
目錄

| | |
|----|-----|
| 简介 | 1.1 |
| 安装 | 1.2 |

调度系统客户端使用

| | |
|----------|-----|
| 注册用户 | 2.1 |
| 调度系统界面简介 | 2.2 |
| 创建和上传地图 | 2.3 |
| 机器操控 | 2.4 |
| 绘制地图路网 | 2.5 |
| 调度系统控制 | 2.6 |
| 调度任务管理 | 2.7 |

调度系统API说明

| | |
|------|-----|
| 快速开始 | 3.1 |
| 文档说明 | 3.2 |

- [1. 引言](#)

1. 引言

拉格朗日调度系统是一个优秀的机器人调度系统。具有动态路径资源分配，机器人管理，地图管理等功能。本文档包含了调度系统客户端使用方法和API说明文档。

调度系统支持用户私有服务器和云服务器两种模式。在机器人有良好的网络情况下可以使用云服务器。

[在线体验地址](#)

这是一个云服务器Demo，可以直接体验。用户名是guest密码是guest123。

对于私有服务器，配置要求内存至少8G，硬盘至少100G，操作系统Windows或Ubuntu都可以。

- 1. 安装

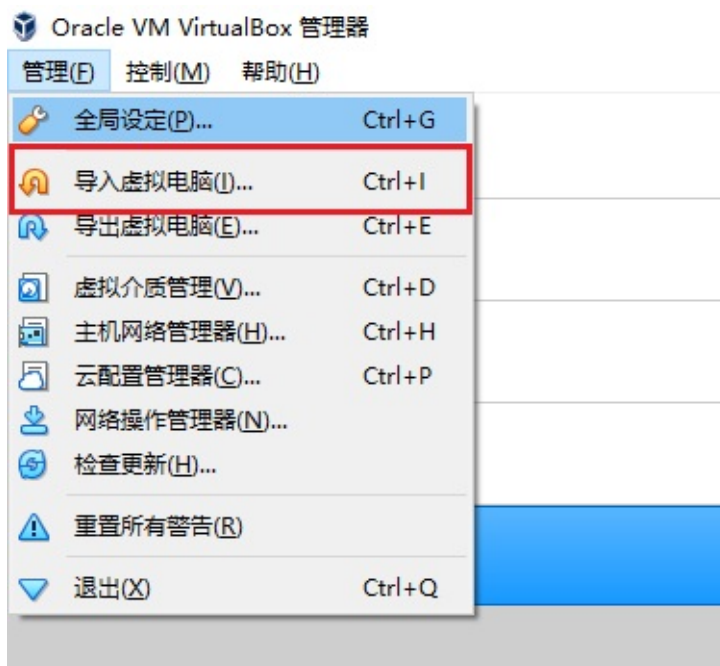
1. 安装

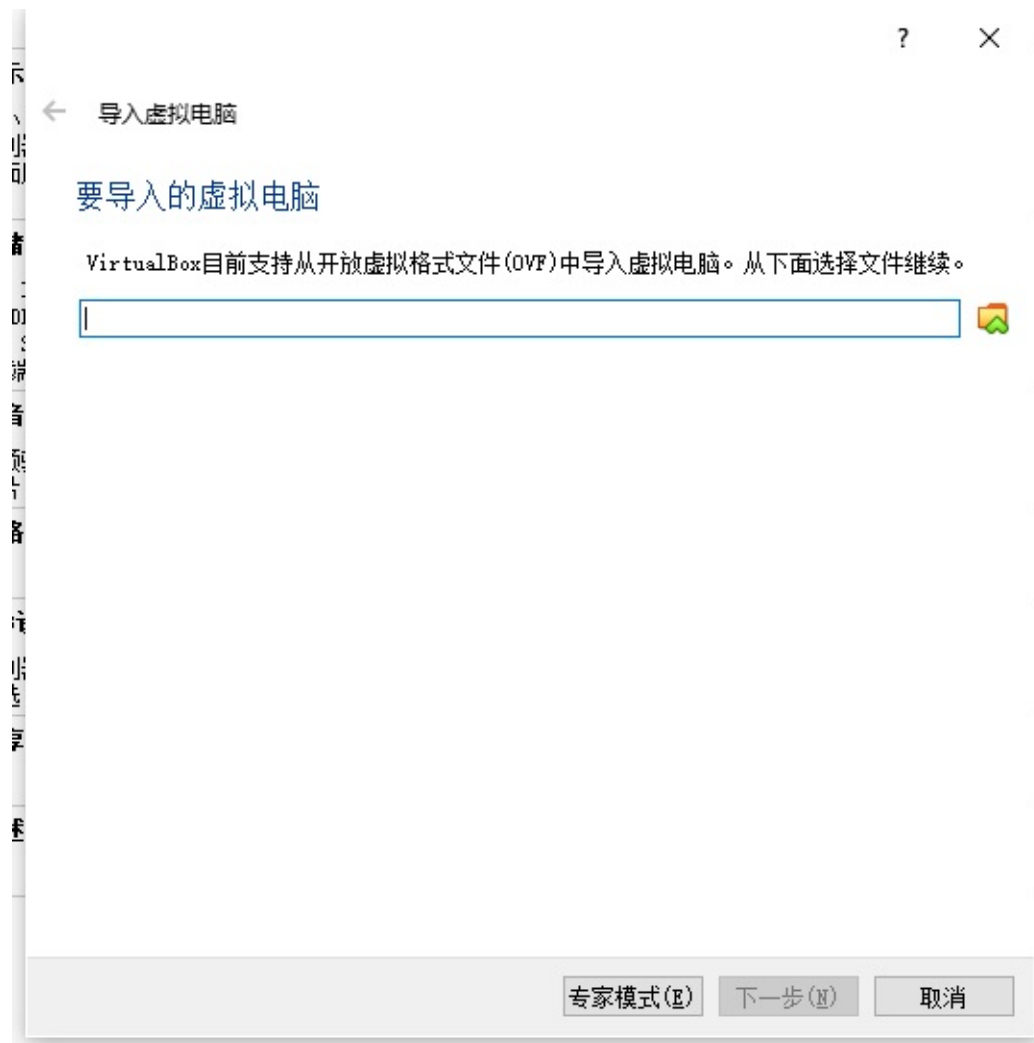
对于使用云服务器的用户，可以直接打开我们提供的服务器地址，无需安装

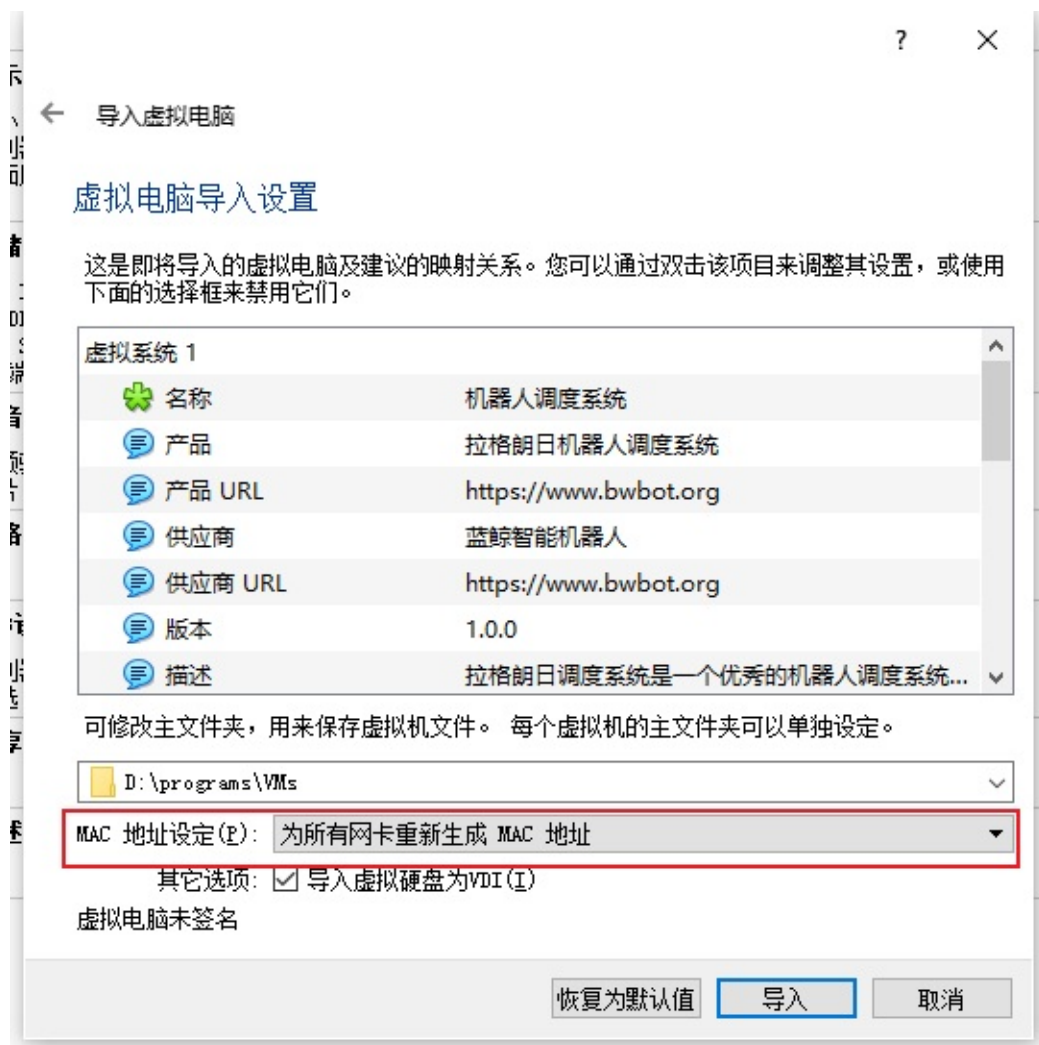
[点击此处](#)下载调度系统服务程序

安装Virtualbox

安装完成后导入之前下载的调度系统服务程序







继续导入设置，注意网卡部分要按照图中设置。

等待导入完成，之后选择导入的虚拟机然后点击显示

等待虚拟机启动完成

```
Ubuntu 18.04.3 LTS lagrange tty1  
lagrange login: _
```

使用用户名 lagrange 密码 lagrange 登录。登录进入后执行 `ip addr` 获取虚拟机ip

```
Ubuntu 18.04.3 LTS lagrange tty1
lagrange login: lagrange
Password:
Last login: Mon Nov 25 10:41:52 CST 2019 on tty1
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-66-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
lagrange@lagrange:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6a:c6:0e brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.172/24 brd 192.168.0.255 scope global dynamic enp0s3
        valid_lft 7014sec preferred_lft 7014sec
    inet6 fe80::a00:27ff:fe6a:c60e/64 scope link
        valid_lft forever preferred_lft forever
lagrange@lagrange:~$
```

从输出中获取到服务器ip。比如上图中的服务器ip为192.168.0.172

在自己的浏览器中输入 `http://192.168.0.172:24958` 即可打开调度系统控制页面。

在获取到ip后为了方便以后使用可以到路由器设置页面中把虚拟的ip设置成固定ip。

以后就可以打开虚拟机直接使用了，不需要再获取虚拟机ip。

- 1. 用户注册
 - 1.1. 权限管理
 - 1.1.1. 申请权限
 - 1.1.2. 管理权限申请

1. 用户注册

当首次打开调度系统页面时系统会自动跳转到登录界面。默认的管理员账号密码为admin。请登录及时修改密码。

如果没有账号要先注册账号。点击注册按钮开始注册用户。普通用户注册后需要向管理员申请权限才能够进行对应操作。

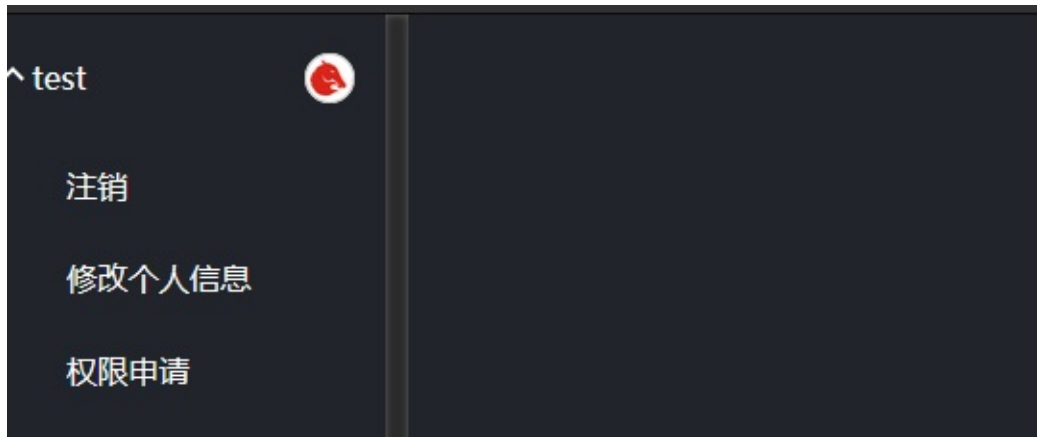


注册完成后会自动跳转到登录界面，在输入用户名和按钮后点击登录按钮即可登录。

1.1. 权限管理

普通用户在进行不同的操作之前要获取对应的权限。权限可以向管理员申请。

1.1.1. 申请权限



点击左侧用户菜单中的权限申请打开权限申请面板。



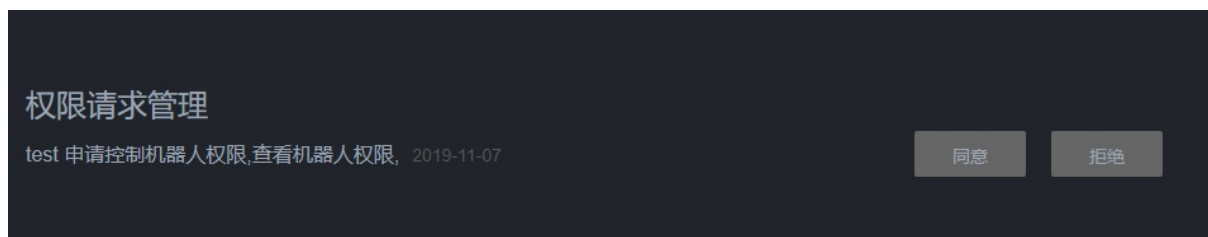
点击自己需要的权限然后点击确认申请按钮即可申请权限。确认后管理员会收到用户的权限请求信息。当管理员同意后对应用户即可获得权限。

1.1.2. 管理权限申请

管理员登录后，左侧菜单栏会出现用户管理菜单



点击权限管理菜单，打开权限管理面板。此时会显示用户的权限申请请求。



点击同意或拒绝按钮即可完成对权限申请操作。

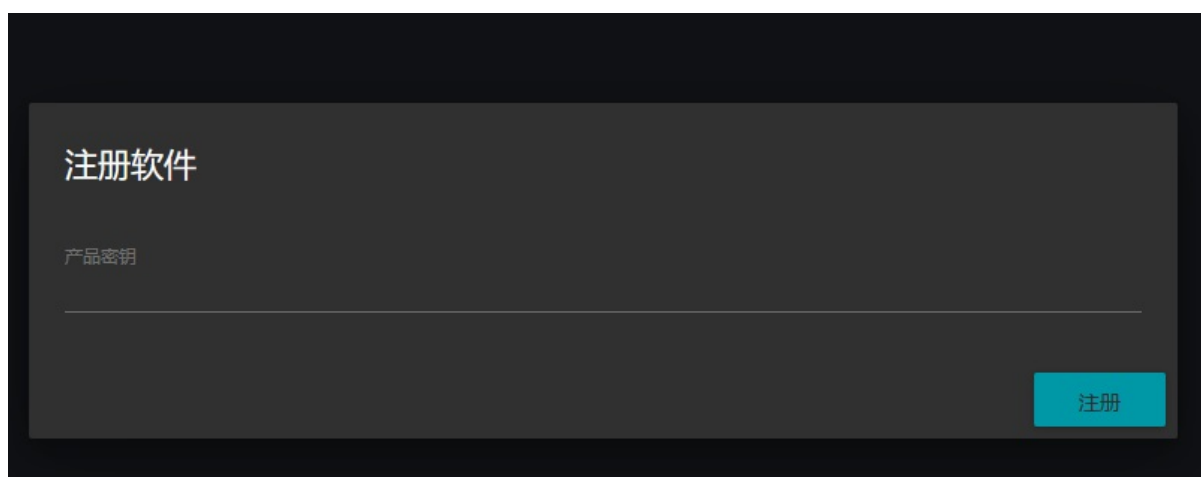
- 1. 调度系统界面简介
 - 1.1. 调度系统软件注册
 - 1.2. 系统主界面介绍
 - 1.3. 基础操作

1. 调度系统界面简介

1.1. 调度系统软件注册

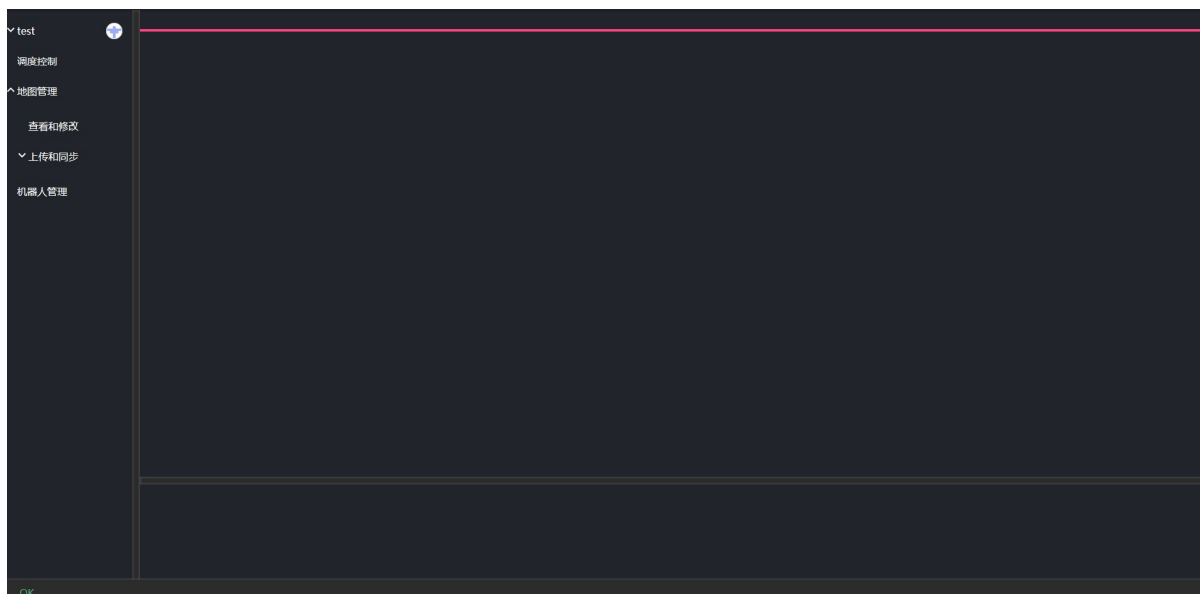
首次使用时请保证服务器能够正常访问网络

在用户登录后如果软件未注册则会提示注册信息，如下图所示。



输入产品密钥字符串，然后点击注册按钮，等待注册完成。

1.2. 系统主界面介绍

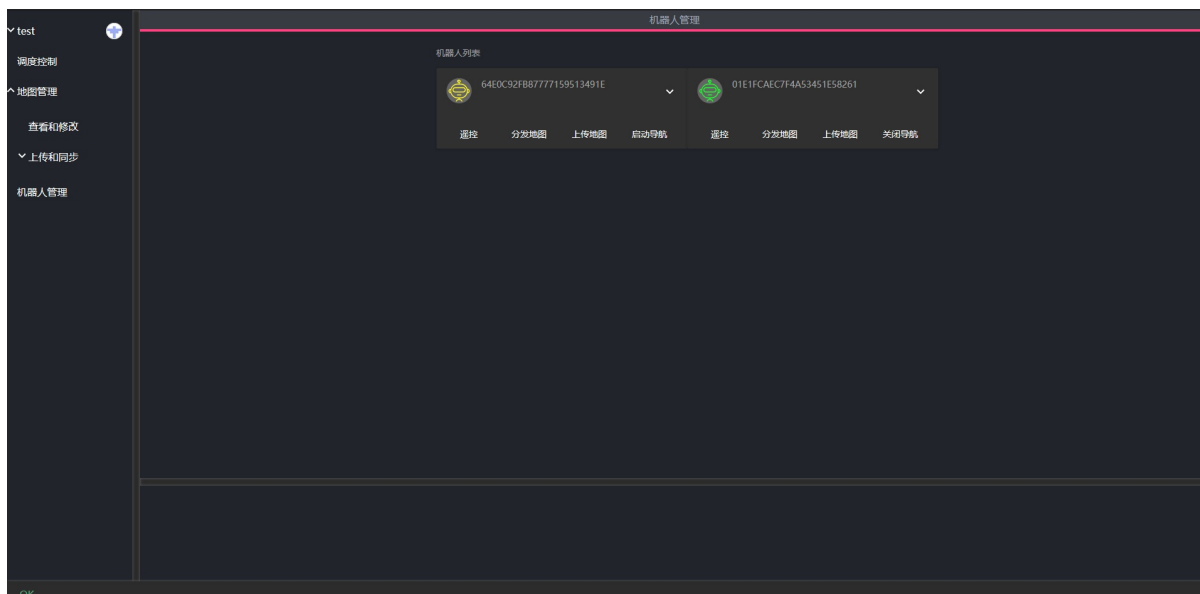


系统左侧为菜单栏。可以点击对应的菜单开启不同的功能面板。右侧的大块面板区域为主面板区。用于显示不同的功能面板。下方为底部面板，用于改变选择目标的属性和发布任务。

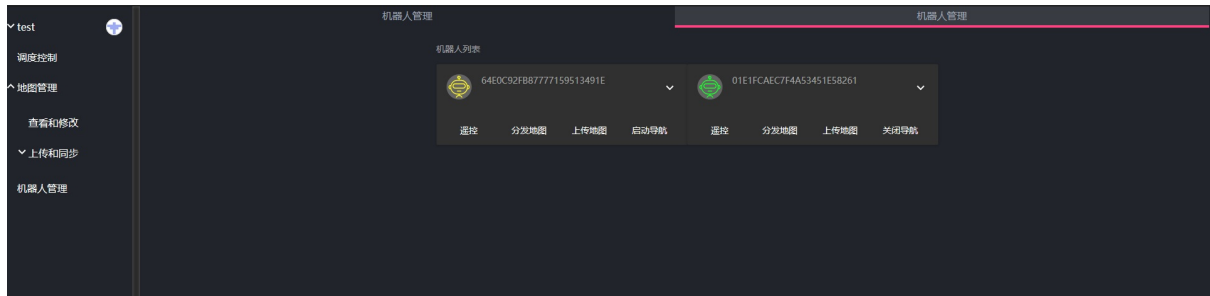
1.3. 基础操作

打开和关闭面板

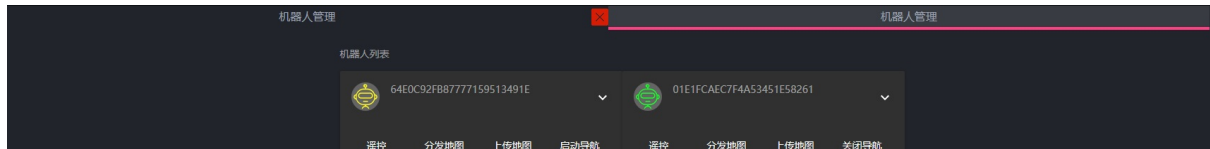
点击对应的左侧菜单会打开不同的面板，比如点击机器人管理后会如下显示



再次点击左侧菜单可以再次打开一个面板。多个面板可以同时打开，且相互独立。



鼠标移动到面板标题栏的时候会显示关闭按钮，点击关闭按钮即可关闭对应面板。



- 1. 创建和上传地图
 - 1.1. 创建地图
 - 1.2. 激活机器人
 - 1.3. 上传地图
 - 1.4. 分发地图
 - 1.5. 查看地图

1. 创建和上传地图

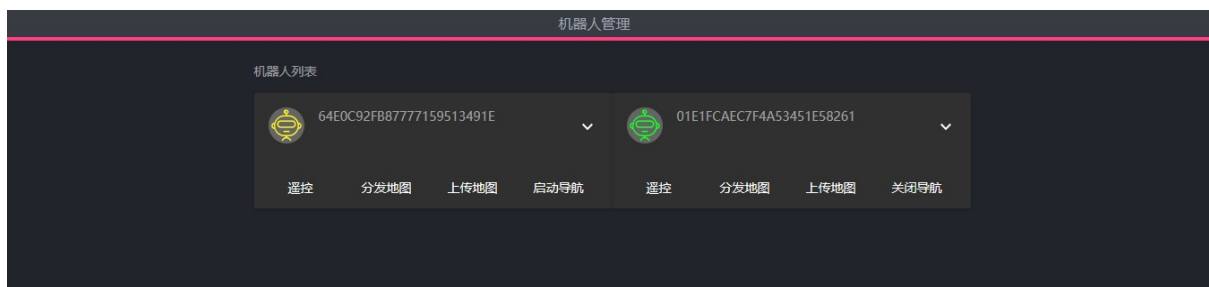
1.1. 创建地图

创建地图使用伽利略导航客户端。具体的地图创建方法参考[伽利略导航客户端手册](#)

地图创建完成后即可关闭伽利略客户端。

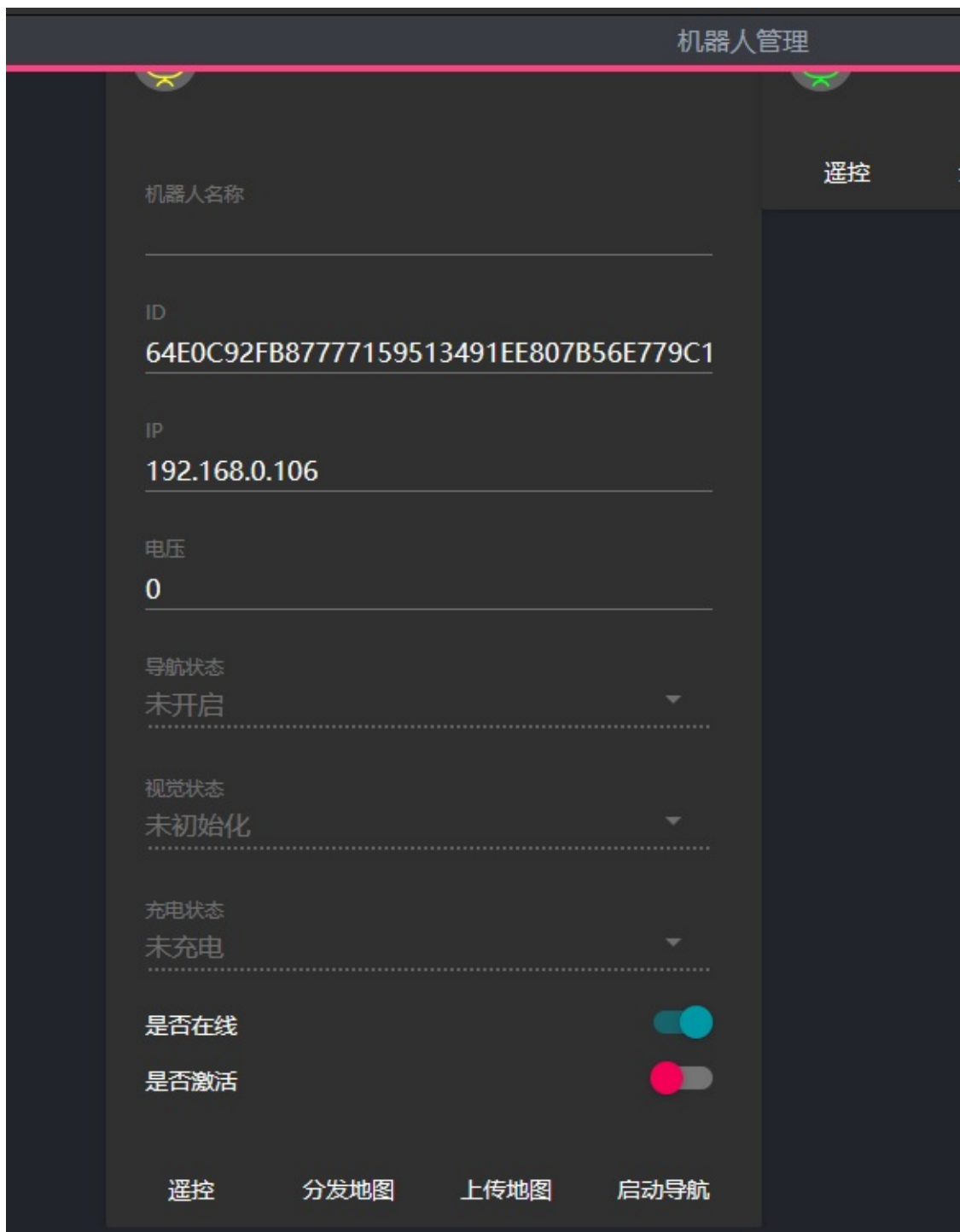
1.2. 激活机器人

在调度系统主界面中，点击左侧的机器人按钮。此时主界面中会显示当前局域网内的机器人。



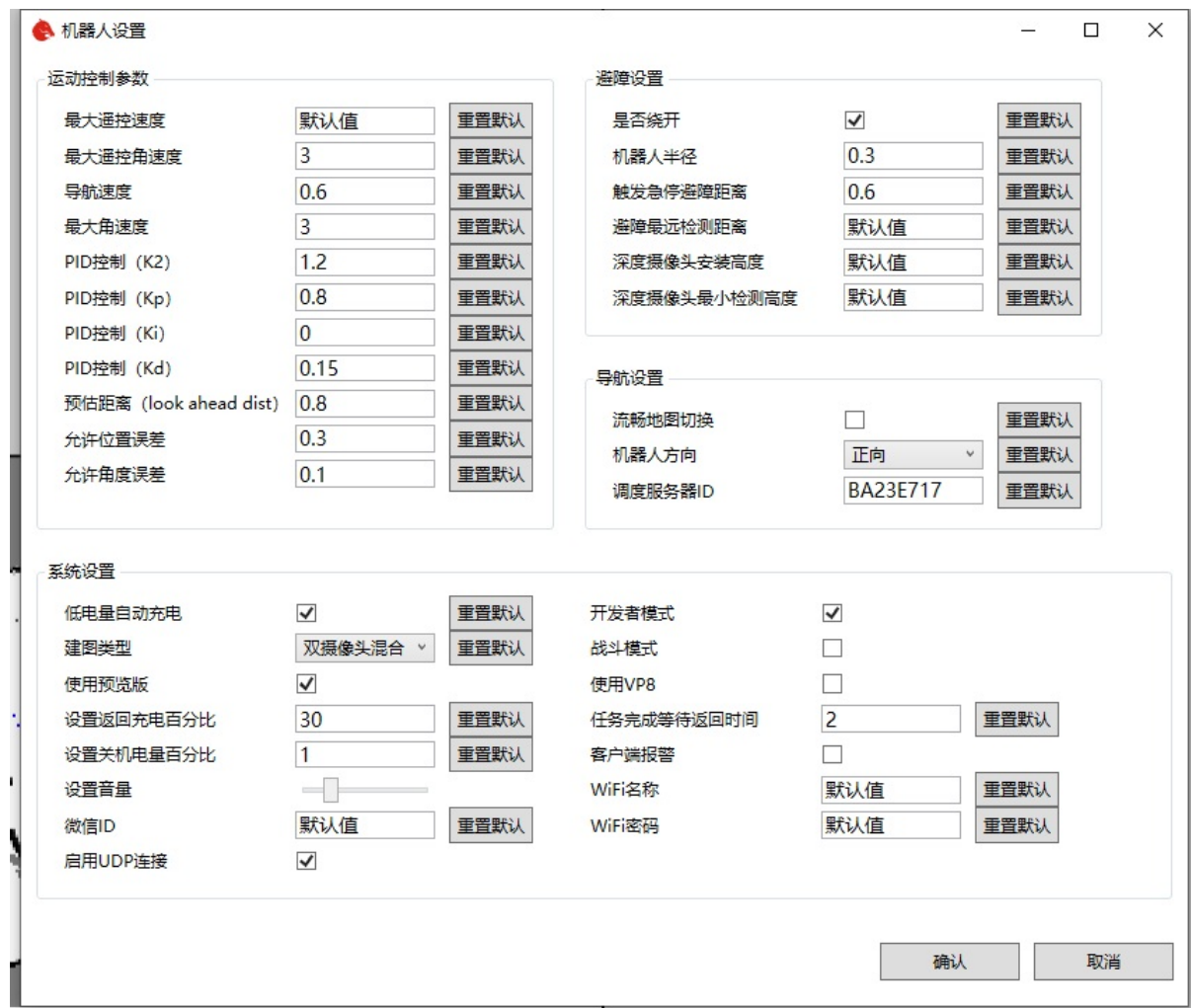
其中灰色代表不在线的机器人。黄色表示未激活的机器人。绿色表示已经激活的机器人。机器人只有在激活后才能被调度系统控制管理。

点击向下的三角，展开机器人控制面板



点击下面的是否激活状态按钮。此时机器人即被激活了。

对于使用云服务器器的情况，机器人不会自动在机器人管理中显示。首先需要把调度服务器的ID设置到机器人中。用伽利略导航客户端连接到机器人，并打开机器人设置界面。如下图所示

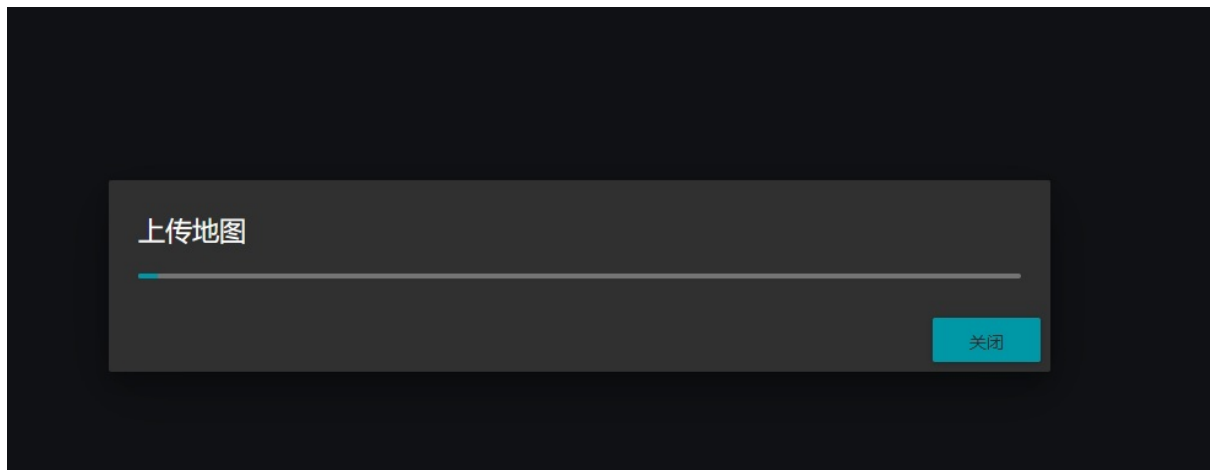


在导航设置，调度服务器ID中添加自己的云服务器ID。云服务器ID是云服务器地址的前八位。如 `http://7ce230e1.24958.robot1.bwbot.org:10427` 对应的ID就是 `7ce230e1`。

添加后过几秒就可以在机器人管理中看到机器人了。剩余操作和局域网机器人一样。

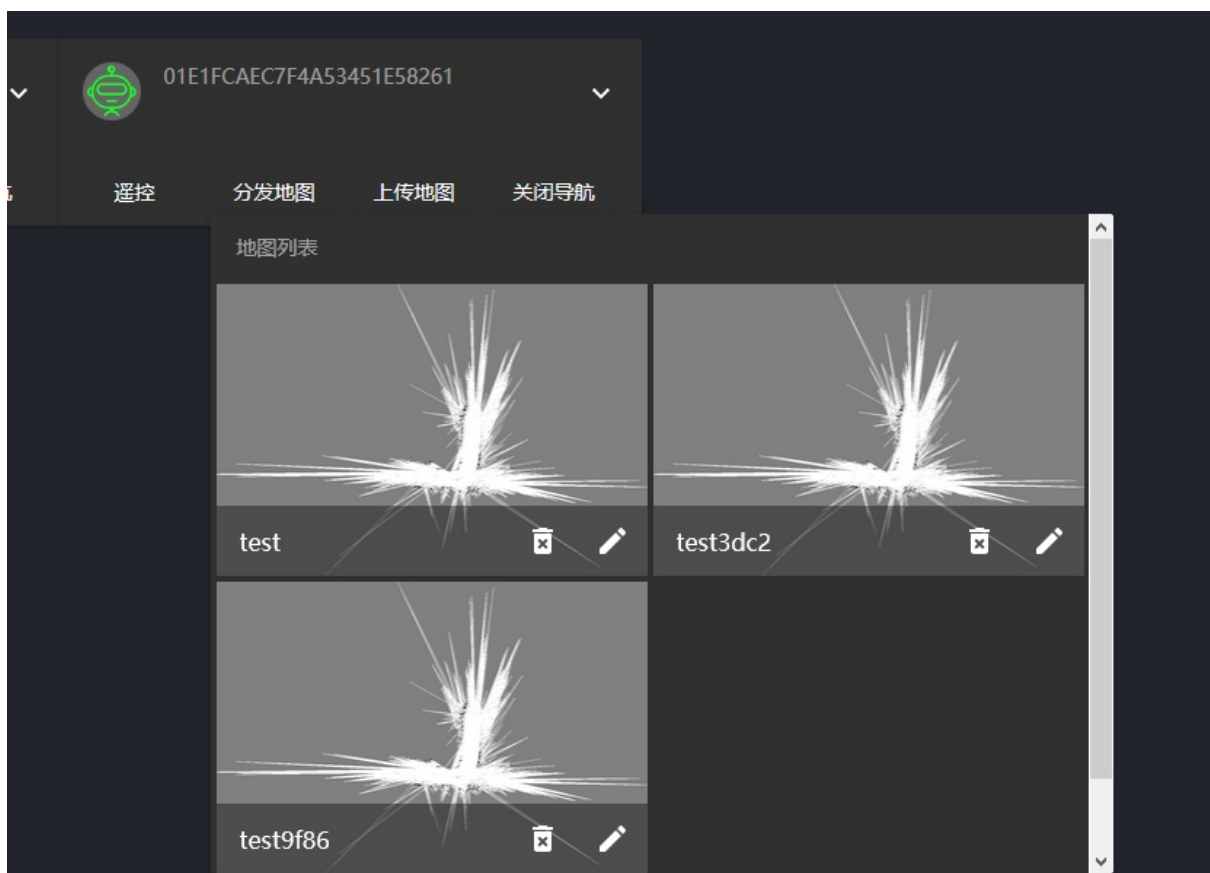
1.3. 上传地图

点击机器人上传地图按钮即开始上传。上传地图时会把机器人之前创建的地图上传到服务器中。



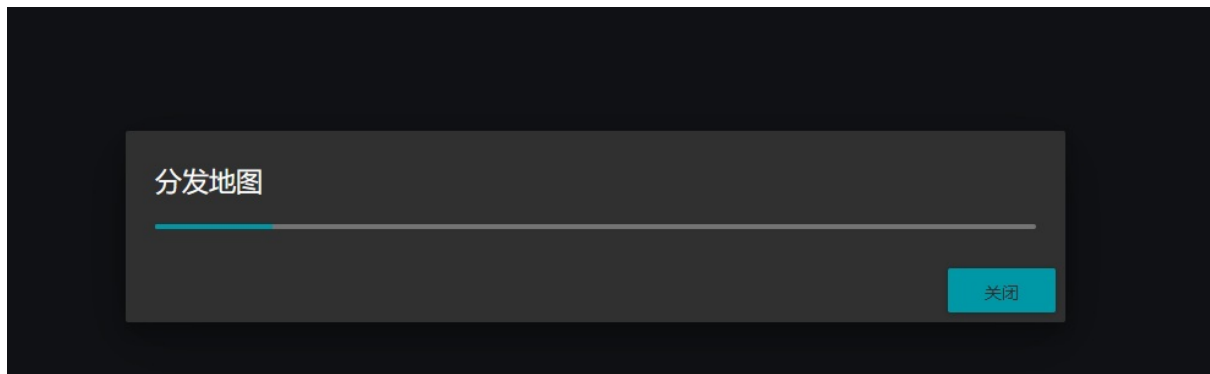
1.4. 分发地图

点击机器人面板中的分发地图即可分发地图。分发地图时会将地图从服务器复制到目标机器人，用于机器人之间共享地图。



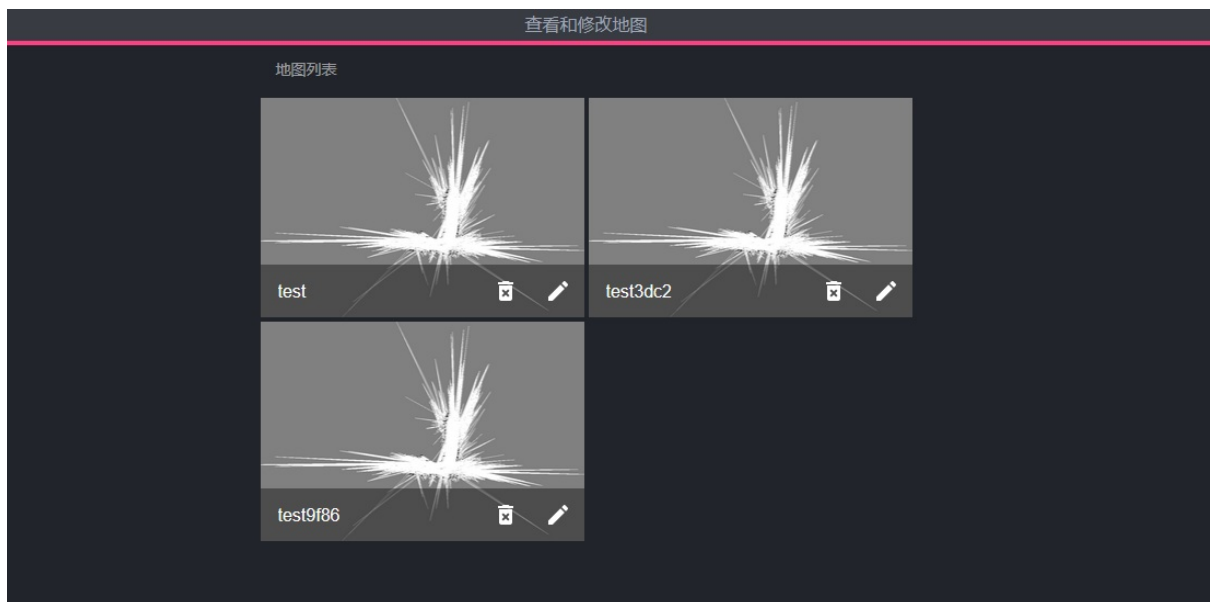
从地图中选择要分发的地图。

等待分发地图完成



1.5. 查看地图

地图上传完成后可以通过地图查看面板查看地图。点击左侧的地图管理选项，然后选择查看和修改。



界面中会列出当前服务器中所有的地图。你可以点击对应的按钮进行删除和重命名操作。

- 1. 机器操控
 - 1.1. 遥控机器人
 - 1.2. 控制机器人导航

1. 机器操控

调度系统提供了简单的机器人手动控制功能，用于更加方便的处理一些异常情况。

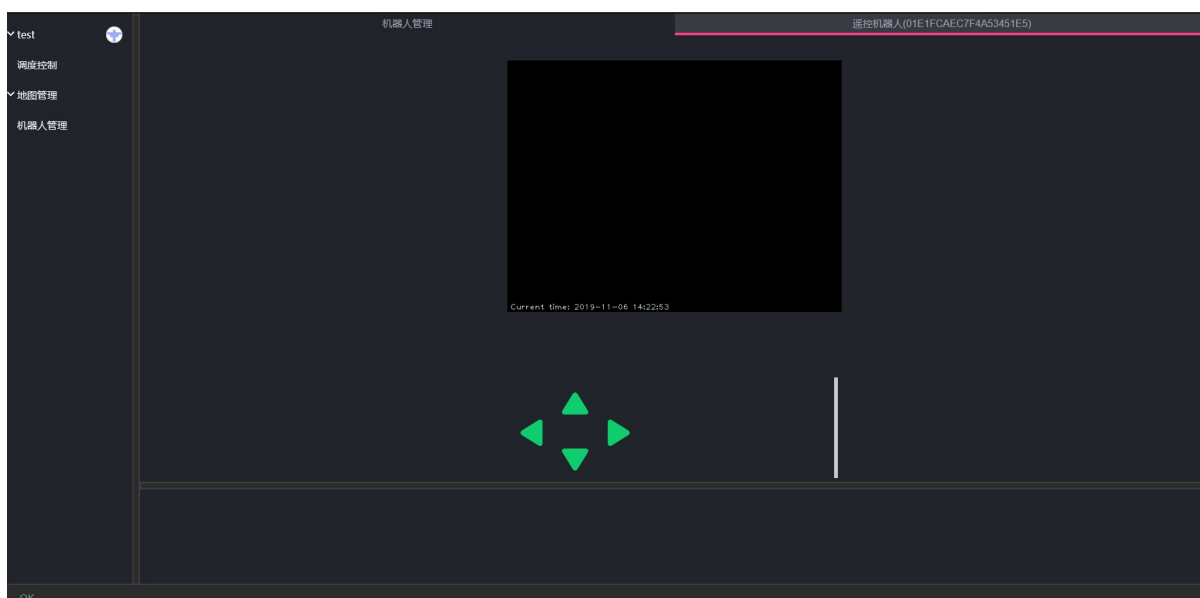
点击左侧的机器人管理菜单，打开机器人控制面板。



在创建地图说明中我们已经介绍了如果激活机器人。下面介绍遥控和启动导航功能功能。

1.1. 遥控机器人

点击需要遥控的机器人的遥控按钮，此时会打开机器人遥控面板。



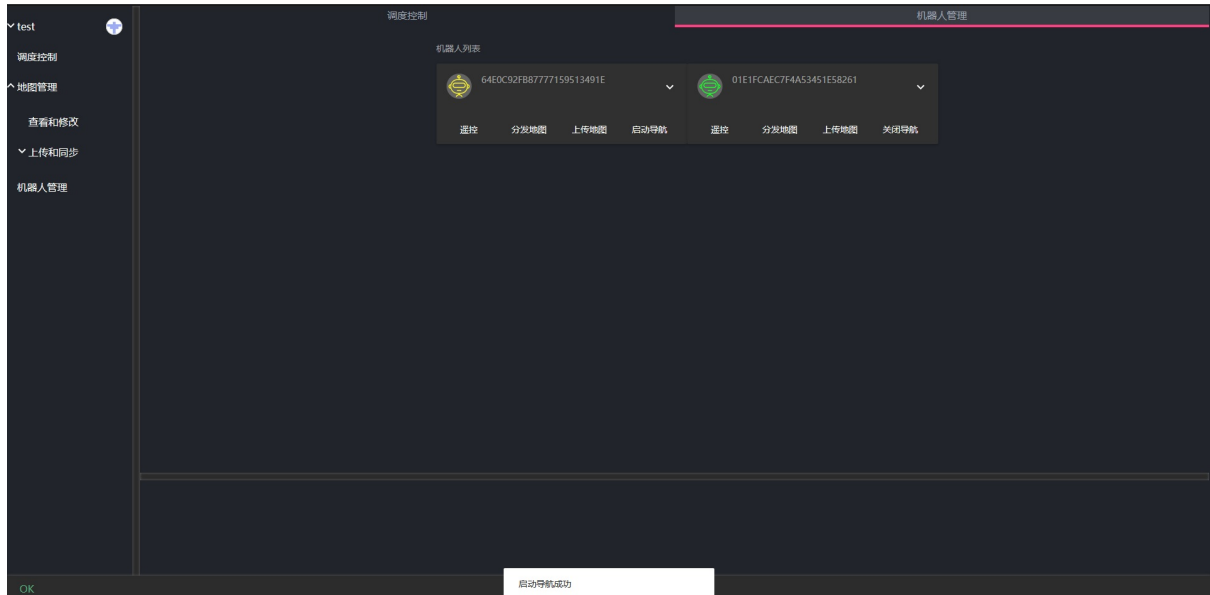
中间图像为机器人图传图像。点击左下角方向键可以控制机器人前进后退左转右转。也可以通过 WSAD 按钮控制对应方向。上下箭头控制机器人移动速度。右下角的色条显示当前机器人的速度档位。

注意当遥控机器人面板在后台时遥控仍然有效。只有遥控面板被关闭时才会停止遥控

我们可以利用这个功能边进行其他操作边进行遥控。

1.2. 控制机器人导航

在导航系统已经开启，但机器人尚未开启导航程序时。我们可以利用开启导航按钮开启机器人导航。



点击导航按钮等待提示启动导航成功。如果启动失败系统也会提示对应的失败原因。

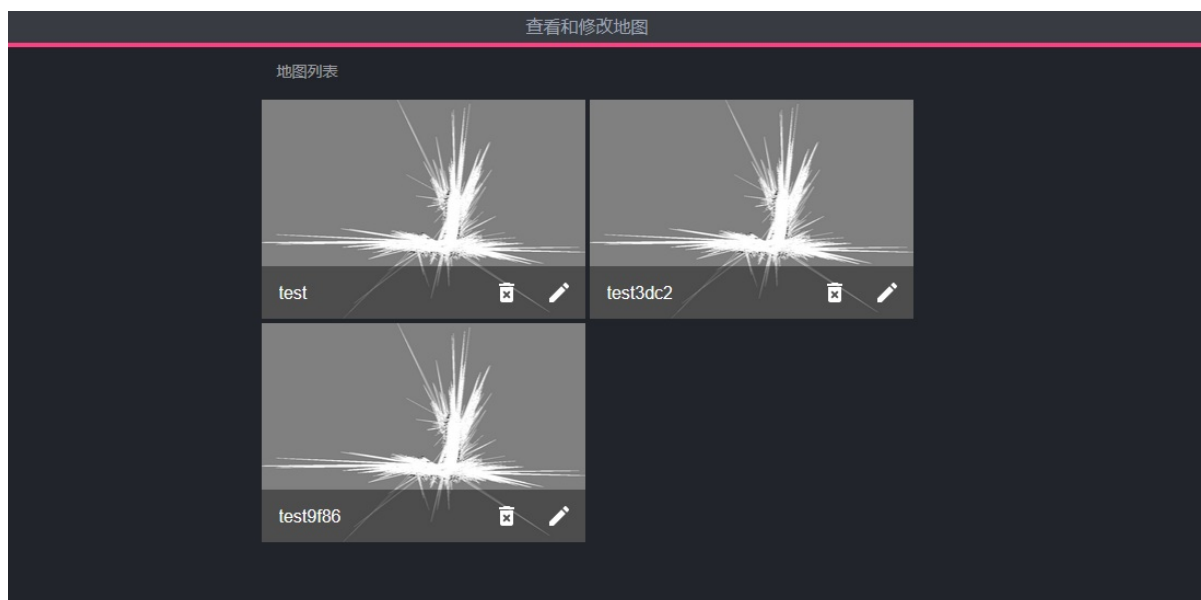
同样在机器人处于导航状态下时我们也可以点击关闭导航按钮，关闭机器人导航功能。

在这里点开始导航机器人默认会采用上次导航的地图和路径。请注意相关参数是否正确。

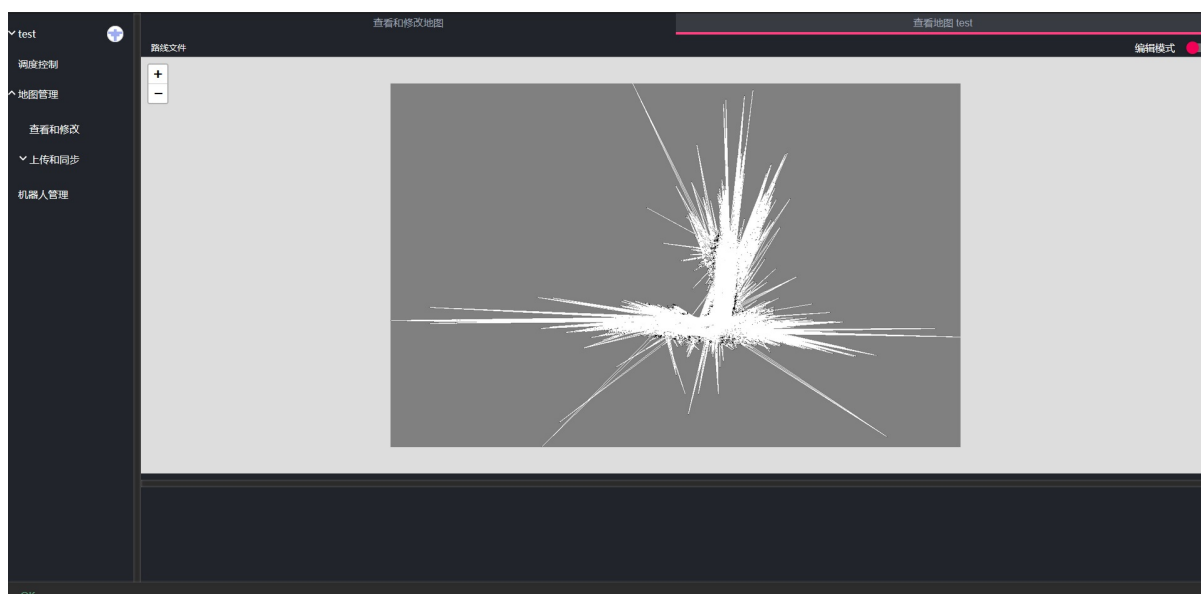
- 1. 绘制地图路网
 - 1.1. 直线工具
 - 1.2. 点工具
 - 1.2.1. 设置点的属性
 - 1.3. 区域工具
 - 1.3.1. 保存路网信息

1. 绘制地图路网

点击左侧的地图管理菜单，然后点击查看和修改菜单，打开查看和修改地图面板。此时面板会显示当前服务器中的所有地图。



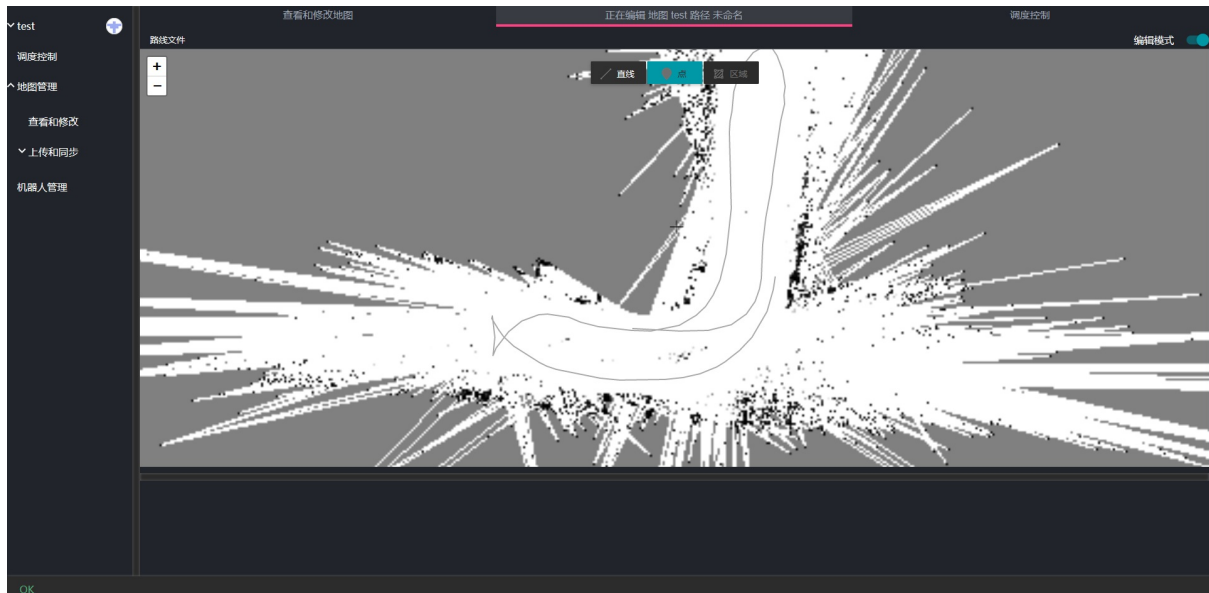
点击自己要绘制路网信息的地图，进入路网编辑面板。



界面中的地图即为机器人创建的地图，我们现在需要在此地图上绘制机器人能够走的路线。同时标定机器人待机点和机器人充电点等等。

点击右上角的编辑模式按钮进入编辑模式

鼠标拖动可以移动地图。鼠标滚轴进行地图缩放。

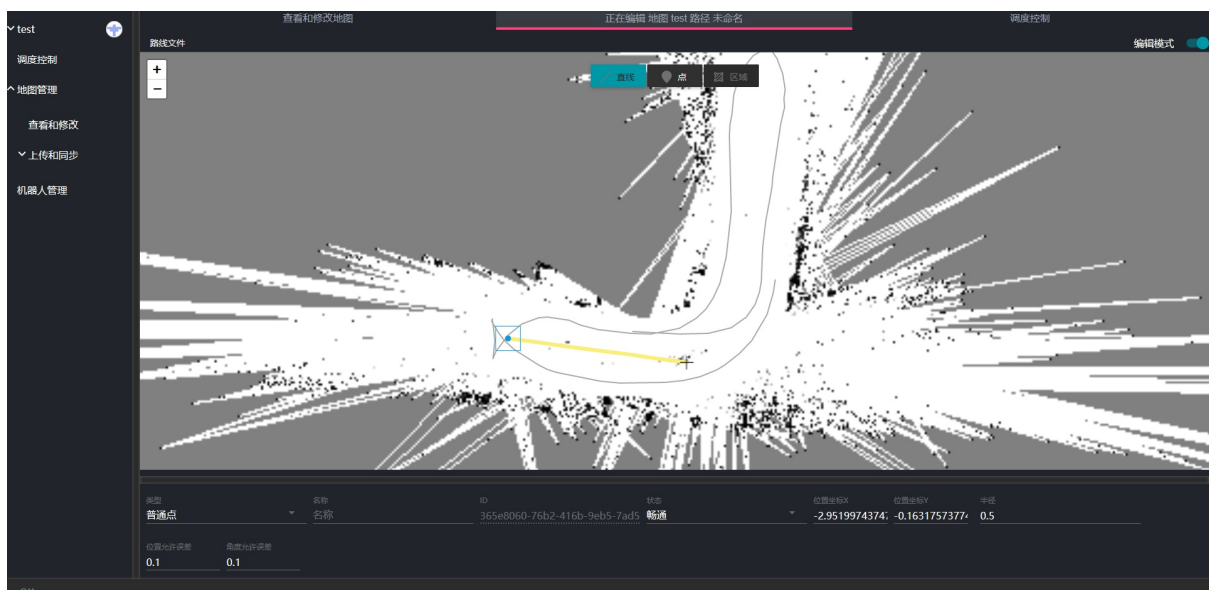


中上方的按钮为绘图工具按钮。通过点击对应的工具可以绘制直线或者点。地图上灰色的路线为创建地图时机器人所行走的路径。可以作为绘制路径的位置参考。

1.1. 直线工具

点击中间上方的直线工具，进入绘制直线的状态。

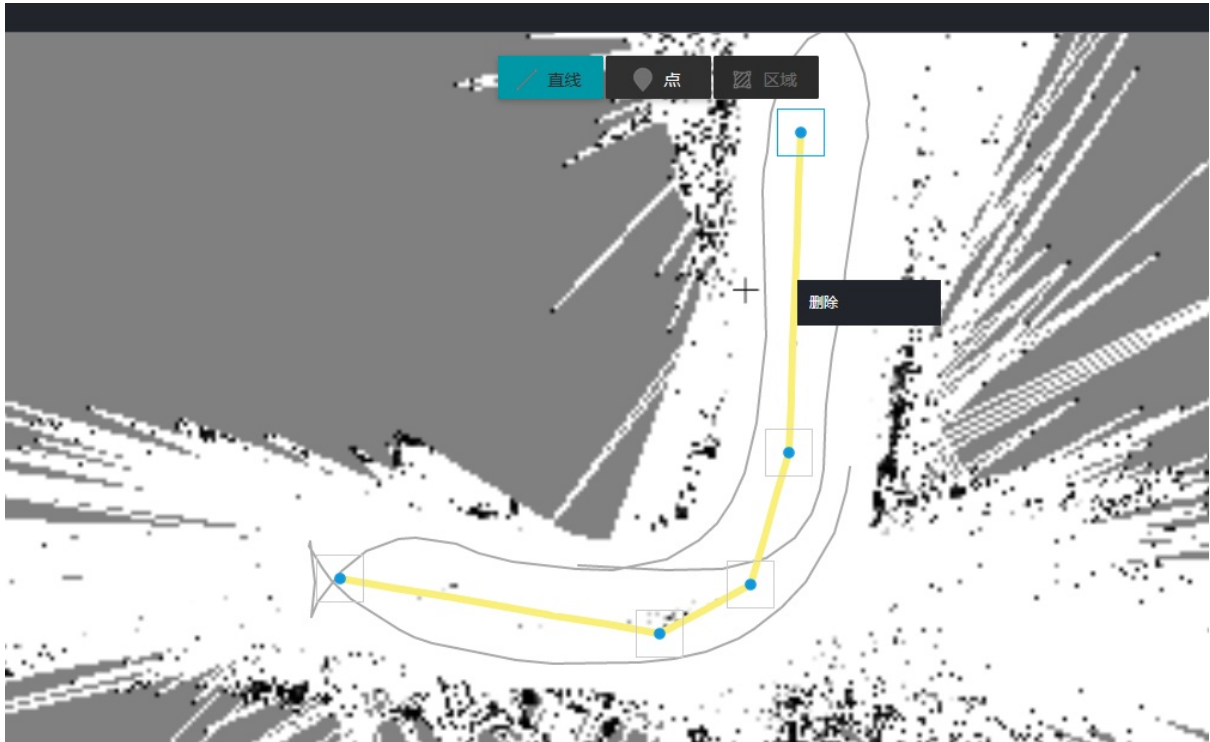
点击地图中的一个点作为起点，此时移动鼠标，会发现一条黄色的线跟随鼠标移动。



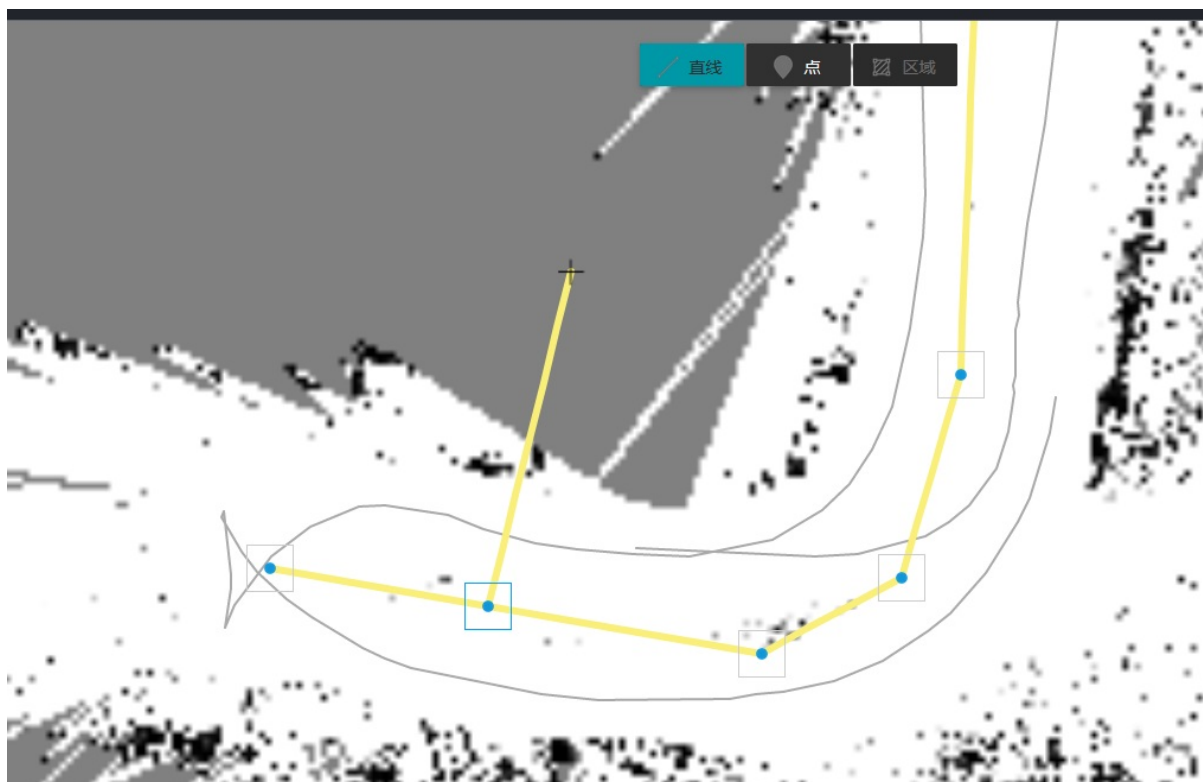
之后依次点击地图上的点绘制路径。最后可以按键盘ESC键结束绘图。

如果想要接上之前绘制的直线，只需要点击之前绘制的直线中的端点然后就可以继续以同样的方式绘图了。

如果想要删除直线，只需要在目标线段上点击右键。此时会弹出一个删除菜单，点击删除菜单即可删除对应线段。



如果想在之前绘制的线段中间连出新的一段线段，可以直接在线段上想要插入的位置点击鼠标（注意要在鼠标变成小手的形状后才点击，否则新插入的点不在线上）。



1.2. 点工具

点工具用于在目标位置添加点。比如可以用于设置充电桩位置，待机点位置等等。

1.2.1. 设置点的属性

路网中的点有时具有特殊的意义。比如充电位置，待机位置等等。我们可以通过点工具选中想要设置的点，然后在底部属性面板中编辑点的属性。点被选中后其边框会变成蓝色。

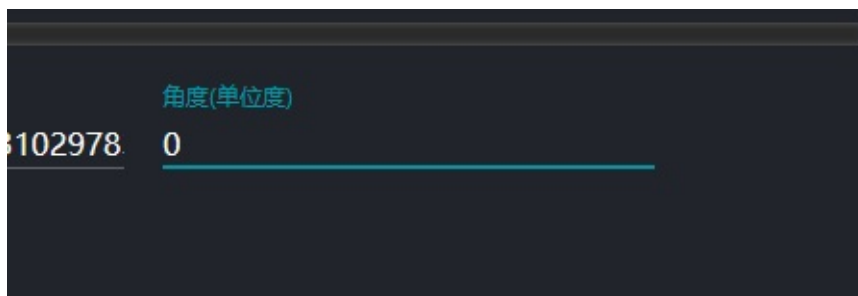


下面以设置充电桩位置为例子介绍

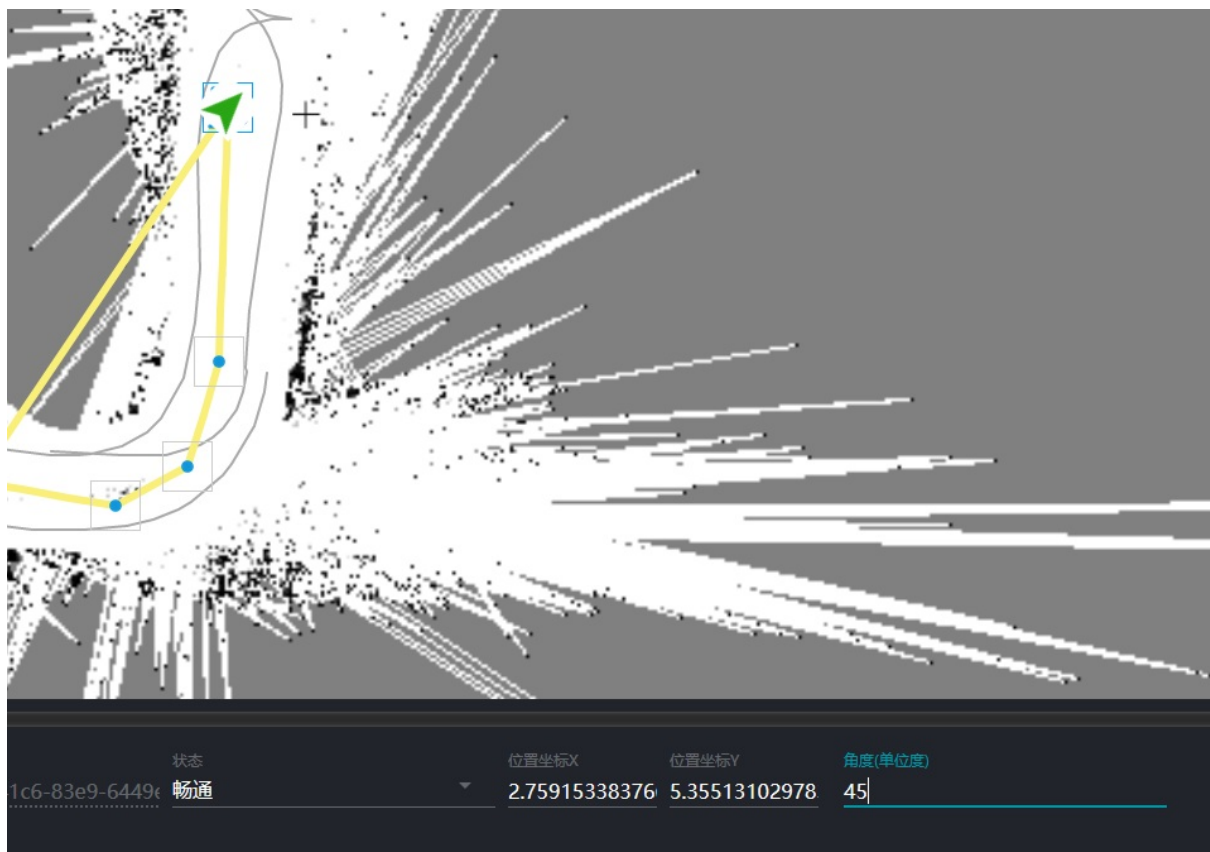
在点类型中选择充电点



由于充电桩有特定的方向，我们需要设置充电桩的正确角度后机器人才能成功的自动充电。通过修改底部面板角度值，设置充电桩角度。



可以看到我们在把角度设置成45°之后地图中的充电桩角度也跟着变化。



1.3. 区域工具

此工具暂未开放

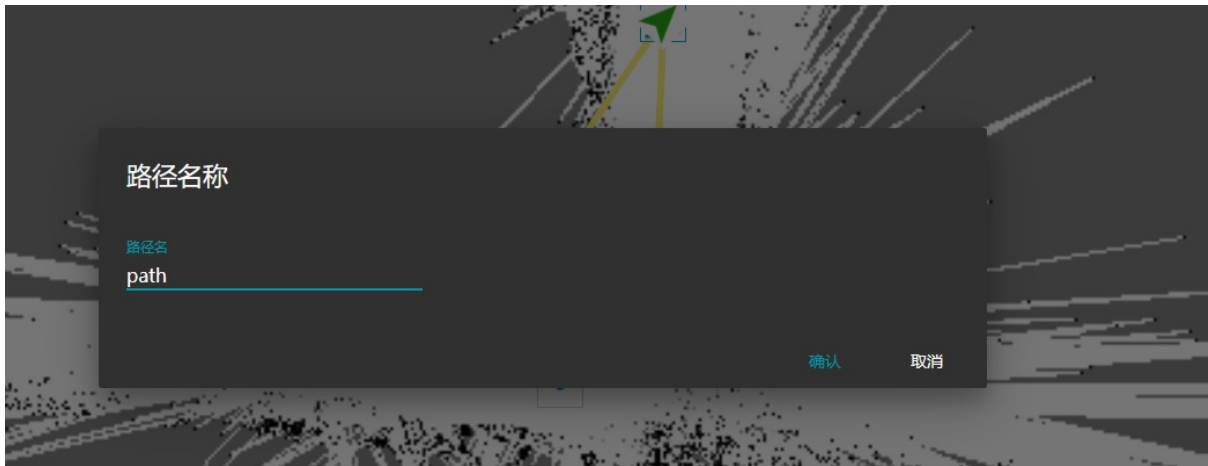
1.3.1. 保存路网信息

地图绘制完成后点击左上角的路径文件菜单。



根据提示进行操作即可保存。

这里我们保存成path



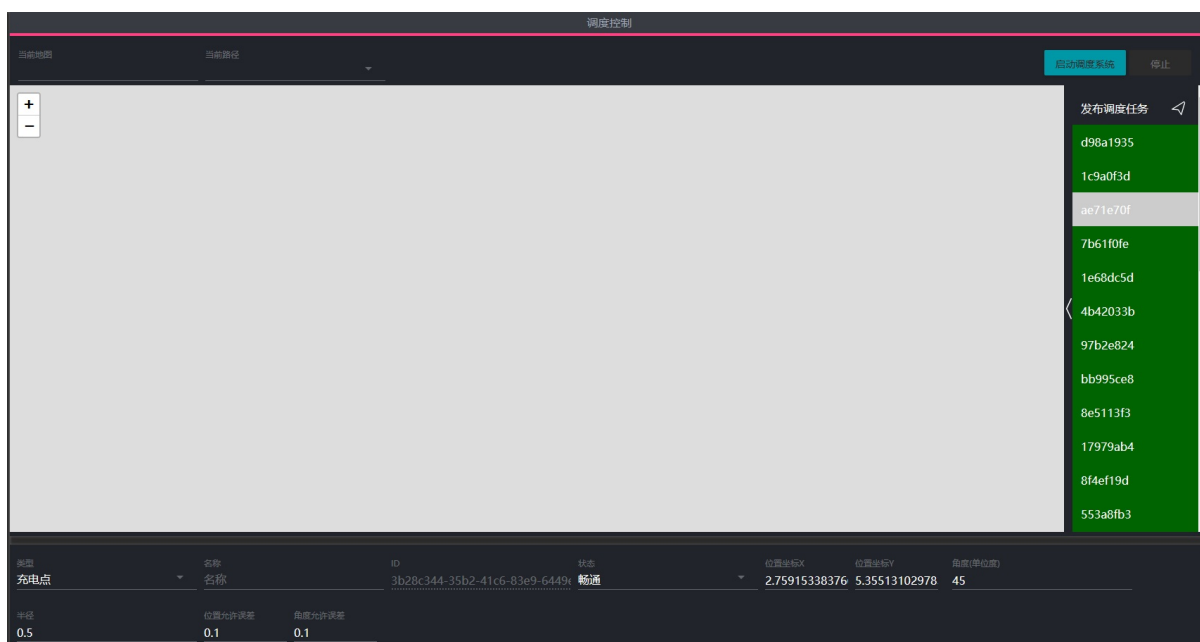
至此路网信息绘制完成。

- 1. 调度系统控制
 - 1.1. 调度系统控制界面介绍
 - 1.2. 调度系统启动与关闭
 - 1.2.1. 开启调度系统
 - 1.2.2. 关闭调度系统

1. 调度系统控制

1.1. 调度系统控制界面介绍

调度空控制主界面如下图所示



其中最上部为选择地图和导航路径的区域。右侧按钮可以控制启动和关闭导航系统。

中间大片空白区域是地图显示区域。可以试试显示当前调度系统中地图和各个机器人状态。

右侧边栏显示当前调度任务状态。

底部面板显示选择的对象的属性。同时可以用来发布新的调度任务

1.2. 调度系统启动与关闭

1.2.1. 开启调度系统

在最上方的选择地图和路径选择菜单中选择目标导航地图和路径。

此时中间地图区域会显示当前的地图和路网信息。

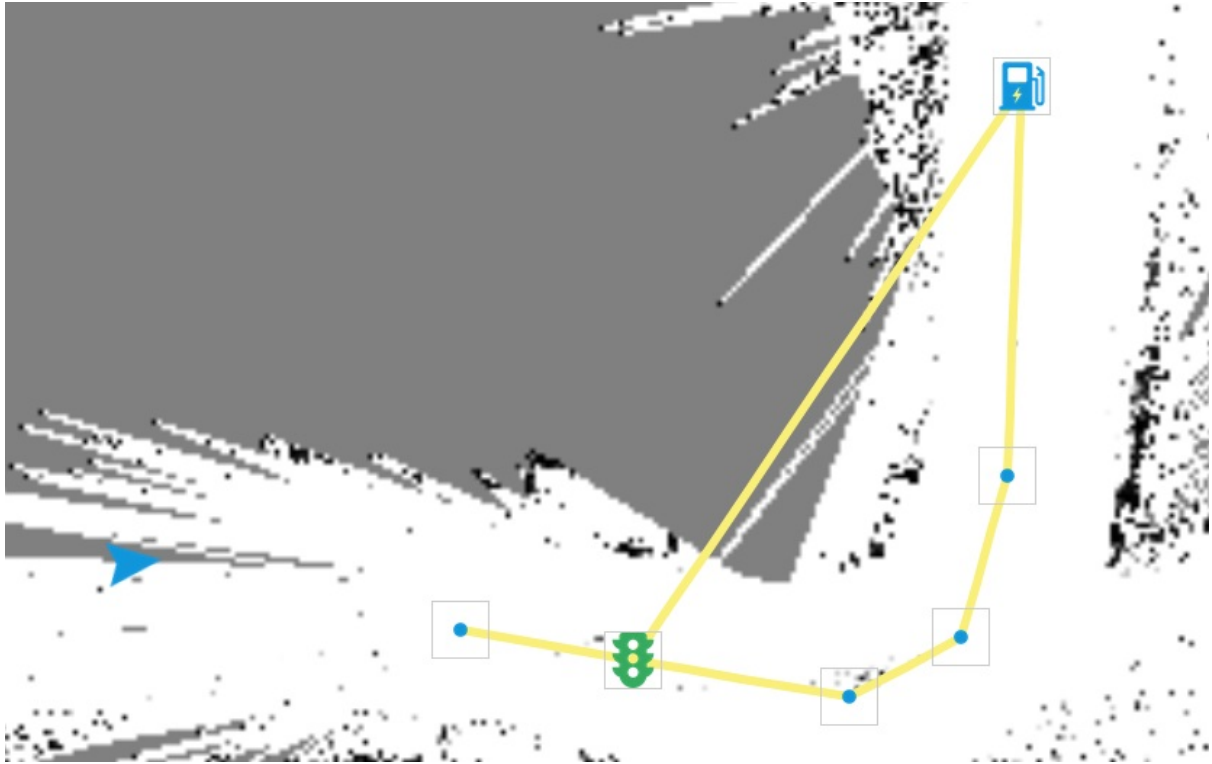


点击右侧启动调度系统按钮，此时会弹出调度系统自检信息。根据提示查看各机器人状态



其中显示准备就绪的机器人会在启动调度系统之后自动进入工作状态。之后发布调度任务后机器人可以正常执行任务。未准备就绪的机器人调度系统将不会对其分配任务。

在点击确认开启导航系统之后可以发现地图中出现蓝色箭头。此箭头代表机器人当前位置。



1.2.2. 关闭调度系统

点击右上角停止按钮即可关闭调度系统

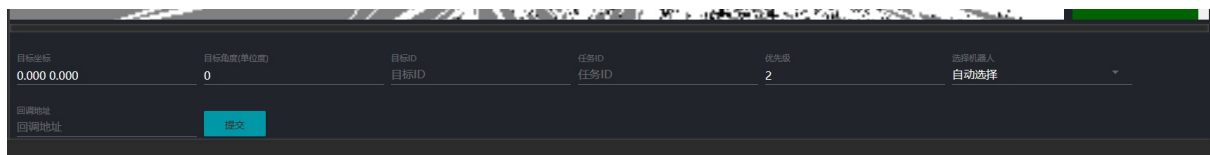
- 1. 调度任务管理
 - 1.1. 发布调度任务
 - 1.2. 取消调度任务
 - 1.3. 暂停和继续调度任务
 - 1.4. 返回待机
 - 1.5. 返回充电
 - 1.6. 配合GalileoAPI使用

1. 调度任务管理

一个调度任务就是控制某个机器人移动到某个位置然后完成某动作。我们可以通过系统界面方便的发布调度任务。

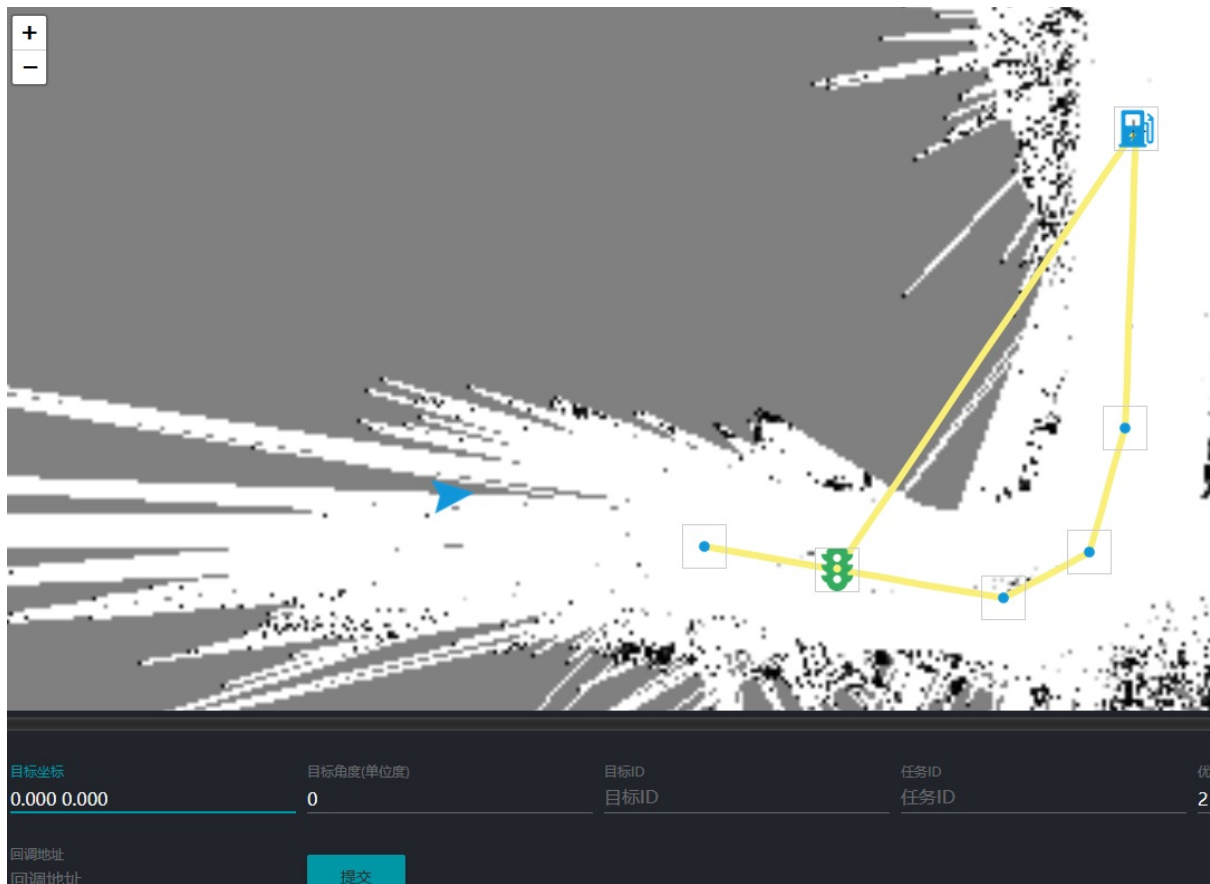
1.1. 发布调度任务

在启动调度系统之后，点击右侧的发布调度任务按钮。此时底部面板会变成调度任务发布面板。

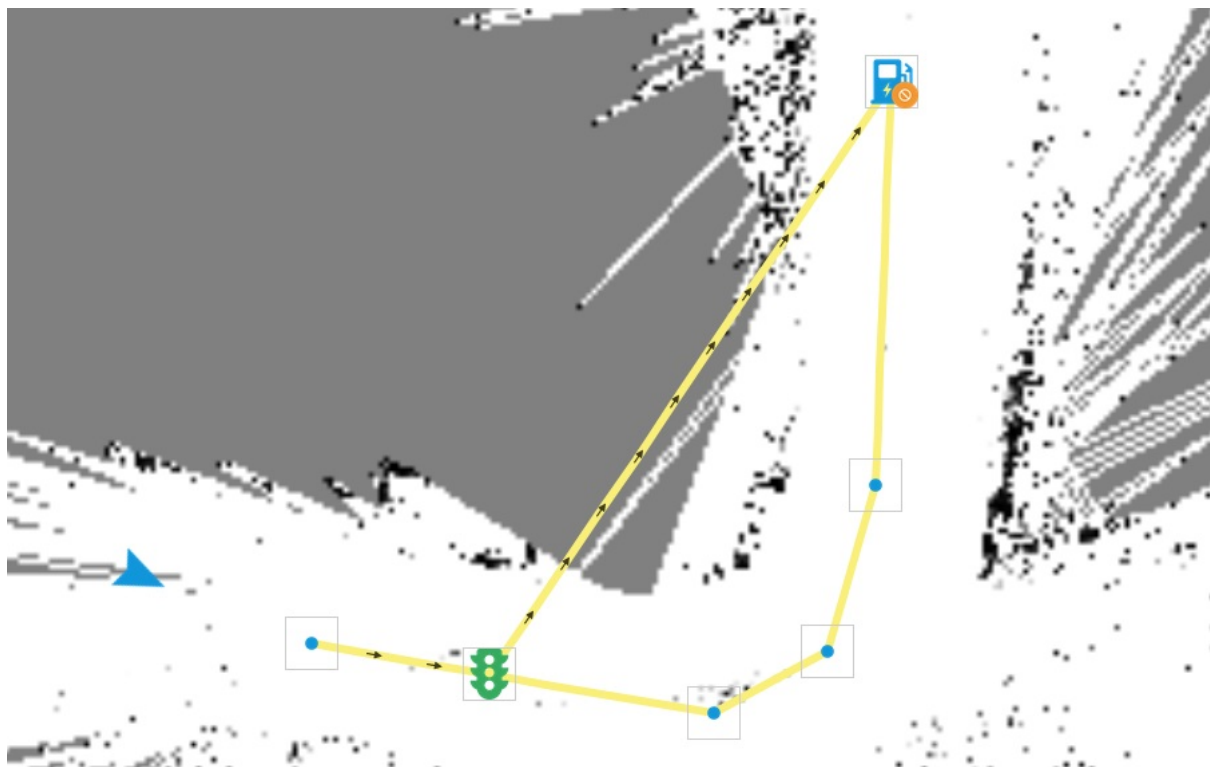


在设置好对应的任务信息之后，点击提交按钮。调度任务就会被添加到系统的队列之中。

比如现在我们要机器人移动到最上方的充电桩位置。首先点击底部面板最左边的目标坐标文本框，然后在地图上点击充电桩所在位置。



点击后点击提交按钮，如果一切正常，现在机器人就会自动向目标点移动了。



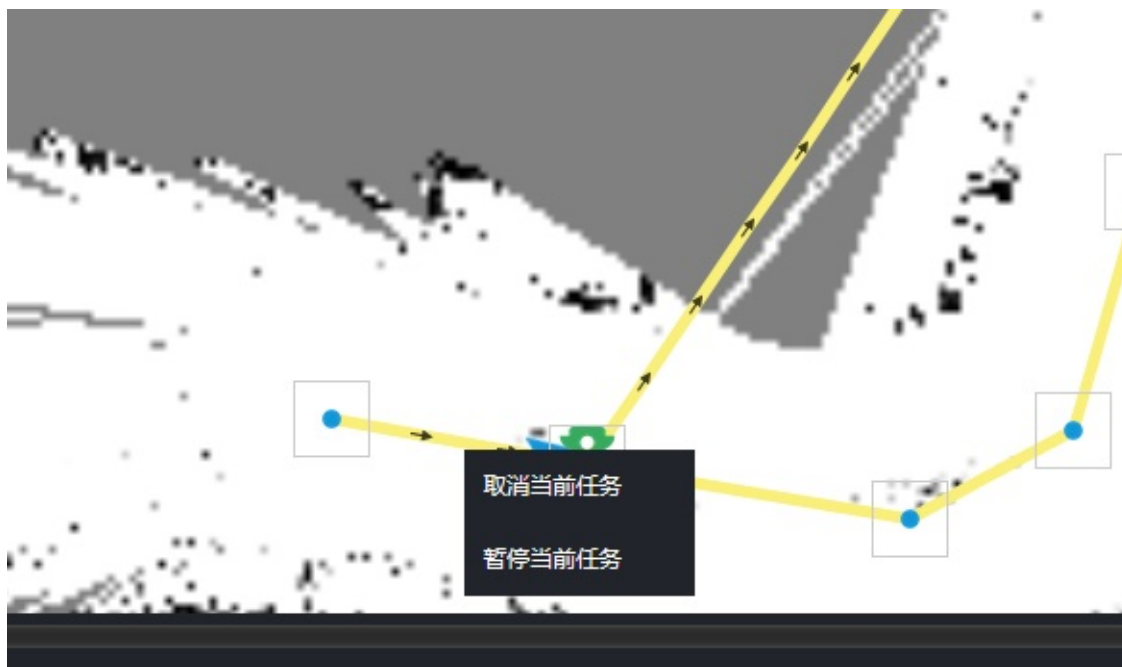
其中带箭头的路线即为调度系统为机器人分配的路线。

1.2. 取消调度任务

在机器人执行调度任务的时候，右侧的调度状态栏会显示当前执行的任務。点击其中的取消按钮即可取消任务。



如果需要取消机器人当前执行的任務，也可以右键点击地图上的机器人图标。在弹出的菜单中选择取消当前任务



1.3. 暂停和继续调度任务

暂停和继续任务的操作方式和取消任务类似，也是点击右侧的调度任务状态按钮。当任务处于不同的状态时会显示出不同的颜色。

1.4. 返回待机

当机器人处于空闲状态时，可以右键点击机器人图标在弹出的菜单中选择返回待机选项



1.5. 返回充电

同样如果需要机器人返回充电，也可以右键点击机器人图标，选择菜单中的返回充电按钮。

1.6. 配合GalileoAPI使用

在调度系统开启后，对于已经激活的机器人，可以通过伽利略导航客户端开启导航。客户端选择地图和路径时选择和调度服务器一致的地图路径。剩下的操作和单台使用完全一样。

机器人在开启导航后，如果检测到调度服务正在运行会有语音提示启用调度系统。此时即使通过伽利略客户端发布导航任务也可以实现多台避让路径规划。

如果机器人当前导航地图或路径和调度服务器不一样。机器人的坐标仍然会在调度地图上显示，但是会以红色图标显示。

- 1. 快速开始
 - 1.1. 启动调度系统
 - 1.2. 关闭调度系统
 - 1.3. 发布调度任务
 - 1.4. 获取调度任务状态
 - 1.5. 暂停, 继续, 取消调度任务
 - 1.6. 返回充电
 - 1.7. 返回待机

1. 快速开始

假设你已经成功创建地图并绘制完成路网信息。下面的内容可以帮助你快速开始使用API控制调度系统。

所有API的URL都以 `/api/v1` 开头下面的表述中将省略此前缀。调度服务器默认端口号为24958

1.1. 启动调度系统

不建议用API进行此操作。最好调度系统一直保持运行状态。URL: `/start`

请求方式: GET

说明: 启动调度系统, 并使用对应的地图和路线数据。如果机器人没有采用当前设置的数据则自动同步和切换机器人地图数据, 并启动机器人至目标地图的导航状态。

请求参数:

| 参数 | 类型 | 说明 |
|---------|--------|-----------------|
| map_id | string | 调度系统所采用的地图id |
| path_id | string | 调度系统所采用的path_id |

返回参数:

| 参数 | 类型 | 说明 |
|--------|--------|---------------|
| layout | object | 成功返回的layout数据 |

例子

```
import requests
```

```
requests.get("http://192.168.0.128:24958/api/v1/start?map_id=7ac96843-d696-40ba-9807-0d4036d6824f&path_id=9613ce18-5fb8-4eab-8d37-2bb2724b790a")
```

其中一长串字符串为启动的目标地图和路径的ID，具体的地图和路径ID可以通过 `http://192.168.0.128:24958/api/v1/layout/map` 这个URL获取。

1.2. 关闭调度系统

不建议用API进行此操作。最好调度系统一直保持运行状态。

URL: /stop

请求方式: GET

说明: 停止调度系统，并停止当前所有任务，关闭机器人导航状态

请求参数: 无

返回参数:

| 参数 | 类型 | 说明 |
|--------|--------|--------|
| status | string | 关闭是否成功 |

例子

```
import requests
requests.get("http://192.168.0.128:24958/api/v1/stop")
```

1.3. 发布调度任务

URL: /schedule 请求方式: POST 请求参数:

| 参数 | 类型 | 说明 |
|-------------|----------------------|--|
| location_id | string | 机器人移动目标点ID，location_id和location参数只用设置其中一个。充电和待机任务不用设置。 |
| location | Object, 包含 x,y,theta | 机器人移动目标点坐标，location_id和location参数只用设置其中一个。充电和待机任务不用设置。 |
| task_id | string | 机器人到达目标点后需要执行的任务id |
| | | 任务优先级。默认优先级如下回归待机点任务 优先级 0, 充 |

| | | |
|--------------|--------|--|
| priority | int | 电任务且当前机器人电量较高 优先级1, 一般移动任务 优先级2, 用户交互任务 优先级3, 低电量充电任务 优先级4 |
| robot_id | string | 可选参数, 当指定此参数时, 只有指定的机器人能够接此任务 |
| callback_url | string | 任务状态回调, 当任务状态发生变化时服务会对对应url发送post请求, 内容为schedule状态数据 |

返回参数:

返回参数: schedule json 数据对象

调用例子

```
import requests
# 移动到id为3b28c344-35b2-41c6-83e9-6449ebe886a0的点
requests.post("http://192.168.0.128:24958/api/v1/schedule", json={
    "location_id": "3b28c344-35b2-41c6-83e9-6449ebe886a0",
    "priority": 2
})
# 移动到坐标为(2.7, 5.3)的目标点, 同时角度为45°
requests.post("http://192.168.0.128:24958/api/v1/schedule", json={
    "location": {
        "x": 2.7,
        "y": 5.3,
        "theta": 45
    },
    "priority": 2
})
```

具体的点的ID和坐标值可以从调度客户端里面获取。可以手动发一个调度任务, 然后查看目标点坐标。

1.4. 获取调度任务状态

URL: /schedule

请求方式: GET

请求参数

| 参数 | 类型 | 说明 |
|----|--------|--------|
| id | string | 调度任务ID |

返回参数: schedule json 数据对象

请求例子

```
import requests
requests.get("http://192.168.0.128:24958/api/v1/schedule?id=56d8e596-d
b41-4bf0-b9a5-03a7e6a1dd1d")
# 返回参数例子

"""
{
  "create_time": 1573043104080,
  "start_time": 0,
  "end_time": 0,
  "id": "56d8e596-db41-4bf0-b9a5-03a7e6a1dd1d",
  "destination": {
    "x": 2.7591533837666145,
    "y": 5.355131029783591,
    "theta": 45
  },
  "destination_id": "3b28c344-35b2-41c6-83e9-6449ebe886a0",
  "priority": 2,
  "callback_url": "",
  "robot": null,
  "state": "CANCELLED",
  "task": null,
  "result": ""
}
"""
```

1.5. 暂停, 继续, 取消调度任务

URL:

取消任务 /schedule/cancel

暂停任务 /schedule/pause

继续任务 /schedule/resume

请求参数

| 参数 | 类型 | 说明 |
|----|--------|-----------|
| id | string | 目标调度任务的ID |

返回参数

目标调度任务内容

例子

```
import requests
# 暂停任务
requests.get("http://192.168.0.128:24958/api/v1/schedule/pause?id=56d8e596-db41-4bf0-b9a5-03a7e6a1dd1d")
# 继续任务
requests.get("http://192.168.0.128:24958/api/v1/schedule/resume?id=56d8e596-db41-4bf0-b9a5-03a7e6a1dd1d")
# 取消任务
requests.get("http://192.168.0.128:24958/api/v1/schedule/cancel?id=56d8e596-db41-4bf0-b9a5-03a7e6a1dd1d")
```

其中的调度任务ID即为创建调度任务时返回的调度ID

1.6. 返回充电

URL: /schedule/go_charge

请求方式

请求方式: GET

说明: 当机器人收到返回充电请求时, 机器人会自动选择一个最近的空闲充电桩进行充电。根据电量的高低, 充电任务可能会抢占其他低优先级的任务。

请求参数

| 参数 | 类型 | 说明 |
|----|--------|------------|
| id | string | 需要返回充电的机器人 |

返回参数

返回充电调度任务数据

例子

```
import requests
requests.get("http://192.168.0.128:24958/api/v1/schedule/go_charge?id=01E1FCAEC7F4A53451E58261EB3DE9A7BDC8C70E367C7102E081F1A65F6970600896D0")
```

```
7A95EB")
```

```
""
```

```
返回数据例子
```

```
{
  "create_time": 1573045014873,
  "start_time": 0,
  "end_time": 0,
  "id": "8288c2c4-b2e5-4661-b38a-a73e88455c0a",
  "destination": {
    "x": -2.951997437470733,
    "y": -0.1631757377496385,
    "theta": 90
  },
  "destination_id": "365e8060-76b2-416b-9eb5-7ad537c5aa1a",
  "priority": 0,
  "callback_url": "",
  "robot": {
    "id": "01E1FCAEC7F4A53451E58261EB3DE9A7BDC8C70E367C7102E081F1A65F6970600896D07A95EB",
    "mac": "00:e0:4c:68:00:dc",
    "ip": "192.168.0.158",
    "port": 3546,
    "version": "4.0.0",
    "galileo_status": {
      "mapStatus": 0,
      "controlSpeedX": 0.0,
      "currentSpeedX": 0.0,
      "targetNumID": -1,
      "power": 12.0,
      "navStatus": 1,
      "loopStatus": 0,
      "gbaStatus": 0,
      "targetDistance": -1.0,
      "controlSpeedTheta": 0.0,
      "currentSpeedTheta": 0.0,
      "header": {
        "stamp": {
          "secs": 1573045014,
          "nsecs": 810050964
        },
        "frame_id": "map",
        "seq": 1039738
      },
    },
    "busyStatus": 0,
    "chargeStatus": 0,
```



```

        "currentAngle": 0.9793111085891724,
        "currentPosX": 1.5362364053726196,
        "currentPosY": 3.550684690475464,
        "gcStatus": 0,
        "angleGoalStatus": 1,
        "visualStatus": 0,
        "targetStatus": 0
    },
    "is_enabled": true,
    "is_online": true,
    "server_url": "http://192.168.0.158:3546/api/v1/",
    "current_schedule": null,
    "last_update_time": 1573045014864,
    "name": ""
},
"state": "EXECUTING",
"task": null,
"result": ""
}
"""

```

机器人的电压信息和ID信息可以通过 `/api/v1/robot/status` 接口获取

1.7. 返回待机

URL: `/schedule/go_rest`

请求方式: GET

说明: 机器人收到返回待机的任务请求后, 机器人会自动选择离自己最近的一个空闲待机点返回待机。

请求参数

| 参数 | 类型 | 说明 |
|----|--------|------------|
| id | string | 需要返回待机的机器人 |

返回待机调度任务数据

例子

```

import requests
requests.get("http://192.168.0.128:24958/api/v1/schedule/go_rest?id=01E1FCAEC7F4A53451E58261EB3DE9A7BDC8C70E367C7102E081F1A65F6970600896D07A

```

```
95EB")

"""
返回数据例子
{
  "create_time": 1573045014873,
  "start_time": 0,
  "end_time": 0,
  "id": "8288c2c4-b2e5-4661-b38a-a73e88455c0a",
  "destination": {
    "x": -2.951997437470733,
    "y": -0.1631757377496385,
    "theta": 90
  },
  "destination_id": "365e8060-76b2-416b-9eb5-7ad537c5aa1a",
  "priority": 0,
  "callback_url": "",
  "robot": {
    "id": "01E1FCAEC7F4A53451E58261EB3DE9A7BDC8C70E367C7102E081F1A
65F6970600896D07A95EB",
    "mac": "00:e0:4c:68:00:dc",
    "ip": "192.168.0.158",
    "port": 3546,
    "version": "4.0.0",
    "galileo_status": {
      "mapStatus": 0,
      "controlSpeedX": 0.0,
      "currentSpeedX": 0.0,
      "targetNumID": -1,
      "power": 12.0,
      "navStatus": 1,
      "loopStatus": 0,
      "gbaStatus": 0,
      "targetDistance": -1.0,
      "controlSpeedTheta": 0.0,
      "currentSpeedTheta": 0.0,
      "header": {
        "stamp": {
          "secs": 1573045014,
          "nsecs": 810050964
        },
        "frame_id": "map",
        "seq": 1039738
      },
      "busyStatus": 0,
      "chargeStatus": 0,
```

```
        "currentAngle": 0.9793111085891724,  
        "currentPosX": 1.5362364053726196,  
        "currentPosY": 3.550684690475464,  
        "gcStatus": 0,  
        "angleGoalStatus": 1,  
        "visualStatus": 0,  
        "targetStatus": 0  
    },  
    "is_enabled": true,  
    "is_online": true,  
    "server_url": "http://192.168.0.158:3546/api/v1/",  
    "current_schedule": null,  
    "last_update_time": 1573045014864,  
    "name": ""  
},  
"state": "EXECUTING",  
"task": null,  
"result": ""  
}  
"""
```

- 1. API文档
 - 1.1. 调度系统整体工作流程
 - 1.2. 数据结构
 - 1.2.1. 地图数据结构
 - 1.2.2. 调度任务状态
 - 1.3. API 说明
 - 1.3.1. 调度中心控制接口
 - 1.3.2. 调度任务相关
 - 1.3.3. 地图状态更新相关
 - 1.3.4. 机器人状态相关
 - 1.3.5. Action 和 Task 相关 API
 - 1.3.6. 证书相关

1. API文档

1.1. 调度系统整体工作流程

用户向调度系统发起任务请求。任务包含移动机器人到某位置并完成某操作。调度系统根据当前机器人资源分配任务至机器人 机器人根据任务计算出所需占用的道路，并将道路占用更新至调度系统

1.2. 数据结构

1.2.1. 地图数据结构

```
{
  points : [
    {
      type: REST_POINT, // 可选类型 NORMAL_POINT, CHARGE_POINT, RE
      ST_POINT, TRAFFIC_POINT, TARGET_POINT
      location: {
        x: 0.0, // 机器人坐标X, 单位为米
        y: 0.0, // 机器人坐标Y, 单位为米
        theta: 0.0, // 机器人坐标角度, 单位为度
      },
      name: "", // 点名称, 可以为空
      id: "xxx-xxx-xxx", // 随机生成的字符串, 点的唯一标识
      status: FREE, // 当前点状态, 可选值为 FREE, BLOCK
      radius: 0.5, // 点半径大小
    }
  ]
}
```

```
        tolerance: {
            xy: 0.1, // 坐标容差
            theta: 0.1 // 角度容差
        }
    },
    ways: [
        {
            id: "xxx-xxx-xxx", // 随机生成的字符串, 线段的唯一标识
            points: ["xxx-xxx-xxx", "xxx-xxx-xxx"], // 线段起点和终点id
            status: "FORWARD", // 当前路线状态, 可选值为 FORWARD, BACKWARD
            , FREE, BLOCK
        }
    ],
}
```

1.2.2. 调度任务状态

ERROR 任务出错

DISPATCHING 任务分配中, 尚未分配到机器人

EXECUTING 任务执行中

PAUSED 任务暂停

CANCELLED 任务取消

COMPLETE 任务完成

1.3. API 说明

1.3.1. 调度中心控制接口

启动调度系统

URL: /start

请求方式: GET

说明: 启动调度系统, 并使用对应的地图和路线数据。如果机器人没有采用当前设置的数据则自动同步和切换机器人地图数据, 并启动机器人至目标地图的导航状态。

请求参数:

| 参数 | 类型 | 说明 |
|---------|--------|-----------------|
| map_id | string | 调度系统所采用的地图id |
| path_id | string | 调度系统所采用的path_id |

返回参数:

| 参数 | 类型 | 说明 |
|--------|--------|-----------------|
| layout | Object | 启动成功后返回layout数据 |

示例layout数据

```
{
  "status": "ok",
  "layout": {
    "name": "path5",
    "points": [
      {
        "type": "TRAFFIC_POINT",
        "location": {
          "x": -7.0625,
          "y": -15.5625,
          "theta": 0
        },
        "name": "",
        "id": "fb9b5da4-d3f8-46af-9f06-6740d18b1350",
        "status": "FREE",
        "radius": 0.5,
        "tolerance": {
          "xy": 0.1,
          "theta": 0.1
        }
      },
      ...
    ],
    "ways": [
      {
        "id": "b4a0cdaa-5f52-4191-9340-36d19a1cb0a6",
        "points": [
          "29d8aea0-ea06-4b61-9b73-cea4ee6833a5",
          "d8db564c-dede-4bfc-8456-a7dfee9e66bb"
        ],
        "status": "FREE",
        "robots": []
      }
    ]
  }
}
```

```
    ...
  ]
},
"map": {
  "_id": {
    "$oid": "5dd34397869f0667d15799e1"
  },
  "id": "04f777b8-ff9d-4303-87ac-334dab2e0ffe",
  "name": "largemap",
  "create_time": 1574126487670,
  "origin": {
    "x": -29.764,
    "y": -33.38139
  },
  "image_width": 1366,
  "image_height": 778,
  "resolution": 0.05,
  "image_png": "/media/map/map_ET4FjhJ.png",
  "image_pgm": "/media/map/map_ET4FjhJ.pgm",
  "keyframedb_bson": "/media/map/keyframedb_hHK57gx.bson",
  "keyframes_bson": "/media/map/keyframes_FnNLite.bson",
  "map_bson": "/media/map/map_T2J2lrF.bson",
  "mappoints_bson": "/media/map/mappoints_zFg5qGm.bson",
  "robot_trajectory": "/media/map/RobotTrajectory_4XjHdHI.txt",
  "map_yaml": "/media/map/largemap.yaml",
  "md5sum": "dd0c7a1c2da0194e66bbe8aee3b5d891",
  "paths": [
    {
      "_id": {
        "$oid": "5dd3442c869f0667d15799e2"
      },
      "map_id": "04f777b8-ff9d-4303-87ac-334dab2e0ffe",
      "create_time": 1574126636820,
      "layout": {
        "name": "path4",
        "points": [
          {
            "type": "TRAFFIC_POINT",
            "location": {
              "x": -7.0625,
              "y": -15.5625,
              "theta": 0
            },
            "name": "",
            "id": "fb9b5da4-d3f8-46af-9f06-6740d18b135"
          }
        ]
      }
    }
  ]
},
0"
```

```
        "status": "FREE",
        "radius": 0.5,
        "tolerance": {
            "xy": 0.1,
            "theta": 0.1
        }
    },
    ...
],
"ways": [
    {
        "id": "10d1d0b3-23d3-4b1d-a16e-0e2f86be6a4
d",
        "points": [
            "fb9b5da4-d3f8-46af-9f06-6740d18b1350",
            "e84adf2b-4c56-419a-beb0-945b60102772"
        ],
        "status": "FREE"
    },
    ...
]
},
{id": "598c5ed2-cb2f-40de-b806-5c94b8200ccf",
"name": "path4"
},
{
    "_id": {
        "$oid": "5ddc8b3d65c125fbd33166d9"
    },
    "map_id": "04f777b8-ff9d-4303-87ac-334dab2e0ffe",
    "create_time": 1574734653138,
    "layout": {
        "name": "path5",
        "points": [
            {
                "type": "TRAFFIC_POINT",
                "location": {
                    "x": -7.0625,
                    "y": -15.5625,
                    "theta": 0
                },
                "name": "",
                "id": "fb9b5da4-d3f8-46af-9f06-6740d18b135
0",
                "status": "FREE",
```



```
        "radius": 0.5,
        "tolerance": {
            "xy": 0.1,
            "theta": 0.1
        }
    },
    ...
],
"ways": [
    {
        "id": "10d1d0b3-23d3-4b1d-a16e-0e2f86be6a4
d",
        "points": [
            "fb9b5da4-d3f8-46af-9f06-6740d18b1350",
            "e84adf2b-4c56-419a-beb0-945b60102772"
        ],
        "status": "FREE"
    },
    ...
]
},
{id": "503ab888-ae67-4981-8840-25f48feeeb9f",
"name": "path5"
}
]
},
"path": {
    "_id": {
        "$oid": "5ddc8b3d65c125fbd33166d9"
    },
    "map_id": "04f777b8-ff9d-4303-87ac-334dab2e0ffe",
    "create_time": 1574734653138,
    "layout": {
        "name": "path5",
        "points": [
            {
                "type": "TRAFFIC_POINT",
                "location": {
                    "x": -7.0625,
                    "y": -15.5625,
                    "theta": 0
                },
                "name": "",
                "id": "fb9b5da4-d3f8-46af-9f06-6740d18b1350",
                "status": "FREE",
```

```
        "radius": 0.5,
        "tolerance": {
            "xy": 0.1,
            "theta": 0.1
        }
    },
    ...
],
"ways": [
    {
        "id": "10d1d0b3-23d3-4b1d-a16e-0e2f86be6a4d",
        "points": [
            "fb9b5da4-d3f8-46af-9f06-6740d18b1350",
            "e84adf2b-4c56-419a-beb0-945b60102772"
        ],
        "status": "FREE",
        "robots": []
    },
    ...
]
},
"id": "503ab888-ae67-4981-8840-25f48feeeb9f",
"name": "path5"
},
"lock_id": "",
"robot_id": ""
}
```

停止调度系统

URL: /stop

请求方式: GET

说明: 停止调度系统, 并停止当前所有任务, 关闭机器人导航状态

请求参数: 无

返回参数:

| 参数 | 类型 | 说明 |
|--------|--------|--------|
| status | string | 关闭是否成功 |

1.3.2. 调度任务相关

发起任务

URL: /schedule

请求方式: POST

请求参数:

| 参数 | 类型 | 说明 |
|--------------|-----------------------|---|
| location_id | string | 机器人移动目标点ID, location_id和location参数只用设置其中一个。充电和待机任务不用设置。 |
| location | Object包含 x,y,theta | 机器人移动目标点坐标, location_id和location参数只用设置其中一个。充电和待机任务不用设置。 |
| task_id | string | 机器人到达目标点后需要执行的任务id |
| priority | int | 任务优先级。默认优先级如下 回归待机点任务 优先级 0 充电任务且当前机器人电量较高 优先级1 一般移动任务 优先级2 用户交互任务 优先级3 低电量充电任务 优先级4 |
| robot_id | string | 可选参数, 当指定此参数时, 只有指定的机器人能够接收此任务 |
| callback_url | string | 任务状态回调, 当任务状态发生变化时服务会对对应url发送post请求, 内容为schedule状态数据 |

返回参数: schedule json 数据对象

示例返回数据:

```
{
  "create_time": 1576127957717,
  "start_time": 0,
  "end_time": 0,
  "id": "6f32ceeb-1a85-4ef6-9608-246098a4ac70",
  "destination": {
    "x": -5.29884054000141,
    "y": -2.7553970808007335,
    "theta": 0
  },
  "destination_id": "",
  "priority": 2,
  "callback_url": "",
  "robot": {
    "id": "16000CD479EEFF2B29B8AC54E96F29802E50E724E95834F5935FE9"
  }
}
```

```
99BE8DB2816E01A482211",
  "mac": "00:e0:4c:ce:83:87",
  "ip": "192.168.0.196",
  "port": 3546,
  "version": "4.0.0",
  "galileo_status": {
    "mapStatus": 0,
    "controlSpeedX": 0.0,
    "currentSpeedX": 0.0,
    "targetNumID": -1,
    "power": 38.76279067993164,
    "navStatus": 1,
    "loopStatus": 0,
    "gbaStatus": 0,
    "targetDistance": -1.0,
    "controlSpeedTheta": 0.0,
    "currentSpeedTheta": 0.0006672566523775458,
    "header": {
      "stamp": {
        "secs": 1576128007,
        "nsecs": 241101980
      },
      "frame_id": "map",
      "seq": 943
    },
    "busyStatus": 0,
    "chargeStatus": 0,
    "currentAngle": 2.859792709350586,
    "currentPosX": -4.2341084480285645,
    "currentPosY": 0.9780586957931519,
    "gcStatus": 0,
    "angleGoalStatus": 1,
    "visualStatus": 1,
    "targetStatus": 0
  },
  "is_enabled": true,
  "is_online": true,
  "server_url": "http://192.168.0.196:3546/api/v1/",
  "current_schedule": null,
  "last_update_time": 1576127957812,
  "name": ""
},
"state": "EXECUTING",
"task": null,
"result": ""
}
```

取消任务

URL: /schedule/cancel

请求方式: GET

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|--------|
| id | string | 调度任务id |

返回: schedule json 数据对象

暂停任务

URL: /schedule/pause

请求方式: GET

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|--------|
| id | string | 调度任务id |

返回: schedule json 数据对象

继续任务

URL: /schedule/resume

请求方式: GET

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|--------|
| id | string | 调度任务id |

返回: schedule json 数据对象

返回待机

URL: /schedule/go_rest

请求方式: GET

说明： 机器人从空闲的待机点中选择直线距离最近的点待机

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|------------|
| id | string | 返回待机的机器人ID |

返回参数： 返回待机的调度任务信息

返回充电

URL: /schedule/go_charge

请求方式: GET

说明： 机器人从空闲的充电点钟选择直线距离最近的点充电

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|-----------|
| id | string | 返回充电机器人ID |

返回参数： 返回充电的调度任务信息

1.3.3. 地图状态更新相关

锁定地图

URL: /layout/lock

请求方式: GET

请求参数

| 参数 | 类型 | 说明 |
|----------|--------|-------|
| robot_id | string | 机器人id |

返回参数

| 参数 | 类型 | 说明 |
|---------|--------|----------|
| lock_id | string | 获取的锁的id |
| layout | object | 当前地图状态数据 |

返回示例数据:

```
{
  "status": "ok",
  "lock_id": "3d1e6049-750d-483e-aa32-dd7b937373ec",
  "layout": {
    "name": "path5",
    "points": [
      {
        "type": "TRAFFIC_POINT",
        "location": {
          "x": -7.0625,
          "y": -15.5625,
          "theta": 0
        },
        "name": "",
        "id": "fb9b5da4-d3f8-46af-9f06-6740d18b1350",
        "status": "FREE",
        "radius": 0.5,
        "tolerance": {
          "xy": 0.1,
          "theta": 0.1
        }
      },
      ...
    ],
    "ways": [
      {
        "id": "10d1d0b3-23d3-4b1d-a16e-0e2f86be6a4d",
        "points": [
          "fb9b5da4-d3f8-46af-9f06-6740d18b1350",
          "e84adf2b-4c56-419a-beb0-945b60102772"
        ],
        "status": "FREE",
        "robots": []
      },
      ...
    ]
  }
}
```

解锁地图

URL: `/layout/unlock`

请求参数: GET

| 参数 | 类型 | 说明 |
|----------|--------|------------------------------------|
| lock_id | string | 地图锁id, 如果id不能对应当前地图锁id, 将无法解锁 |
| robot_id | string | 机器人id, 如果id无法对应当前获得锁的机器人id, 将无法解锁。 |

返回参数

| 参数 | 类型 | 说明 |
|--------|--------|----------|
| layout | object | 当前地图状态数据 |
| status | string | 是否解锁成功 |

更新地图

URL: /layout

请求方式: PUT

请求参数:

| 参数 | 类型 | 说明 |
|----------|--------|---------------------------------|
| robot_id | string | 当前机器人id |
| lock_id | string | 当前机器人获取到的锁id |
| layout | object | 更改的地图状态信息。如果有未包含的地图状态, 则认为没有改动。 |
| unlock | bool | 是否自动解锁地图 |

返回参数:

| 参数 | 类型 | 说明 |
|--------|--------|----------|
| layout | object | 当前地图状态数据 |

获取地图

URL: /layout

请求方式: GET

请求参数:

| 参数 | 类型 | 说明 |
|------|------|----------|
| lock | bool | 是否自动锁定地图 |
| | | |

| | | |
|----------|--------|-----------------------|
| robot_id | string | 当开启自动锁定地图时需要此参数，否则不需要 |
|----------|--------|-----------------------|

返回参数:

| 参数 | 类型 | 说明 |
|--------|--------|--------|
| layout | object | 地图状态信息 |

获取地图数据

URL: /layout/map

请求方式: GET

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|-------------------------|
| id | string | 地图id,可以为空。为空则返回当前所有地图数据 |

返回参数

| 参数 | 类型 | 说明 |
|-----------------|----------------|-------------------|
| id | string | 地图id, 唯一的字符串 |
| name | string | 地图名称 |
| origin | object, 有x,y属性 | 地图左上角的坐标 |
| resolution | float | 1像素对应的实际距离 |
| image | string | 地图图片url, 图片格式为png |
| keyframedb.bson | string | 地图数据url格式为bson |
| keyframes.bson | string | 地图数据url格式为bson |
| map.bson | string | 地图数据url格式为bson |
| mappoints.bson | string | 地图数据url格式为bson |
| image_pgm | string | 地图图片url,格式为pgm |
| image_yaml | string | 地图说明文件, 格式为yaml |
| image_png | string | 地图图片url,格式为png |

上传地图数据

URL: /layout/map

请求方式: POST

请求参数:

| 参数 | 类型 | 说明 |
|-----------------|--------|--|
| name | string | 地图名称 |
| origin_x | float | 地图左上角X坐标 |
| origin_y | float | 地图左上角Y坐标 |
| resolution | float | 1像素对应实际的距离 |
| image_pgm | file | 地图文件数据, 图片格式为pgm |
| keyframedb_bson | file | 地图数据文件, 格式为bson |
| keyframes_bson | file | 地图数据文件, 格式为bson |
| map_bson | file | 地图数据文件, 格式为bson |
| mappoints_bson | file | 地图数据文件,格式为bson |
| force | bool | 遇到重名地图时是否覆盖以前地图。如果为true则覆盖, false则返回错误 |

返回参数

| 参数 | 类型 | 说明 |
|-----------------|----------------|-------------------|
| id | string | 地图id, 唯一的字符串 |
| name | string | 地图名称 |
| origin | object, 有x,y属性 | 地图左上角的坐标 |
| resolution | float | 1像素对应的实际距离 |
| image | string | 地图图片url, 图片格式为png |
| keyframedb.bson | string | 地图数据url格式为bson |
| keyframes.bson | string | 地图数据url格式为bson |
| map.bson | string | 地图数据url格式为bson |
| mappoints.bson | string | 地图数据url格式为bson |
| image_pgm | string | 地图图片url,格式为pgm |
| image_yaml | string | 地图说明文件, 格式为yaml |
| image_png | string | 地图图片url,格式为png |

删除地图数据

URL: `/layout/map`

请求方式: DELETE

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|------|
| id | string | 地图id |

返回参数

| 参数 | 类型 | 说明 |
|--------|--------|----------|
| status | string | 表明是否删除成功 |

获取保存的路径数据

URL: /layout/path

请求方式: GET

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|---------------------------------|
| id | string | Path id, 可选参数。若没有此参数则返回所有path信息 |

返回参数

| 参数 | 类型 | 说明 |
|-------------|--------|------------|
| map_id | string | 所属地图id |
| create_time | string | 创建时间 |
| layout | object | layout数据结构 |
| name | string | 路径名称 |
| id | string | 路径id |

上传路径数据

URL: /layout/path

请求方式: POST

请求参数:

| 参数 | 类型 | 说明 |
|------|--------|------|
| name | string | 路径名称 |

| | | |
|--------|--------|------------|
| map_id | string | 地图id |
| layout | object | layout数据结构 |

返回数据:

| 参数 | 类型 | 说明 |
|-------------|--------|------------|
| map_id | string | 所属地图id |
| create_time | string | 创建时间 |
| layout | object | layout数据结构 |
| name | string | 路径名称 |
| id | string | 路径id |

修改路径数据

URL: /layout/path

请求方式: PUT

请求参数:

| 参数 | 类型 | 说明 |
|--------|--------|-----------------|
| id | string | path id, 唯一的字符串 |
| map_id | string | 所属地图的id |
| layout | object | layout数据结构 |
| name | string | 路径名称 |

返回参数

| 参数 | 类型 | 说明 |
|-------------|--------|------------|
| map_id | string | 所属地图id |
| create_time | string | 创建时间 |
| layout | object | layout数据结构 |
| name | string | 路径名称 |
| id | string | 路径id |

删除路径数据

URL: /layout/path

请求方法: DELETE

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|----------------|
| id | string | path id, 唯一字符串 |

1.3.4. 机器人状态相关

获取机器人状态1

URL: /robot

请求方式: GET

请求参数:

| 参数 | 类型 | 说明 |
|----------|--------|---|
| robot_id | string | 机器人id, 当此参数未指定时返回所有机器人状态。当指定时返回指定的机器人状态 |

返回参数:

| 参数 | 类型 | 说明 |
|--------|--------|----------|
| status | string | 是否成功获取状态 |
| robot | object | 机器人状态信息 |

返回示例信息

```
{
  "status": "ok",
  "robots": [
    {
      "id": "16000CD479EEFF2B29B8AC54E96F29802E50E724E95834F5935FE999BE8DB2816E01A482211",
      "mac": "00:e0:4c:ce:83:87",
      "ip": "192.168.0.196",
      "port": 3546,
      "version": "4.0.0",
      "galileo_status": {
        "mapStatus": 0,
        "controlSpeedX": 0.0,
        "currentSpeedX": -1.4528632164001465e-07,

```

```
    "targetNumID": -1,  
    "power": 38.7115478515625,  
    "navStatus": 1,  
    "loopStatus": 0,  
    "gbaStatus": 0,  
    "targetDistance": -1.0,  
    "controlSpeedTheta": 0.0,  
    "currentSpeedTheta": -0.00039996925625018775,  
    "header": {  
      "stamp": {  
        "secs": 1576128308,  
        "nsecs": 894376993  
      },  
      "frame_id": "map",  
      "seq": 9989  
    },  
    "busyStatus": 0,  
    "chargeStatus": 0,  
    "currentAngle": -0.04512203484773636,  
    "currentPosX": -5.170620441436768,  
    "currentPosY": -2.721562385559082,  
    "gcStatus": 0,  
    "angleGoalStatus": 1,  
    "visualStatus": 2,  
    "targetStatus": 0  
  },  
  "is_enabled": true,  
  "is_online": true,  
  "server_url": "http://192.168.0.196:3546/api/v1/",  
  "current_schedule": null,  
  "last_update_time": 1576128259367,  
  "name": ""  
},  
{  
  "id": "66D23E4AC785F675EB3615D0B1C1CA63AF80B6F6B285751CCC4  
A075B10FF7E2C2BD26B89D75A",  
  "mac": "00:e0:4c:13:4f:21",  
  "ip": "192.168.0.121",  
  "port": 3546,  
  "version": "4.0.0",  
  "galileo_status": null,  
  "is_enabled": false,  
  "is_online": false,  
  "server_url": "http://192.168.0.121:3546/api/v1/",  
  "current_schedule": null,  
  "last_update_time": 1576122896046,
```

```

      "name": ""
    }
  ]
}

```

修改机器人名称

URL: /robot

请求方式: PUT

请求参数:

| 参数 | 类型 | 说明 |
|------|--------|-------|
| name | string | 机器人名称 |
| id | string | 机器人ID |

返回参数: 机器人状态信息

获取机器人状态2

URL: /robot/status

请求方式: GET

请求参数:

| 参数 | 类型 | 说明 |
|----------|--------|---|
| robot_id | string | 机器人id, 当此参数未指定时返回所有机器人状态。当指定时返回指定的机器人状态 |

返回参数

| 参数 | 类型 | 说明 |
|--------|------|---------|
| robots | list | 机器人状态列表 |

上传机器人状态

URL: /robot/status

请求方式: POST

请求参数:

| | | |
|--|--|--|
| | | |
|--|--|--|

| 参数 | 类型 | 说明 |
|--------------|--------|---------|
| robot_id | string | 机器人id |
| robot_status | object | 机器人状态信息 |

返回参数

| 参数 | 类型 | 说明 |
|--------------|--------|---------|
| robot_id | string | 机器人id |
| robot_status | object | 机器人状态信息 |

返回参数

| 参数 | 类型 | 说明 |
|--------------|--------|-------------|
| robot_id | string | 机器人id |
| robot_status | object | 更新后的机器人状态信息 |

使能机器人

URL: `/robot/enable`

请求方式: `PUT`

说明: 使能机器人或禁用机器人。只有使能后的机器人系统才会为其分配任务

| 参数 | 类型 | 说明 |
|------------|--------|---------|
| is_enabled | bool | 是否使能机器人 |
| robot_id | string | 机器人ID |

返回参数

| 参数 | 类型 | 说明 |
|-------|--------|---------|
| robot | object | 机器人状态信息 |

调用某机器人API

URL: `/robot/< ID >/< API_URL >`

说明:

ID为机器人id, API_URL为对应的调用机器人URL。如获取机器人状态信息可以使用 `http://xxx.xxx.xxx.xxx/AABBCCDD/api/v1/syste/status` 进行访问。其中 `xxx.xxx.xxx.xxx`为调度中心ip。AABBCCDD为机器人id。

请求方式: `GET` , `POST` , `PUT` , `DELETE`

请求参数: 和对应机器人API一致

返回参数: 和机器人API返回参数一致

1.3.5. Action 和 Task 相关 API

Action代表机器人执行的一个动作, 比如等待5秒, 或播放一段语音, 或左转30°。当一系列action编排成一组动作的时候就是一个Task。通过action和task相关的API我们可以方便的控制机器人实现我们想要的动作。

支持的Action类型和对应属性

| 类型 | 说明 | 属性 |
|-------------------|--------------------------------------|---|
| callback_action | 回调action, 执行此action时会向指定的url发送http请求 | url:http请求的地址 method:http 请求的方式 data:http请求的数据 |
| sleep_action | 等待动作, 机器人等待对应时间后执行之后的action | wait_time: 机器人等待时间 |
| charge_action | 自动充电动作, 只有在机器人在充电桩附近时才可以执行此动作 | x: 充电桩x坐标 y: 充电桩y坐标 theta: 充电桩角度, 单位为弧度 robot_id: 需要执行此动作的机器人id |
| local_move_action | 局部移动动作。控制机器人小范围移动, 适用于精准对接场景 | robot_id: 需要执行此动作的机器人id distance: 机器人向前移动距离, 当为负值时则机器人后退对应距离 angle:机器人转动对应角度, 单位为弧度 method:定位方式。0使用纯惯性导航定位。1采用激光雷达辅助定位 |

获取Action

URL: `/action`

请求方式: `GET`

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|---|
| id | string | action id, 唯一字符串。id为可选参数, 如果有id则返回目标action信息, 如果没有id则返回所有action信息 |

返回参数: 目标action信息

示例返回数据

```
{
  "type": "charge_action",
  "id": "45affe13-37de-4041-b592-70db27ec4e38",
  "x": -3.799302718853892,
  "y": 0.9755894865773105,
  "theta": 1.48352986419518,
  "robot": {
    "id": "16000CD479EEFF2B29B8AC54E96F29802E50E724E95834F5935FE999BE8DB2816E01A482211",
    "mac": "00:e0:4c:ce:83:87",
    "ip": "192.168.0.196",
    "port": 3546,
    "version": "4.0.0",
    "galileo_status": {
      "mapStatus": 0,
      "controlSpeedX": 0,
      "currentSpeedX": -2.3283064365386963e-7,
      "targetNumID": -1,
      "power": 38.631343841552734,
      "navStatus": 1,
      "loopStatus": 0,
      "gbaStatus": 0,
      "targetDistance": 1.2436540126800537,
      "controlSpeedTheta": 0,
      "currentSpeedTheta": 3.206077963113785e-7,
      "header": {
        "stamp": {
          "secs": 1576128895,
          "nsecs": 762253046
        }
      },
      "frame_id": "map",
      "seq": 27595
    },
    "busyStatus": 0,
  }
}
```

```

        "chargeStatus": 0,
        "currentAngle": 2.205643653869629,
        "currentPosX": -5.04252815246582,
        "currentPosY": 1.0082411766052246,
        "gcStatus": 0,
        "angleGoalStatus": 1,
        "visualStatus": 1,
        "targetStatus": 2
    },
    "is_enabled": true,
    "is_online": true,
    "server_url": "http://192.168.0.196:3546/api/v1/",
    "current_schedule": null,
    "last_update_time": 1576128846285,
    "name": ""
},
"state": "WAITTING",
"result": "",
"progress": 0
}

```

创建Action

URL: /action

请求方式: POST

请求参数:

| 参数 | 类型 | 说明 |
|---------|--------|--|
| type | string | action类型, 目前可选的action类型为 callback_action, sleep_action, charge_action, local_move_action |
| task_id | string | action 所属task的id, 如果没有task_id则会自动创建一个包含目标action的task |
| 其余参数 | | 其余参数根据不同的action类型要求添加 |

返回参数: 新创建的action信息。如果task_id不存在则返回自动创建的task的信息

注意对于自动创建的task, 系统将至作为临时task不会长期保存在数据库中。当调度系统重启之后对应的task和action都会消失

修改Action

URL: /action

请求方式: PUT

请求参数:

| 参数 | 类型 | 说明 |
|-----------------|--------|-----------|
| id | string | action id |
| 其他需要修改的action属性 | | |

返回参数: 修改后的action对象信息

删除Action

URL: /action

请求方式: DELETE

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|------------------|
| id | string | action的id, 唯一字符串 |

查找Task

URL: /task

请求方式: GET

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|---|
| id | string | task id, task唯一字符串。可选参数, 如果没有id则返回最后创建的20个task。如果有id则返回目标task信息 |

返回参数: task 信息

示例返回信息:

```
{
  "id": "50d79436-50c6-43da-9117-d569684f6205",
  "name": "navigation task",
  "loop_flag": false,
  "current_task": null,
```

```
"state": "WAITTING",
"sub_tasks": [
  {
    "type": "charge_action",
    "x": -3.799302718853892,
    "y": 0.9755894865773105,
    "theta": 1.48352986419518,
    "robot": {
      "id": "16000CD479EEEEFF2B29B8AC54E96F29802E50E724E95834F5935FE9
99BE8DB2816E01A482211",
      "mac": "00:e0:4c:ce:83:87",
      "ip": "192.168.0.196",
      "port": 3546,
      "version": "4.0.0",
      "galileo_status": {
        "mapStatus": 0,
        "controlSpeedX": 0,
        "currentSpeedX": -2.0489096641540527e-7,
        "targetNumID": -1,
        "power": 38.68146896362305,
        "navStatus": 1,
        "loopStatus": 0,
        "gbaStatus": 0,
        "targetDistance": 1.0949629545211792,
        "controlSpeedTheta": 0,
        "currentSpeedTheta": -0.0005335922469384968,
        "header": {
          "stamp": {
            "secs": 1576128494,
            "nsecs": 861792087
          },
          "frame_id": "map",
          "seq": 15568
        },
        "busyStatus": 0,
        "chargeStatus": 0,
        "currentAngle": 1.8335497379302979,
        "currentPosX": -4.688186168670654,
        "currentPosY": 0.3361913859844208,
        "gcStatus": 0,
        "angleGoalStatus": 1,
        "visualStatus": 1,
        "targetStatus": 2
      },
      "is_enabled": true,
      "is_online": true,
```

```

    "server_url": "http://192.168.0.196:3546/api/v1/",
    "current_schedule": null,
    "last_update_time": 1576128445446,
    "name": ""
  },
  "state": "WAITTING",
  "result": "",
  "progress": 0
},
{
  "id": "90575557-f993-4146-96d6-3450d84457ba",
  "type": "simple_action",
  "state": "WAITTING",
  "progress": 0
}
],
"progress": 0
}

```

创建Task

URL: /task

请求方式: POST

请求参数

| 参数 | 类型 | 说明 |
|-----------|--------|--------------------------------------|
| name | string | task的名称, 可选参数 |
| loop_flag | bool | 是否自动循环执行,可选参数 |
| sub_tasks | list | 子任务的id列表, 子任务可以是action也可以是task, 可选参数 |

返回参数: 创建后的task信息

修改Task

URL: /task

请求方式: PUT

请求参数

| 参数 | 类型 | 说明 |
|----|--------|----------------------|
| id | string | 目标task id, task唯一字符串 |

| | | |
|-----------|--------|--------------------------------------|
| name | string | task的名称, 可选参数 |
| loop_flag | bool | 是否自动循环执行,可选参数 |
| sub_tasks | list | 子任务的id列表, 子任务可以是action也可以是task, 可选参数 |

返回参数: 修改后的task信息

删除Task

URL: /task

请求方式: DELETE

请求参数

| 参数 | 类型 | 说明 |
|----|--------|----------------------|
| id | string | 目标task id, task唯一字符串 |

返回参数

| 参数 | 类型 | 说明 |
|--------|--------|-------------|
| status | string | task是否被成功删除 |

启动Task

URL: /task/start

请求方式: GET

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|----------------------|
| id | string | task id, task唯一字符串标识 |

返回参数: 启动后的task信息

暂停Task

URL: /task/pause

请求方式: GET

请求参数:

| 参数 | 类型 | 说明 |
|----|----|----|
|----|----|----|

| | | |
|----|--------|----------------------|
| id | string | task id, task唯一字符串标识 |
|----|--------|----------------------|

返回参数: 暂停后的task信息

继续Task

URL: /task/resume

请求方式: GET

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|----------------------|
| id | string | task id, task唯一字符串标识 |

返回参数: 继续后的task信息

停止Task

URL: /task/stop

请求方式: GET

请求参数:

| 参数 | 类型 | 说明 |
|----|--------|----------------------|
| id | string | task id, task唯一字符串标识 |

返回参数: 停止后的task信息

1.3.6. 证书相关

注册证书

URL: /cert/register

请求方式: GET

请求参数

| 参数 | 类型 | 说明 |
|---------|--------|---------|
| cert_id | string | 证书注册序列号 |

返回参数

| | | |
|--|--|--|
| | | |
|--|--|--|

| 参数 | 类型 | 说明 |
|--------|------|--------|
| result | bool | 证书注册结果 |

获取注册后的证书

URL: /cert

请求方式: GET

请求参数: 无

返回参数

| 参数 | 类型 | 说明 |
|-----------|--------|--------|
| cert_data | string | 注册证书内容 |

验证证书

URL: /cert/check

请求方式: GET

请求参数: 无

返回参数:

| 参数 | 类型 | 说明 |
|-----------|------|--------|
| check_res | bool | 证书验证结果 |