

Slamware RESTful API开发手册

 李振峰 | 4月1日修改

 AI 速览

本文讨论了 Slamware RESTful API 的开发手册，涵盖其基本信息、接口规范、分类、部署流程、业务流程示例以及二维码精准对接等内容。关键点包括：1. API ...

1. 概述

4.0 及更高版本的 Slamware 固件提供 `RESTful API`，相比 `C++ SDK`，该接口更易于使用且提供更丰富的功能，并且兼容任意客户端系统和编程语言。

该服务的端口为 `1448`，本文介绍API的基本使用方法，接口的具体定义请参阅在线文档：<https://docs.slamtec.com/#/>。

如果固件版本在4.6.0以上，通过在浏览器输入 `IP:1448`，即可在线调试API。例：`192.168.11.1:1448`

例如，连上机器人热点并获取机器人电源状态的接口如下：

■ `GET http://192.168.11.1:1448/api/core/system/v1/power/status`

返回内容为

```
■ {
  "batteryPercentage": 90,
  "dockingStatus": "on_dock",
  "isCharging": true,
  "isDCCConnected": false,
  "powerStage": "running",
  "sleepMode": "awake"
}
```

2. 接口规范

2.1 命名方式

API 接口的 `endpoint` 规范

大部分接口都以以下的结构组织：

■ `/api/{plugin}/{feature}/{version}/{resource...}`

plugin

- `core`：Agent核心框架和通用服务
- `platform`：通用底盘的插件，提供上报设备事件、上传日志等基础功能
- `multi_floor`：提供多楼层地图管理和跨楼层移动能力的插件，同时兼容单楼层地图
- `delivery`：提供配送服务的插件，可应用于餐厅、酒店等场景

feature

- 机器人功能大类

version

- 版本号

2.2 Method类型

目前用到 GET、PUT、POST、DELETE 四类方法

- **GET** 获取资源(安全、幂等)
- **PUT** 创建、更新资源(非安全、幂等)
- **POST** 创建资源或执行一个动作(非安全、非幂等)
- **DELETE** 删除资源(非安全、幂等)

2.3 API参数

query类型

`query` 参数是问号后面带一对或者多对 key=value，如下表示获取E撞2楼的POI：

■ **GET** <http://127.0.0.1:1448/api/multi-floor/map/v1/pois?floor=2F&building=E>

path类型

`path` 参数直接放在路径中, 形如：`DELETE /api/core/artifact/v1/lines/{usage}/{id}`，

如下表示删除id为199的虚拟墙：

■ **DELETE** <http://127.0.0.1:1448/api/core/artifact/v1/lines/walls/199>

Request Body

即 API 请求的 `Content-Type` 为 `application/json`，如：

代码块

```
1  curl -X 'POST' \  
2  'http://127.0.0.1:1448/api/core/motion/v1/actions' \  
3  -H 'accept: application/json' \  
4  -H 'Content-Type: application/json' \  
5  -d '{  
6      "action_name": "slamtec.agent.actions.MoveToAction",  
7      "options": {  
8          "target": {  
9              "x": 0,  
10             "y": 0,  
11             "z": 0  
12         },  
13         "move_options": {  
14             "mode": 0,  
15             "flags": [],  
16             "yaw": 0,  
17             "acceptable_precision": 0,  
18             "fail_retry_count": 0  
19         }  
20     }  
21 }'
```

2.4 返回状态码

2xx：成功

- `200 (OK)` 表示成功执行了客户端请求的任何操作。
- `204 (无内容)` 表示服务器已成功完成请求，并且在响应有效负载正文中没有要发送的内容。

4xx：客户端错误

- `400 (不良请求)` 通用客户端错误状态，在没有其他 `4xx` 错误代码时使用。
- `404 (未找到)` 无法找到 REST API 请求的 URI 资源。

5xx: 服务器错误

■ 500 服务器内部错误

2.5 返回值

当接口返回状态码是200时, `Content-Type` 有如下几种类型:

- `application/json`

大部分接口的返回数据都是这种json格式

- `application/octet-stream`

二进制流, 获取explore map 和stcm map的返回值是二进制流

- `text/plain`

部分接口的返回值是单纯的字符串

2.6 如何在API文档中查看接口

以 `创建新的运动行为 (/api/core/motion/v1/actions)` 接口为例

请求/响应概览:

代码块

```
1  POST http://127.0.0.1:1448/api/core/motion/v1/actions
2  Curl:
3  curl -X 'POST' \
4  'http://127.0.0.1:1448/api/core/motion/v1/actions' \
5  -H 'accept: application/json' \
6  -H 'Content-Type: application/json' \
7  -d '{
8      "action_name": "slamtec.agent.actions.MoveToAction",
9      "options": {
10         "target": {
11             "x": 0.1,
12             "y": 0.2
13         },
14         "move_options": {
15             "mode": 0,
16             "flags": [
17                 "with_yaw"
18             ],
19             "yaw": 1
20         }
21     }
22 }'
```

23

24 Response Body

```
25 {
26     "action_id": 0,
27     "action_name": "string",
28     "stage": "GOING_TO_TARGET",
29     "state": {
30         "status": 0,
31         "result": 0,
32         "reason": ""
33     }
34 }
```

接口 `Request Body` 和 `Responses Body` 详细说明

下图左侧从上到下三个箭头, 第一个箭头处有 `required` 字样, 接口文档中凡是带有该字样的都是必填项, 第二个和第三个箭头处有 `Schema` 和 `Example Value` 字样, 默认显示 `Example Value` 字样下的详细内容, 也即是 `Request Body` 或 `Responses Body` 的 Json 结构体, `Schema` 字样可

以点开，点击 Schema 即可看到 Example Value 的 Json 结构的生成规则。

POST /api/core/motion/v1/actions Create new motion behaviors

Parameters

No parameters

Request body required

application/json

Try it out

action_name is queried through the /core/motion/v1/action-factories interface, and the specific content of options depends on the action type.

Example Value Schema

```
{
  "action_name": "slamtec.agent.actions.MoveToAction",
  "options": {
    "target": {
      "x": 0,
      "y": 0,
      "z": 0
    },
    "move_options": {
      "mode": 0,
      "flags": [],
      "yaw": 0,
      "acceptable_precision": 0,
      "fail_retry_count": 0,
      "speed_ratio": 0
    }
  }
}
```

Responses

Code	Description	Links
200	OK	No links
<div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div><div><div>Example Value Schema</div><div><pre>{ "action_id": 0, "action_name": "string", "stage": "GOING_TO_TARGET", "state": { "status": 0, "result": 0, "reason": "" } }</pre></div></div></div>		
400	Can not create action	No links

接口文档中 Schema 的查看方式

Schema 在 Request Body 和 Responses 中都有可能出现，这里以 Request Body 来举例说明，两者查看方式类似。

下图是 Request body 的 Schema 展开后的细节。

带有红色 required 字样 和 * 的参数都是必填项

红框内的参数是 Json 结构的字段 key，蓝色框的字样只是辅助说明文字，如果有大括号内省略号项，可点开查看详情

图中一对绿色框为相对应的值：如 action_name 值是图中一对绿色框为相对应的值：如

action_name 值是： slamtec.agent.actions.MultiFloorMoveAction ,则 option 中的 oneOf 则取对应的 MultiFloorMoveActionOptions 结构

https://slamtec.feishu.cn/docx/StfqdFhcmo8ivSxXQYcceagTnen

4/19

Response Body 详情

Response Body 展示了返回 200 的 Response Body 、以及 Body 对应的解析规则 Shcema 或者返回其他状态码及其描述

Responses

Curl

```
curl -X 'POST' \
'http://127.0.0.1:1448/api/core/motion/v1/actions' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "action_name": "slamtec.agent.actions.MoveToAction",
  "options": {
    "target": {
      "x": 0,
      "y": 0,
      "z": 0
    },
    "move_options": {
      "mode": 0,
      "flags": [],
      "yaw": 0,
      "acceptable_precision": 0,
      "fail_retry_count": 0
    }
  }
}
```

Request URL

```
http://127.0.0.1:1448/api/core/motion/v1/actions
```

Server response

Code	Details
Undocumented	<div><div>Failed to fetch.</div><div>Possible Reasons:</div><ul style="list-style-type: none">• CORS• Network Failure• URL scheme must be "http" or "https" for CORS request.</div>

Responses

Code	Description	Links
200	<div><div>OK</div><div><div>Media type</div><div><div>application/json</div></div><div>Controls Accept header.</div><div><div>Example Value</div><div>Schema</div></div></div><div><pre>{ "action_id": 0, "action_name": "string", "stage": "GOING_TO_TARGET", "state": { "status": 0, "result": 0, "reason": "" } }</pre></div></div> <div>No links</div>	No links
400	<div>Can not create action</div>	

No links

 No links |

Server response

Code

Details

Undocumented

Failed to fetch.

Possible Reasons:

CORS

Network Failure

URL scheme must be "http" or "https" for CORS request.

Responses

Code

Description

Links

200

OK

No links

Media type

application/json

Controls Accept header.

Example Value

Schema

ActionInfo

integer

string

string

example: GOING_TO_TARGET

state

ActionState

integer

0:NewBorn, 1:Working, 3:Paused, 4:Done

Enum:

[0, 1, 3, 4]

integer

0:Success, -1: Failed, -2: Aborted

Enum:

[0, -1, -2]

string

default:

400

Can not create action

No links

3. 接口分类

即2.1节 feature 的分类

3.1 system 系统资源

该类接口可以访问机器人的系统级资源，如读取电源状态，重启机器，设置系统参数等。

3.2 slam 定位建图相关功能

获取机器人位姿，获取/注册充电桩，打开/关闭建图，获取地图数据等。

完成建图并添加所需的 POI 后，可调用 获取复合地图 接口导出地图：

■

GET /api/core/slam/v1/maps/stcm

3.3 artifact 人工标记地图元素

可以在地图中添加如下元素

- 虚拟轨道(tracks)： 通过参数控制可以使得机器人沿预设轨道行进。
- 虚拟墙(walls): 禁止机器人进入某些区域。
- 禁行区域(forbidden_area)： 效果类似虚拟墙，支持自动逃脱功能，机器人在外部时不会进入禁区，一旦被人为推进去，则可以往最近的边缘方向逃离禁区。
- 电梯区域(elevator_area)： 应用于多楼层环境时，需要添加电梯信息，并且通过RS将多个楼层的地图合并为一个文件。
- 危险区域(dangerous_area)： 有斜坡、窄道等，可以限制机器人的最大移动速度。
- 覆盖区域(coverage_area): 机器人会规划路径覆盖整个区域，其行为类似扫地机。
- 运维区域(maintenance_area)： 机器人重新打开建图时，只会更新运维区域内的地图。

https://slamtec.feishu.cn/docx/StfqdFhcmo8ivSxXQYcceagTnen

7/19

3.4 motion 机器人运控控制

该类接口提供：获取机器人支持的所有运动行为；机器人获取/终止/创建新的运动行为；查询路径与目标点；获取/设置运动策略；开启/查询手动重定位等行为控制。

- 机器人需要创建新的运动行为开始运动。

创建新的运动行为：POST `/api/core/motion/v1/actions`

- 机器人运动过程中，需要不断查询Action状态，以判断当前运动状态。

查询Action状态：GET `/api/core/motion/v1/actions/{action_id}`

3.5 firmware 固件升级

该类接口提供机器人固件升级和查询相关升级信息的功能。

3.6 statistics 运行数据统计

该类接口主要是获取机器人的运动里程和运行时间。

3.7 application 安卓应用程序管理（仅限ARM平台）

该类接口提供机器人安装/卸载App，以及获取已安装的App的功能。

3.8 platform 通用底盘和平台相关的功能

该类接口提供获取机器人系统时间戳和获取机器人事件信息的功能。

- 机器人运动过程中，可能会遇到一系列如：遇到障碍物，电量过低，等情况，所以调用方需要不断地获取事件信息实时掌握机器人情况，`获取事件信息`：GET `/api/platform/v1/events`。

3.9 multi-floor 地图管理，跨楼层移动

多楼层地图管理，乘电梯等功能：如查找距离机器人最近的充电桩，持久化保存当前地图，重新加载地图，同步地图等。

3.10 delivery 配送服务相关接口

注意事项：delivery相关接口仅限整机使用，通用底盘默认不支持，如需使用，请联系FAE。

整体上分为系统配置、cargo管理、任务管理等三大类

该类接口提供针对配送、引领、迎宾等机器人场景集成好的功能，如果有类似需求也可直接使用deliver服务接口。主要包含功能：创建任务、查询任务、取消任务、暂停/继续执行任务、结束任务、开始取物、结束取物、完成操作等

两类地图操作说明

- `/api/core/slam` 类地图操作接口：
如果设置地图，则会设置到定位导航统内存中，地图不会持久化保存，可以导出地图文件。
- `/api/multi-floor/map` 类地图操作接口：
可以上传地图或者从定位导航统内存中获取地图，并持久化保存到磁盘中。

两类poi操作说明

- `/api/core/artifact` 类POI操作接口：
可对poi进行增删改查操作。
- `/api/multi-floor/map` 类POI操作接口：
只能查找poi信息，其中`/api/multi-floor/map/v1/pois`可获取额外的building、floor、poi_name、type信息。

4 部署流程示例

该阶段是完成机器人初始化操作，给机器人使能，使其处于就绪待用状态。

主要包含：开启/关闭建图、添加pois、添加禁区、导出地图 等操作。

注意事项：

- 1.如采用RoboStudio部署方式，可忽略此流程；
- 2.设置禁行区域时，相比虚拟墙，建议使用虚拟禁区。

4.1 建图

开启/关闭建图：

■ **PUT** `/api/core/slam/v1/mapping/:enable`

请求体中 enable 设为 false 则为关闭建图

返回值 `true` 表示操作成功

4.2 添加 poi

添加POI：

■ **POST** `/api/core/artifact/v1/pois`

调用方应当随机生成一个UUID作为id, metadata中的display_name用于界面显示, type用于区分POI类型。

在建图过程中添加POI时，建议不包含Pose，此时会用机器人当前位置创建POI，并且记录传感器观测信息，在闭环后会进行位姿调整。

请点击接口文档中 `Schema` 查看详细说明

4.3 添加虚拟墙

添加虚拟线段

■ **POST** `/api/core/artifact/v1/lines/{usage}`

添加时id为无效字段，可为任意值。

请点击接口文档中 `Schema` 查看详细说明

4.4 设置虚拟禁区

添加矩形区域

■ **POST** `/api/core/artifact/v1/rectangle-areas/{usage}`

不同类型的矩形区域，所需要的 `metadata` 也不同，请点击接口文档中 `Schema` 查看详细说明

4.5 导出地图

获取复合地图：

■ **GET** `/api/core/slam/v1/maps/stcm`

包含所有数据的复合地图

响应报文为二进制字节流，可直接保存为stcm文件。

4.6 保存地图

如果想将4.5导出的地图，通过API的方式保存到机器人中，有两种方式可以保存地图。

方法1:

将地图导出后，通过API的方式保存到机器人中，请按照下述接口调用顺序保存地图：

1. 上传地图到机器人

上传的地图会持久化保存在文件系统中，但不会加载到Slamware中。【注意】当机器人由云端管理时，从云端下载的地图会覆盖本地地图。

■ `POST /api/multi-floor/map/v1/stcm`

2. 重新加载地图

■ `POST /api/multi-floor/map/v1/stcm:reload`

3. 持久化保存当前地图

■ `POST /api/multi-floor/map/v1/stcm:save`

方法2:

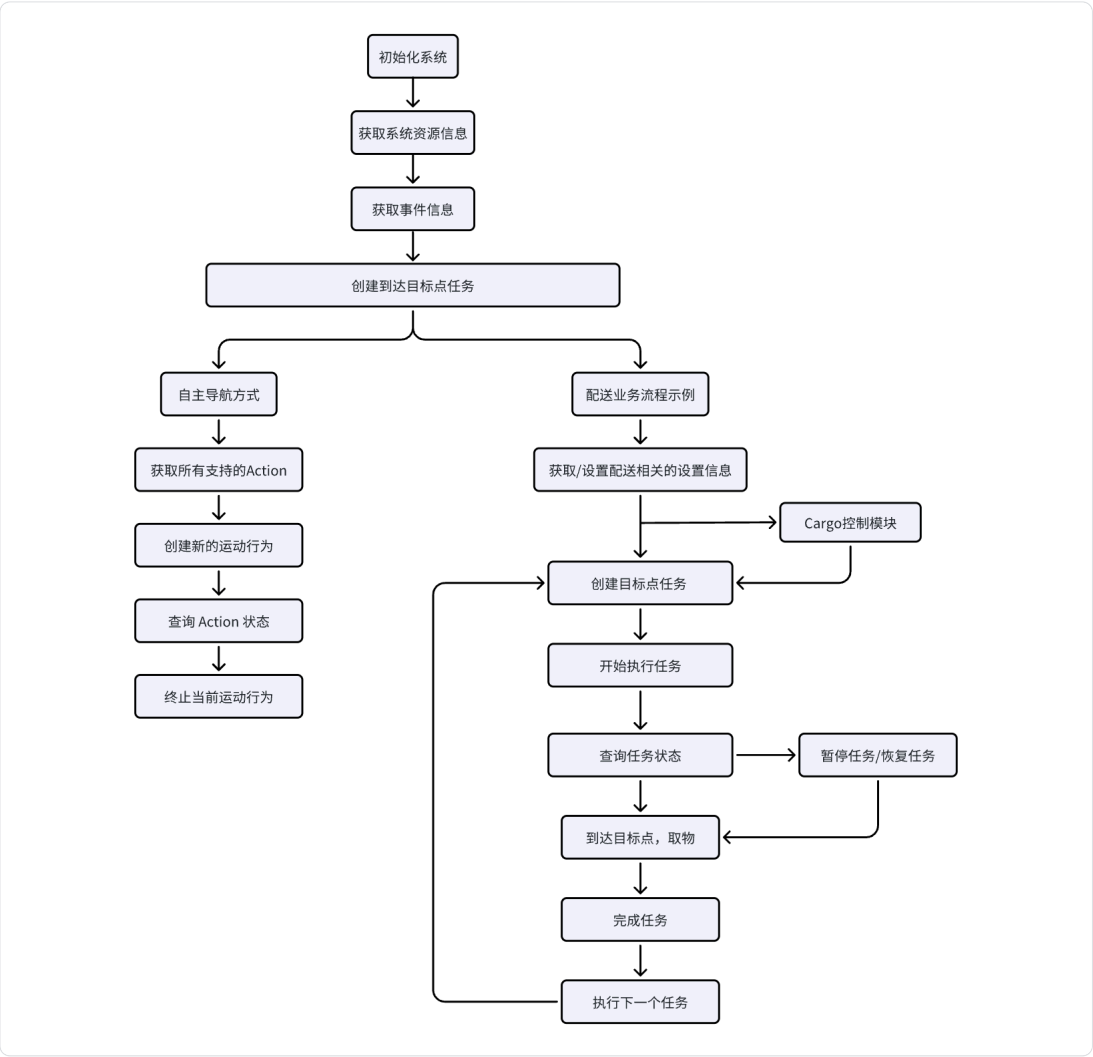
通过RoboStudio把地图加载到固件后；调用端只需要调用同步地图接口即可保存地图

1. 同步地图

【注意】：保存当前地图到文件，并重新加载，相当于save和reload 2个接口的组合。多楼层环境中禁止该操作，否则会丢失其他楼层的地图。

■ `POST /api/multi-floor/map/v1/stcm:sync`

5.业务流程示例



5.1 初始化系统

机器人启动时，通过轮询如下接口判断系统是否完成初始化。

只有在每个组件都 enable 后，机器人才能进入正常的业务逻辑。

■ GET /api/core/system/v1/capabilities

5.2 获取系统资源信息

使用API可以实现对机器人系统级资源的读取和设置，从而更加高效地控制和管理机器人。通过API，您可以获取设备信息、设备健康状况、电源状态等关键数据，确保机器人的正常运行和合理使用。此外，您还可以关闭或重启机器人以及获取和修改系统参数，以满足特定的使用需求。

获取机器人电源状态：

■ GET /api/core/system/v1/power/status

获取设备信息：

■ GET /api/core/system/v1/robot/info

获取设备健康状态信息：

■ GET /api/core/system/v1/robot/health

获取系统参数：

■ GET /api/core/system/v1/parameter

设置系统参数：

■ PUT /api/core/system/v1/parameter

关闭或重启机器人：

■ POST /api/core/system/v1/power/:shutdown

5.3 获取POI信息

获取地图中设置的POI信息列表，即机器人可以前往的目标点。

■ GET /api/multi-floor/map/v1/pois

通过参数可以获取指定楼层、建筑物、POI类型、POI分组的POI列表，不带参数时获取所有POI。

参数名	类型	必填	说明
floor	string	否	楼层名
building	string	否	建筑物名
type	string	否	poi类型
group	string	否	poi分组

5.4 获取事件信息

机器人开机状态下，获取机器人当前状态、充电状态、运行状态、可能会遇到障碍物，进出电梯、健康状态报警等情况，建议调用方轮询此接口，实时知晓机器人遇到的情况。

获取事件信息：

■ GET /api/platform/v1/events

机器人通知调用端自身发生的事件，上位机可以播报语音或进行别的交互，启用不同的插件会扩展出不同的事件类型。

机器人事件信息，type在不同场景下会扩展新的定义，调用方只需处理自己关心的事件即可。GeneralEventType为通用的事件，ElevatorEventType为进出电梯相关的事件，DeliveryEventType为配送相关事件。

请点击接口文档中 Schema 查看详细说明

5.5 用户开始操作机器人

用户操作机器人时，首先调用以下接口 `enable_task_execution` 设为false来禁止机器人移动,用户完成操作时，`enable_task_execution` 设为true允许机器人运动，此时机器人有任务则执行任务，没任务则回桩或回到类型为PARKING的POI。

■ **PUT** `/api/delivery/v1/tasks/:task_execution`

request body 示例如下

代码块

```
1 {
2   "enable_task_execution": false
3 }
```

5.6 自主导航流程示例

本示例，展示一个由调用方创建action控制机器人移动到指定POI 的例子。

5.6.1 获取所有支持的Action

■ **GET** `/api/core/motion/v1/action-factories`

- `slamtec.agent.actions.MoveToAction` 自主导航移动
- `slamtec.agent.actions.MultiFloorMoveAction` 自主导航移动，支持跨楼层、POI目标点、多级调度
- `slamtec.agent.actions.MultiFloorBackHomeAction` 跨楼层自主回桩
- `slamtec.agent.actions.SeriesMoveToAction` 包含多个目标点的自主导航移动
- `slamtec.agent.actions.MoveByAction` 遥控移动, 需要定时调用以达到连续运动效果
- `slamtec.agent.actions.GoHomeAction` 自主回桩
- `slamtec.agent.actions.RotateToAction` 原地旋转，转到指定角度
- `slamtec.agent.actions.RotateAction` 原地旋转，转动指定角度
- `slamtec.agent.actions.MoveToTagAction` 二维码精准对接
- `slamtec.agent.actions.BackOffFromTagAction` 从二维码前后退,防止碰撞。
- `slamtec.agent.actions.RecoverLocalizationAction` 重定位
- `slamtec.agent.actions.ManualRelocalizationAction` 手动重定位
- `slamtec.agent.actions.SweepAction` 覆盖规划运动，适用于清扫、消毒等场景，所需固件版本为4.4
- `slamtec.agent.actions.ReturnToParkingAction` 自主返航回到待命点（POI类型为PARKING），支持多机避障和排队功能（需要Lora模块），所需固件版本为4.5.5

5.6.2 创建新的运动行为

■ **POST** `/api/core/motion/v1/actions`

action_name选择 `slamtec.agent.actions.MultiFloorMoveAction`，不同运动行为可通过5.6.1中获取到的action类型选择

5.6.2.1 Request Body

示例如下，表示以精确到点、精确到角（yaw值为弧度1.0）的方式移动到A101。

代码块

```
1 {
2   "action_name": "slamtec.agent.actions.MultiFloorMoveAction",
```

```

3      "options":{
4          "target":{
5              "poi_name":"A101"
6          },
7          "move_options":{
8              "mode":2,
9              "flags":[
10                 "with_yaw",//
11                 "precise"
12             ],
13              "yaw":1,
14              "acceptable_precision":0,
15              "fail_retry_count":0
16          }
17      }
18  }

```

1. mode:默认是 0

- 0：自由导航
- 1：严格轨道模式(遇障碍物停止并等待)
- 2：轨道优先模式(遇障碍物下轨绕行)

2. flags:

- precise 精确到点模式，使机器人到点时更加精准
- with_yaw 精确到角模式，只有包含该flag，yaw字段的值才会生效
- fail_retry_count 指定搜路失败后的重试次数，不指定时采用默认配置
- find_path_ignoring_dynamic_obstacles 搜路时忽略动态障碍物，适用于人群拥挤、通道狭窄的区域

3. yaw:

到目标点后机器人的朝向，精确到角

4. acceptable_precision:

可接受的到点范围，当目标点被占据时，机器人离目标点距离在该范围内都算成功，默认值为0.1米或0.18米，该值不影响机器人到点精度。

5. fail_retry_count:

失败重试次数

5.6.2.2 Response Body

代码块

```

1  {
2      "action_id": 1,
3      "action_name":"slamtec.agent.actions.MultiFloorMoveAction",
4      "stage": "GOING_TO_TARGET",
5      "state": {
6          "status": 0,
7          "result": 0,
8          "reason": ""
9      }
10 }

```

1. action_id:

为整数，是Action的标识，后续查询状态要用该值；

2. action_name:

和 Request Body 请求传入的参数一致；

3. stage:表示Action执行的阶段;

4. state:表示action的状态

status :

0: 新创建, 1: 正在运行, 4: 已结束。

result :

0: 成功, -1: 失败, -2: 被取消。

reason 如果result是-1, 则该字段表示失败的原因

先判断state.status, 如果不是4, 则Action还在运行, 如果是4再根据result判断是成功还是失败。

5.6.3 查询 Action 状态

创建action时会返回一个action_id, 根据此id查询action的当前状态, 机器人运行过程中, 需要通过轮询此接口, 实时查看 Action 状态, 建议调用间隔1秒。

该接口的返回数据与POST /api/core/motion/v1/actions接口一样。

```
■ GET /api/core/motion/v1/actions/{action_id}
```

5.6.4 终止当前行为

```
■ DELETE /api/core/motion/v1/actions/:current
```

5.7 配送业务流程示例（需支持delivery配送服务）

注意事项: delivery相关接口仅限整机使用, 通用底盘默认不支持, 如需使用, 请联系FAE

接口 endpoint 前缀为 /api/delivery

整体上分为系统配置、cargo管理、任务管理等三大类

► 5.7.1 用户开始操作机器人

► 5.7.2 获取配送相关的设置信息

5.7.3 设置任务的超时时间

```
■ PUT /api/delivery/v1/settings/timeout
```

request body 示例如下, food_pickup_timeout表示到达目标点后等待时间, 单位为秒。

```
代码块
1  {
2    "food_pickup_timeout": 0
3  }
```

5.7.4 操作Cargo

如果是H2酒店配送机器人, APP通过cargo接口来打开/关闭舱门。如果没有cargo, 请忽略此操作。

```
■ PUT /api/delivery/v1/cargos/{cargo_id}/boxes/{box_id}/{op}
```

5.7.5 创建任务

用户放入物品并关闭舱门后, 调用以下接口创建Task。

```
■ POST /api/delivery/v1/tasks
```

目前支持的Task类型有 TAKEOUT（外卖配送）、GUIDE（引领）、FOOD_DELIVERY（送餐）、RECYCLE（回收）、RETURN（返航）、DISINFECT（消毒）

如果要一次性创建多个任务, 调用如下接口

■ **POST** /api/delivery/v1/tasks/:batch

5.7.6 开始执行任务

用户完成操作时，`enable_task_execution` 设为true允许机器人运动，此时机器人有任务则执行任务，没任务则回桩或回到类型为PARKING的POI。

■ **PUT** /api/delivery/v1/tasks/:task_execution

request body 示例如下

代码块

```
1  {
2    "enable_task_execution": true
3  }
```

5.7.7 暂停/继续执行任务

■ **PUT** /api/delivery/v1/tasks/:task_execution

参数名	类型	必填	说明
enable_task_execution	boolean	是	true :继续执行任务 false:暂停任务

request body 示例如下

代码块

```
1  {
2    "enable_task_execution": false
3  }
```

5.7.8 查询任务状态

在机器人执行任务过程中，APP需要定时查询任务状态来切换自身的界面

■ **GET** /api/delivery/v1/stage

- `DEVICE_ERROR` 设备故障，底盘上报了Error信息，机器人无法移动，调用端应当显示故障页面。
- `GOING_TO_TASK_POINT` 正在前往任务点，有些任务（如回盘、取物配送）需要中途停靠某些任务点，完成操作后再前往目标点。
- `ARRIVED_AT_TASK_POINT` 到达任务点，机器人会等待操作完成或超时后再继续下一阶段。
- `ON_DELIVERING` 正在前往目标点，为了兼容采用该名称，实际不一定是配送任务。
- `ARRIVED_AT_TARGET` 到达最终目标点。
- `ON_RETURNING` 正在返航，当机器人有默认停靠点时，该状态表示机器人正在前往该停靠点。
- `GOING_HOME` 正在回桩。
- `IDLE` 空闲，机器人在默认停靠点或桩上时处于该状态

此外还需要轮询事件接口来查询一些突发事件，比如路径被挡、收到新任务等，可参考5.4 获取事件信息

■ GET /api/platform/v1/events

5.7.9 开始取物

当APP查询到 Stage 为 ARRIVED_AT_TARGET 时，应当显示取物界面。

如果机器人有Cargo，需要显示【开舱】按钮，用户点击后，调用如下接口

■ PUT /api/delivery/v1/tasks/:start_pickup

接着调用cargo接口来打开舱门

5.7.10 完成取物

用户完成取物后调用该接口通知机器人

■ PUT /api/delivery/v1/tasks/:end_pickup

5.7.11 执行下一个任务

APP调用以下接口来允许机器人自主移动，如果有下一个任务则执行，如果没有则回桩或回待命点（POI类型为PARKING）

■ PUT /api/delivery/v1/tasks:task_execution

5.8 回充电桩

■ POST /api/core/motion/v1/actions

action_name选择 slamtec.agent.actions.MultiFloorBackHomeAction

request body 示例如下。

代码块

```
1  {
2      "action_name": "slamtec.agent.actions.MultiFloorBackHomeAction",
3      "options": {}
4  }
5
```

5.9 机器人异常恢复

将机器人位姿设置到指定的POI上，一般用于发生异常后的恢复操作，如恢复机器人定位信息。

注意事项：此接口通常在定位丢失后调用，如果把机器人推回充电桩后，则无需调用此接口，系统会自动做恢复动作。

■ PUT /api/multi-floor/localization/v1/pose

5.10 运动策略

运动策略为Slamware的一系列内部参数，涉及到运动速度、避障行为等各个方面，不同的策略可适用于不同的场景。一般情况下采用默认策略即可。

所需最低固件版本 4.2.4

获取支持的所有运动策略：

■ GET /api/core/motion/v1/strategies

获取当前运动策略：

■ GET /api/core/motion/v1/strategies/:current

设置运动策略：

■ PUT /api/core/motion/v1/strategies/:current

5.11 设备健康状态常见问题

描述	触发条件	触发后底盘动作	等级	错误代码	显示信息	消除方法
急停开关	急停开关触发	紧急停止，下发速度不再响应	ERROR	0x02010100	"system emergency stop"	释放急停开关恢复
低电量报警	电量低于15%	无	WARNING	0x01020100	"power low"	电量高于15%
低电量报警	电量低于5%	自动关机，所有电源关闭			"power low"	电量高于5%
刹车释放	刹车释放按钮触发	电机松轴，下发速度不再响应	ERROR	0x02010700	"motor brake released"	刹车释放按钮恢复
驱动电机报警	驱动电机驱动器报警	依赖驱动器固件，如过流报警时会松轴。	WARNING	0x01030100	"motor alarm"	底盘固件会尝试清除驱动器报警后，再次检测是否有报警，若无再次报警则自动清除health
电机里程计报警	电机在停止时（无下发速度，或下发速度为0），电机移动超过500mm	无	WARNING	0x0103010x	"motor[y]odometry alarm"	有速度下发，或刹车释放触发
底盘开门狗触发	底盘固件看门狗触发，固件重启	固件重启后，下发速度不再响应	ERROR	0x02030400	"watchdog overflow"	手动消除
磁传感器触发	磁传感器触发	立即停止后，下发的速度不再响应	ERROR	0x0204050x	"magtape[x] triggered"	急停开关触发，或手动清除
磁传感器通信错误	磁传感器通信错误	立即停止后，下发的速度不再响应	FATAL	0x0404060x	"magnetic[x]:y."	检查连接线是否可靠连接，检查传感器是否损坏，手动清除错误，必要时重启底盘
tof跌落通信错误	tof跌落通信错误	立即停止后，下发的速度不再响应	FATAL	0x0404020x	"cliff[x]:y."	检查连接线是否可靠连接，检查传感器是否损坏，手动清除错误，必要时重启底盘
碰撞传感器错误	碰撞持续触发且向前行走200mm	立即停止后，下发的	ERROR	0x0204010x	"bumper sensor error"	碰撞信号消除

		速度不再响应				
tof跌落信号错误	tof跌落持续触发且向前行走200mm	立即停止后，下发的速度不再响应	ERROR	0x0204020x	"cliff sensor error"	tof跌落信号消除
低定位错误			ERROR	0x02010900	"Low localization due to great environmental change, because visual coarse pose received" 或"Low localization quality"	
重定位错误			ERROR	0x02010800	"Relocalization has failed last time, clear the error to move"	
onlineslam reboot 定位异常	onlineslam 重新启动	不执行action	ERROR	0x02010600	"slamware has rebooted, clear the error to move"	将机器人推回充电桩会回复（若在同一区域重复出现多次，请重新建图）

6. 二维码精准对接

关于精准对接功能二维码部署流程请参考《二维码精准对接部署手册》。

本文只讲解如何通过API调用

6.1 获取部署好的POI信息

■ GET /api/core/artifact/v1/pois

找到type为TAG的POI，根据display_name找到所需POI，记下pose和tag_ids，填入MoveToTagAction参数中。如果缺少tag_ids信息，请检查POI部署流程。

6.2. 创建精准对接运动行为

■ POST /api/core/motion/v1/actions

action_name选择 slamtec.agent.actions.MoveToTagAction

request body 示例如下

代码块

```
1  {
2      "action_name": "slamtec.agent.actions.MoveToTagAction",
3      "options": {
4          "target": {
5              "x": 0.590,
6              "y": 0.110,
7              "yaw": -3.130
8          },
9          "tag_ids": [0,50],
10         "relative_pose_to_tag":{
11             "x":0.4,
12             "y":0.0
13         }
14     }
15 }
16
```

其中target和tag_ids为6.1记录的数据。

`relative_pose_to_tag`字段可以不填，`x`表示离二维码中心的纵向距离，`y`表示离二维码中心的横向偏差。

若不填写，二维码对接默认使`precise_move_to_tag.safe_distance_to_tag`（7.5cm）作为`x`的默认值，`y`值默认为0。

6.3. 操作完成后先调用后退action，以免碰撞

■ `POST /api/core/motion/v1/actions`

action_name选择 `slamtec.agent.actions.BackOffFromTagAction`

▼ 代码块

Java | 复制

```
1 {  
2   "action_name": "slamtec.agent.actions.BackOffFromTagAction",  
3   "options": {}  
4 }
```