

Tracer 200-ROBOT-API

Version	Edit Time	Editor	Edit Content
v1.0.5	2023-3-24	HuiMin	Draft
v1.0.9	2023-4-26		Modify communication protocol
v1.1.1	2023-6-26		Add websocket protocol description
v1.1.2	2023-7-3		Add Basic authentication
v1.1.38	2024-4-23		Supplement operational statistics related interfaces

Document Description

The `data dictionary` in this document is `woosh_robot_data_dictionary`.

URL

- `http`: `http://{robot or DMS IP address}:5480`
- `websocket`: `ws://{robot or DMS IP address}:5480`

Protocol Description

Currently supports `http` and `websocket`, both request and response are `application/json`.

woosh.Request

```
{
  "type": "",           // Request message type
  "body": {},           // Request message content
  "sn": number          // Message sequence number,
                        // can be used to determine message order
}
```

- For `http` requests, simply pass the `body` object.
- The `sn` field is only valid for `websocket` requests.
- `sn` is an optional field

woosh.Response

```
{
  "type": "",           // Response message type
  "body": {},           // Response message content
  "ok": boolean,        // Response result, TRUE: success, FALSE: failure
  "msg": "",            // Response status message, e.g. "REQUEST SUCCESS"
  "sn": number          // Message sequence number,
                        // assigned by the requesting end, returned as is
}
```

woosh.Subscription

```
{
  "type": "woosh.Subscription",
  "body": {
    "sub": boolean, // TRUE: subscribe, FALSE: unsubscribe
    "topics": [
      "" // Subscribe to topic, e.g. "woosh.robot.PoseSpeed"
    ]
  }
}
```

- Subscribe/Publish only valid for `websocket`.

woosh.Notify

```
{
  "type": "", // Message type to publish
  "body": {} // Message content to publish
}
```

woosh.Topic

Name	topic	Robot	DMS
Configuration	woosh.robot.Setting	✓	✓
Robot State	woosh.robot.RobotState	✓	✓
Robot Mode	woosh.robot.Mode	✓	✓
Pose Speed	woosh.robot.PoseSpeed	✓	✓
Battery Information	woosh.robot.Battery	✓	✓
Network Information	woosh.robot.Network	✓	✓
Scene Information	woosh.robot.Scene	✓	✓
Task Progress	woosh.robot.TaskProc	✓	✓
Device Status	woosh.robot.DeviceState	✓	✓
Hardware Status	woosh.robot.HardwareState	✓	✓
Operation Status	woosh.robot.OperationState	✓	✓
Robot Model	woosh.robot.Model	✓	✓
Navigation Path	woosh.robot.NavPath	✓	✓
Status Code	robot.count.StatusCode	✓	✓
Unresolved Exception Code	woosh.robot.count.AbnormalCode	✓	✓
Mapping Data	woosh.robot.BuildMapData	✓	

Name	topic	Robot	DMS
Lidar Point Cloud Data	woosh.robot.ScannerData	✓	
Runtime Operation Statistics	woosh.robot.count.Operation	✓	
Runtime Task Statistics	woosh.robot.count.Task	✓	
Runtime Status Statistics	woosh.robot.count.Status	✓	
Robot Dispatch Information	woosh.dispatch.robot.Robot		✓
Dispatch Scene Configuration Information	woosh.dispatch.system.SceneSettings		✓
Dispatch charging configuration information	woosh.dispatch.system.ChargeSettings		✓
Task set status update	woosh.dispatch.task.TaskSetState		✓
Robot operation statistics	woosh.dispatch.kanban.RobotOperation		✓
Average route consumption time	woosh.dispatch.kanban.RouteAvg		✓
Task statistics (total)	woosh.dispatch.kanban.TaskCount		✓
Robot task statistics	woosh.dispatch.kanban.RobotTaskCount		✓

Example

Websocket request example

Request robot information

```
{
  "type": "woosh.robot.RobotInfo",
  "body": {
    "robotId": 30001
  }
}
```

Websocket subscription example

Subscribe to robot pose, velocity, and battery level

```
{
  "type": "woosh.Subscription",
  "body": {
    "sub": true,
    "topics": [
      "woosh.robot.PoseSpeed",
      "woosh.robot.Battery"
    ]
  }
}
```

Authentication Instructions

`http` and `websocket` both use `Basic Auth` encryption authentication, the username and password depend on the project.

`websocket` only requires authentication when connecting. If a connection requires authentication but is not authenticated, the server will disconnect it automatically.

Explanation of `Basic Auth` authentication:

Add the parameter `Authorization` in the `Header`.

Its value is the `Username:Password` concatenated with a space after `Basic`, and then encoded in `Base64`.

Example: `Authorization: Basic YWRtaW46MTIzNDU2`

Robot Information

POST Get Robot Information

POST /woosh/robot/RobotInfo

Body Request Parameters

```
{
  "robotId": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```

{
  "body": {
    "abnormalCodes": {
      "scs": []
    },
    "battery": {
      "batteryCycle": 0,
      "chargeCycle": 0,
      "chargeState": 1,
      "health": 0,
      "power": 93,
      "robotId": 30001,
      "tempMax": 0
    },
    "deviceState": {
      "state": 3
    },
    "genral": {
      "displayModel": "Robase200",
      "modelData": {
        "height": 1550,
        "length": 1250,
        "load": 150,
        "weight": 50,
        "width": 670
      },
      "serialNumber": 30001,
      "serviceId": "RB2000030100000000z",
      "type": 1,
      "urdfName": "wooshmodel_robbase200_cp_y",
      "version": {
        "rc": "1.0.0-1",
        "system": "UROS20210720"
      }
    },
    "hardwareState": {
      "beacon": 0,
      "board": 0,
      "camera": [
        0,
        0
      ],
      "crash": 0,
      "esb": 0,
      "imu": 0,
      "lidar": [
        0,
        0
      ],
      "lift": [
        0,
        0
      ],
      "light": 0,
      "magnetism": 0,

```

[illegible]

```

        "x": 0.4,
        "y": -0.215,
        "z": 0
    },
    {
        "x": 0.35,
        "y": -0.265,
        "z": 0
    },
    {
        "x": -0.35,
        "y": -0.265,
        "z": 0
    },
    {
        "x": -0.4,
        "y": -0.215,
        "z": 0
    },
    {
        "x": -0.4,
        "y": 0.215,
        "z": 0
    },
    {
        "x": -0.35,
        "y": 0.265,
        "z": 0
    }
    ],
    "robotId": 30001
},
"network": {
    "isConnected": false,
    "robotIp": "172.20.254.63",
    "schIp": ""
},
"operationState": {
    "nav": 2,
    "robotId": 30001
},
"posespeed": {
    "mapId": 0,
    "mileage": 0,
    "pose": {
        "theta": -0.255771846,
        "x": 4.23276472,
        "y": -6.6727581
    },
    "robotId": 30001
},
"robotId": 30001,
"scene": {
    "mapId": 0,
    "mapName": "wooshmap",
    "robotId": 30001,

```

```

    "sceneName": "wooshmap",
    "version": "147"
  },
  "setting": {
    "allow": {
      "autoCharge": true,
      "autoPark": true,
      "goodsCheck": true,
      "mechanismCheck": false
    },
    "identity": {
      "name": "30001"
    },
    "power": {
      "alarm": 5,
      "full": 98,
      "idle": 80,
      "low": 20
    },
    "server": {
      "ip": "172.20.8.85",
      "port": 5420
    }
  },
  "state": 4,
  "statusCodes": {
    "scs": [
      {
        "code": "130104000000",
        "level": 0,
        "msg": "Received new navigation task, navigation and obstacle avoidance
started",
        "robotId": 30001,
        "state": 4,
        "taskId": "68240893101",
        "time": "1682409385879",
        "type": 1
      },
      {
        "code": "130104000031",
        "level": 1,
        "msg": "Obstacle avoidance successful",
        "robotId": 30001,
        "state": 4,
        "taskId": "68240893101",
        "time": "1682409386081",
        "type": 1
      }
    ]
  },
  "taskProc": {
    "action": {
      "state": 1,
      "type": 1,
      "waitId": 0
    }
  },

```



```

    "dest": "U5",
    "msg": "",
    "robotId": 30001,
    "robotTaskId": "68240893101",
    "state": 3,
    "time": 1682409349,
    "type": 1
  }
},
"msg": "",
"ok": true,
"type": "woosh.robot.RobotInfo"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.RobotInfo</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.RobotInfo</code>

POST Get General Information

POST /woosh/robot/General

Body Request Parameters

```

{
  "robotId": 30001
}

```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "displayModel": "Robase200",
    "modelData": {
      "height": 1550,
      "length": 1250,
      "load": 150,
      "weight": 50,
      "width": 670
    },
    "serialNumber": 30001,
    "serviceId": "RB2000030100000000Z",
    "type": 1,
    "urdfName": "wooshmodel_robbase200_cp_y",
    "version": {
      "rc": "1.0.0-1",
      "system": "UROS20210720"
    }
  },
  "msg": "",
  "ok": true,
  "type": "woosh.robot.General"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.General</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.General</code>

POST Get Configuration Information

POST /woosh/robot/Setting

Body Request Parameters

```
{
  "robotId": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "allow": {
      "autoCharge": true,
      "autoPark": true,
      "goodsCheck": true,
      "mechanismCheck": false
    },
    "identity": {
```

```

    "name": "30001"
  },
  "power": {
    "alarm": 5,
    "full": 98,
    "idle": 80,
    "low": 20
  },
  "server": {
    "ip": "172.20.8.85",
    "port": 5420
  }
},
"msg": "",
"ok": true,
"type": "woosh.robot.Setting"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.Setting</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.Setting</code>

POST Get Robot Status

POST /woosh/robot/RobotState

Body Request Parameters

```
{
  "robotId": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» robotId	body	integer	optional	Robot ID	none

Return example

Success

```
{
  "body": {
    "robotId": 30001,
    "state": 4
  },
  "msg": "",
  "ok": true,
  "type": "woosh.robot.RobotState"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.RobotState</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.RobotState</code>

POST Get Mode Information

POST /woosh/robot/Mode

Body Request Parameters

```
{
  "robotId": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "ctrl": 1,
    "work": 3
  },
  "msg": "",
  "ok": true,
  "type": "woosh.robot.Mode"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.Mode</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.Mode</code>

POST Get Pose Speed

POST /woosh/robot/PoseSpeed

Body Request Parameters

```
{
  "robotId": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "mapId": 0,
    "mileage": 0,
    "pose": {
      "theta": 2.89655304,
      "x": -8.88137531,
      "y": 0.518660784
    },
    "robotId": 30001
  }
}
```

```

},
"msg": "",
"ok": true,
"type": "woosh.robot.PoseSpeed"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.PoseSpeed</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.PoseSpeed</code>

POST Get Battery Information

POST /woosh/robot/Battery

Body Request Parameters

```

{
  "robotId": 30001
}

```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none

Name	Location	Type	Required	Chinese Name	Description
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "batteryCycle": 0,
    "chargeCycle": 0,
    "chargeState": 1,
    "health": 0,
    "power": 91,
    "robotId": 30001,
    "tempMax": 0
  },
  "msg": "",
  "ok": true,
  "type": "woosh.robot.Battery"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.Battery</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.Battery</code>

POST Get Network Information

POST /woosh/robot/Network

Body Request Parameters

```
{
  "robotId": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "isConnected": false,
    "robotIp": "172.20.254.63",
    "schIp": ""
  },
  "msg": "",
  "ok": true,
  "type": "woosh.robot.Network"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.Network</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.Network</code>

POST Get scene information

POST /woosh/robot/Scene

Body Request Parameters

```
{
  "robotId": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "mapId": 0,
    "mapName": "wooshmap",
    "sceneName": "wooshmap",
    "version": "147"
  },
  "msg": "",
  "ok": true,
  "type": "woosh.robot.Scene"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.Scene</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.Scene</code>

POST Get task progress

POST /woosh/robot/TaskProc

Body Request Parameters

```
{
  "robotId": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "action": {
      "state": 1,
      "type": 1,
      "waitId": 0
    },
    "dest": "K10",
    "msg": "",
    "robotId": 30001,
    "robotTaskId": "68241434201",
    "state": 3,
    "time": 1682416646,
    "type": 1
  },
  "msg": "",
  "ok": true,
  "type": "woosh.robot.TaskProc"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.TaskProc</code> for details

Name	Type	Required	Constraint	Chinese name	Explanation
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	woosh.robot.TaskProc

POST Get device status

POST /woosh/robot/DeviceState

Body Request Parameters

```
{
  "robotId": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "robotId": 30001,
    "state": 3
  },
  "msg": "",
  "ok": true,
  "type": "woosh.robot.DeviceState"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.DeviceState</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.DeviceState</code>

POST Get hardware status

POST /woosh/robot/HardwareState

Body Request Parameters

```
{
  "robotId": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

[illegible]


```
    0,
    0,
    0
  ],
  "tractor": 0
},
"msg": "",
"ok": true,
"type": "woosh.robot.HardwareState"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.HardwareState</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.HardwareState</code>

POST Get Running State

POST /woosh/robot/OperationState

Body Request Parameters

```
{
  "robotId": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "nav": 2,
    "robotId": 30001
  },
  "msg": "",
  "ok": true,
  "type": "woosh.robot.operationState"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.operationState</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.operationState</code>

POST Get Robot Model

POST /woosh/robot/Model

Body Request Parameters

```
{
  "robotId": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "model": [
      {
        "x": 0.35,
        "y": 0.265,
        "z": 0
      },
      {
        "x": 0.4,
        "y": 0.215,
        "z": 0
      },
      {
        "x": 0.4,
        "y": -0.215,
        "z": 0
      },
      {
        "x": 0.35,
        "y": -0.265,
        "z": 0
      },
      {
        "x": -0.35,
        "y": -0.265,
        "z": 0
      }
    ]
  }
}
```

```

    },
    {
      "x": -0.4,
      "y": -0.215,
      "z": 0
    },
    {
      "x": -0.4,
      "y": 0.215,
      "z": 0
    },
    {
      "x": -0.35,
      "y": 0.265,
      "z": 0
    }
  ],
  "robotId": 30001
},
"msg": "",
"ok": true,
"type": "woosh.robot.Model"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.Model</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.Model</code>

POST Get Task History

POST /woosh/robot/TaskHistory

Task status information (last fifty records)

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none

Return example

Success

```
{
  "body": {
    "tes": [
      {
        "dest": "",
        "msg": "",
        "robotTaskId": "68242105502",
        "state": 7,
        "time": 1682421626,
        "type": 0
      },
      {
        "dest": "",
        "msg": "",
        "robotTaskId": "68242105503",
        "state": 7,
        "time": 1682421661,
        "type": 0
      },
      {
        "dest": "",
        "msg": "",
        "robotTaskId": "-68242105601",
        "state": 8,
        "time": 1682421673,
        "type": 0
      },
      {
        "dest": "",
        "msg": "",
        "robotTaskId": "68242105701",
        "state": 7,
        "time": 1682421718,
        "type": 0
      }
    ]
  }
}
```

```

{
  "dest": "",
  "msg": "",
  "robotTaskId": "68242105702",
  "state": 7,
  "time": 1682421823,
  "type": 0
},
{
  "dest": "",
  "msg": "",
  "robotTaskId": "68242105703",
  "state": 7,
  "time": 1682421910,
  "type": 0
},
{
  "dest": "",
  "msg": "",
  "robotTaskId": "68242105704",
  "state": 7,
  "time": 1682421949,
  "type": 0
},
{
  "dest": "",
  "msg": "",
  "robotTaskId": "68242105705",
  "state": 9,
  "time": 0,
  "type": 0
},
{
  "action": {
    "state": 3,
    "type": 1,
    "waitId": 0
  },
  "dest": "P2",
  "msg": "",
  "robotId": 30001,
  "robotTaskId": "-68242105801",
  "state": 8,
  "time": 1682422342,
  "type": 2
},
{
  "action": {
    "state": 4,
    "type": 6,
    "waitId": 1
  },
  "dest": "K11",
  "msg": "",
  "robotId": 30001,
  "robotTaskId": "68242105901",

```

```

    "state": 7,
    "time": 1682422438,
    "type": 1
  },
  {
    "action": {
      "state": 0,
      "type": 1,
      "waitId": 0
    },
    "dest": "U1",
    "msg": "",
    "robotTaskId": "68242105902",
    "state": 3,
    "time": 0,
    "type": 1
  }
]
},
"msg": "Request succeed",
"ok": true,
"type": "woosh.robot.TaskHistory"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.TaskHistory</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.TaskHistory</code>

POST Get status code information

POST /woosh/robot/count/StatusCodes

Body Request Parameters

```
{
  "robotId": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "scs": [
      {
        "code": "130104000061",
        "level": 1,
        "msg": "Go straight",
        "state": 4,
        "taskId": "68241437402",
        "time": "1682420920688",
        "type": 1
      },
      {
        "code": "130104000031",
        "level": 1,
        "msg": "Obstacle avoidance successful",
        "state": 4,
        "taskId": "68241437402",
        "time": "1682420920690",
        "type": 1
      },
      {
        "code": "140102000002",
        "level": 2,
        "msg": "Disconnected from dispatch",
        "state": 4,
        "taskId": "68241437402",
        "time": "1682421049593",
        "type": 1
      }
    ]
  }
}
```



```

    "code": "140102000002",
    "level": 2,
    "msg": "Disconnected from dispatch",
    "state": 5,
    "taskId": "68241437402",
    "time": "1682421050002",
    "type": 1
  },
  {
    "code": "140102000000",
    "level": 0,
    "msg": "Connected to dispatch",
    "state": 5,
    "taskId": "68241437402",
    "time": "1682421054840",
    "type": 1
  },
  {
    "code": "140300000000",
    "level": 0,
    "msg": "Navigation resumed",
    "state": 5,
    "taskId": "68241437402",
    "time": "1682421054845",
    "type": 1
  },
  {
    "code": "130104000000",
    "level": 0,
    "msg": "Received new navigation task, navigation and obstacle avoidance
started",
    "state": 4,
    "taskId": "68241437402",
    "time": "1682421054950",
    "type": 1
  }
]
},
"msg": "",
"ok": true,
"type": "woosh.robot.count.StatusCodes"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.count.StatusCodes</code>
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.count.StatusCodes</code>

POST Get abnormal code information

POST /woosh/robot/count/AbnormalCodes

Body Request Parameters

```
{
  "robotId": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "scs": []
  },
  "msg": "",
  "ok": true,
  "type": "woosh.robot.count.AbnormalCodes"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.count.AbnormalCodes</code>
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.count.AbnormalCodes</code>

POST Request navigation path

POST /woosh/robot/NavPath

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Request global planning path

POST /woosh/robot/PlanPath

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Robot Configuration

POST Set Robot Identity

POST /woosh/robot/setting/Identity

Body Request Parameters

```
{
  "robotId": 30001,
  "name": "A666"
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» name	body	string	Yes	Robot Name	none
» robotId	body	integer	Yes	Robot ID	only assign when requesting to the dispatch system

Return example

Success

```
{
  "body": {
    "name": "A666",
    "robotId": 30001
  },
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.setting.Identity"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response Message Body	See dictionary <code>woosh.robot.setting.Identity</code>
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure

Name	Type	Required	Constraint	Chinese name	Explanation
» type	string	true	none	Response Message Type	woosh.robot.setting.Identity

POST Set Server Connection Configuration

POST /woosh/robot/setting/Server

Body Request Parameters

```
{
  "robotId": 30001,
  "ip": "172.20.8.85",
  "port": 5420
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» ip	body	string	Yes	Dispatch IP	none
» port	body	integer	Yes	Dispatch Port	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "ip": "172.20.8.85",
    "port": 5420,
    "robotId": 30001
  },
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.setting.Server"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response Message Body	See dictionary <code>woosh.robot.setting.Server</code>
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response Message Type	<code>woosh.robot.setting.Server</code>

POST Switch Autonomous Recharge

POST /woosh/robot/setting/AutoCharge

Body Request Parameters

```
{
  "robotId": 30001,
  "allow": true
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» allow	body	boolean	is	Whether to allow	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "allow": true,
    "robotId": 30001
  },
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.setting.AutoCharge"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.setting.AutoCharge</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.setting.AutoCharge</code>

POST Switch Autonomous Parking

POST /woosh/robot/setting/AutoPark

Body Request Parameters

```
{
  "allow": true,
  "robotId": 0
}
```


Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» allow	body	boolean	is	Whether to allow	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "allow": true,
    "robotId": 30001
  },
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.setting.AutoPark"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.setting.AutoPark</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure

Name	Type	Required	Constraint	Chinese name	Explanation
» type	string	true	none	Response message type	woosh.robot.setting.AutoPark

POST Switch Goods Detection

POST /woosh/robot/setting/GoodsCheck

Body Request Parameters

```
{
  "allow": true,
  "robotId": 0
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» allow	body	boolean	is	Whether to allow	none
» robotId	body	integer	No	Robot ID	Only required when requesting the dispatch system

Return example

Success

```
{
  "body": {
    "allow": true,
    "robotId": 30001
  },
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.setting.GoodsCheck"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.setting.GoodsCheck</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.setting.GoodsCheck</code>

POST Robot Power Configuration

POST /woosh/robot/setting/Power

Body Request Parameters

```
{
  "robotId": 30001,
  "alarm": 10,
  "low": 20,
  "idle": 80,
  "full": 98
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» alarm	body	integer	is	Warning power value	none
» low	body	integer	is	Low power value	none
» idle	body	integer	is	Idle power value	none
» full	body	integer	is	Full power value	none

Name	Location	Type	Required	Chinese Name	Description
» robotId	body	integer	is	Robot ID	none

Return example

Success

```
{
  "body": {
    "alarm": 10,
    "full": 98,
    "idle": 80,
    "low": 20,
    "robotId": 30001
  },
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.setting.Power"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.setting.Power</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.setting.Power</code>

Scene Map

POST Get Scene List

POST /woosh/map/SceneList

Body Request Parameters

{}

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none

Return example

Success

```
{
  "body": {
    "scenes": [
      {
        "maps": [
          "wooshmap"
        ],
        "name": "wooshmap"
      },
      {
        "maps": [
          "t"
        ],
        "name": "test"
      }
    ]
  },
  "msg": "",
  "ok": true,
  "type": "woosh.map.SceneList"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.map.SceneList</code> for details

Name	Type	Required	Constraint	Chinese name	Explanation
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.map.SceneList</code>

POST Get scene data

POST /woosh/map/SceneData

Body Request Parameters

```
{
  "name": "wooshmap"
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» name	body	string	true	none	Scene name

Return example

Success

```
{
  "body": {
    "maps": [
      {
        "end": {
          "theta": 0.473383,
          "x": -2.57125568,
          "y": -8.37596226
        },
        "id": 0,
        "keepoutPng": "",
        "mapPng": {},
        "markInfo": {},
        "name": "wooshmap",
        "origin": {
          "theta": 0,
          "x": -13.8952227,
          "y": -16.3379803
        }
      }
    ]
  }
}
```

```

    },
    "resolution": 0.05,
    "timewrite": "1678843323",
    "traffickeepoutPng": ""
  }
],
"name": "wooshmap",
"taskInfo": {},
"actionGroup": ""
},
"msg": "Request succeed",
"ok": true,
"type": "woosh.map.SceneData"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.map.SceneData</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.map.SceneData</code>

POST Get scene data (Easy)

POST /woosh/map/SceneDataEasy

Body Request Parameters

```

{
  "name": ""
}

```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» name	body	string	true	none	Scene name

Return example

Success

```
{
  "body": {
    "maps": [
      {
        "floor": "",
        "id": 0,
        "name": "L1",
        "storages": {
          "armWorktops": [],
          "bases": [
            {
              "custom": "",
              "identity": {
                "desc": "",
                "id": 729210423,
                "no": "Department 1"
              },
              "pose": {
                "dock": {
                  "theta": 0.0243854094,
                  "x": 4.06974411,
                  "y": 2.65976524
                },
                "real": {
                  "theta": -0.0166651234,
                  "x": 4.09826422,
                  "y": 2.67348981
                }
              }
            }
          ]
        }
      },
      {
        "custom": "",
        "identity": {
          "desc": "",
          "id": 729212789,
          "no": "Department 2"
        },
        "pose": {
          "dock": {
            "theta": 0,
            "x": 7.93,
            "y": 2.59
          }
        }
      }
    ]
  }
}
```



```

        "real": {
            "theta": 3.13,
            "x": 7.93055534,
            "y": 2.6125
        }
    },
    {
        "custom": "",
        "identity": {
            "desc": "",
            "id": 729218198,
            "no": "Delivery point"
        },
        "pose": {
            "dock": {
                "theta": 3.12,
                "x": 1.46,
                "y": -0.38
            },
            "real": {
                "theta": 3.14,
                "x": 1.49546933,
                "y": -0.381765574
            }
        }
    }
],
"chargers": [
    {
        "custom": "",
        "identity": {
            "desc": "",
            "id": 2327928145,
            "no": "Charging station 10001"
        },
        "pose": {
            "dock": {
                "theta": 3.1415925,
                "x": 2.83154774,
                "y": -1.70588875
            },
            "real": {
                "theta": 3.14,
                "x": 4.33,
                "y": -1.71
            }
        }
    }
],
"liftTrolleys": [],
"palletPlatforms": [],
"parkspots": [
    {
        "custom": "",
        "identity": {

```

```

        "desc": "",
        "id": 2144646872,
        "no": "Parking spot 10001"
    },
    "pose": {
        "dock": {
            "theta": -3.13739085,
            "x": 2.96408176,
            "y": -1.73506939
        },
        "real": {
            "theta": -3.13,
            "x": 2.54,
            "y": -1.7299999
        }
    }
},
"racks": [],
"rollerStations": [],
"tractorTrailers": []
},
"version": "249"
}
],
"name": "lejuwoosh"
},
"msg": "",
"ok": true,
"type": "woosh.map.SceneDataEasy"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.map.SceneDataEasy</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure

Name	Type	Required	Constraint	Chinese name	Explanation
» type	string	true	none	Response message type	<code>woosh.map.SceneData</code>

POST Download map

POST /woosh/map/Download

Body Request Parameters

```
{
  "sceneName": "wooshmap"
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none
» sceneName	body	string	true	none	Scene name

Return example

Success

```
{
  "body": {},
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.map.DownloadResponse"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.map.DownloadResponse</code> for details

Name	Type	Required	Constraint	Chinese name	Explanation
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.map.DownloadResponse</code>

POST Upload Map

POST /woosh/map/Upload

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Modify Map or Scene Name

POST /woosh/map/Rename

Body Request Parameters

```
{
  "old_scene_name": "string",
  "new_scene_name": "string",
  "old_map_name": "string",
  "new_map_name": "string"
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none
» old_scene_name	body	string	No		none
» new_scene_name	body	string	No		none
» old_map_name	body	string	No		none
» new_map_name	body	string	No		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Delete Scene or Map

POST /woosh/map/Delete

Body Request Parameters

```
{
  "scene_name": "string",
  "map_name": "string"
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none
» scene_name	body	string	Yes		none
» map_name	body	string	No		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Scene File MD5 Request

POST /woosh/map/SceneMd5

Body Request Parameters

```
{
  "sceneName": "wooshmap"
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none
» sceneName	body	string	Yes		none

Return example

Success

```
{
  "body": {
    "scenes": [
      {
        "maps": [
          {
            "mapMd5s": [
```

```

{
  "md5": "d21f307ffd6bf86ca1c17ef684248d33",
  "name": "traffic_keepout_w.png"
},
{
  "md5": "d21f307ffd6bf86ca1c17ef684248d33",
  "name": "keepout.png"
},
{
  "md5": "d21f307ffd6bf86ca1c17ef684248d33",
  "name": "keepout_w.png"
},
{
  "md5": "d21f307ffd6bf86ca1c17ef684248d33",
  "name": "localization_map.png"
},
{
  "md5": "d21f307ffd6bf86ca1c17ef684248d33",
  "name": "localization_map.yaml"
},
{
  "md5": "d21f307ffd6bf86ca1c17ef684248d33",
  "name": "localization_map_w.png"
},
{
  "md5": "d21f307ffd6bf86ca1c17ef684248d33",
  "name": "map.png"
},
{
  "md5": "d21f307ffd6bf86ca1c17ef684248d33",
  "name": "map.yaml"
},
{
  "md5": "d21f307ffd6bf86ca1c17ef684248d33",
  "name": "map_info.data"
},
{
  "md5": "d21f307ffd6bf86ca1c17ef684248d33",
  "name": "map_w.png"
},
{
  "md5": "d21f307ffd6bf86ca1c17ef684248d33",
  "name": "mark_info.data"
},
{
  "md5": "d21f307ffd6bf86ca1c17ef684248d33",
  "name": "nav_area.data"
},
{
  "md5": "d21f307ffd6bf86ca1c17ef684248d33",
  "name": "path_info.data"
},
{
  "md5": "d21f307ffd6bf86ca1c17ef684248d33",
  "name": "traffic_keepout.png"
}

```

```

    ],
    "mapName": "wooshmap",
    "version": 140
  }
],
"sceneName": "wooshmap",
"scenesMd5s": [
  {
    "md5": "d21f307ffd6bf86ca1c17ef684248d33",
    "name": "action_groups.json"
  },
  {
    "md5": "d21f307ffd6bf86ca1c17ef684248d33",
    "name": "scene.json"
  },
  {
    "md5": "d21f307ffd6bf86ca1c17ef684248d33",
    "name": "task_info.data"
  }
]
}
]
},
"msg": "",
"ok": true,
"type": "woosh.map.SceneMd5Response"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none		none
» msg	string	true	none		none
» ok	boolean	true	none		none
» type	string	true	none		none

POST Scene Synchronization

POST /woosh/map/SceneSync

Body Request Parameters


```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Get Task Information (Predefined)

POST /woosh/task/TaskInfo

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

Success

```
{
  "body": {
    "callTasks": [
      {
        "callInfo": {
          "events": [
            {
              "caller": {
                "keyId": 1,
                "macAddr": "AA-BB-CC-DD-EE-A2",
                "triggered": false
              }
            }
          ]
        }
      }
    ]
  }
}
```

```

        "guard": {
            "callers": [],
            "type": 1
        },
        "release": 0
    }
],
"guard": {
    "callers": [
        {
            "keyId": 1,
            "macAddr": "AA-BB-CC-DD-EE-A1",
            "triggered": false
        }
    ],
    "type": 1
},
"type": 1
},
"taskSet": {
    "base": {
        "actuator": "",
        "adapter": "",
        "id": -2060076769,
        "name": "CallTask",
        "no": "",
        "priority": 0,
        "robots": [],
        "route": "",
        "rtype": 1,
        "type": ""
    },
    "custom": {
        "autoComplete": true
    },
    "tasks": [
        {
            "base": {
                "cannotCancel": false,
                "custom": "",
                "direction": 0,
                "id": 1,
                "markNo": "Operating Room 3",
                "name": "Task 1",
                "type": 1,
                "typeNo": 0,
                "waitTime": 0
            },
            "consta": {
                "canSkip": false,
                "cancelEnd": false,
                "cancelNext": false,
                "cancelSkip": false,
                "failEnd": false,
                "failNext": false,
                "failRedo": false,

```

```

        "failSkip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
}
]
}
],
"repeatTasks": [
{
    "taskSet": {
        "base": {
            "actuator": "",
            "adapter": "",
            "id": -2127845149,
            "name": "PRETASK-1",
            "no": "",
            "priority": 0,
            "robots": [],
            "route": "",
            "rtype": 21,
            "type": ""
        },
        "custom": {
            "autoComplete": true
        },
        "tasks": [
            {
                "base": {
                    "cannotCancel": false,
                    "custom": "",
                    "direction": 0,
                    "id": 1,
                    "markNo": "Anesthesia Preparation Room",
                    "name": "Task 1",
                    "type": 1,
                    "typeNo": 0,
                    "waitTime": 0
                },
                "consta": {
                    "canSkip": false,
                    "cancelEnd": true,
                    "cancelNext": false,
                    "cancelSkip": false,
                    "failEnd": true,
                    "failNext": false,
                    "failRedo": false,
                    "failSkip": false
                },
                "custom": {
                    "executable": true,
                    "priority": 0
                }
            }
        ]
    }
}
]

```

```

    },
    {
      "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 2,
        "markNo": "Nurse Station",
        "name": "Task 2",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
      },
      "consta": {
        "canSkip": false,
        "cancelEnd": false,
        "cancelNext": false,
        "cancelSkip": false,
        "failEnd": false,
        "failNext": false,
        "failRedo": false,
        "failSkip": false
      },
      "custom": {
        "executable": true,
        "priority": 0
      }
    }
  ]
},
"times": -1
},
{
  "taskSet": {
    "base": {
      "actuator": "",
      "adapter": "",
      "id": -2126068894,
      "name": "PRETASK-2",
      "no": "",
      "priority": 0,
      "robots": [],
      "route": "",
      "rtype": 21,
      "type": ""
    },
    "custom": {
      "autoComplete": true
    },
    "tasks": [
      {
        "base": {
          "cannotCancel": false,
          "custom": "",
          "direction": 0,
          "id": 1,

```

```

        "markNo": "Operating Room 5",
        "name": "Task 1",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
    },
    "consta": {
        "canSkip": false,
        "cancelEnd": false,
        "cancelNext": false,
        "cancelSkip": false,
        "failEnd": false,
        "failNext": false,
        "failRedo": false,
        "failSkip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
},
{
    "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 2,
        "markNo": "Operating Room 3",
        "name": "Task 2",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
    },
    "consta": {
        "canSkip": false,
        "cancelEnd": false,
        "cancelNext": false,
        "cancelSkip": false,
        "failEnd": false,
        "failNext": false,
        "failRedo": false,
        "failSkip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
}
]
},
"times": -1
}
]
},
"msg": "",
"ok": true,

```

```
"type": "woosh.task.TaskInfo"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Get Loop Task (Predefined)

POST /woosh/task/RepeatTasks

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

Success

```
{
  "body": {
    "callTasks": [
      {
        "callInfo": {
          "events": [
            {
              "caller": {
                "keyId": 1,
                "macAddr": "AA-BB-CC-DD-EE-A2",
                "triggered": false
              },
              "guard": {
                "callers": [],
                "type": 1
              },
              "release": 0
            }
          ],
          "guard": {
            "callers": [
              {
                "keyId": 1,
                "macAddr": "AA-BB-CC-DD-EE-A1",
```

```

        "triggered": false
    }
],
    "type": 1
},
    "type": 1
},
"taskSet": {
    "base": {
        "actuator": "",
        "adapter": "",
        "id": -2060076769,
        "name": "CallTask",
        "no": "",
        "priority": 0,
        "robots": [],
        "route": "",
        "rtype": 1,
        "type": ""
    },
    "custom": {
        "autoComplete": true
    },
    "tasks": [
        {
            "base": {
                "cannotCancel": false,
                "custom": "",
                "direction": 0,
                "id": 1,
                "markNo": "Operating Room 3",
                "name": "Task 1",
                "type": 1,
                "typeNo": 0,
                "waitTime": 0
            },
            "consta": {
                "canSkip": false,
                "cancelEnd": false,
                "cancelNext": false,
                "cancelSkip": false,
                "failEnd": false,
                "failNext": false,
                "failRedo": false,
                "failSkip": false
            },
            "custom": {
                "executable": true,
                "priority": 0
            }
        }
    ]
}
},
    "repeatTasks": [

```

```

{
  "taskSet": {
    "base": {
      "actuator": "",
      "adapter": "",
      "id": -2127845149,
      "name": "PRETASK-1",
      "no": "",
      "priority": 0,
      "robots": [],
      "route": "",
      "rtype": 21,
      "type": ""
    },
    "custom": {
      "autoComplete": true
    },
    "tasks": [
      {
        "base": {
          "cannotCancel": false,
          "custom": "",
          "direction": 0,
          "id": 1,
          "markNo": "Anesthesia Preparation Room",
          "name": "Task 1",
          "type": 1,
          "typeNo": 0,
          "waitTime": 0
        },
        "consta": {
          "canSkip": false,
          "cancelEnd": true,
          "cancelNext": false,
          "cancelSkip": false,
          "failEnd": true,
          "failNext": false,
          "failRedo": false,
          "failSkip": false
        },
        "custom": {
          "executable": true,
          "priority": 0
        }
      },
      {
        "base": {
          "cannotCancel": false,
          "custom": "",
          "direction": 0,
          "id": 2,
          "markNo": "Nurse Station",
          "name": "Task 2",
          "type": 1,
          "typeNo": 0,
          "waitTime": 0
        }
      }
    ]
  }
}

```



```

    },
    "consta": {
      "canSkip": false,
      "cancelEnd": false,
      "cancelNext": false,
      "cancelSkip": false,
      "failEnd": false,
      "failNext": false,
      "failRedo": false,
      "failSkip": false
    },
    "custom": {
      "executable": true,
      "priority": 0
    }
  }
]
},
"times": -1
},
{
  "taskSet": {
    "base": {
      "actuator": "",
      "adapter": "",
      "id": -2126068894,
      "name": "PRETASK-2",
      "no": "",
      "priority": 0,
      "robots": [],
      "route": "",
      "rtype": 21,
      "type": ""
    },
    "custom": {
      "autoComplete": true
    },
    "tasks": [
      {
        "base": {
          "cannotCancel": false,
          "custom": "",
          "direction": 0,
          "id": 1,
          "markNo": "Operating Room 5",
          "name": "Task 1",
          "type": 1,
          "typeNo": 0,
          "waitTime": 0
        },
        "consta": {
          "canSkip": false,
          "cancelEnd": false,
          "cancelNext": false,
          "cancelSkip": false,
          "failEnd": false,

```

```

        "failNext": false,
        "failRedo": false,
        "failSkip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
},
{
    "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 2,
        "markNo": "Operating Room 3",
        "name": "Task 2",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
    },
    "consta": {
        "canSkip": false,
        "cancelEnd": false,
        "cancelNext": false,
        "cancelSkip": false,
        "failEnd": false,
        "failNext": false,
        "failRedo": false,
        "failSkip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
}
]
},
"times": -1
}
]
},
"msg": "",
"ok": true,
"type": "woosh.task.TaskInfo"
}

```

```

{
  "body": {
    "repeatTasks": [
      {
        "taskSet": {
          "base": {
            "actuator": "",
            "adapter": "",

```

```

    "id": -2127845149,
    "name": "PRETASK-1",
    "no": "",
    "priority": 0,
    "robots": [],
    "route": "",
    "rtype": 21,
    "type": ""
  },
  "custom": {
    "autoComplete": true
  },
  "tasks": [
    {
      "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 1,
        "markNo": "Anesthesia Preparation Room",
        "name": "Task 1",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
      },
      "consta": {
        "canSkip": false,
        "cancelEnd": true,
        "cancelNext": false,
        "cancelSkip": false,
        "failEnd": true,
        "failNext": false,
        "failRedo": false,
        "failSkip": false
      },
      "custom": {
        "executable": true,
        "priority": 0
      }
    },
    {
      "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 2,
        "markNo": "Nurse Station",
        "name": "Task 2",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
      },
      "consta": {
        "canSkip": false,
        "cancelEnd": false,
        "cancelNext": false,

```

```

        "cancelSkip": false,
        "failEnd": false,
        "failNext": false,
        "failRedo": false,
        "failSkip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
}
]
},
"times": -1
},
{
    "taskSet": {
        "base": {
            "actuator": "",
            "adapter": "",
            "id": -2126068894,
            "name": "PRETASK-2",
            "no": "",
            "priority": 0,
            "robots": [],
            "route": "",
            "rtype": 21,
            "type": ""
        },
        "custom": {
            "autoComplete": true
        },
        "tasks": [
            {
                "base": {
                    "cannotCancel": false,
                    "custom": "",
                    "direction": 0,
                    "id": 1,
                    "markNo": "Operating Room 5",
                    "name": "Task 1",
                    "type": 1,
                    "typeNo": 0,
                    "waitTime": 0
                },
                "consta": {
                    "canSkip": false,
                    "cancelEnd": false,
                    "cancelNext": false,
                    "cancelSkip": false,
                    "failEnd": false,
                    "failNext": false,
                    "failRedo": false,
                    "failSkip": false
                },
                "custom": {

```

```

        "executable": true,
        "priority": 0
    },
    {
        "base": {
            "cannotCancel": false,
            "custom": "",
            "direction": 0,
            "id": 2,
            "markNo": "Operating Room 3",
            "name": "Task 2",
            "type": 1,
            "typeNo": 0,
            "waitTime": 0
        },
        "consta": {
            "canSkip": false,
            "cancelEnd": false,
            "cancelNext": false,
            "cancelSkip": false,
            "failEnd": false,
            "failNext": false,
            "failRedo": false,
            "failSkip": false
        },
        "custom": {
            "executable": true,
            "priority": 0
        }
    }
],
},
"times": -1
}
]
},
"msg": "",
"ok": true,
"type": "woosh.task.RepeatTasks"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Get Call Task (Predefined)

POST /woosh/task/CallTasks

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

Success

```
{
  "body": {
    "callTasks": [
      {
        "callInfo": {
          "events": [
            {
              "caller": {
                "keyId": 1,
                "macAddr": "AA-BB-CC-DD-EE-A2",
                "triggered": false
              },
              "guard": {
                "callers": [],
                "type": 1
              },
              "release": 0
            }
          ],
          "guard": {
            "callers": [
              {
                "keyId": 1,
                "macAddr": "AA-BB-CC-DD-EE-A1",
                "triggered": false
              }
            ],
            "type": 1
          },
          "type": 1
        },
        "taskSet": {
          "base": {
            "actuator": "",
            "adapter": "",
            "id": -2060076769,
            "name": "CallTask",
            "no": "",
            "priority": 0,
            "robots": []
          }
        }
      }
    ]
  }
}
```

```

        "route": "",
        "rtype": 1,
        "type": ""
    },
    "custom": {
        "autoComplete": true
    },
    "tasks": [
        {
            "base": {
                "cannotCancel": false,
                "custom": "",
                "direction": 0,
                "id": 1,
                "markNo": "Operating Room 3",
                "name": "Task 1",
                "type": 1,
                "typeNo": 0,
                "waitTime": 0
            },
            "consta": {
                "canSkip": false,
                "cancelEnd": false,
                "cancelNext": false,
                "cancelSkip": false,
                "failEnd": false,
                "failNext": false,
                "failRedo": false,
                "failSkip": false
            },
            "custom": {
                "executable": true,
                "priority": 0
            }
        }
    ]
},
],
"repeatTasks": [
    {
        "taskSet": {
            "base": {
                "actuator": "",
                "adapter": "",
                "id": -2127845149,
                "name": "PRETASK-1",
                "no": "",
                "priority": 0,
                "robots": [],
                "route": "",
                "rtype": 21,
                "type": ""
            },
            "custom": {
                "autoComplete": true
            }
        }
    }
]

```

```

},
"tasks": [
  {
    "base": {
      "cannotCancel": false,
      "custom": "",
      "direction": 0,
      "id": 1,
      "markNo": "Anesthesia Preparation Room",
      "name": "Task 1",
      "type": 1,
      "typeNo": 0,
      "waitTime": 0
    },
    "consta": {
      "canSkip": false,
      "cancelEnd": true,
      "cancelNext": false,
      "cancelSkip": false,
      "failEnd": true,
      "failNext": false,
      "failRedo": false,
      "failSkip": false
    },
    "custom": {
      "executable": true,
      "priority": 0
    }
  },
  {
    "base": {
      "cannotCancel": false,
      "custom": "",
      "direction": 0,
      "id": 2,
      "markNo": "Nurse Station",
      "name": "Task 2",
      "type": 1,
      "typeNo": 0,
      "waitTime": 0
    },
    "consta": {
      "canSkip": false,
      "cancelEnd": false,
      "cancelNext": false,
      "cancelSkip": false,
      "failEnd": false,
      "failNext": false,
      "failRedo": false,
      "failSkip": false
    },
    "custom": {
      "executable": true,
      "priority": 0
    }
  }
]

```



```

    ]
  },
  "times": -1
},
{
  "taskSet": {
    "base": {
      "actuator": "",
      "adapter": "",
      "id": -2126068894,
      "name": "PRETASK-2",
      "no": "",
      "priority": 0,
      "robots": [],
      "route": "",
      "rtype": 21,
      "type": ""
    },
    "custom": {
      "autoComplete": true
    },
    "tasks": [
      {
        "base": {
          "cannotCancel": false,
          "custom": "",
          "direction": 0,
          "id": 1,
          "markNo": "Operating Room 5",
          "name": "Task 1",
          "type": 1,
          "typeNo": 0,
          "waitTime": 0
        },
        "consta": {
          "canSkip": false,
          "cancelEnd": false,
          "cancelNext": false,
          "cancelSkip": false,
          "failEnd": false,
          "failNext": false,
          "failRedo": false,
          "failSkip": false
        },
        "custom": {
          "executable": true,
          "priority": 0
        }
      },
      {
        "base": {
          "cannotCancel": false,
          "custom": "",
          "direction": 0,
          "id": 2,
          "markNo": "Operating Room 3",

```

```

        "name": "Task 2",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
    },
    "consta": {
        "canSkip": false,
        "cancelEnd": false,
        "cancelNext": false,
        "cancelSkip": false,
        "failEnd": false,
        "failNext": false,
        "failRedo": false,
        "failSkip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
}
]
},
"times": -1
}
]
},
"msg": "",
"ok": true,
"type": "woosh.task.TaskInfo"
}

```

```

{
  "body": {
    "repeatTasks": [
      {
        "taskSet": {
          "base": {
            "actuator": "",
            "adapter": "",
            "id": -2127845149,
            "name": "PRETASK-1",
            "no": "",
            "priority": 0,
            "robots": [],
            "route": "",
            "rtype": 21,
            "type": ""
          },
          "custom": {
            "autoComplete": true
          },
          "tasks": [
            {
              "base": {
                "cannotCancel": false,

```

```

        "custom": "",
        "direction": 0,
        "id": 1,
        "markNo": "Anesthesia Preparation Room",
        "name": "Task 1",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
    },
    "consta": {
        "canSkip": false,
        "cancelEnd": true,
        "cancelNext": false,
        "cancelSkip": false,
        "failEnd": true,
        "failNext": false,
        "failRedo": false,
        "failSkip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
},
{
    "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 2,
        "markNo": "Nurse Station",
        "name": "Task 2",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
    },
    "consta": {
        "canSkip": false,
        "cancelEnd": false,
        "cancelNext": false,
        "cancelSkip": false,
        "failEnd": false,
        "failNext": false,
        "failRedo": false,
        "failSkip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
}
]
},
"times": -1
},
{

```

```

"taskSet": {
  "base": {
    "actuator": "",
    "adapter": "",
    "id": -2126068894,
    "name": "PRETASK-2",
    "no": "",
    "priority": 0,
    "robots": [],
    "route": "",
    "rtype": 21,
    "type": ""
  },
  "custom": {
    "autoComplete": true
  },
  "tasks": [
    {
      "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 1,
        "markNo": "Operating Room 5",
        "name": "Task 1",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
      },
      "consta": {
        "canSkip": false,
        "cancelEnd": false,
        "cancelNext": false,
        "cancelSkip": false,
        "failEnd": false,
        "failNext": false,
        "failRedo": false,
        "failSkip": false
      },
      "custom": {
        "executable": true,
        "priority": 0
      }
    },
    {
      "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 2,
        "markNo": "Operating Room 3",
        "name": "Task 2",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
      },

```

```

        "consta": {
            "canSkip": false,
            "cancelEnd": false,
            "cancelNext": false,
            "cancelSkip": false,
            "failEnd": false,
            "failNext": false,
            "failRedo": false,
            "failSkip": false
        },
        "custom": {
            "executable": true,
            "priority": 0
        }
    }
    ],
    },
    "times": -1
}
]
},
"msg": "",
"ok": true,
"type": "woosh.task.RepeatTasks"
}

```

```

{
  "body": {
    "callTasks": [
      {
        "callInfo": {
          "events": [
            {
              "caller": {
                "keyId": 1,
                "macAddr": "AA-BB-CC-DD-EE-A2",
                "triggered": false
              },
              "guard": {
                "callers": [],
                "type": 1
              },
              "release": 0
            }
          ],
          "guard": {
            "callers": [
              {
                "keyId": 1,
                "macAddr": "AA-BB-CC-DD-EE-A1",
                "triggered": false
              }
            ],
            "type": 1
          }
        }
      }
    ],
    "type": 1
  },

```

```

    "type": 1
  },
  "taskSet": {
    "base": {
      "actuator": "",
      "adapter": "",
      "id": -2060076769,
      "name": "CallTask",
      "no": "",
      "priority": 0,
      "robots": [],
      "route": "",
      "rtype": 1,
      "type": ""
    },
    "custom": {
      "autoComplete": true
    },
    "tasks": [
      {
        "base": {
          "cannotCancel": false,
          "custom": "",
          "direction": 0,
          "id": 1,
          "markNo": "Operating Room 3",
          "name": "Task 1",
          "type": 1,
          "typeNo": 0,
          "waitTime": 0
        },
        "consta": {
          "canSkip": false,
          "cancelEnd": false,
          "cancelNext": false,
          "cancelSkip": false,
          "failEnd": false,
          "failNext": false,
          "failRedo": false,
          "failSkip": false
        },
        "custom": {
          "executable": true,
          "priority": 0
        }
      }
    ]
  },
  "msg": "",
  "ok": true,
  "type": "woosh.task.CallTasks"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Robot Request

POST Initialize Robot

POST /woosh/robot/InitRobot

Body Request Parameters

```
{
  "isRecord": true
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» isRecord	body	boolean	No	Whether to reset at record point	If <code>TRUE</code> , initialize the robot with the reset point

Return example

Success

```
{
  "body": {},
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.InitRobot"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	Empty
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.InitRobot</code>

POST Set Robot Pose

POST /woosh/robot/SetRobotPose

Body Request Parameters

```
{
  "pose": {
    "x": 12.3,
    "y": 5.6,
    "theta": 1.57
  }
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» pose	body	object	Yes	Pose	Empty, default initialization to map origin <0, 0, 0>
»» x	body	number	is	x	none
»» y	body	number	is	y	none
»» theta	body	number	is	orientation	none

Return example

Success


```
{
  "body": {},
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.SetRobotPose"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	Empty
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	response message type	<code>woosh.robot.SetRobotPose</code>

POST Set Robot Occupancy

POST /woosh/robot/SetOccupancy

Body Request Parameters

```
{
  "occupy": true
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» occupy	body	boolean	is		none

Return example

Success

```
{
  "body": {},
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.SetOccupancy"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	Empty
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	response message type	<code>woosh.robot.SetRobotPose</code>

POST Switch Control Mode

POST /woosh/robot/SwitchControlMode

- **req:** `woosh::robot::SwitchControlMode`
- **rsp:** `woosh::robot::Mode`

Body Request Parameters

```
{
  "mode": 1
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» mode	body	integer	is	control mode	none

Return example

Success

```
{
  "body": {
    "ctrl": 2,
    "work": 3
  },
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.Mode"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.Mode</code> for details
»» ctrl	integer	true	none		none
»» work	integer	true	none		none
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.Mode</code>

POST Switch Work Mode

POST /woosh/robot/SwitchWorkMode

Body Request Parameters

```
{
  "mode": 3
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» mode	body	integer	is	work mode	none

Return example

Success

```
{
  "body": {
    "ctrl": 1,
    "work": 2
  },
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.Mode"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	response message body	see dictionary <code>woosh.robot.Mode</code>
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure

Name	Type	Required	Constraint	Chinese name	Explanation
» type	string	true	none	Response message type	<code>woosh.robot.Mode</code>

POST Switch Map

POST /woosh/robot/SwitchMap

Body Request Parameters

```
{
  "sceneName": "wooshmap",
  "mapName": "wooshmap"
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» sceneName	body	string	is	scene name	none
» mapName	body	string	No	Map name	none

Return example

Success

```
{
  "body": {},
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.SwitchMap"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	Empty
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.SwitchMap</code>

POST Construct Map Request

POST /woosh/robot/BuildMap

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Deployment Request

POST /woosh/robot/Deployment

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Radar Data Request

POST /woosh/robot/ScannerData

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Execute Predefined Task

POST /woosh/robot/ExecPreTask

Body Request Parameters

```
{
  "taskSetId": 0
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» taskSetId	body	integer	Yes	Task set ID	none

Return example

Success

```
{
  "body": {},
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.ExecPreTask"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	Empty
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure

Name	Type	Required	Constraint	Chinese name	Explanation
» type	string	true	none	Response message type	woosh.robot.ExecPreTask

POST Execute Task

POST /woosh/robot/ExecTask

Body Request Parameters

```
{
  "taskId": 0,
  "type": 0,
  "direction": 0,
  "taskTypeNo": 0,
  "markNo": "string"
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» taskId	body	integer	Yes	Task ID	none
» type	body	integer	Yes	Task type	none
» direction	body	integer	No	Task direction	none
» taskTypeNo	body	integer	No	Type combination	none
» markNo	body	string	Yes	Target point number	none

Return example

Success

```
{
  "body": {},
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.ExecTask"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	Empty
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.robot.ExecTask</code>

POST Action Command

POST /woosh/robot/ActionOrder

Body Request Parameters

```
{
  "order": 0
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» order	body	integer	yes	Command	See dictionary <code>woosh.action.order</code>

Return example

Success

```
{
  "body": {},
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.ActionOrder"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	Empty
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	woosh.robot.ActionOrder

POST Plan Navigation Path

POST /woosh/robot/PlanNavPath

Body Request Parameters

```
{
  "start": {
    "x": 0,
    "y": 0,
    "theta": 0
  },
  "end": {
    "x": 0,
    "y": 0,
    "theta": 0
  },
  "tolerance": 0
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» start	body	object	yes	Starting point	none
»» x	body	integer	yes		none
»» y	body	integer	yes		none
»» theta	body	integer	yes		none
» end	body	object	yes	Target point	none
»» x	body	number	yes		none
»» y	body	number	yes		none
»» theta	body	integer	yes		none
» tolerance	body	integer	yes	Allowable detour range	none

Return example

Success

```
{
  "body": {
    "path": {
      "poses": [
        {
          "theta": 0,
          "x": -0.0202224571,
          "y": -0.0129800271
        },
        {
          "theta": 0,
          "x": -0.0702224597,
          "y": -0.00829252694
        },
        {
          "theta": 0,
          "x": -0.118576623,
          "y": 0.0103710685
        },
        {
          "theta": 0,
          "x": -0.158427283,
          "y": 0.0394931808
        },
        {
          "theta": 0,
```

```

        "x": -0.183533967,
        "y": 0.0704372749
    },
    {
        "theta": 0,
        "x": -0.2,
        "y": 0.1
    }
]
},
"msg": "Request succeed",
"ok": true,
"type": "woosh.robot.NavPath"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.robot.NavPath</code> for details
»» path	object	true	none		none
»»» poses	[object]	true	none		none
»»»» theta	integer	true	none		none
»»»» x	number	true	none		none
»»»» y	number	true	none		none
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure

Name	Type	Required	Constraint	Chinese name	Explanation
» type	string	true	none	Response message type	<code>woosh.robot.NavPath</code>

POST Change Navigation Path

POST /woosh/robot/ChangeNavPath

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Change Navigation Mode

POST /woosh/robot/ChangeNavMode

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Voice Broadcast

POST /woosh/robot/Speak

Body Request Parameters

```
{
  "text": "start heading to the target point"
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» text	body	string	yes	Broadcast content	none

Return example

Success

```
{
  "body": {},
  "msg": "Request succeed",
  "ok": true,
  "type": "woosh.robot.Speak"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	Empty
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	woosh.robot.speak

POST Speed Control

POST /woosh/robot/Twist

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Follow

POST /woosh/robot/Follow

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST WiFi Request

POST /woosh/robot/RobotWiFi

Body Request Parameters

```
{
  "order": 2,
  "content":
  "eyJ1c3NpZCI6IldPTlNIIFBTSyIsImFldGhfYmV0aG9kIjoid3BhLXBzayIsImhpZGRlbiI6ZmFsc2Us
  Im1wNF9zZXROaw5ncyI6eyJtZXRob2QiOiJhdXRvIn0sIndpZm1fc2VjdXJpdHkiOnsicHNrIjoid29vc
  2hyb2JvdG1jczg4OCJ9fQ==",
  "isConnectNow": true
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Name	Location	Type	Required	Chinese Name	Description
» order	body	integer	yes	Request type	2: Add, 3: Reconnect
» content	body	string	yes	WiFi configuration information	encoded as json data in base64 format
» isConnectNow	body	boolean	is	Add and reconnect immediately	none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Dispatch request

POST Get dispatch robot list

POST /woosh/dispatch/robot/Robots

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none

Return example

Success

```
{
  "body": {
    "robots": [
      {
        "commu": {
```

```

        "ip": "172.20.254.63",
        "online": true,
        "port": 30001,
        "time": 1682474265
    },
    "id": 30001,
    "schedulable": true,
    "task": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 2,
        "markNo": "U3",
        "name": "U3",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
    },
    "taskSetId": 682421115
}
]
},
"msg": "",
"ok": true,
"type": "woosh.dispatch.robot.Robots"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.dispatch.robot.Robots</code> for details
»» robots	[object]	true	none	Robot list	none
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure

Name	Type	Required	Constraint	Chinese name	Explanation
» type	string	true	none	Response message type	woosh.dispatch.robot.Robots

POST Get dispatch robot information

POST /woosh/dispatch/robot/Robot

Body Request Parameters

```
{
  "id": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» id	body	integer	is	Robot ID	none

Return example

Success

```
{
  "body": {
    "commu": {
      "ip": "172.20.254.63",
      "online": true,
      "port": 30001,
      "time": 1682474265
    },
    "id": 30001,
    "schedulable": true,
    "task": {
      "cannotCancel": false,
      "custom": "",
      "direction": 0,
      "id": 5,
      "markNo": "k1",
      "name": "k1",
      "type": 1,
      "typeNo": 0,
      "waitTime": 0
    },
    "taskSetId": 682421115
  },
  "msg": "",
  "ok": true,
}
```

```
"type": "woosh.dispatch.robot.Robot"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.dispatch.robot.Robot</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.dispatch.robot.Robot</code>

POST Switch scene

POST /woosh/dispatch/system/SwitchScene

Body Request Parameters

```
{
  "name": "wooshmap"
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» name	body	string	true	none	Scene name

Return example

Success

```
{
  "body": {},
  "msg": "",
  "ok": true,
  "type": "woosh.dispatch.system.SwitchScene"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	Empty
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	none
» type	string	true	none	Response message type	woosh.dispatch.system.SwitchScene

POST Get current scene

POST /woosh/dispatch/system/SceneSettings

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none

Return example

Success

```
{
  "body": {
    "name": "wooshmap"
  },
  "msg": "",
  "ok": true,
  "type": "woosh.dispatch.system.SceneSettings"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.dispatch.system.SceneSettings</code> for details
»» name	string	true	none	Current scene name	none
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.dispatch.system.SceneSettings</code>

POST Specify robot charging

POST /woosh/dispatch/system/GotoCharge

Body Request Parameters

```
{
  "robot": 30001
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none

Name	Location	Type	Required	Chinese Name	Description
» robot	body	integer	yes	Robot ID	none

Return example

Success

```
{
  "body": {
    "no": "",
    "robot": 0
  },
  "msg": "",
  "ok": true,
  "type": "woosh.dispatch.system.GotoCharge"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	Empty
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	woosh.dispatch.system.GotoCharge

POST Charging configuration

POST /woosh/dispatch/system/ChargeSettings

Body Request Parameters


```
{
  "settings": [
    {
      "level": 1,
      "guardPower": 10,
      "lowPower": 20,
      "workPower": 60,
      "fullPower": 100,
      "time": 0
    }
  ]
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none
» settings	body	[object]	yes		none
»» level	body	integer	no	Automatic charging level	none
»» guardPower	body	integer	No	Guard Power	none
»» lowPower	body	integer	No	Low Power	none
»» workPower	body	integer	No	Work Power	none
»» fullPower	body	integer	No	Full Power	none
»» time	body	integer	No	Time Period	none

Return example

Success

```
{
  "body": {
    "settings": [
      {
        "fullPower": 100,
        "guardPower": 10,
        "level": 1,
        "lowPower": 20,
        "time": 0,
        "workPower": 60
      }
    ]
  },
  "msg": "",
  "ok": true,
}
```

```
"type": "woosh.dispatch.system.ChargeSettings"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST PacAccountList

POST /woosh/dispatch/system/PacAccountList

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST PacAccount

POST /woosh/dispatch/system/PacAccount

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Get Current Charging Configuration

POST /woosh/dispatch/system/ChargeSettings/Setting

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Dispatch Request / Dashboard

POST Average Time Consumption of Task Routes

POST /woosh/dispatch/kanban/RouteAvgs

Body Request Parameters

```
{
  "robots": [],
  "bucket": {
    "start": 0,
    "end": 0
  }
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none
» robots	body	[string]	Yes	Robot List	If empty, query all
» bucket	body	object	Yes	Time Period	If empty, query all
»» start	body	integer	Yes		none
»» end	body	integer	Yes		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Task Statistics (All)

POST /woosh/dispatch/kanban/TaskCount

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Robot Task Statistics Set

POST /woosh/dispatch/kanban/RobotTaskCounts

Body Request Parameters

```
{
  "robots": [
    "string"
  ],
  "bucket": {
    "start": 0,
    "end": 0
  }
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none
» robots	body	[string]	Yes	Robot List	If empty, query all
» bucket	body	object	Yes	Time Period	If empty, query all
»» start	body	integer	Yes		none
»» end	body	integer	Yes		none

Return example

200 Response

{}

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Daily Task Statistics Set (Internal)

POST /woosh/dispatch/kanban/DailyTasks

Body Request Parameters

{}

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

{}

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Exception Information Set

POST /woosh/dispatch/kanban/Abnormals

Body Request Parameters

{}

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Robot Runtime Statistics

POST /woosh/dispatch/kanban/RobotOperation

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Robot Runtime Statistics Set

POST /woosh/dispatch/kanban/RobotOperations

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Dispatch Request/Task

POST Find Task

POST /woosh/dispatch/task/FindTask

Body Request Parameters

```
{
  "id": 0,
  "state": {
    "gte": 1,
    "lte": 3
  },
  "time": {
    "gte": 0,
    "lte": 0
  },
  "sort": 0,
  "robot": 0,
  "route": "",
  "page": 0
}
```


}

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» id	body	integer	No	Task Set ID	If not empty, only query this task set
» page	body	integer	No	Pagination	If not specified, return the 0th page
» state	body	object	No	Task State	Query task set based on state condition
»» gte	body	integer	Yes		none
»» lte	body	integer	Yes		none
» time	body	object	No		Query task generation time range (in seconds)
»» gte	body	integer	Yes		none
»» lte	body	integer	Yes		none
» sort	body	integer	no	Sort method	none
» robot	body	integer	no	Robot ID	none
» route	body	string	no	Route	none
» onlyExternal	body	boolean	no	Only external	Filtering external tasks such as parking and charging

Return example

Success

```
{
  "body": {
    "tsets": [
      {
        "base": {
          "actuator": "woosh",
```

```

    "adapter": "RANDOM",
    "id": 682421117,
    "name": "RANDOM",
    "no": "682421117",
    "priority": 0,
    "robots": [],
    "route": "",
    "rtype": 1,
    "type": "woosh.task.WooshTaskSet"
  },
  "exec": {
    "curTaskId": 4,
    "preTaskId": 3,
    "robot": 30001,
    "state": 2
  },
  "tasks": [
    {
      "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 1,
        "markNo": "K7",
        "name": "K7",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
      },
      "exec": {
        "actionWaitId": 0,
        "state": 7
      }
    },
    {
      "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 2,
        "markNo": "B6",
        "name": "B6",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
      },
      "exec": {
        "actionWaitId": 0,
        "state": 7
      }
    }
  ],
  {
    "base": {
      "cannotCancel": false,
      "custom": "",
      "direction": 0,

```

```

        "id": 3,
        "markNo": "B2",
        "name": "B2",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
    },
    "exec": {
        "actionwaitId": 0,
        "state": 7
    }
},
{
    "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 4,
        "markNo": "K12",
        "name": "K12",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
    },
    "exec": {
        "actionwaitId": 0,
        "state": 3
    }
}
],
"time": {
    "end": 1682474826,
    "generate": 1682474595,
    "start": 1682474595
}
}
]
},
"msg": "",
"ok": true,
"type": "woosh.task.TaskSetList"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.task.TaskSetList</code> for details
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.task.TaskSetList</code>

POST Task Instruction

POST /woosh/dispatch/task/TaskOrder

Body Request Parameters

```
{
  "id": 681348376,
  "order": 2
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none
» id	body	integer	yes	Task set ID	none
» order	body	integer	yes	Instruction	none
» robot	body	integer	yes	Robot ID	none

Return example

Success

```
{
  "body": {
    "id": 0,
    "order": 0,
    "taskId": 0
  },
  "msg": "",
  "ok": true,
  "type": "woosh.dispatch.task.TaskOrder"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	Empty
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.dispatch.task.TaskOrder</code>

POST Top Task

POST /woosh/dispatch/task/StickTask

Body Request Parameters

```
{
  "id": 681348378,
  "unstick": false
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» id	body	integer	yes	Task set ID	none
» unstick	body	boolean	is	Unstick	none

Return example

Success

```
{
  "body": {
    "id": 0,
    "unstick": false
  },
  "msg": "",
  "ok": true,
  "type": "woosh.dispatch.task.StickTask"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	Empty
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.dispatch.task.StickTask</code>

POST Add Task Set

POST /woosh/task/AddTask

Body Request Parameters

```
{
  "tset": {
    "base": {
      "name": "A10-k11",
      "priority": 0,
      "robots": [],
      "rtype": 1
    },
    "custom": {
      "auto_complete": true
    },
    "tasks": [
      {
```

```

    "base": {
      "cannot_cancel": false,
      "custom": "",
      "direction": 0,
      "id": 1,
      "mark_no": "A10",
      "name": "A10",
      "type": 1,
      "type_no": 0,
      "wait_time": 0
    },
    "consta": {
      "can_skip": true,
      "cancel_end": true,
      "cancel_next": false,
      "cancel_skip": false,
      "fail_end": true,
      "fail_next": true,
      "fail_redo": false,
      "fail_skip": false
    },
    "custom": {
      "executable": true,
      "priority": 0
    }
  },
  {
    "base": {
      "cannot_cancel": false,
      "custom": "",
      "direction": 0,
      "id": 2,
      "mark_no": "K11",
      "name": "K11",
      "type": 1,
      "type_no": 0,
      "wait_time": 0
    },
    "consta": {
      "can_skip": true,
      "cancel_end": true,
      "cancel_next": false,
      "cancel_skip": false,
      "fail_end": true,
      "fail_next": true,
      "fail_redo": false,
      "fail_skip": false
    },
    "custom": {
      "executable": true,
      "priority": 0
    }
  }
]
}

```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» tset	body	object	is	Task set	none
»» base	body	object	is		none
»»» rtype	body	integer	is	Robot type	none
»»»» priority	body	integer	is	Priority	The higher the value, the higher the priority
»»»» robots	body	[string]	is	Specify robot ID	none
»»»» adapter	body	string	is		none
»»»» actuator	body	string	is		none
»»»» name	body	string	is		none
»»»» type	body	string	is		none
»» tasks	body	[object]	is	subtask collection	none
»»» base	body	object	is		none
»»»» id	body	integer	is	task ID	range 1~99, unique within task set
»»»»» name	body	string	is	name	none
»»»»» type	body	integer	is	task type	none
»»»»» direction	body	integer	is	action direction	none
»»»»» cannot_cancel	body	boolean	is		none
»»»»» custom	body	string	is		none
»»»»» mark_no	body	string	is		none
»»»»» type_no	body	integer	is		none
»»»»» wait_time	body	integer	is		none

Name	Location	Type	Required	Chinese Name	Description
»»» custom	body	object	is		none
»»»» executable	body	boolean	is		none
»»»» priority	body	integer	is		none
»»» consta	body	object	is		none
»»»» can_skip	body	boolean	is		none
»»»» cancel_end	body	boolean	is		none
»»»» cancel_next	body	boolean	is		none
»»»» cancel_skip	body	boolean	is		none
»»»» fail_end	body	boolean	is		none
»»»» fail_next	body	boolean	is		none
»»»» fail_redo	body	boolean	is		none
»»»» fail_skip	body	boolean	is		none
»» custom	body	object	is		none
»»» auto_complete	body	boolean	is		none

Return example

Success

```
{
  "body": {
    "tset": {
      "base": {
        "actuator": "woosh",
        "adapter": "woosh",
        "id": 682477540,
        "name": "A10-K11",
        "no": "",
        "priority": 0,
        "robots": [],
        "route": "",
        "rtype": 1,
        "type": "woosh.task.WooshTaskSet"
      },
    },
    "custom": {
      "autoComplete": true
    }
  }
}
```

```

},
"tasks": [
  {
    "base": {
      "cannotCancel": false,
      "custom": "",
      "direction": 0,
      "id": 1,
      "markNo": "A10",
      "name": "A10",
      "type": 1,
      "typeNo": 0,
      "waitTime": 0
    },
    "consta": {
      "canSkip": true,
      "cancelEnd": true,
      "cancelNext": false,
      "cancelSkip": false,
      "failEnd": true,
      "failNext": true,
      "failRedo": false,
      "failSkip": false
    },
    "custom": {
      "executable": true,
      "priority": 0
    }
  },
  {
    "base": {
      "cannotCancel": false,
      "custom": "",
      "direction": 0,
      "id": 2,
      "markNo": "K11",
      "name": "K11",
      "type": 1,
      "typeNo": 0,
      "waitTime": 0
    },
    "consta": {
      "canSkip": true,
      "cancelEnd": true,
      "cancelNext": false,
      "cancelSkip": false,
      "failEnd": true,
      "failNext": true,
      "failRedo": false,
      "failSkip": false
    },
    "custom": {
      "executable": true,
      "priority": 0
    }
  }
]
}

```

```

    ]
  }
},
"msg": "",
"ok": true,
"type": "woosh.task.AddTask"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.task.AddTask</code> for details
»» tset	object	true	none	Task collection	none
»»» base	object	true	none		none
»»»» actuator	string	true	none		none
»»»» adapter	string	true	none		none
»»»» id	integer	true	none		none
»»»» name	string	true	none		none
»»»» no	string	true	none		none
»»»» priority	integer	true	none		none
»»»» robots	[string]	true	none		none
»»»» route	string	true	none		none
»»»» rtype	integer	true	none		none
»»»» type	string	true	none		none
»»» custom	object	true	none		none
»»»» autoComplete	boolean	true	none		none
»»» tasks	[object]	true	none		none
»»»» base	object	true	none		none
»»»»» cannotCancel	boolean	true	none		none

Name	Type	Required	Constraint	Chinese name	Explanation
»»»»» custom	string	true	none		none
»»»»» direction	integer	true	none		none
»»»»» id	integer	true	none		none
»»»»» markNo	string	true	none		none
»»»»» name	string	true	none		none
»»»»» type	integer	true	none		none
»»»»» typeNo	integer	true	none		none
»»»»» waitTime	integer	true	none		none
»»»»» consta	object	true	none		none
»»»»» canSkip	boolean	true	none		none
»»»»» cancelEnd	boolean	true	none		none
»»»»» cancelNext	boolean	true	none		none
»»»»» cancelSkip	boolean	true	none		none
»»»»» failEnd	boolean	true	none		none
»»»»» failNext	boolean	true	none		none
»»»»» failRedo	boolean	true	none		none
»»»»» failSkip	boolean	true	none		none
»»»»» custom	object	true	none		none
»»»»» executable	boolean	true	none		none
»»»»» priority	integer	true	none		none
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.task.AddTask</code>

POST Insert Task

POST /woosh/task/InsertTask

Body Request Parameters

```

{
  "tset_id": 705112497,
  "task": {
    "base": {
      "cannot_cancel": false,
      "custom": "",
      "direction": 0,
      "id": 11,
      "mark_no": "B3",
      "name": "B3",
      "type": 1,
      "type_no": 0,
      "wait_time": 0
    },
    "consta": {
      "can_skip": true,
      "cancel_end": true,
      "cancel_next": false,
      "cancel_skip": false,
      "fail_end": true,
      "fail_next": true,
      "fail_redo": false,
      "fail_skip": false
    },
    "custom": {
      "executable": true,
      "priority": 0
    }
  }
}

```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» tset_id	body	integer	is		none
» task	body	object	is		none
»» base	body	object	is		none
»»» cannot_cancel	body	boolean	is		none
»»» custom	body	string	is		none
»»» direction	body	integer	is		none
»»» id	body	integer	is		none
»»» mark_no	body	string	is		none
»»» name	body	string	is		none

Name	Location	Type	Required	Chinese Name	Description
»»» type	body	integer	is		none
»»» type_no	body	integer	is		none
»»» wait_time	body	integer	is		none
»» consta	body	object	is		none
»»» can_skip	body	boolean	is		none
»»» cancel_end	body	boolean	is		none
»»» cancel_next	body	boolean	is		none
»»» cancel_skip	body	boolean	is		none
»»» fail_end	body	boolean	is		none
»»» fail_next	body	boolean	is		none
»»» fail_redo	body	boolean	is		none
»»» fail_skip	body	boolean	is		none
»» custom	body	object	is		none
»»» executable	body	boolean	is		none
»»» priority	body	integer	is		none

Return example

Success

```
{
  "body": {
    "tset": {
      "base": {
        "actuator": "woosh",
        "adapter": "woosh",
        "id": 682477540,
        "name": "A10-K11",
        "no": "",
        "priority": 0,
        "robots": [],
        "route": "",
        "rtype": 1,
        "type": "woosh.task.WooshTaskSet"
      },
      "custom": {
        "autoComplete": true
      },
      "tasks": [
        {
          "base": {
            "cannotCancel": false,
```

```

        "custom": "",
        "direction": 0,
        "id": 1,
        "markNo": "A10",
        "name": "A10",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
    },
    "consta": {
        "canSkip": true,
        "cancelEnd": true,
        "cancelNext": false,
        "cancelSkip": false,
        "failEnd": true,
        "failNext": true,
        "failRedo": false,
        "failSkip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
},
{
    "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 2,
        "markNo": "K11",
        "name": "K11",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
    },
    "consta": {
        "canSkip": true,
        "cancelEnd": true,
        "cancelNext": false,
        "cancelSkip": false,
        "failEnd": true,
        "failNext": true,
        "failRedo": false,
        "failSkip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
}
]
}
},
"msg": "",
"ok": true,

```

```
"type": "woosh.task.AddTask"
}
```

```
{
  "body": {
    "base": {
      "actuator": "woosh",
      "adapter": "wooshCall",
      "id": 705112497,
      "name": "A2-B1-A2",
      "no": "705112497",
      "priority": 0,
      "robots": [],
      "route": "",
      "rtype": 21,
      "type": ""
    },
    "custom": {
      "autoComplete": true
    },
    "exec": {
      "curTaskId": 1,
      "preTaskId": 0,
      "robot": 30001,
      "state": 2
    },
    "tasks": [
      {
        "base": {
          "cannotCancel": false,
          "custom": "",
          "direction": 1,
          "id": 1,
          "markNo": "A2",
          "name": "A2 loading",
          "type": 4,
          "typeNo": 0,
          "waitTime": 0
        },
        "consta": {
          "canSkip": false,
          "cancelEnd": false,
          "cancelNext": false,
          "cancelSkip": false,
          "failEnd": false,
          "failNext": false,
          "failRedo": true,
          "failSkip": false
        },
        "custom": {
          "executable": true,
          "priority": 0
        },
        "exec": {
          "actionWaitId": 0,

```



```

        "state": 3
    }
},
{
    "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 2,
        "markNo": "B1",
        "name": "B1 call",
        "type": 1,
        "typeNo": 0,
        "waitTime": -1
    },
    "consta": {
        "canSkip": false,
        "cancelEnd": false,
        "cancelNext": false,
        "cancelSkip": false,
        "failEnd": false,
        "failNext": false,
        "failRedo": true,
        "failSkip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
},
{
    "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 2,
        "id": 3,
        "markNo": "A2",
        "name": "A2 unloading",
        "type": 4,
        "typeNo": 0,
        "waitTime": 0
    },
    "consta": {
        "canSkip": false,
        "cancelEnd": false,
        "cancelNext": false,
        "cancelSkip": false,
        "failEnd": false,
        "failNext": false,
        "failRedo": true,
        "failSkip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
}

```

```

    },
    {
      "base": {
        "cannotCancel": false,
        "custom": "",
        "direction": 0,
        "id": 11,
        "markNo": "B3",
        "name": "B3",
        "type": 1,
        "typeNo": 0,
        "waitTime": 0
      },
      "consta": {
        "canSkip": true,
        "cancelEnd": true,
        "cancelNext": false,
        "cancelSkip": false,
        "failEnd": true,
        "failNext": true,
        "failRedo": false,
        "failSkip": false
      },
      "custom": {
        "executable": true,
        "priority": 0
      }
    }
  ],
  "time": {
    "end": 1705112613,
    "generate": 1705112610,
    "start": 1705112611
  }
},
"msg": "",
"ok": true,
"type": "woosh.task.WooshTaskSet"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.task.AddTask</code> for details
»» tset	object	true	none	Task collection	none
»»» base	object	true	none		none
»»»» actuator	string	true	none		none
»»»» adapter	string	true	none		none
»»»» id	integer	true	none		none
»»»» name	string	true	none		none
»»»» no	string	true	none		none
»»»» priority	integer	true	none		none
»»»» robots	[string]	true	none		none
»»»» route	string	true	none		none
»»»» rtype	integer	true	none		none
»»»» type	string	true	none		none
»»» custom	object	true	none		none
»»»» autoComplete	boolean	true	none		none
»»» tasks	[object]	true	none		none
»»»» base	object	true	none		none
»»»»» cannotCancel	boolean	true	none		none
»»»»» custom	string	true	none		none
»»»»» direction	integer	true	none		none
»»»»» id	integer	true	none		none
»»»»» markNo	string	true	none		none
»»»»» name	string	true	none		none
»»»»» type	integer	true	none		none
»»»»» typeNo	integer	true	none		none
»»»»» waitTime	integer	true	none		none
»»»» consta	object	true	none		none
»»»»» canSkip	boolean	true	none		none
»»»»» cancelEnd	boolean	true	none		none

Name	Type	Required	Constraint	Chinese name	Explanation
»»»»» cancelNext	boolean	true	none		none
»»»»» cancelSkip	boolean	true	none		none
»»»»» failEnd	boolean	true	none		none
»»»»» failNext	boolean	true	none		none
»»»»» failRedo	boolean	true	none		none
»»»»» failSkip	boolean	true	none		none
»»»»» custom	object	true	none		none
»»»»» executable	boolean	true	none		none
»»»»» priority	integer	true	none		none
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.task.AddTask</code>

POST Add predefined task

POST /woosh/dispatch/task/ExecPreTask

Body Request Parameters

```
{
  "taskSetId": -2127845149,
  "robot": 30001,
  "repeatTimes": 2,
  "order": 0
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» taskSetId	body	integer	is	Predefined task ID	none
» robot	body	integer	is	Specify robot	none

Name	Location	Type	Required	Chinese Name	Description
» repeatTimes	body	integer	is	Number of times to repeat execution	none
» order	body	integer	is	0 add, 1 delete	none

Return example

Success

```
{
  "body": {
    "tset": {
      "base": {
        "actuator": "woosh",
        "adapter": "woosh",
        "id": 682477540,
        "name": "A10-K11",
        "no": "",
        "priority": 0,
        "robots": [],
        "route": "",
        "rtype": 1,
        "type": "woosh.task.WooshTaskSet"
      },
      "custom": {
        "autoComplete": true
      },
      "tasks": [
        {
          "base": {
            "cannotCancel": false,
            "custom": "",
            "direction": 0,
            "id": 1,
            "markNo": "A10",
            "name": "A10",
            "type": 1,
            "typeNo": 0,
            "waitTime": 0
          },
          "consta": {
            "canSkip": true,
            "cancelEnd": true,
            "cancelNext": false,
            "cancelSkip": false,
            "failEnd": true,
            "failNext": true,
            "failRedo": false,
            "failSkip": false
          },
          "custom": {
            "executable": true,
            "priority": 0
          }
        }
      ]
    }
  }
}
```

```

    }
  },
  {
    "base": {
      "cannotCancel": false,
      "custom": "",
      "direction": 0,
      "id": 2,
      "markNo": "K11",
      "name": "K11",
      "type": 1,
      "typeNo": 0,
      "waitTime": 0
    },
    "consta": {
      "canSkip": true,
      "cancelEnd": true,
      "cancelNext": false,
      "cancelSkip": false,
      "failEnd": true,
      "failNext": true,
      "failRedo": false,
      "failSkip": false
    },
    "custom": {
      "executable": true,
      "priority": 0
    }
  }
]
}
},
"msg": "",
"ok": true,
"type": "woosh.task.AddTask"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.task.AddTask</code> for details

Name	Type	Required	Constraint	Chinese name	Explanation
»» tset	object	true	none	Task collection	none
»»» base	object	true	none		none
»»»» actuator	string	true	none		none
»»»» adapter	string	true	none		none
»»»» id	integer	true	none		none
»»»» name	string	true	none		none
»»»» no	string	true	none		none
»»»» priority	integer	true	none		none
»»»» robots	[string]	true	none		none
»»»» route	string	true	none		none
»»»» rtype	integer	true	none		none
»»»» type	string	true	none		none
»»» custom	object	true	none		none
»»»» autoComplete	boolean	true	none		none
»»» tasks	[object]	true	none		none
»»»» base	object	true	none		none
»»»»» cannotCancel	boolean	true	none		none
»»»»» custom	string	true	none		none
»»»»» direction	integer	true	none		none
»»»»» id	integer	true	none		none
»»»»» markNo	string	true	none		none
»»»»» name	string	true	none		none
»»»»» type	integer	true	none		none
»»»»» typeNo	integer	true	none		none
»»»»» waitTime	integer	true	none		none
»»»» consta	object	true	none		none
»»»»» canSkip	boolean	true	none		none
»»»»» cancelEnd	boolean	true	none		none
»»»»» cancelNext	boolean	true	none		none

Name	Type	Required	Constraint	Chinese name	Explanation
»»»»» cancelSkip	boolean	true	none		none
»»»»» failEnd	boolean	true	none		none
»»»»» failNext	boolean	true	none		none
»»»»» failRedo	boolean	true	none		none
»»»»» failSkip	boolean	true	none		none
»»»»» custom	object	true	none		none
»»»»» executable	boolean	true	none		none
»»»»» priority	integer	true	none		none
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.task.AddTask</code>

POST Current execution of predefined task set

POST /woosh/dispatch/task/ExecPreTasks

Body Request Parameters

```
{
  "tset": {
    "base": {
      "name": "A10-k11",
      "priority": 0,
      "robots": [],
      "rtype": 1
    },
    "custom": {
      "auto_complete": true
    },
    "tasks": [
      {
        "base": {
          "cannot_cancel": false,
          "custom": "",
          "direction": 0,
          "id": 1,
          "mark_no": "A10",
          "name": "A10",
          "type": 1,

```



```

        "type_no": 0,
        "wait_time": 0
    },
    "consta": {
        "can_skip": true,
        "cancel_end": true,
        "cancel_next": false,
        "cancel_skip": false,
        "fail_end": true,
        "fail_next": true,
        "fail_redo": false,
        "fail_skip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
},
{
    "base": {
        "cannot_cancel": false,
        "custom": "",
        "direction": 0,
        "id": 2,
        "mark_no": "K11",
        "name": "K11",
        "type": 1,
        "type_no": 0,
        "wait_time": 0
    },
    "consta": {
        "can_skip": true,
        "cancel_end": true,
        "cancel_next": false,
        "cancel_skip": false,
        "fail_end": true,
        "fail_next": true,
        "fail_redo": false,
        "fail_skip": false
    },
    "custom": {
        "executable": true,
        "priority": 0
    }
}
]
}
}

```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	No	Request message body	none
» tset	body	object	is	Task set	none
»» base	body	object	is		none
»»» rtype	body	integer	is	Robot type	none
»»» priority	body	integer	is	Priority	The higher the value, the higher the priority
»»» robots	body	[string]	is	Specify robot ID	none
»»» adapter	body	string	is		none
»»» actuator	body	string	is		none
»»» name	body	string	is		none
»»» type	body	string	is		none
»» tasks	body	[object]	is	subtask collection	none
»»» base	body	object	is		none
»»»» id	body	integer	is	task ID	range 1~99, unique within task set
»»»» name	body	string	is	name	none
»»»» type	body	integer	is	task type	none
»»»» direction	body	integer	is	action direction	none
»»»» cannot_cancel	body	boolean	is		none
»»»» custom	body	string	is		none
»»»» mark_no	body	string	is		none
»»»» type_no	body	integer	is		none
»»»» wait_time	body	integer	is		none

Name	Location	Type	Required	Chinese Name	Description
»»» custom	body	object	is		none
»»»» executable	body	boolean	is		none
»»»» priority	body	integer	is		none
»»» consta	body	object	is		none
»»»» can_skip	body	boolean	is		none
»»»» cancel_end	body	boolean	is		none
»»»» cancel_next	body	boolean	is		none
»»»» cancel_skip	body	boolean	is		none
»»»» fail_end	body	boolean	is		none
»»»» fail_next	body	boolean	is		none
»»»» fail_redo	body	boolean	is		none
»»»» fail_skip	body	boolean	is		none
»» custom	body	object	is		none
»»» auto_complete	body	boolean	is		none

Return example

Success

```
{
  "body": {
    "tset": {
      "base": {
        "actuator": "woosh",
        "adapter": "woosh",
        "id": 682477540,
        "name": "A10-K11",
        "no": "",
        "priority": 0,
        "robots": [],
        "route": "",
        "rtype": 1,
        "type": "woosh.task.WooshTaskSet"
      },
    },
    "custom": {
      "autoComplete": true
    }
  }
}
```

```

},
"tasks": [
  {
    "base": {
      "cannotCancel": false,
      "custom": "",
      "direction": 0,
      "id": 1,
      "markNo": "A10",
      "name": "A10",
      "type": 1,
      "typeNo": 0,
      "waitTime": 0
    },
    "consta": {
      "canSkip": true,
      "cancelEnd": true,
      "cancelNext": false,
      "cancelSkip": false,
      "failEnd": true,
      "failNext": true,
      "failRedo": false,
      "failSkip": false
    },
    "custom": {
      "executable": true,
      "priority": 0
    }
  },
  {
    "base": {
      "cannotCancel": false,
      "custom": "",
      "direction": 0,
      "id": 2,
      "markNo": "K11",
      "name": "K11",
      "type": 1,
      "typeNo": 0,
      "waitTime": 0
    },
    "consta": {
      "canSkip": true,
      "cancelEnd": true,
      "cancelNext": false,
      "cancelSkip": false,
      "failEnd": true,
      "failNext": true,
      "failRedo": false,
      "failSkip": false
    },
    "custom": {
      "executable": true,
      "priority": 0
    }
  }
]
}

```

```

    ]
  }
},
"msg": "",
"ok": true,
"type": "woosh.task.AddTask"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none	Response message body	See dictionary <code>woosh.task.AddTask</code> for details
»» tset	object	true	none	Task collection	none
»»» base	object	true	none		none
»»»» actuator	string	true	none		none
»»»» adapter	string	true	none		none
»»»» id	integer	true	none		none
»»»» name	string	true	none		none
»»»» no	string	true	none		none
»»»» priority	integer	true	none		none
»»»» robots	[string]	true	none		none
»»»» route	string	true	none		none
»»»» rtype	integer	true	none		none
»»»» type	string	true	none		none
»»» custom	object	true	none		none
»»»» autoComplete	boolean	true	none		none
»»» tasks	[object]	true	none		none
»»»» base	object	true	none		none
»»»»» cannotCancel	boolean	true	none		none

Name	Type	Required	Constraint	Chinese name	Explanation
»»»»» custom	string	true	none		none
»»»»» direction	integer	true	none		none
»»»»» id	integer	true	none		none
»»»»» markNo	string	true	none		none
»»»»» name	string	true	none		none
»»»»» type	integer	true	none		none
»»»»» typeNo	integer	true	none		none
»»»»» waitTime	integer	true	none		none
»»»»» consta	object	true	none		none
»»»»» canSkip	boolean	true	none		none
»»»»» cancelEnd	boolean	true	none		none
»»»»» cancelNext	boolean	true	none		none
»»»»» cancelSkip	boolean	true	none		none
»»»»» failEnd	boolean	true	none		none
»»»»» failNext	boolean	true	none		none
»»»»» failRedo	boolean	true	none		none
»»»»» failSkip	boolean	true	none		none
»»»»» custom	object	true	none		none
»»»»» executable	boolean	true	none		none
»»»»» priority	integer	true	none		none
» msg	string	true	none	Response status message	none
» ok	boolean	true	none	Response result	TRUE: Success, FALSE: Failure
» type	string	true	none	Response message type	<code>woosh.task.AddTask</code>

Device request/callbox

POST Online

POST /woosh/device/callbox/Online

```
{
  "callbox": {
    "no": "cb110",
    "keyNum": 4,
    "keys": [
      {
        "id": 1
      },
      {
        "id": 2
      },
      {
        "id": 3
      },
      {
        "id": 4
      }
    ]
  }
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none
» callbox	body	object	is		none
»» no	body	string	is		none
»» keyNum	body	integer	is		none
»» keys	body	[object]	is		none
»»» id	body	integer	is		none

Return example

Success

```
{
  "body": {
    "keyNum": 4,
    "keys": [
      {
        "id": 1,
        "state": 0,
        "taskId": 0,
        "time": 0,
        "tsetId": 0
      },
      {

```

```

        "id": 2,
        "state": 0,
        "taskId": 0,
        "time": 0,
        "tsetId": 0
    },
    {
        "id": 3,
        "state": 0,
        "taskId": 0,
        "time": 0,
        "tsetId": 0
    },
    {
        "id": 4,
        "state": 0,
        "taskId": 0,
        "time": 0,
        "tsetId": 0
    }
],
"no": "cb110",
"online": true,
"state": 0
},
"msg": "",
"ok": true,
"type": "woosh.device.callbox.Callbox"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none		none
»» keyNum	integer	true	none		none
»» keys	[object]	true	none		none
»»» id	integer	true	none		none
»»» state	integer	true	none		none
»»» taskId	integer	true	none		none
»»» time	integer	true	none		none

Name	Type	Required	Constraint	Chinese name	Explanation
»»» tsetId	integer	true	none		none
»» no	string	true	none		none
»» online	boolean	true	none		none
»» state	integer	true	none		none
» msg	string	true	none		none
» ok	boolean	true	none		none
» type	string	true	none		none

POST Get Call Button List

POST /woosh/device/callbox/Callers

Body Request Parameters

```
{}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

Success

```
{
  "body": {
    "callers": [
      {
        "key": 4,
        "no": "cb110",
        "state": 0,
        "taskId": 0,
        "time": 0,
        "tsetId": 0
      },
      {
        "key": 2,
        "no": "cb110",
        "state": 0,
        "taskId": 0,
        "time": 0,
        "tsetId": 0
      },
      {
        "key": 3,
```

```

    "no": "cb110",
    "state": 0,
    "taskId": 0,
    "time": 0,
    "tsetId": 0
  },
  {
    "key": 1,
    "no": "cb110",
    "state": 1,
    "taskId": 0,
    "time": 1686727674,
    "tsetId": 0
  }
]
},
"msg": "",
"ok": true,
"type": "woosh.device.callbox.Callers"
}

```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none		none
»» callers	[object]	true	none		none
»»» key	integer	true	none		none
»»» no	string	true	none		none
»»» state	integer	true	none		none
»»» taskId	integer	true	none		none
»»» time	integer	true	none		none
»»» tsetId	integer	true	none		none
» msg	string	true	none		none
» ok	boolean	true	none		none
» type	string	true	none		none

POST Offline

POST /woosh/device/callbox/Offline

Body Request Parameters

```
{
  "no": "cb110"
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none
» no	body	string	is		none

Return example

Success

```
{
  "body": {
    "no": "cb110"
  },
  "msg": "",
  "ok": true,
  "type": "woosh.device.callbox.Offline"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none		none
»» no	string	true	none		none
» msg	string	true	none		none
» ok	boolean	true	none		none
» type	string	true	none		none

POST Call

POST /woosh/device/callbox/Call

Body Request Parameters

```
{
  "event": {
    "no": "cb110",
    "key": 1,
    "type": 1
  }
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none
» event	body	object	is		none
»» no	body	string	is		none
»» key	body	integer	is		none
»» type	body	integer	is		none

Return example

200 Response

```
{}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

POST Get Specified Call Box

POST /woosh/device/callbox/Callbox

Body Request Parameters

```
{
  "no": "cb110"
}
```

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none
» no	body	string	is		none

Return example

Success

```
{
  "body": {
    "keyNum": 4,
    "keys": [
      {
        "id": 1,
        "state": 3,
        "taskId": 0,
        "time": 0,
        "tsetId": 686712522
      },
      {
        "id": 2,
        "state": 0,
        "taskId": 0,
        "time": 0,
        "tsetId": 0
      },
      {
        "id": 3,
        "state": 0,
        "taskId": 0,
        "time": 0,
        "tsetId": 0
      },
      {
        "id": 4,
        "state": 0,
        "taskId": 0,
        "time": 0,
        "tsetId": 0
      }
    ],
    "no": "cb110",
    "online": false,
    "state": 0
  },
  "msg": "",
  "ok": true,
  "type": "woosh.device.callbox.Callbox"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none		none
»» keyNum	integer	true	none		none
»» keys	[object]	true	none		none
»»» id	integer	true	none		none
»»» state	integer	true	none		none
»»» taskId	integer	true	none		none
»»» time	integer	true	none		none
»»» tsetId	integer	true	none		none
»» no	string	true	none		none
»» online	boolean	true	none		none
»» state	integer	true	none		none
» msg	string	true	none		none
» ok	boolean	true	none		none
» type	string	true	none		none

POST Get Call Box List

POST /woosh/device/callbox/Callboxes

Body Request Parameters

<code>{}</code>

Request Parameters

Name	Location	Type	Required	Chinese Name	Description
body	body	object	no		none

Return example

```
{
  "body": {
    "callboxes": [
      {
        "keyNum": 4,
        "keys": [
          {
            "id": 1,
            "state": 3,
            "taskId": 0,
            "time": 0,
            "tsetId": 686712522
          },
          {
            "id": 2,
            "state": 0,
            "taskId": 0,
            "time": 0,
            "tsetId": 0
          },
          {
            "id": 3,
            "state": 0,
            "taskId": 0,
            "time": 0,
            "tsetId": 0
          },
          {
            "id": 4,
            "state": 0,
            "taskId": 0,
            "time": 0,
            "tsetId": 0
          }
        ],
        "no": "cb110",
        "online": false,
        "state": 0
      }
    ]
  },
  "msg": "",
  "ok": true,
  "type": "woosh.device.callbox.Callboxes"
}
```

Return result

Status code	Status code meaning	Explanation	Data model
200	OK	Success	Inline

Return data structure

Status code **200**

Name	Type	Required	Constraint	Chinese name	Explanation
» body	object	true	none		none
»» callboxs	[object]	true	none		none
»»» keyNum	integer	false	none		none
»»» keys	[object]	false	none		none
»»»» id	integer	true	none		none
»»»» state	integer	true	none		none
»»»» taskId	integer	true	none		none
»»»» time	integer	true	none		none
»»»» tsetId	integer	true	none		none
»»» no	string	false	none		none
»»» online	boolean	false	none		none
»»» state	integer	false	none		none
» msg	string	true	none		none
» ok	boolean	true	none		none
» type	string	true	none		none