 Full
cleanWaterTank	Integer	Fresh water tank status	-1 No hardware 0 Empty 1 Low 2 Medium 3 Full

dustBag	Integer	Dust bag status	0 Not in place 1 In place
rollingBrushPushRod	Integer	Roller brush push rod status	0 Lift 1 Put down
faulting	Boolean	Fault Status	true Faulty false No fault
locationSuc	Boolean	Positioning status	true: Positioning successful false: Positioning failed
lock	Boolean	Unlocked state	true unlock successful false unlock failed
rosConnect	Boolean	ROS connection status	true if the connection is successful false if the connection is failed
scheduling	Boolean	Scheduling status	trueScheduling falseNo scheduling
scram	Boolean	Emergency stop state	true Emergency stop false No emergency stop
upgrading	Boolean	Upgrade Status	true Upgrading false No upgrade
lifting	Boolean	Elevator status	true: Taking the elevator false: Not taking the elevator
navigating	Boolean	Navigation Status	true Navigating false No navigation

Main state mainState enumeration

Main Status	Status value

idle	1
operate	2
Work	3
Charge	4
Offline	-1

Secondary state subState enumeration

Secondary master status	Status value
Idle	11
In operation	twenty one
Charging default	30
Charging matching	31
Charging	32
Under pile	33
Line charging	34
Working Default	40
Navigating	41
Cleaning	42
Self-cleaning	43
Returning	44

Cleaning Pause	45
Return to flight suspended	46
Hand push work	47
Offline	-1

Query the cleaning robot cleaning area interface

Path:/api/open/ custom/clean/robot/area/list

Method: GET

Headers

Parameter name	Parameter Value	Is it necessary	Example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

name	type	Required	illustrate
storeId	Strring	yes	Store ID
robot Sn	Strring	no	Robot SN

name	type	Required	illustrate
currentPage	Integer	no	Page number, the default value is 1
pageSize	Integer	no	Page size, default is 20, maximum value is 100

Return data example

```
{
      "code": 610000,
   "msg": "success",
      "data": {
"currentPage": 1,
"pageSize": 20,
"count":1,
"entities": [{
                  "storeId": "C00000102",
                  "robotSn": "34:7D:E4:98:A1:79",
"mapId": "9ce13177f5b7469b8eb428929af94364",
"floor": 1,
"areaIdList": ["9ce13177f5b7469b8eb428929af94364"],
"areaNameList": ["area 1"]
}]
}
      }
}
```

name	type	default value	Remark	Additional Information
storeId	String		Store ID	
robotSn	String		Robot SN	
mapId	String		Map ID	
floor	Integer		floor	
areaIdList	Array		Region ID	
areaNameList	Array		Region Name	

Remotely send cleaning robot backwash task command interface

Path: /api/open/custom/clean/robot/recharge/task

Method: POST

Headers

Parameter name	Parameter Value	Is it necessary	Example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Body

name	type	Required	illustrate
robotSn	Strring	yes	Robot SN

Return data example

```
{

"msg": "success",

"code": 610000,

"data": {

"code": 610000,

"robotSn": "34:7D:E4:98:A1:BA",

"bizType": "CleanRobotRechargeTask"

},

"errorMsg": ""

}
```

name	type	default value	Remark	Additional Information
code	String		Remote control receipt code	
robotSn	String		Robot SN	
bizType	String		Business Type	

Remotely send cleaning robot end task command interface

Path: /api/open/custom/clean/robot/finish/task

Method: POST

Headers

Parameter name	Parameter Value	Is it necessary	Example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Body

name	type	Required	illustrate
robotSn	Strring	yes	Robot SN

Return data example

```
{

"msg": "success",

"code": 610000,

"data": {

"code": 610000,

"robotSn": "34:7D:E4:98:A1:BA",

"bizType": "CleanRobotFinishTask"

},

"errorMsg": ""

}
```

name	type	default value	Remark	Additional Information
code	String		Remote control receipt code	
robotSn	String		Robot SN	
bizType	String		Business Type	

Remotely send cleaning robot pause task command interface

Path: /api/open/custom/clean/robot/pause/task

Method: POST

Headers

Parameter name	Parameter Value	Is it necessary	Example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Body

name	type	Required	illustrate
robotSn	Strring	yes	Robot SN

Return data example

```
{
"msg": "success",
"code": 610000,
"data": {
"code": 610000,
"robotSn": "34:7D:E4:98:A1:BA",
"bizType": "CleanRobotPauseTask"
},
"errorMsg": ""
}
```

name	type	default value	Remark	Additional Information
code	String		Remote control receipt code	
robotSn	String		Robot SN	
bizType	String		Business Type	

Query the cleaning mode interface of the cleaning robot

Path: /api/open /custom/clean/robot/strategy/clean/model

Method: GET

Headers

Parameter name	Parameter Value	Is it necessary	Example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

name	type	Required	illustrate
robot Sn	Strring	no	Robot SN

Return data example

```
{
"msg": "success",
"code": 610000,
"data": {
"robotSn": "34:7D:E4:98:A3:A1",
"cleanModelList": [
{
"cleanModelId": "1",
"cleanModelName": "Vacuuming"
}
]
},
"errorMsg": ""
}
```

name	type	default value	Remark	Additional Information
robotSn	String		Robot SN	
cleanModelId	String		Cleaning Mode Id	
cleanModelName	String		Cleaning mode name	

Query the cleaning robot's return point interface

Path: /api/open /custom/clean/robot/strategy/back/point

Method: GET

Headers

Parameter name	Parameter Value	Is it necessary	Example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Param

name	type	Required	illustrate
robot Sn	Strring	no	Robot SN

Return data example

```
{
"msg": "success",
"code": 610000,
"data": {
"robotSn": "34:7D:E4:98:A3:A1",
"backPointList": [
{
"backPointId": "11",
"backPointName": "Charging pile",
"backPointType": "charge"
}
]
},
"errorMsg": ""
}
```

name	type	default value	Remark	Additional Information
robotSn	String		Robot SN	
backPointId	String		Return point Id	

backPointName	String		Return point name	
backPointType	String		Return point type	

Remotely send temporary task command interface to cleaning robots

Path: /api/open /custom/clean/robot/strategy/temporary/task

Method: POST

Headers

Parameter name	Parameter Value	Is it necessary	Example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Body Parameter Example

```
{

"robotSn": "34:7D:E4:98:A3:A1",

"areaIdList":["5c2f330e6a1d4999ae631fb79e0f6c5d"],

"cleanModelId":"1",

"cleanTimes": 1,

"backPointId": "11"

}
```

name	type	Required	illustrate
robotSn	Strring	yes	Robot SN
areaIdList	Array	yes	Cleaning area ID collection (can cross floors, does not support automatic area optimization, the total number of floors where the area is located cannot be greater than 6)
cleanModelId	String	yes	Cleaning Mode Id
cleanTimes	Integer	yes	Cleaning times: maximum 5 times, minimum 1 time
backPointId	Integer	yes	Return point Id

Return data example

```
{

"msg": "success",

"code": 610000,

"data": {

"code": 610000,

"robotSn": "34:7D:E4:98:A1:BA",

"bizType": "CleanStrategyTemporary"

},

"errorMsg": ""

}
```

name	type	default value	Remark	Additional Information
code	String		Remote control receipt code	
robotSn	String		Robot SN	
bizType	String		Business Type	

Remotely send cleaning robot to create a scheduling task instruction interface

Path: /api/open /custom/clean/robot/strategy/task

Method: POST

Headers

Parameter name	Parameter Value	Is it necessary	Example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Body Parameter Example

```json
{
"cleanName": "21",
"timeCleanType": 1,
"startTime": "12:09",
"endTime": "13:56",
"cycleType": "1",
" cycleDetail ": "1,3,5,7",
"cleanModelId": "0e5ae0c96fc44cb98a20ac297db05d33",
"areaIdList": [
"3993823cd5f3479c82701c9306c9bbf8",
"5d8126e73aff414c8022f2bd65b4ab54"
],
"cleanTimes": 1,
"backPointId": "811",
"robotSn": "34:7D:E4:98:A1:BA"
}
```

name	type	Required	illustrate
cleanName	Strring	yes	Task Name

name	type	Required	illustrate
areaIdList	Array	yes	Cleaning area ID collection (can cross floors, does not support automatic area optimization, the total number of floors where the area is located cannot be greater than 6)
timeCleanType	Integer	yes	Task type 1 quantitative 2 scheduled
startTime	Strring	yes	Start time format hh:mm
endTime	Strring	no	End time format hh:mm , timeCleanType must be 2
cycleType	String	yes	Frequency 0 only once, 1 recurring, 2 daily
cycleDetail	String	no	Cleaning cycle (if the frequency is cycle, you must pass pure numbers 1~7 separated by commas)
cleanModelId	String	yes	Cleaning Mode Id
cleanTimes	Integer	yes	Cleaning times: maximum 5 times, minimum 1 time
backPointId	Integer	yes	Return point Id
robotSn	String	yes	Robot SN

Return data example

```
{
"msg": "success",
"code": 610000,
"data": {
"code": 610000,
"robotSn": "34:7D:E4:98:A1:BA",
"bizType": "AddCleanStrategy"
},
"errorMsg": ""
}
```

name	type	default value	Remark	Additional Information
code	String		Remote control receipt code	
robotSn	String		Robot SN	
bizType	String		Business Type	

Remotely send cleaning robot to modify the task scheduling command interface

Path: /api/open/custom/clean/robot/strategy/task

Method: PUT

Headers

Parameter name	Parameter Value	Is it necessary	Example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Body Parameter Example

```
{
"strategyId":"ba77a65cb060412bae638fd35ccae572",
    "cleanName": "21",
    "timeCleanType": 1,
    "startTime": "12:09",
    "endTime": "13:56",
"cycleType": "1",
"cycleDetail": "1,3,5,7",
    "cleanModelId": "0e5ae0c96fc44cb98a20ac297db05d33",
    "areaIdList": [
        "3993823cd5f3479c82701c9306c9bbf8",
        "5d8126e73aff414c8022f2bd65b4ab54"
    ],
    "cleanTimes": 1,
    "backPointId": "811",
    "robotSn": "34:7D:E4:98:A1:BA"
}
```

name	type	Required	illustrate
strategyId	String	yes	Scheduling Task Id
cleanName	Strring	yes	Task Name
areaIdList	Array	yes	Cleaning area ID collection (can cross floors, does not support automatic area optimization, the total number of floors where the area is located cannot be greater than 6)
timeCleanType	Integer	yes	Task type 1 quantitative 2 scheduled
startTime	Strring	yes	Start time format hh:mm
endTime	Strring	no	End time format hh:mm , timeCleanType must be 2
cycleType	String	yes	Frequency 0 only once, 1 recurring, 2 daily
cycleDetail	String	no	Cleaning cycle (if the frequency is cycle, you must pass pure numbers 1~7 separated by commas)
cleanModelId	String	yes	Cleaning Mode Id
cleanTimes	Integer	yes	Cleaning times: maximum 5 times, minimum 1 time
backPointId	Integer	yes	Return point Id
robotSn	String	yes	Robot SN

Remotely send cleaning robot deletion, enable or disable scheduling task command interface

Path: /api/open/custom/clean/robot/strategy/task

Method: DELETE

Headers

Parameter name	Parameter Value	Is it necessary	Example	Remark
Authorization	bearer access_token	yes	bearer 009a8332-634e-4886-9c99-324caac55166	

Body Parameter Example

```
{
"strategyId":"f93ac2fd45bd42c1bb41d8faa4e54f52",
"robotSn":"34:7D:E4:98:A1:BA",
"strategyEnable": 0
}
```

name	type	Required	illustrate
strategyId	String	yes	Scheduling Task Id
strategyEnable	Integer	no	Task switch 0 off 1 on (enable or disable required)
deleted	Integer	no	0 Delete (required for deletion)
robotSn	String	yes	Robot SN

Return data example

```
{

"msg": "success",

"code": 610000,

"data": {

"code": 610000,

"robotSn": "34:7D:E4:98:A1:BA",

"bizType": "DeleteCleanStrategy"

},

"errorMsg": ""

}
```

name	type	default value	Remark	Additional Information
code	String		Remote control receipt code	
robotSn	String		Robot SN	
bizType	String		Business Type	

Remote robot control receipt code value table

Note: The code values marked in yellow are provided by the app.

Code value	describe
610000	success
615001	Robot offline
615025	Robot response timeout
615028	The maximum number of floors to be cleaned is 5.
615029	Incorrect cleaning frequency range
615030	Incorrect cleaning cycle
615031	Please select an area with a path
615032	Incorrect cleaning task type
5000	Robot in operation
5001	The robot is not idle
5003	The robot is performing the current task
5004	There are no tasks for the robot to perform
5005	Backflush task self-check failed
5006	Continue task self-check failed
5007	Robot is working

5008	The robot is matching the charging pile
5009	Robot taking the elevator
5010	The robot is paused
5011	Remote operation is too frequent
5012	Robot dispatching
5013	Robot self-cleaning
5014	Robot positioning lost
5015	The machine APP program is running in the background
5016	Failed to create the shift task
5017	Failed to modify the shift task
5018	Temporary task startup self-check failed
5019	The return point does not exist
5020	Region does not exist
5021	Duplicate task name
5022	Region ID does not exist
5023	Temporary task startup failed

5024	Failed to enable/disable scheduled tasks
5025	Cleaning mode does not exist
5026	The end time must be greater than the start time
5027	Wrong time format
5028	Data verification failed
5029	Failed to delete task
290001	Unknown error

The following are the states in which instructions can be issued in the secondary state:

Robot Status	Secondary status code value	pause	Recharge	continue	Finish
Offline	-1	——	——	——	——
In the elevator	none	——	——	——	——
Emergency stop	none	——	——	——	——
Unlocking	none	——	——	——	——
Scheduling	none	——	——	——	——
Upgrading	none	——	——	——	——
Positioning failed	none	——	——	——	——
Failure	none	——	——	——	——
Idle	11	——	√	——	——

In operation	twenty one	——	——	——	——
Charging matching	31	——	——	——	——
Charging	32	——	——	——	——
Off the pile	33	——	——	——	——
Line charging	34	——	——	——	——
Navigating	41	√	——	——	——
Cleaning	42	√	——	——	——
Self-cleaning	43	——	——	——	——
Returning	44	√	——	——	——
Cleaning Pause	45	——	——	√	√
Return to flight suspended	46	——	——	√	√
Hand push work	47	——	——	——	——

http status code value enumeration

Code value	describe
610000	success
610500	System abnormality
610401	Invalid token
610403	No access permission

4 Appendix 1

4.1 Remote call callback example

1. The user provides the webhook interface address, and when the task changes, the message is pushed through the webhook interface address
2. The format of the body pushed to the user interface is:

name	type	Default Value	Remark	Additional Information
bizType	String		business type	CreateTask : remote call
clientId	String			
data	object			
├─taskNo	String		TaskId	
├─taskStatus	Integer		task status	0 - task failed 1 - task queued 2 - calling 3 - in progress 4 - task completed 5 - task canceled 6 - target point reached 7 - waiting

├─errorCode	Integer		robot error code	300, "The remote call function is not enabled on the robot side" 301, "The robot side is being set up" 302, "The robot side is executing a task and cannot be called" 303, "The robot side refuses to execute the task" 304, "There is no corresponding task on the robot side , unable to operate" 305, "The robot cannot recognize the target point" 400, "The robot manually canceled the task" 401, "The corresponding task record was not found on the robot" 402, "An unknown exception occurred on the robot and the task ended" 403, "The robot The terminal judged that the task timed out" 500, "The server did not find an available robot" 501, "The task has not been updated for a long time, and the task has ended in the background" 502, "The robot is offline, resulting in a task that has not been updated for a long time" 503, "The robot has not registered to call the service"
├─remark	String		task notes	
├─deviceId	String		User ID	return client_id

├─taskData	object		task current data	
├─├─remainingDistance	Double		Remaining mileage in meters	
├─├─robotSn	String		Robot Unique ID	
├─├─waitQueuing	Integer		current queue order	
├─├─lastUpdateTime	String		last update time	

{
"bizType": "CreateTask",
"clientId": "tianmao",
"data": {
"taskNo": "demoData",
"taskStatus": 1,
"errorCode": 1,
"remark": "demoData",
"deviceId": "demoData",
"taskData": {
"remainingDistance": 1,
"robotSn": "demoData",
"waitQueuing": 1,
"remainingWaitTime": 1,
"lastUpdateTime": "2022-02-02 12:22:22"

```
}
}
}
```

4.2Example of Machine Presence Callback

1. The user provides the webhook interface address, and when the online status of the machine changes, the message is pushed through the webhook interface address
2. The body format pushed to the user interface

name	type	Default Value	Remark	Additional Information
bizType	String		business type	RobotOnlineStatus : machine online status
clientId	String		account	
data	object			
├─onlineStatus	Boolean		online status	false: offline , true: online
├─robotSn	String		Machine sn number	

```
{
" bizType ": " RobotOnlineStatus ",
" clientId ": " tianmao ",
"data": {" onlineStatus ": false,
" robotSn ": "8C:FC:A0:17:EF:C0"
}}
```

4.3 Example of machine online callback

1. The user provides the webhook interface address, and when the online mode of the machine changes, the message is pushed through the webhook interface address
2. The body format pushed to the user interface

name	type	Default Value	Remark	Additional Information
bizType	String		business type	RobotOnlineType : machine online mode
clientId	String		account	
data	object			
├─onlineType	String		online	Values: 2G, 4G, Wi-Fi, unknown
├─robotSn	String		Machine sn number	

```
{
" bizType ": " RobotOnlineType ",
" clientId ": " tianmao ",
"data": {" onlineType ": " Wi-Fi ",
" robotSn ": "0C:C6:55:77:32:84"
}}
```

4.4 Example of machine power information callback

1. The user provides the address of the webhook interface, and when the machine reports battery power information, the message is pushed through the address of the webhook interface
2. The body format pushed to the user interface

name	type	Default Value	Remark	Additional Information
bizType	String		business type	RobotPowerInfo : machine power information
clientId	String		account	
data	object			
├─robotSn	String		Machine sn number	
├─power	object			
├─chargeStatus	Integer		charging	
├─batteryLevel	Integer		battery power	

```
{
" bizType ": " RobotPowerInfo ",
" clientId ": " tianmao ",
"data": {
" robotSn ": "0C:C6:55:77:32:84",
"power": {" chargeStatus ": - 1,
" batteryLevel ": 95
}}}
```

4.5 Remote dispatch order status callback example

1. The user provides the address of the webhook interface, and the robot pushes the message through the address of the webhook interface when performing the remote dispatch process
2. The body format pushed to the user interface

name	type	Default Value	Remark	Additional Information
bizType	String		business type,this field is not pushed	HotelOrderStatus: remote order status
clientId	String		Account,this field is not pushed	
data	object			
├─orderNo	String		order number	
├─orderStatus	Integer		Order Status	See order status enumeration
├─logisticsList	Array		logistics list	
├─logisticsNo	String		shipment number	
├─cabin	Integer		cabin number	as 1
├─cabinType	Integer		Cabin	See cabin type enumeration
├─logisticsStatus	Integer		logistics order status	See Logistics Order Status Enumeration

Reference push data

```
{
"orderNo": "20221229213425091",
"orderStatus": 100,
"logisticsList": [{
"logisticsNo": "20221229213425612",
"cabin": 1,
"cabinType": 1,
"logisticsStatus": 501
}]
}
```

Order status enumeration value

status code	status description
100	in the line
109	Go to pick up
110	to be delivered
111	delivery
180	completed
190	Order exception
193	Cancelled
390	System cancel
999	unknown state
114	Arrived

Logistic order status enumeration value

status code	status description	Corresponding order status

501	to be confirmed	100
502	to be delivered	100
503	delivery	111
505	arrive to placement point	109
510	completed	180
511	Cancelled	193
590	Abnormal (reason)	390
504	Arrived	114

Cabin enumeration

Cabin	describe
1	four cabins
2	double cabin
3	Three-cabin T-type
4	Three cabin inverted T type
5	single cabin

Interface Compatibility:

Applicable models	W series
Minimum APP version	Hotel v1.12.0

4.6 An example of a robot delivery task callback

1. The user provides the webhook interface address, and the robot pushes messages through the webhook interface address when performing the delivery task
2. The body format pushed to the user interface

name	type	Default Value	Remark	Additional Information
bizType	String		business type,this field is not pushed	HotelRobotBizTask : robot delivery task
clientId	String		account,this field is not pushed	
data	object			
├─taskState	Integer		Main task status (0: initial value, 1 : task successful, 2 : task not completed	
├─totalTime	Double		Main task execution time	
├─startTime	Long		Main task start time	

├─endTime	Long		Main task end time	
├─robotSn	String		Robot SN	
├─storeId	String		Store Id	
├─storeName	String		store name	
├─uuid	String		task uuid	
├─totalMileage	Double		total mileage	
├─taskDetails	Array		subtask list	
├──duration	Double		task time consuming	
├──logisticsNo	String		shipment number	
├──startTime	Long		Subtask start time	
├──endTime	Long		Subtask end time	
├──doorId	Integer		hatch id	
├──takeCode	String		Pick up code	
├──mileage	Double		mileage	
├──status	Integer		Subtask status	-1: initial value 0: success 1: failure 2: cancel task 3: passive cancellation 4: start delivery 5: delivery arrival
├──target	String		delivery point	

├──taskId	String		Subtask id	
├──parentTaskId	String		Main task id	

├──reason	Integer		Reason	1: The user retrieves the item. 2: Timeout to automatically confirm pick-up 101: uncollected 102: Delivery timeout 103: shutdown restart 104: app blink 105: elevator ride timeout 116: outbound ground elevator call failure 117: internal elevator call failure 201: Information input error 202: Robot not moving 203: Elevator fault 208: Wrong destination 209: Item misplaced 210: Wrong phone number 211: Add new delivery task 212: Compartment door fault 213: Phone-related fault 214: Elevator under maintenance 217: unexpected charge 224: Functional test or demonstration 226: Floor error 227: active exit APP 229: connectivity test cancelled 301: Passive cancellation 307: enter the supersetting 308: User modification task

├──taskType	String		Subtask type (1: Delivery)	

Reference push data

```
{
"taskState": 0,
"totalTime": 35.105,
"startTime": 1672312116669,
"storeName": "测试门店",
"endTime": 1672312151774,
"robotSn": "8C:18:D9:9D:D0:FA",
"storeId": "S00003434",
"totalMileage": 0.0,
"uuid": "fd28f36b533141d896933d2e07930619",
"taskDetails": [{
"duration": 0.0,
"logisticsNo": "20221229190829593",
"startTime": 1672312153948,
"endTime": 0,
"doorId": -1,
"takeCode": "1111",
"mileage": 0.0,
"status": 4,
"taskId": 12344,
"parentTaskId": 12345,
"reason": 301,
"taskType": "12"
}]
}
```

Interface Compatibility:

Applicable models	W series
Minimum APP version	Hotel v1.12.0

4.7 Machine working status, callable status callback

1. Callback business type RobotWorkState
2. The user provides the webhook interface address, and the robot pushes messages through the webhook interface address when the working status or callable status changes.
3. The body format pushed to the user interface

name	**type**	**default value**	**Remark**	**Additional Information**
bizType	String		Business Type	RobotWorkState : Machine working state
clientId	String			
data	Object			
├─ robotSn	String		Machine sn	
├─ robotState	Integer		Working status	1: On task, 2: Idle, 3: Operating, 4: Scheduling, 5: Charging, 6: Powering on and positioning
├─ canBeCalled	Boolean		Call Status	true: callable, false: not callable

Callback data example:

```
{

      "bizType": " RobotWorkState ",

      "clientId": "tianmao",

      "data": {

            " robotSn ": " 54:EF:33:CA:E4:FF ",

            " robotState ": 2 ,

            " canBeCalled ": true

      }

}
```

4.8 Machine coordinate position and nearby point callback

1. Callback business type RobotPositionType
2. The user provides the webhook interface address, and the robot pushes messages through the webhook interface address when the position changes
3. The body format pushed to the user interface

name	**type**	**default value**	**Remark**	**Additional Information**
bizType	String		Business Type	RobotPositionType : Real-time position and point of the machine

clientId	String			Open platform account
data	Object			
-robotPos	Object		Coordinate information	
--x	String		x-coordinate	
--y	String		y coordinate	
--rotation	String		angle	
-params	Object		Nearby point information	
--dstId	Integer		Nearby point ID	
--dstName	String		Nearby point name	
--dstType	String		Nearby point type	
--distance	Double		Attachment point distance	

Callback data example

```
{
   "msg" : "Request successful" ,
   "code" : 610000 ,
   "data" : {
      "robotPos" : {
         "x" : "1.23" ,
         "y": "-2.34",
         "rotation": "35"
```

```
        },
        "params": {
           "dstId": 1,
           "dstName":"桌号 3",
           "dstType":"出菜口",
           "distance":1.23,
}
}
```

4.9 Remote task real-time status callback

1. Callback business type RobotTaskState
2. The user provides the webhook interface address, and the status of the robot's remote task is pushed through the webhook interface address
3. The body format pushed to the user interface

Parameter name	**Parameter Type**	**Is it necessary**	**Example**	**Remark**
bizType	String	yes		RobotTaskState : Task real-time status type
clientId	String	yes		
data	Object	yes		
-robotSn	String	yes		Machine sn
-taskNo	String	yes		Task Number

-taskType	String	yes		Task Type 0: non-task, 1: food delivery, 2: multiple orders, 3: remote task, 4: remote dish return task, 5: snack, 6: cyclic dish return, 7: go to the dishwashing room, 8: welcome, 9: blessing, 10: return, 11: charging, 12: charging return task , 13: direct
-taskState	String	yes		Main task status 0 : failed , 1: queued , 2: calling 3: In progress , 4 : Completed , 5 : Task cancelled
-taskNoType	Integer	yes		Task source, 1: remote task, 2: local task

-errorCode	Integer	yes		Error code. When the main task status is failed, the error code is reported 300 : "The remote call function is not enabled on the robot" 301, "The robot is being set up" 302, "The robot is executing a task and cannot be called" 303, "The robot refuses to execute the task" 304, "The robot has no corresponding task and cannot be operated" 305, "The robot cannot identify the target point" 400, "The robot manually cancels the task" 401, "The robot does not find the corresponding task record" 402, "An unknown exception occurred on the robot, causing the task to end" 403, "The robot determines that the task has timed out" 500, "The server does not find an available robot"
-subTaskInfoList	Object[]	yes		Subtask status information
--uuid	String	yes		The point uuid of the current subtask
--pointId	String	yes		The point id of the current subtask
--pointName	String	yes		The point name of the current subtask
--type	Integer	yes		The point type of the current subtask 1: Target point, 2: Dishwashing room, 3: Food outlet, 4: Supplementary point

--taskState	String	yes		The task status of the current subtask 1: Waiting, 2: Running, 3: Paused, 4: Arrive at the destination, 5: Pick up the meal in advance, 6: Cancel the task, 7: Task abnormality
--taskDistance	String	yes		The task distance of the current subtask, in meters

Callback data example

```
{
   "robotSn" : "54:EF:33:CA:E4:FF" ,
   "taskNo" : "Zdpbzv5LehGMAZ5h" ,
   "taskNoType" : 1 ,
   "taskState" : 3 ,
   "taskType" : 1 ,
   "errorCode" : 400 ,
   "extendedInfo": "",
   "subTaskInfoList": [
      {
         "uuid": "e23rt5ty",
         "pointId": 2,
         "pointName": "桌号 1",
         "type": 1,
         "taskState": 1,
         "taskDistance": "12.3"
      },
      {
         "uuid": "y2ftt5er",
         "pointId": 3,
         "pointName": "桌号 3",
         "type": 2,
         "taskState" : 1 ,
        "taskDistance" : "12.3"
}
]
}
```

4.10 Cleaning robot real-time status push callback

1. Callback service type CleanRobotStatus
2. The user provides the webhook interface address, and the robot status changes are pushed through the webhook interface address
3. The body format pushed to the user interface

Parameter name	Parameter Type	Remark	Additional Information
mainState	Integer	**Main Status**	See mainState enumeration
subState	Integer	Secondary status	See subState enumeration
robotSn	String	Robot SN	
armrests	Integer	Armrest status	0 Not in place 1 In place
bilgeTankState	Integer	Wastewater tank status	-1 No hardware 0 Empty 1 Middle 2 Full
cleanWaterTank	Integer	Fresh water tank status	-1 No hardware 0 Empty 1 Low 2 Medium 3 Full
dustBag	Integer	Dust bag status	0 Not in place 1 In place

rollingBrushPushRod	Integer	Roller brush push rod status	0 Lift 1 Put down
faulting	Boolean	Fault Status	true Faulty false No fault
locationSuc	Boolean	Positioning status	true: Positioning successful false: Positioning failed
lock	Boolean	Unlocked state	true unlock successful false unlock failed
rosConnect	Boolean	ROS connection status	true if the connection is successful false if the connection is failed
scheduling	Boolean	Scheduling status	trueScheduling falseNo scheduling
scram	Boolean	Emergency stop state	true Emergency stop false No emergency stop
upgrading	Boolean	Upgrade Status	true Upgrading false No upgrade
lifting	Boolean	Elevator status	true: Taking the elevator false: Not taking the elevator
navigating	Boolean	Navigation Status	true Navigating false No navigation

Callback data example

```
{
"globalState": {
"rosConnect": true,
"faulting": false,
        "upgrading": false,
        "locationSuc": true,
        "scheduling": false,
        "lock": false,
        "scram": true
    },
    "hardwareState": {
        "bilgeTankState": -1,
        "dustBag": 1,
        "rollingBrushPushRod": 0,
        "armrests": 1,
        "cleanWaterTank": -1
    },
    "subState": 21,
    "mainState": 2,
    "childState": {
        "lifting": false,
"navigating": false
},
"updateTime": 1736228801348,
"robotSn": "2C:C3:E6:E8:33:78"
}
```

4.11 Remotely send cleaning robot backflush task callback

1. Callback service type CleanRobotRechargeTask
2. The user provides the webhook interface address, and the robot sends the task receipt through the webhook interface address.
3. The body format pushed to the user interface

Parameter name	**Parameter Type**	**Remark**	**Additional Information**

code	String	Remote control receipt code	
robotSn	String	Robot SN	
bizType	String	Business Type	Default CleanRobotRechargeTask

Callback data example

```
{
"bizType": "CleanRobotRechargeTask",
"code": 610000,
"robotSn": "2C:C3:E6:E8:33:78"
}
```

4.12 Remotely send the cleaning robot to complete the task callback

1. Callback business type CleanRobotFinishTask
2. The user provides the webhook interface address, and the robot sends the task completion receipt through the webhook interface address.
3. The body format pushed to the user interface

Parameter name	**Parameter Type**	**Remark**	**Additional Information**
code	String	Remote control receipt code	
robotSn	String	Robot SN	
bizType	String	Business Type	Default CleanRobotFinishTask

Callback data example

```
{
"bizType": "CleanRobotFinishTask",
"code": 610000,
"robotSn": "2C:C3:E6:E8:33:78"
}
```

4.13 Remotely send cleaning robot pause task callback

1. Callback service type CleanRobotPauseTask
2. The user provides the webhook interface address, and the robot pause task receipt is sent through the webhook interface address.
3. The body format pushed to the user interface

Parameter name	**Parameter Type**	**Remark**	**Additional Information**
code	String	Remote control receipt code	
robotSn	String	Robot SN	
bizType	String	Business Type	Default CleanRobotPauseTask

Callback data example

```
{
"bizType": " CleanRobotPauseTask ",
"code": 610000,
"robotSn": "2C:C3:E6:E8:33:78"
}
```

4.14 Remotely send cleaning robot temporary task callback

1. Callback business type CleanStrategyTemporary
2. The user provides the webhook interface address, and the robot's temporary task receipt is pushed through the webhook interface address
3. The body format pushed to the user interface

Parameter name	**Parameter Type**	**Remark**	**Additional Information**
code	String	Remote control receipt code	
robotSn	String	Robot SN	
bizType	String	Business Type	CleanStrategyTemporary

Callback data example

```
{
"bizType": " CleanStrategyTemporary ",
"code": 610000,
"robotSn": "2C:C3:E6:E8:33:78"
}
```

4.15 Remotely send cleaning robot to create a scheduling task callback

1. Callback business type AddCleanStrategy
2. The user provides the webhook interface address, and the robot creates a task schedule receipt and pushes it through the webhook interface address
3. The body format pushed to the user interface

Parameter name	Parameter Type	Remark	Additional Information
code	String	Remote control receipt code	
robotSn	String	Robot SN	
bizType	String	Business Type	AddCleanStrategy

Callback data example

```
{
"bizType": " AddCleanStrategy ",
"code": 610000,
"robotSn": "2C:C3:E6:E8:33:78"
}
```

4.16 Remotely send cleaning robot to modify the task schedule callback

1. Callback business type UpdateCleanStrategy
2. The user provides the webhook interface address, and the robot sends the modified shift task receipt through the webhook interface address.
3. The body format pushed to the user interface

Parameter name	Parameter Type	Remark	Additional Information
code	String	Remote control receipt code	
robotSn	String	Robot SN	
bizType	String	Business Type	UpdateCleanStrategy

Callback data example

```
{
"bizType": " UpdateCleanStrategy ",
"code": 610000,
"robotSn": "2C:C3:E6:E8:33:78"
}
```

4.17 Remotely send cleaning robot deletion, enable or disable scheduling task callbacks

1. Callback business type DeleteCleanStrategy
2. The user provides the webhook interface address, sends the robot to delete, enables or disables the scheduling task receipt to be pushed through the webhook interface address
3. The body format pushed to the user interface

Parameter name	**Parameter Type**	**Remark**	**Additional Information**
code	String	Remote control receipt code	
robotSn	String	Robot SN	
bizType	String	Business Type	Default DeleteCleanStrategy

Callback data example

```
{
"bizType": " DeleteCleanStrategy ",
"code": 610000,
"robotSn": "2C:C3:E6:E8:33:78"
}
```