

高仙导航模块API使用方法快速入门

- 本文档仅作为快速入门使用, 更详细的API介绍请参考[GS-API说明手册](#)
- 机器人服务器IP地址与端口号: 10.7.5.88:8080
- 每个URL对应GET请求或者POST请求. 均已标注

如何扫描地图

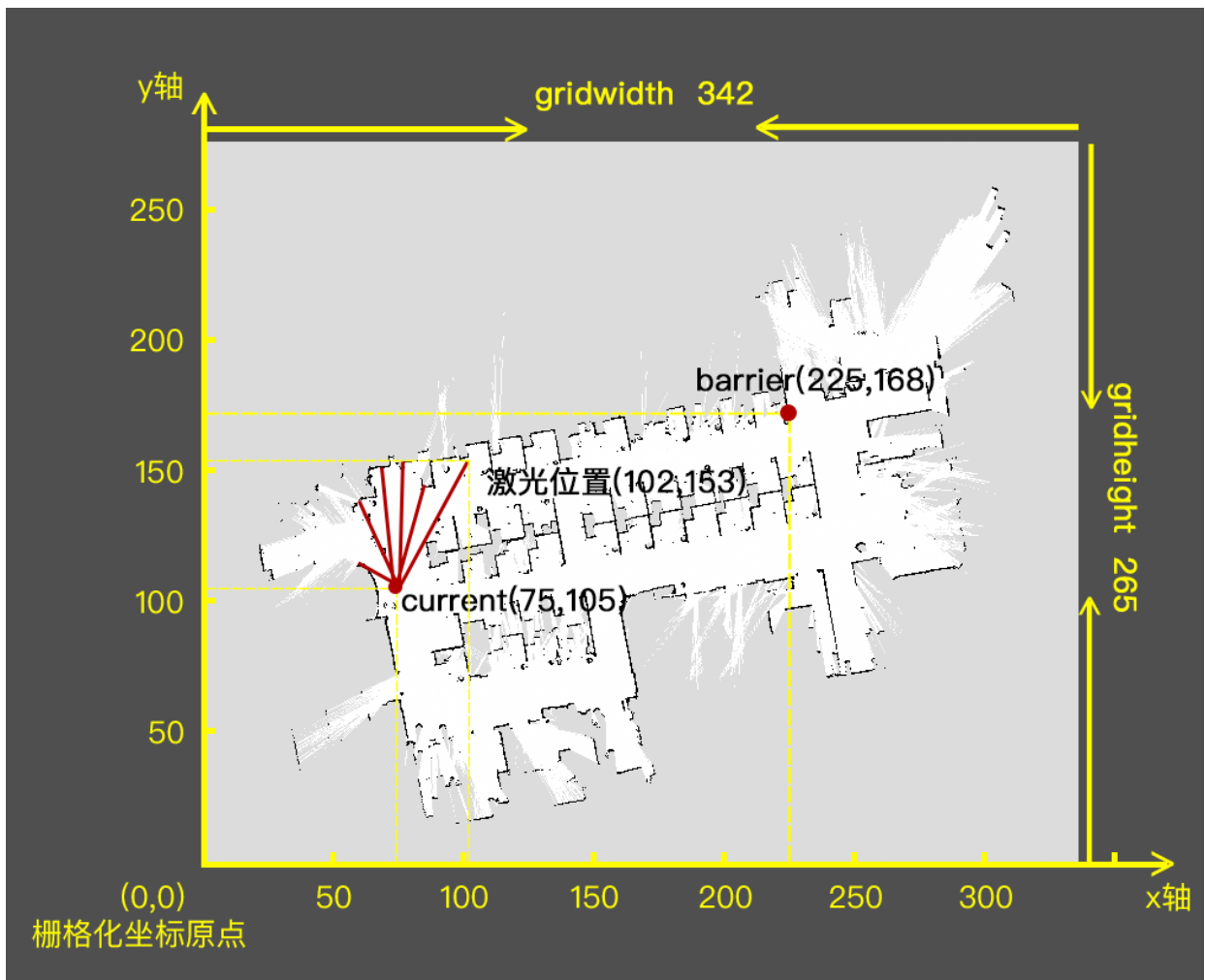
首先您得通知机器要开始扫地图啦

开始扫描地图 GET请求 `/gs-robot/cmd/start_scan_map?map_name=(自定义要构建地图的名字)`

机器需要您手动控制它在您希望它工作的地点完整跑上一圈以构建地图

```
机器移动控制 POST请求 /gs-robot/cmd/move  
request body:  
{  
  "speed":{  
    "linearSpeed":0.5,  
    //线速度,取值在[-0.6,0.6],正值则前进,负值则后退  
    "angularSpeed":0.2  
    //角速度,取值在[-0.6,0.6],正值则左转,负值则右转  
  }  
}
```

在扫描地图过程中您可以按您的需求定时访问提供的接口以实时获取机器当前的位置、激光传感器返回的数据、构建完成的地图数据等信息(考虑到很多用户并不需要去展示地图构建的过程,原本并不想在这里介绍这三个API,但是为了让用户弄清楚如何使用机器返回的坐标数据,还是在这里介绍了吧:))



获取根据地图栅格化之后的激光数据：GET请求 `/gs-robot/real_time_data/laser_phit`
response body:

```
{
  "gridPhits": [{"x":111,"y":111}, {"x":111,"y":111},
{"x":111,"y":111}], //栅格化后的激光位置
  "mapInfo": {
    "gridWidth":100. //栅格地图宽度
    "gridHeight":100 //栅格地图长度
  }
}
```

获取机器人当前位置：GET请求 `/gs-robot/real_time_data/position`
response body:

```
{
  "angle":111.41528128678252, //角度
  "gridPosition": { //栅格化坐标
    "x":100,
    "y":100
  }
}
```

获取扫描地图过程中构建完成地图数据：GET请求 `/gs-robot/real_time_data/scan_map_data`
response body:

```
{
  {
    "data": [ //栅格化后的障碍物坐标的index,详细计算过程看下方注解
      57681,
      51486,
      51572,
      51500,
      51656,
      51812,
      51068,
      56480,
      52736,
      52574,
    ],
    "mapInfo": {
      "gridHeight": 265, //栅格地图高度
      "gridWidth": 342, //栅格地图宽度
    },
  }
}
```

注解：以第一个index'57681'为例， $57681 \div \text{gridWidth} = 57681 \div 342 = 168 \text{余} 225$ ，即栅格化坐标(225,168)位置为障碍物

如果您在扫描地图过程中想取消此次扫描并且不打算保存

取消扫描且不保存地图：GET请求 `/gs-robot/cmd/cancel_scan_map`

如果您已经扫描结束并且确认扫描正确可以保存

结束扫描并保存地图：GET请求 `/gs-robot/cmd/async_stop_scan_map`

您需要通过这个接口来确认机器是否成功保存刚刚构建的地图, 保存的过程有可能会比较长, 所以您可以选择在发送完结束扫描并保存地图命令后过一段时间去检查是否保存成功, 或者选择发送完结束扫描并保存地图命令后定时轮询机器是否保存成功, 推荐使用第二种方法

查看是否成功构建地图并将构建的地图保存：GET请求 `/gs-robot/cmd/is_stop_scan_finished`

您可以获取机器现有地图的列表

获取地图列表：GET请求 `/gs-robot/data/maps`

如果您有显示地图的需要, 您可以获取地图的png格式图片

获取地图png格式图片：GET请求 `/gs-robot/data/map_png?map_name=(要获取的地图名字)`

您可以给任何地图换一个名字

修改地图名称：GET请求 `/gs-robot/cmd/rename_map?origin_map_name=(需要重命名地图的名字)&new_map_name=(地图的新名字)`

您可以删除除了正在使用的地图之外的任何地图

删除地图：GET请求 `/gs-robot/cmd/delete_map?map_name=(要删除地图的名字)`

如何初始化

为什么要初始化？

初始化的作用即确认机器当前使用的地图和机器在地图上的位置以及机器朝向的方向是否正确, 只有在都正确的情况下机器才能准确完成导航、跟线等它擅长的工作. 除了上述扫描地图相关API和另外极少一些API外, 其他API都要在初始化之后才能够使用.

三个默认的初始化点

每张地图都会默认生成Origin点和EndPoint, 即扫地图时的开始点和结束点, 用作初始化用. 正在使用的地图会实时更新Current点, 即机器人当前的位置. Current点只有正在使用的地图才有, 一旦切换地图就不能再用, 此点的作用是机器重启后或者搭载在机器上的应用程序重启后, 机器可以自动完成初始化, 而不需要人为参与

首先您需要为机器选择一张当前工作地点的地图

加载地图 GET请求 `/gs-robot/cmd/use_map?map_name=(要加载地图的名字)`

然后需要您帮助机器定位它在当前地图上的位置

第一种初始化方法:

您可以在扫描地图过程中或者用后面即将要介绍的编辑地图功能为机器在地图上设置任意个初始化点, 初始化时将机器移动到其中一个初始化点然后进行初始化, 只要位置正确即可, 机器朝向的方向可以不对, 因为机器会转动一圈自己完成准确定位

①扫描地图时添加机器当前所在点作为一个初始化点: GET请求 `/gs-robot/cmd/add_init_point?init_point_name=(自定义的初始化点的名字)&type=0`

②地图编辑时在地图上添加一个初始化点: POST请求 `/gs-robot/cmd/init_point/add_init_point`

request body:

```
{
  "angle":100, //自定义初始化点的角度
  "gridX":100, //自定义初始化点X方向上的栅格化坐标
  "gridY":100, //自定义初始化点Y方向上的栅格化坐标
  "mapName":"workPlace", //在哪张地图上定义初始化点
  "name":"initPoint", //自定义初始化点名称
  "type":0 //type字段须定义为0
}
```

③将机器放置在初始化点进行初始化: GET请求 `/gs-robot/cmd/initialize?map_name=(当前正在使用的地图的名称)&init_point_name=(初始化点名称)`

第二种初始化方法:

您可以使用机器当前所在的位置对机器进行初始化，允许传入的参数存在一定的误差，机器会转动一圈自己完成准确定位

直接传入机器的位置参数对机器进行初始化：POST请求 `/gs-robot/cmd/initialize_customized`

```
request body:
{
  "mapName":"workPlace" //当前使用的地图的名字
  "point":{
    "angle":100, //机器人角度
    "gridPosition":{ //机器人栅格化坐标
      "x":100,
      "y":100
    }
  }
}
```

第三种初始化方法:

如若从上次机器关闭或者机器上的应用程序关闭以来，机器的位置以及方向并没有改变，您可以直接使用本文档开头介绍的Current点对机器进行初始化

直接使用某个初始化点进行初始化，机器不会转动一圈自身进行定位，所以必须保证此初始点的坐标与方向和机器是一致的：GET请求 `/gs-robot/cmd/initialize_directly?map_name=(当前机器使用的地图的名字)&init_point_name=(要使用的初始化点的名字)`

所以，我们推荐您尽可能使用 `/gs-robot/cmd/initialize_directly?map_name=(当前机器使用的地图的名字)&init_point_name=Current` 来对机器进行初始化

接下来您需要检查机器是否已经初始化成功，初始化的过程有可能会比较长，所以您可以选择在发送完初始化命令后过一段时间去检查初始化是否成功，或者选择发送完初始化命令后定时轮询机器是否初始化成功，推荐使用第二种方法

检查初始化是否成功：GET请求 `/gs-robot/cmd/is_initialize_finished`

```
response body:
{
  "data":"true" // "true"表示初始化完成， "false"表示未完成初始化
}
```

您可以获取已有的初始化点的列表

获取特定地图的初始化点列表：GET请求 `/gs-robot/data/positions?map_name=(获取该地图的所有初始化点)&type=0`

您可以为某个初始化点重新命名

修改初始化点名称：GET请求 `/gs-robot/cmd/rename_init_point?map_name=(初始化点所在的地图名字)&origin_init_point_name=(初始化点原先的名字)&new_init_point_name=(您要给初始化点取的新名字)`

您可以将已经不需要的初始化点删除

删除初始化点：GET请求 `/gs-robot/cmd/delete_init_point?map_name=(要删除初始化点所在的地图的名字)&init_point_name=(要删除的初始化点的名字)`

如何添加导航点

为什么要添加导航点？

在地图上添加导航点, 并为这个导航点取一个名字, 之后便可以使用这个导航点的名字方便地命令机器去到达这个导航点, 关于如何命令机器去到达导航点将在后文'如何执行任务'中介绍

您也可以在扫描地图过程中或者使用后面即将介绍的地图编辑功能将地图上任意合理位置添加为导航点

①扫描地图过程中将机器所在的位置添加为导航点：GET请求 `/gs-robot/cmd/add_position?position_name=(自定义导航点的名字)&type=2`

②添加任意合理位置为导航点：POST请求 `/gs-robot/cmd/position/add_position`
request body:

```
{
  "angle":100,
  "gridX":100,
  "gridY":100,
  "mapName":"workPlace",
  "name":"navigatePoint",
  "type":2 //除了type字段需定义为2外，以上字段均为自定义
}
```

您可以查看某个地图上已经添加的所有导航点

获取地图上导航点列表：GET请求 `/gs-robot/data/positions?map_name=(获取该地图上的导航点列表)&type=2`

您可以将不再需要的导航点删除

删除导航点：GET请求 `/gs-robot/cmd/delete_position?map_name=(要删除的导航点所在地图的名字)&position_name=(要删除的导航点的名字)`

如何添加录制路径

为什么要添加录制路径？

手动控制机器跑一个路径，并将这个路径取一个名字，机器可以将这个路径录制下来，之后可以通过这个路径的名字方便地命令机器重复任意次去跑这个录制下来的路径。关于如何命令机器去跑这个录制的路径将在后文‘如何执行任务’中介绍

首先您需要保证机器已经初始化过了，然后通知机器要开始去录制路径了

开始录制路径：GET请求 `/gs-robot/cmd/start_record_path?map_name=(当前正在使用的地图的名字)&path_name=(自定义路径的名字)`

机器需要您手动控制它跑一个您希望它录制的路径

```
机器移动控制：POST请求 /gs-robot/cmd/move  
request body:  
{  
  "speed":{  
    "linearSpeed":0.5,  
    //线速度,取值在[-0.6,0.6],正值则前进,负值则后退  
    "angularSpeed":0.2  
    //角速度,取值在[-0.6,0.6],正值则左转,负值则右转  
  }  
}
```

在录制过程中您可以随时取消录制

取消录制路径：GET请求 `/gs-robot/cmd/cancel_record_path`

若您录制完成并且确认可以保存

停止录制并保存文件：GET请求 `/gs-robot/cmd/stop_record_path`

您可以获取某个地图上已经录制好的全部路径信息

获取指定地图的录制路径：GET请求 `/gs-robot/data/paths?map_name=(指定地图的名字)`

您可以为已经录制好的路径重新取一个名字

重命名录制路径：GET请求 `/gs-robot/cmd/rename_path?map_name=(录制的路径所在地图的名字)&origin_path_name=(该录制路径原先的名字)&new_path_name=(为该录制路径取的新名字)`

如果您认为某个已经录制好的路径已经不在需要, 您可以删除它

删除已经录制好的某个录制路径：GET请求 `/gs-robot/cmd/delete_path?map_name=(该录制路径所在地图的名字)&path_name=(该录制路径的名字)`

如何执行任务

若希望机器去执行导航、跟线等一系列任务, 您需要先将该一系列任务添加到机器的任务队列中并给该任务取一个名字, 之后便可以使用该任务的名字去命令机器执行该任务, 目前已经有的任务类型有: `PlayPathTask`, `PlayAreaPathtask`, `NavigationTask`, `PlayGraphPathTask`, `PlayGraphPathGroupTask`, `PlayUnfilledPathTask`

添加任务到任务队列. 比如您希望机器有一个任务mainTask: 先到一个已经添加的导航点pointA, 再去栅格化坐标为(100,100)的位置, 然后跑一段之前录制好的路径pathB.

```
添加任务到任务队列并将该任务保存到任务队列中：POST请求 /gs-robot/cmd/save_task_queue
request body:
{
  "name":"mainTask", //自定义任务名字
  "map_name":"workPlace", //该任务所使用的地图
  "map_id":"", //地图id
  "loop":false, //是否循环执行任务
  "tasks":[
    {
      "name":"NavigationTask", //导航任务
      "start_param":{
        "map_name":"workPlace",
        "position_name":"pointA" //指定导航点
      }
    },
    {
      "name":"NavigationTask", //导航任务
      "start_param":{
        "destinnation":{
          "angle":100,
          "gridPosition":{
            "x":100,
            "y":100
          }
        }
      }
    },
    {
      "name":"PlayPathTask", //跟线任务(录制的路径)
      "start_param":{
        "map_name":"workPlace",
        "path_name":"pathB" //指定录制好的路径
      }
    }
  ]
}
```

现在可以使用任务的名字去命令机器执行您上面添加的这个mainTask任务了

```
开始执行一个任务：POST请求 /gs-robot/cmd/start_task_queue
request body:
{
  "name":"mainTask",
  "loop":false,
  "map_name":"workPlace"
}
```

如果这个mainTask任务您只想执行一次, 并不想长期执行, 您也跳过将该任务保存到任务队列这一步, 而是直接命令机器去执行

开始执行一个任务: POST请求 /gs-robot/cmd/start_task_queue

request body:

```
{
  "name": "",
  "loop": false,
  "map_name": "workPlace"
  "tasks": [
    {
      "name": "NavigationTask", //导航任务
      "start_param": {
        "map_name": "workPlace",
        "position_name": "pointA" //指定导航点
      }
    },
    {
      "name": "NavigationTask", //导航任务
      "start_param": {
        "destinnation": {
          "angle": 100,
          "gridPosition": {
            "x": 100,
            "y": 100
          }
        }
      }
    },
    {
      "name": "PlayPathTask", //跟线任务(录制的路径)
      "start_param": {
        "map_name": "workPlace",
        "path_name": "pathB" //指定录制好的路径
      }
    }
  ]
}
```

在执行导航任务过程中, 机器会通过websocket给您实时推送导航状态. 在执行跟线任务过程中, 机器会通过websocket给您实时推送跟线状态.

导航、跟线状态推送(web-socket)

/gs-robot/notice/status port:8089

导航、跟线状态推送数据结构:

```
{
  "statusCode" : 601,
  "statusMsg" : "rotating",
  "statusData" : {}
}
```

跟线推送样例: 301-304才有data数据, 其他为null

```
{
  "statusCode" : 302,
  "statusData" :
  {
    "graphName" : "test",
    "index" : "8",
    "nextPoint" : "h",
    "passedPoints" : "['a','b']",
    "pathName" : "rty",
    "remainingMilleage" : "7.790543",
    "remainingTime" : "16.230299",
    "totalMilleage" : "8.190543",
    "totalTime" : "17.063632",
    "type" : 2
  },
  "statusMsg" : "following path"
}
```

导航推送样例: 404-408才有data, 其他为null

```
{
  "statusCode":405,
  "statusData":{
    "angle":0,
    "gridPosition":{
      "x":820,
      "y":352
    },
    "name":"NO_GOAL",
    "worldPosition":{
      "orientation":{
        "w":1,
        "x":0,
        "y":0,
        "z":0
      },
      "position":{
        "x":-0.5749993886798634,
        "y":1.625000262632966,
        "z":0
      }
    }
  },
  "statusMsg":"navigating"
```

```

}
000 空闲中           idle
100 暂停状态         pause

300 路径无效         path is invalid
301 导航去路径开始点 navigation to start point
302 跟线中           following path
303 避障等待中       waiting to avoid obstacle
304 避障中           avoiding obstacle
305 路径终点到达不了 path is not safe
306 跑路径结束       follow path finished

401 定位异常         localization error
402 目标点不安全     goal point not safe
403 前方有障碍物     obstacle in front
404 目标点不可达     goal point unreachable
405 导航行走中       navigating
406 规划路径中       planning
407 到达目标点       reached
408 未到达目标点     unreached
409 寻找充电桩       searching charger
410 移动到充电桩前   moving to charger
411 后退对接充电桩   backward to charger

600 正在转圈         rotating
601 不能转圈         cannot rotate
602 转圈结束         roate finished
603 转圈失败         rotate failed
604 下桩中（桩指充电桩） leaving charger
605 不能下桩（转圈前准备） cannot leaving charger
606 下桩结束（转圈前准备） leaving charger finished
607 下桩失败（转圈前准备） leaving charger failed

1000 定位程序正常
1001 转圈定位中
1002 计算中
1003 定位程序初始化中
1004 定位程序异常，需重启 localization program error, please reload
    map and initialize robot
1005 激光不匹配       laser param not valid
1006 定位丢失         localization lost
1007 定位校正         localization calibration

1008 检测到rfid       rfid detected
1009 rfid验证后，定位有问题 after rfid verify, localization is abnormal
1010 加密狗有问题     usb security not valid

```

在机器执行任务过程中,您可以随时暂停任务和恢复任务

```
暂停执行任务：GET请求 /gs-robot/cmd/pause_task_queue  
恢复执行任务：GET请求 /gs-robot/cmd/resume_task_queue
```

您同样可以随时停止任务的执行

```
停止任务队列：GET请求 /gs-robot/cmd/stop_task_queue
```

For more

如何添加区域路径

在某些应用场景下，录制区域路径功能可以发挥很大的作用。比如将机器用作扫地机器人时，当您想清扫一个房间时，只需要开启录制区域路径功能，然后手动控制机器绕着房间跑一圈，机器便能自动生成可以将房间打扫干净的区域路径

您首先得保证机器已经初始化过了，然后通知机器要开始录制区域路径啦

```
开始录制区域路径：GET请求 /gs-robot/cmd/start_record_area?map_name=(当前使用的地图的名字)&path_name=(自定义区域路径的名字)
```

机器需要您手动控制它绕着您希望生成区域路径的区域的外层跑一圈

```
机器移动控制：POST请求 /gs-robot/cmd/move  
request body:  
{  
  "speed":{  
    "linearSpeed":0.5,  
    //线速度,取值在[-0.6,0.6],正值则前进,负值则后退  
    "angularSpeed":0.2  
    //角速度,取值在[-0.6,0.6],正值则左转,负值则右转  
  }  
}
```

当您录制完成并且确认可以保存

停止并保存区域路径 GET请求 `/gs-robot/cmd/stop_record_area`

如果在录制区域路径过程中您希望取消录制

取消录制区域路径 GET请求 `/gs-robot/cmd/cancel_record_area`

如何添加手画路径

如果您觉得添加录制路径略显麻烦, 那添加手画路径无疑是一种更好的选择.

手画路径需要post的数据除了名字外主要包含四部分:

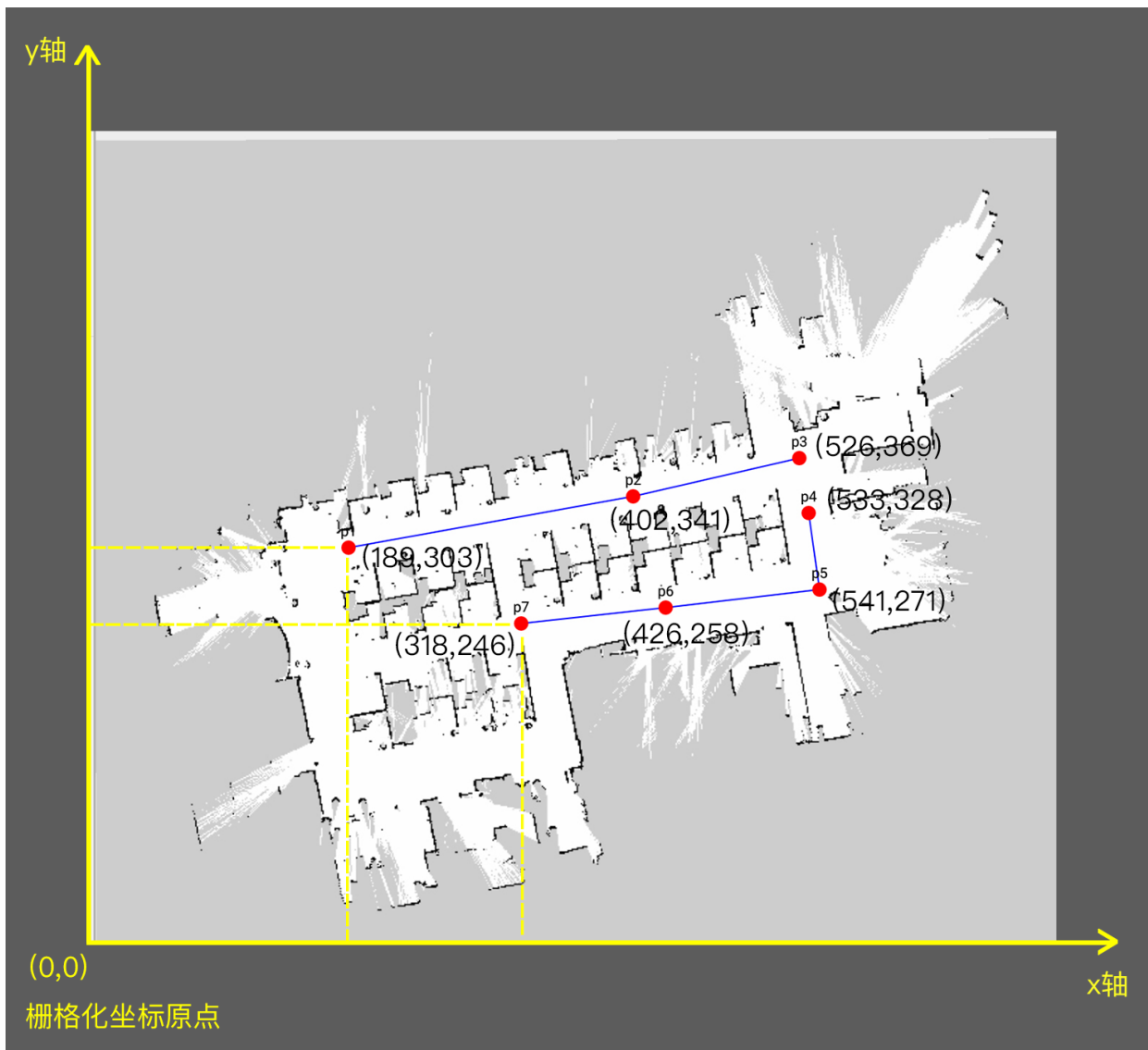
第一部分 是points,即点, 每个点的数据有名字, 栅格化坐标, 和动作列表即到达此点时要执行的动作, 每个动作有名字和成员, 成员是嵌套类型即成员嵌套成员, 成员的数据有名字, 类型, 和值, 动作目前有Pause暂停, Rotate2d平面旋转, Rotate3d立体旋转, RotateTo2d旋转到角度, RotateTo3d立体旋转到角度, PlaySound播放音乐, StopSound停止播放, OperateDevice操作设备。

第二部分 是lines, 即线, 每条线包含名字, 起点, 终点, 和曲率半径。

第三部分 是paths, 即路径, 每条路径包含名字, 线段名字列表, 点列表, 此处点列表不仅只有名字, 是全部点的数据, 为了不同路径相同点可以执行不一样的动作。

第四部分 是pathGroups, 即路径组, 每个路径组包含名字和路径名字列表

比如您希望机器添加如下两条手画路径: p1->p2->p3、 p4->p5->p6->p7



首先您得保证p1和p2、p2和p3、p4和p5...之间可以生成线段, 传入两个点的坐标、曲率半以及地图名字, 如果不能生成则返回errorcode: path is not safe

验证两点之间能否生成线段: POST请求 /gs-robot/cmd/verify_graph_line

request body:

```
{
  "begin" :
  {
    "x" : 189,
    "y" : 303
  },
  "end" :
  {
    "x" : 402,
    "y" : 341
  },
  "mapName" : "workPlace",
  "radius" : 0
}
```

}

如果通过了上述验证, 您得再次验证是否可以生成路径

验证是否可以生成路径: POST请求 /gs-robot/cmd/verify_graph_path

request body:

```
{
  "lines" :
  [
    {
      "begin" :
      {
        "x" : 533,
        "y" : 328
      },
      "end" :
      {
        "x" : 541,
        "y" : 271
      },
      "radius" : 0
    },
    {
      "begin" :
      {
        "x" : 541,
        "y" : 271
      },
      "end" :
      {
        "x" : 426,
        "y" : 258
      },
      "radius" : 0
    },
    {
      "begin" :
      {
        "x" : 426,
        "y" : 258
      },
      "end" :
      {
        "x" : 318,
        "y" : 246
      },
      "radius" : 0
    }
  ],
  "mapName" : "workPlace"
}
```

验证再次通过的话, 就可以生成手画路径了

生成手画路径: post请求 /gs-robot/cmd/generate_graph_path

request body:

```
{
  "points" :
  [
    {
      "actions" : [],
      "angle" : -1,
      "defaultAngle" : false,
      "gridPosition" :
      {
        "x" : 189,
        "y" : 303
      },
      "name" : "p1"
    },
    {
      "actions" : [],
      "angle" : -1,
      "defaultAngle" : false,
      "gridPosition" :
      {
        "x" : 402,
        "y" : 341
      },
      "name" : "p2"
    },
    {
      "actions" : [],
      "angle" : -1,
      "defaultAngle" : false,
      "gridPosition" :
      {
        "x" : 526,
        "y" : 369
      },
      "name" : "p3"
    },
    {
      "actions" : [],
      "angle" : -1,
      "defaultAngle" : false,
      "gridPosition" :
      {
        "x" : 533,
        "y" : 328
      },
      "name" : "p4"
    },
    {
      "actions" : [],
```

```

        "angle" : -1,
        "defaultAngle" : false,
        "gridPosition" :
        {
            "x" : 541,
            "y" : 271
        },
        "name" : "p5"
    },
    {
        "actions" : [],
        "angle" : -1,
        "defaultAngle" : false,
        "gridPosition" :
        {
            "x" : 426,
            "y" : 258
        },
        "name" : "p6"
    },
    {
        "actions" : [],
        "angle" : -1,
        "defaultAngle" : false,
        "gridPosition" :
        {
            "x" : 318,
            "y" : 246
        },
        "name" : "p7"
    }
],
"lines" :
[
    {
        "begin" : "p1",
        "end" : "p2",
        "name" : "p1_p2",
        "radius" : 0
    },
    {
        "begin" : "p2",
        "end" : "p3",
        "name" : "p2_p3",
        "radius" : 0
    },
    {
        "begin" : "p4",
        "end" : "p5",
        "name" : "p4_p5",
        "radius" : 0
    },
],

```

```
{
  "begin" : "p5",
  "end" : "p6",
  "name" : "p5_p6",
  "radius" : 0
},
{
  "begin" : "p6",
  "end" : "p7",
  "name" : "p6_p7",
  "radius" : 0
}
],
"mapName" : "workplace",
"name" : "graph1", //自定义图的名字
"pathGroups" : [],
"paths" :
[
  {
    "lines" :
    [
      {
        "name" : "p1_p2"
      },
      {
        "name" : "p2_p3"
      }
    ],
    "name" : "path1",
    "points" :
    [
      {
        "actions" : [],
        "angle" : -1,
        "defaultAngle" : false,
        "gridPosition" :
        {
          "x" : 189,
          "y" : 305
        },
        "name" : "p1"
      },
      {
        "actions" : [],
        "angle" : -1,
        "defaultAngle" : false,
        "gridPosition" :
        {
          "x" : 402,
          "y" : 267
        },
        "name" : "p2"
      }
    ]
  }
]
```

```
    },
    {
      "actions" : [],
      "angle" : -1,
      "defaultAngle" : false,
      "gridPosition" :
      {
        "x" : 526,
        "y" : 239
      },
      "name" : "p3"
    }
  ]
},
{
  "lines" :
  [
    {
      "name" : "p4_p5"
    },
    {
      "name" : "p5_p6"
    },
    {
      "name" : "p6_p7"
    }
  ],
  "name" : "path2",
  "points" :
  [
    {
      "actions" : [],
      "angle" : -1,
      "defaultAngle" : false,
      "gridPosition" :
      {
        "x" : 533,
        "y" : 280
      },
      "name" : "p4"
    },
    {
      "actions" : [],
      "angle" : -1,
      "defaultAngle" : false,
      "gridPosition" :
      {
        "x" : 541,
        "y" : 337
      },
      "name" : "p5"
    },
  ],
}
```

```
{
  "actions" : [],
  "angle" : -1,
  "defaultAngle" : false,
  "gridPosition" :
  {
    "x" : 426,
    "y" : 350
  },
  "name" : "p6"
},
{
  "actions" : [],
  "angle" : -1,
  "defaultAngle" : false,
  "gridPosition" :
  {
    "x" : 318,
    "y" : 362
  },
  "name" : "p7"
}
]
}
]
```

如何编辑地图

当您想手动修改机器扫描出来的地图, 或者您想在地图上为机器添加虚拟墙, 又或者您有在地图上标记斜坡的需要, 这时您可以使用我们提供的编辑地图功能

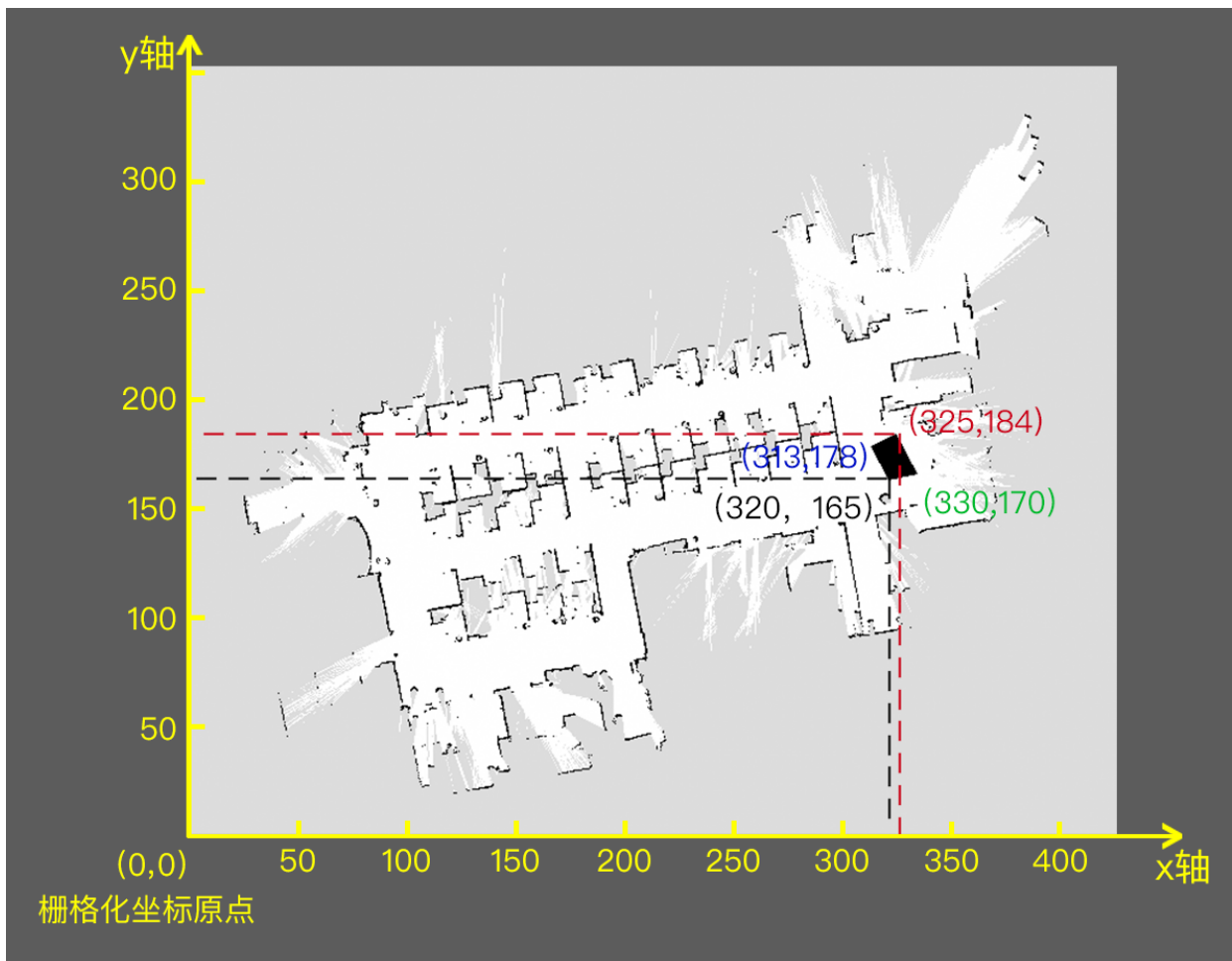
什么时候可以使用虚拟墙?

如果机器工作在一个面积很大的空地, 而此时您希望限制机器的活动范围, 这时您可以在地图上添加虚拟墙

什么时候可以使用标记斜坡?

如果机器工作的地点有较多斜坡, 这会略微增加机器的处理时间去判断斜坡, 而标记斜坡可以使机器运行更加流畅

您可以使用一个任意边的多边形对地图原图做添加障碍物或去除障碍物的操作. 比如您想在地图原图上添加一个如图的四边形障碍物



在地图原图上增加障碍物: POST请求 `/gs-robot/cmd/edit_map?map_name=(要修改的地图的名字)&operation_type=add`

request body:

```
{
  "mapname": "workPlace",
  "obstacles": {
    "polygons": [
      [{"x": 313, "y": 178}, {"x": 320, "y": 165},
      {"x": 330, "y": 170}, {"x": 325, "y": 184}]
    ]
  }
}
```

去除障碍物的操作与添加障碍物雷同, post一个可以把您要去除的障碍物覆盖的任意边的多边形即可, 多边形覆盖范围内的障碍物都将被去除, 比如您需要去除刚刚手动添加的这个障碍物

从地图原图上去除障碍物: POST请求 /gs-robot/cmd/edit_map?map_name=(要修改的地图的名字)&operation_type=remove

request body:

```
{
  "mapname": "workPlace",
  "obstacles": {
    "polygons": [
      [{"x": 313, "y": 178}, {"x": 320, "y": 165},
      {"x": 330, "y": 170}, {"x": 325, "y": 184}]
    ]
  }
}
```

希望在地图上增加障碍物,除了可以直接修改地图,还可以使用添加虚拟墙的方法,添加虚拟墙之前需要先获取一下地图上现有的虚拟墙数据,然后在现有的虚拟墙数据的基础上进行修改,虚拟墙的数据包括线段(线段的起点和终点)、矩形(矩形的对角点)、圆形(圆的圆心和半径)、封闭多边形(多个点)、不封闭多边形(多个点),

获取虚拟墙数据：GET请求 /gs-robot/data/virtual_obstacles?map_name=(获取该地图的虚拟墙数据)

response body:

```
{
  "ID": "",
  "createDate": "",
  "description": "for test",
  "mapName": "test",
  "obstacles": {
    "circles": [ {
      "center": {
        "x": 109,
        "y": 76
      },
      "radius": 60.108235708594876
    } ],
    "lines": [ {
      "start": {
        "x": 449,
        "y": 806
      },
      "end": {
        "x": 560,
        "y": 802
      }
    } ],
    "polygons": [
      [ {
        "x" : 476,
        "y" : 672
      },
      {
        "x" : 489,
        "y" : 748
      },
      {
        "x" : 522,
        "y" : 719
      },
      {
        "x" : 554,
        "y" : 765
      },
      {
        "x" : 569,
        "y" : 676
      }
    ]
  ],
  "polylines": [
    [ {
      "x" : 476,
```

```

        "y" : 672
      },
      {
        "x" : 489,
        "y" : 748
      },
      {
        "x" : 522,
        "y" : 719
      },
      {
        "x" : 554,
        "y" : 765
      },
      {
        "x" : 569,
        "y" : 676
      }
    ]
  ],
  "rectangles": [{
    "end": {
      "x": 613,
      "y": 362
    },
    "start": {
      "x": 529,
      "y": 281
    }
  }
]}
}
}

```

在获取的虚拟墙数据的基础上进行修改, 修改包括添加虚拟墙和删除虚拟墙, 将修改之后的虚拟墙数据重新post给机器即可保存修改

更新虚拟墙数据: post请求 `/gs-robot/cmd/update_virtual_obstacles`

在地图上标明斜坡可以使得机器工作地更加流畅, 在地图上添加斜坡之前应该获取一次地图上现有的斜坡数据, 然后在现有的斜坡数据的基础上进行修改, 将修改之后的斜坡数据post给机器即可完成保存. 获取到的斜坡数据的数据结构与上面获取到的虚拟墙数据的数据结构相同

获取地图上的斜坡数据: GET请求 `/gs-robot/data/slopes`

更新地图上的斜坡数据: post请求 `/gs-robot/cmd/update_slopes`

如何获取机器实时状态

各个状态的字段数量会因为机器型号的不同而不同

获取机器当前执行的任务的状态

获取机器当前执行的任务的状态：GET请求 `/gs-robot/real_time_data/work_status`
response body:

```
{
  "data": {
    "task_status": {
      "map": {
        "createdAt": "2017-03-03 18:16:22",
        "dataFileName": "dc35ecaa-c22c-444b-a1e3-f0750a0ec87f.data",
        "id": "0a40ebe1-1060-4d8e-8c51-72784e8ab934",
        "mapInfo": {
          "gridHeight": 512,
          "gridWidth": 672,
          "originX": -22.400000000000002,
          "originY": -11.200000000000001,
          "resolution": 0.05000000074505806
        },
        "name": "test2",
        "obstacleFileName": "",
        "pgmFileName": "649baae0-e50e-4a21-82ac-cbadd7292d7c.pgm",
        "pngFileName": "9dba6068-eac4-4754-95db-f2dc8939e0bf.png",
        "slopeFileName": "",
        "yamlFileName": "c3930a97-1a4e-46e4-85c9-0174048ced9c.yaml"
      },
      "task": null,
      "task_queue": null
    },
    "work_status": {
      "current_map_name": "",
      "current_path_name": "",
      "map": null,
      "work_type": "IDLE",
      "work_type_id": 7
    }
  },
  "errorCode": "",
  "msg": "succeeded",
  "succeeded": true
}
```

获取系统健康状态

获取系统健康状态：GET请求 `/gs-robot/data/health_status`

response body:

```
{  
  {"MCUConnection":true, "deviceStatusTopic":true, "gyroBoard":true, "laserConnection":true, "laserTopic":true, "odomTopic":true, "powerBoard":true, "protectorTopic":true, "routerConnection":true, "ultrasonic0":false, "ultrasonic1":false, "ultrasonic2":false, "ultrasonic3":false, "ultrasonic4":false, "ultrasonic5":false, "ultrasonicBoard":false, "ultrasonicTopic":true}\n}
```

获取设备状态

获取设备状态：GET请求 `/gs-robot/data/device_status`

response body:

```
{  
  "speed":3.5, // 实时速度  
  "battery":65, // 实时电量(65%)  
  "millage":1231, // 自机器启动的里程 单位：米  
  "totalMillage":8888, // 自机器出厂的里程 单位：米  
  "emergencyStop":false, // 是否是急停状态  
  "charge":false //是否在充电  
}
```