WEB API

通讯接口

目 录

第1章　概述

EAIGO Web API 是基于 EAI 科技 DASHGO 移动底座以及 PATHGO 模块进行开发的，为所有平台提供的一套简单易用的移动控制、建图导航服务接口，为开发者对 EAI 产品进行二次开发提供良好的技术支持。通过使用 Web API 可以轻松为应用程序实现机器人移动控制、建图定位以及自主导航、移动避障等功能。

主要分为以下几类接口：

一、　启动机器人建图/导航服务的接口

二、　机器人的控制接口

三、　获取机器人软硬件状态信息的接口

四、　机器人的建图/导航相关功能接口

*注：　1、接口中不要包含中文。

2、每进行一次接口升级，接口 URL 不会改变，但接口返回 JSON 数据的结构或内容可能会发生改变，所以接口升级后，在开发过程中，应注意 JSON 数据结构是否发生改变。

第2章　机器人服务 API

*注意：不要同时启动多个服务。其中建图/导航服务包含移动服务。

2.1　机器人移动服务

2.1.1 带陀螺仪的移动服务

1.　启动服务

请求 URL，如下：

```
http://192.168.31.200:8080/driverDemoImuStart
```

JSON 数据响应，如下：

```
{
   "result":true,              请求是否成功
   "msg":"driverDemoImuStart"      请求功能类型
}
```

2.　关闭服务

请求 URL，如下：

```
http://192.168.31.200:8080/driverDemoImuStop
```

JSON 数据响应，如下：

```
{
   "result": true,        请求是否成功
   "msg": "driverDemoImuStop"      请求功能类型
}
```

2.1.2 不带陀螺仪的移动服务

1. 启动服务

请求 URL，如下：

```
http://192.168.31.200:8080/driverDemoStart
```

JSON 数据响应，如下：

```
{
   "result": true,       请求是否成功
   "msg": "driverDemoStart"        请求功能类型
}
```

2. 关闭服务

请求 URL，如下：

```
http://192.168.31.200:8080/driverDemoStop
```

JSON 数据响应，如下：

```
{
   "result": true,       请求是否成功
   "msg": "driverDemoStop"         请求功能类型
}
```

2.2 机器人建图服务

2.2.1 带陀螺仪的手动建图服务

1. 启动建图服务

请求 URL，如下：

```
http://192.168.31.200:8080/gmappingImuAppStart
```

JSON 数据响应，如下：

```
{
    "result": true,       请求是否成功
    "msg": "gmappingImuAppStart"        请求功能类型
}
```

2. 关闭建图服务

请求 URL，如下：

```
http://192.168.31.200:8080/gmappingImuAppStop
```

JSON 数据响应，如下：

```
{
   "result": true,       请求是否成功
   "msg": "gmappingImuAppStop"          请求功能类型
}
```

3. 保存地图

请求 URL，如下：（需在建图服务处于运行状态下使用）

```
http://192.168.31.200:8080/addMap?name=eai_map&map_type=imu
```

请求参数说明，如下：

字段名	数据类型	数据描述
name	String	自定义的地图名
map_type	String	要保存的地图类型，此处为 imu

JSON 数据响应，如下：

```
{
```

```
    "isError": false,      请求是否成功
    "errInfo": ""          请求错误信息
}
```

2.2.2 不带陀螺仪的手动建图服务

1. 启动建图服务

请求 URL，如下：

```
http://192.168.31.200:8080/gmappingAppStart
```

JSON 数据响应，如下：

```
{
    "result": true,        请求是否成功
    "msg": "gmappingAppStart"       请求功能类型
}
```

2. 关闭建图服务

请求 URL，如下：

```
http://192.168.31.200:8080/gmappingAppStop
```

JSON 数据响应，如下：

```
{
    "result": true,        请求是否成功
    "msg": "gmappingAppStop"        请求功能类型
}
```

3. 保存地图

请求 URL，如下：（需在建图服务处于运行状态下使用）

```
http://192.168.31.200:8080/addMap?name=eai_map&map_type=odom
```

请求参数说明，如下：

字段名	数据类型	数据描述
name	String	自定义的地图名
map_type	String	要保存的地图类型，此处为 odom

JSON 数据响应，如下：

```
{
   "isError": false,      请求是否成功
   "errInfo": ""          请求错误信息
}
```

2.2.3 带陀螺仪的自主建图服务

1. 启动建图服务

请求 URL，如下：

```
http://192.168.31.200:8080/autoGmappingImuStart
```

JSON 数据响应，如下：

```
{
  "result": true,          请求是否成功
  "msg": "autoGmappingImuStart"        请求功能类型
}
```

2. 关闭建图服务

请求 URL，如下：

```
http://192.168.31.200:8080/autoGmappingImuStop
```

JSON 数据响应，如下：

```
{
    "result": true,        请求是否成功
    "msg": "autoGmappingImuStop"        请求功能类型
}
```

3. 保存地图

请求 URL，如下：（需在建图服务处于运行状态下使用）

```
http://192.168.31.200:8080/addMap?name=eai_map&map_type=autoImu
```

请求参数说明，如下：

字段名	数据类型	数据描述
name	String	自定义的地图名
map_type	String	要保存的地图类型，此处为 autoImu

JSON 数据响应，如下：

```
{
    "isError": false,      请求是否成功
    "errInfo": ""          请求错误信息
}
```

2.2.4 不带陀螺仪的自主建图服务

1. 启动建图服务

请求 URL，如下：

```
http://192.168.31.200:8080/autoGmappingStart
```

JSON 数据响应，如下：

```
{
   "result": true,        请求是否成功
   "msg": "autoGmappingStart"      请求功能类型
}
```

2. 关闭建图服务

请求 URL，如下：

```
http://192.168.31.200:8080/autoGmappingStop
```

JSON 数据响应，如下：

```
{
   "result": true,        请求是否成功
   "msg": "autoGmappingStop"       请求功能类型
}
```

3. 保存地图

请求 URL，如下：

```
http://192.168.31.200:8080/addMap?name=eai_map&map_type=auto
```

请求参数说明，如下：

字段名	数据类型	数据描述
name	String	自定义的地图名
map_type	String	要保存的地图类型，此处为 auto

JSON 数据响应，如下：

```
{
    "isError": false,      请求是否成功
    "errInfo": ""          请求错误信息
}
```

2.3 机器人导航服务

2.3.1 带陀螺仪的导航服务

1. 启动导航服务

请求 URL，如下：

```
http://192.168.31.200:8080/navigationImuAppStart
```

JSON 数据响应，如下：

```
{
    "result": true,        请求是否成功
    "msg": "navigationImuAppStart"       请求功能类型
}
```

2. 关闭导航服务

请求 URL，如下：

```
http://192.168.31.200:8080/navigationImuAppStop
```

JSON 数据响应，如下：

```
{
    "result": true,        请求是否成功
    "msg": "navigationImuAppStop"        请求功能类型
}
```

2.3.2 不带陀螺仪的导航服务

1. 启动导航服务

请求 URL，如下：

```
http://192.168.31.200:8080/navigationAppStart
```

JSON 数据响应，如下：

```
{
   "result": true,        请求是否成功
   "msg": "navigationAppStart"      请求功能类型
}
```

2. 关闭导航服务

请求 URL，如下：

```
http://192.168.31.200:8080/navigationAppStop
```

JSON 数据响应，如下：

```
{
    "result": true,        请求是否成功
    "msg": "navigationAppStop"      请求功能类型
}
```

第3章　机器人控制 API

3.1　移动控制

3.1.1 平滑移动控制

请求 URL，如下：

```
http://192.168.31.200:8080/cmdSmootherMove?lx=0.3&az=0.5
```

参数说明，如下：(两个参数都为 0 时停止)

字段名	数据类型	数据描述
lx	float	直行方向的速度（前正后负）
az	float	自转方向的速度（左正右负）

返回的 JSON 数据如下:{"res":true,"req":"cmdSmootherMove","err":""}

字段名	数据类型	数据描述
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

3.1.2 移动控制

请求 URL，如下：

```
http://192.168.31.200:8080/cmdMove?lx=0.3&az=0.5
```

参数说明，如下：(两个参数都为 0 时停止)

字段名	数据类型	数据描述
lx	float	直行方向的速度（前正后负）
az	float	自转方向的速度（左正右负）

返回的 JSON 数据说明如下:{"res":true,"req":"cmdMove","err":""}

字段名	数据类型	数据描述
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

3.2 自主建图控制

3.2.1 开始/继续扫图

请求 URL，如下：

```
http://192.168.31.200:8080/autoGmappingCtrl?state=start
```

参数说明，如下：（state：start 开始或继续扫图，pause 暂停扫图，stop 停止扫图）

字段名	数据类型	数据描述
state	String	要执行的操作的类型

返回的 JSON 数据说明如下:{"res":true,"req":"autoGmappingCtrl","err":""}

字段名	数据类型	数据描述
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

3.2.2 暂停扫图

请求 URL，如下：

```
http://192.168.31.200:8080/autoGmappingCtrl?state=pause
```

3.2.3 停止扫图

请求 URL，如下：

```
http://192.168.31.200:8080/autoGmappingCtrl?state=stop
```

3.3 自动回充控制

3.3.1 执行回充

请求 URL，如下：

```
http://192.168.31.200:8080/rechargeCtrl&status=1
```

参数说明，如下：（status：1 执行回充，0 停止回充）

字段名	数据类型	数据描述

state	String	要执行的操作的类型

返回的 JSON 数据说明如下:{"res":true,"req":" rechargeCtrl","err":""}

字段名	数据类型	数据描述
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

3.3.2 停止回充

请求 URL，如下：

```
http://192.168.31.200:8080/rechargeCtrl&status=0
```

3.4 简单行走任务

3.4.1 直线行走

请求 URL，如下：

```
http://192.168.31.200:8080/driverAction?method=line&imu=true&total=1&speed=1&error=0.05
```

参数说明，如下：（method：line 走直线 angule 原地旋转 cycle 扇形旋转）

字段名	数据类型	数据描述
method	String	选择行走的方式

imu	Boolean	是否带有 imu
total	double	直线行走的距离<10 米
speed	double	直线行走的速度 0.02-2 m/s
error	double	允许的距离误差（米）

返回的 JSON 数据说明如下: {"res":true,"req":"driverAction","err":""}

字段名	数据类型	数据描述
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

3.4.2 原地旋转

请求 URL，如下：

```
http://192.168.31.200:8080/driverAction?method=angule&imu=true&total=90&speed=1&error=5
```

参数说明，如下：（method：line 走直线 angule 原地旋转 cycle 扇形旋转）

字段名	数据类型	数据描述
method	String	选择行走的方式
imu	Boolean	是否带有 imu
total	double	旋转行走的角度

speed	double	旋转的速度 0.02- pi rad/s
error	double	允许的旋转角度误差（度）

3.4.3 扇形旋转

请求 URL，如下：

```
http://192.168.31.200:8080/driverAction?method=cycle&imu=true&total=90
&radius=1&speed=1&error=5
```

参数说明，如下：（method：line 走直线 angule 原地旋转 cycle 扇形旋转）

字段名	数据类型	数据描述
method	String	选择行走的方式
imu	Boolean	是否带有 imu
total	double	旋转行走的角度
radius	double	旋转行走的半径
speed	double	旋转的速度 0.02- pi rad/s
error	double	允许的旋转角度误差（度）

3.4.4 停止行走

请求 URL，如下：

```
http://192.168.31.200:8080/driverAction?method=stop
```

参数说明，如下：

字段名	数据类型	数据描述
method	String	要执行的操作的类型

3.5 机器人位置初始化

请求 URL，如下：

```
http://192.168.31.200:8080/initialPose?pose=0_0_0_0_0_0_0
```

参数说明，如下：（pose:x_y_z_qx_qy_qz_qw）

字段名	数据类型	数据描述
pose	String	Pose 坐标以_拼接的字符串

返回的 JSON 数据说明如下: {"res":true,"req":"initialPose","err":""}

字段名	数据类型	数据描述
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

第4章　机器人状态 API

4.1　获取硬件状态信息

4.1.1 获取急停开关状态

请求 URL，如下：

```
http://192.168.31.200:8080/getEmergencybtStatus
```

返回的 JSON 数据如下：{"result":"0","res":true,"req":"getEmergencybtStatus","err":""}

字段名	数据类型	数据描述
result	String	状态值(0 未按下，1 按下)
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

4.1.2 获取陀螺仪角度信息

请求 URL，如下：

```
http://192.168.31.200:8080/getImuAngle
```

返回的 JSON 数据如下：{"result":"-0.0","res":true,"req":"getImuAngle","err":""}

字段名	数据类型	数据描述

result	String	角度值(-180~180)
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

4.1.3 获取完整的陀螺仪状态信息

请求 URL，如下：

```
http://192.168.31.200:8080/getImu
```

4.1.4 获取电量百分比

请求 URL，如下：

```
http://192.168.31.200:8080/ getVoltagePercentage
```

返回的 JSON 数据如下：{"result":"80","res":true,"req":"getVoltagePercentage","err":""}

字段名	数据类型	数据描述
result	String	电压百分比
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

4.1.5 获取电源电压值

请求 URL，如下：

```
http://192.168.31.200:8080/getVoltageValue
```

返回的 JSON 数据如下：{"result":"27127","res":true,"req":"getVoltageValue","err":""}

字段名	数据类型	数据描述
result	String	电压值(毫伏)
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

4.1.6 获取红外线数据

请求 URL，如下：

```
http://192.168.31.200:8080/getIrs       获取所有红外线数据信息

http://192.168.31.200:8080/getIr0       获取单个红外线数据信息(Ir0~Ir5 共 6 个)
```

4.1.7 获取超声波数据

请求 URL，如下：

```
http://192.168.31.200:8080/getSonars        获取所有超声波数据信息

http://192.168.31.200:8080/getSonar0        获取单个红外线数据信息(Sonar0~Sonar5 共 6 个)
```

4.1.8 获取充电状态

请求 URL，如下：

```
http://192.168.31.200:8080/getRechargeStatus
```

返回的 JSON 数据如下：{"result":"5","res":true,"req":"getRechargeStatus","err":""}

字段名	数据类型	数据描述
result	String	充电状态
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

result 的值对应代表的充电状态为：

0 启动回充标志　1 未找到充电桩发出的红外信号且未收到握手信号

2 收到充电桩发出的握手信号　3 充电正常进行中

4 充电过程中出现电压不正常　5 电池电量充满/充电结束

4.1.9 获取充电方式

请求 URL，如下：

```
http://192.168.31.200:8080/getRechargeWay
```

返回的 JSON 数据如下：{"result":"0","res":true,"req":"getRechargeWay ","err":""}

字段名	数据类型	数据描述
result	String	充电方式
res	Boolean	请求是否成功

req	String	URL 请求的功能类型
err	String	URL 请求失败信息

result 的值对应代表的充电方式为：

0 未充电　　1 回充充电　　2DC 充电　　3 回充和 DC 充电同时生效

4.1.10　系统使用情况

请求 URL，如下：

```
http://192.168.31.200:8080/machineState
```

JSON 数据响应，如下：

```
{
    "isError": false,          请求是否成功
    "errInfo": "",             请求错误信息
    "bootTime": "",            系统启动时间
    "updateStamp": "",         系统时间戳
    "updateTime": "",          系统时间
    "cpu": "",                 CPU 运行情况
    "load": "",                运存使用情况
    "swapUsed": "",            SWAP 使用情况
    "diskRoot": "",            磁盘使用情况
    "port1": "",               串口 1 是否被使用，等于"1"时为已使用
    "port2": "",               串口 2 是否被使用，等于"1"时为已使用
    "port3": "",               串口 3 是否被使用，等于"1"时为已使用
    "port4": ""                串口 4 是否被使用，等于"1"时为已使用
}
```

4.2 获取软件状态信息

4.2.1 获取上面激光数据信息

请求 URL，如下：

```
http://192.168.31.200:8080/getLaserData
```

返回的 JSON 数据如下：{"result":"{…}","res":true,"req":"getLaserData","err":""}

字段名	数据类型	数据描述
result	String	激光数据
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

4.2.2 获取下面激光数据信息

请求 URL，如下：

```
http://192.168.31.200:8080/getLaser2Data
```

返回的 JSON 数据如下：{"result":"{…}","res":true,"req":"getLaser2Data","err":""}

字段名	数据类型	数据描述
result	String	激光数据

res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

4.2.3 获取地图数据信息

请求 URL，如下：

```
http://192.168.31.200:8080/getMapData
```

返回的 JSON 数据如下：{"result":"{…}","res":true,"req":"getMapData","err":""}

字段名	数据类型	数据描述
result	String	地图数据
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

4.2.4 获取导航时路径规划的数据信息

请求 URL，如下：

```
http://192.168.31.200:8080/getGlobalPlan
```

返回的 JSON 数据如下：{"result":"{…}","res":true,"req":"getGlobalPlan","err":""}

字段名	数据类型	数据描述
result	String	路径数据
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

4.2.5 获取实时保存的当前的坐标位置

请求 URL，如下：(建图、导航通用)

```
http://192.168.31.200:8080/getCurrentSavePose
```

返回的 JSON 数据如下：{"result":"{…}","res":true,"req":"getCurrentSavePose","err":""}

字段名	数据类型	数据描述
result	String	坐标数据
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

4.2.6 获取简单行走任务的执行状态

请求 URL，如下：

```
用处：执行简单行走任务后，结合起来，循环调用这两个接口查看任务是否正常执行并完成

http://192.168.31.200:8080/getActionFeedback     获取任务进度(数字)

http://192.168.31.200:8080/getActionResult  检查任务是否失败(失败时 result 为"False"、否则 True)
```

返回的 JSON 数据如下：{"result":"…","res":true,"req":"getActionResult","err":""}

字段名	数据类型	数据描述
result	String	任务进度/任务执行状态
res	Boolean	请求是否成功
req	String	URL 请求的功能类型
err	String	URL 请求失败信息

4.2.7 机器人服务启动情况

请求 URL，如下：

```
http://192.168.31.200:8080/processState
```

JSON 数据响应，如下：

```
{
    "isError": false,          请求是否成功
    "errInfo": "",             请求错误信息
    "updateStamp": "",         系统时间戳
    "updateTime": "",          系统时间
    "dashgoDriver": "",        机器人底盘驱动服务是否启动
    "driver": "",              不带 IMU 的机器人固定行走服务是否启动
    "driverImu": "",           带 IMU 的机器人固定行走服务是否启动
    "pathgoImu": "",           陀螺仪服务是否启动
```

```
    "flashgo": "",              EAI 激光雷达驱动服务是否启动
    "sickTim": "",              SICK 激光雷达驱动服务是否启动
    "gmapping": "",             不带 IMU 的手动建图服务是否启动
    "gmappingImu": "",          带 IMU 的手动建图服务是否启动
    "navigation": "",           基于手动建图的不带 IMU 的导航服务是否启动
    "navigationImu": "",        基于手动建图的带 IMU 的导航服务是否启动
    "autoGmapping": "",         不带 IMU 的自主建图服务是否启动
    "autoGmappingImu": "",      带 IMU 的自主建图服务是否启动
    "autoGmappingStart": "",    自主建图控制服务是否启动
    "autoNavigation": "",       基于自主建图的不带 IMU 的导航服务是否启动
    "autoNavigationImu": ""     基于自主建图的带 IMU 的导航服务是否启动
}
```

4.2.8 系统日志条数

请求 URL，如下：

```
http://192.168.31.200:8080/getRosErrLogNum
```

JSON 数据响应，如下：

```
{
    "isError": false,       请求是否成功
    "errInfo": "",          请求错误信息
    "number": ""            日志条数
}
```

注意：计算的日志条数都是以警告级别以上的日志信息为基础。

4.2.9 系统日志具体内容

请求 URL，如下：

```
http://192.168.31.200:8080/getRosErrLog
```

JSON 数据响应，如下：

```
{
    "isError": false,      请求是否成功
    "errInfo": "",         请求错误信息
    "log": ""              详细的日志信息
}
```

注意：获取到的日志信息都是警告级别以上的日志信息。

第5章　机器人功能 API

5.1　地图操作

5.1.1 获取地图列表

请求 URL，如下：

```
http://192.168.31.200:8080/getMap
```

JSON 数据响应，如下：

```
{
    "isError": false,      请求是否成功
    "errInfo": "",         请求错误信息
    "maps": ""             所有地图的信息
}
```

maps 包含的 JSON 数组说明，如下：

字段名	数据类型	数据描述
id	String	地图编号
name	String	地图名称
path	String	地图所保存的位置
isUsed	Boolean	是否为当前使用的地图

5.1.2 下载地图等文件

请求 URL，如下：

```
192.168.31.200:8080/downloadFile/文件名.类型?flag=download
```

只有真实存在于/home/eaibot/dashgo_ws/src/dashgo/dashgo_nav/maps 目录下的文件才可以通过此 URL 进行下载。

5.1.3 修改地图名

请求 URL，如下：

```
http://192.168.31.200:8080/updateMap?map_id=2017-09-04_16:09:55&name=one
```

请求参数说明，如下：

字段名	数据类型	数据描述
map_id	String	要修改的地图编号
name	String	新的地图名称

JSON 数据响应，如下：

```
{
    "isError": false,      请求是否成功
    "errInfo": ""          请求错误信息
}
```

5.1.4 删除某个地图

请求 URL，如下：

```
http://192.168.31.200:8080/delMap?map_id=2017-09-04_16:09:55
```

请求参数说明，如下：

字段名	数据类型	数据描述
map_id	String	要删除的地图编号

JSON 数据响应，如下：

```
{
    "isError": false,      请求是否成功
    "errInfo": ""          请求错误信息
}
```

5.1.5 切换到某个地图

1、倘若启动的导航服务为

```
http://192.168.31.200:8080/navigationAppStart
```

则请求 URL，如下：

```
http://192.168.31.200:8080/setMap?map_name=two&map_type=odom
```

请求参数说明，如下：

字段名	数据类型	数据描述
name	String	要切换的地图名
map_type	String	要切换地图的类型，此处为 odom

2、倘若启动的导航服务为

```
http://192.168.31.200:8080/navigationImuAppStart
```

则请求 URL，如下：

```
http://192.168.31.200:8080/setMap?map_name=two&map_type=imu
```

请求参数说明，如下：

字段名	数据类型	数据描述
name	String	要切换的地图名
map_type	String	要切换地图的类型，此处为 imu

3、以上 URL 请求的 JSON 数据响应，都如下：

```
{
    "isError": false,      请求是否成功
    "errInfo": ""          请求错误信息
}
```

5.2 多目标点循环

5.2.1 机器人当前坐标

获取机器人当前坐标（相对于地图），请求 URL，如下：

```
http://192.168.31.200:8080/getCurrentSavePose          建图/导航时都能使用
JSON 数据响应：{"result":"{…pose 数据…}","res":true,"req":"getCurrentSavePose","err":""}
```

或：

```
http://192.168.31.200:8080/getCurrentRobotPose         仅导航时有效
JSON 数据响应如下：
{
    "isError": false,      //请求是否成功
```

```
    "errInfo": "",          //请求错误信息
    "pose": ""   //当前位置信息，包含 x、y、z 坐标与四元数信息。如：1.0_0.0_0.0_0.0_0.0_0.0_1.0
}
```

5.2.2 设置导航间隔时间

在多点导航状态下，到达目标后停留的时间，则请求 URL，如下：

```
http://192.168.31.200:8080/updateGoalState?key=intervalTime&val=5
```

请求参数说明，如下：

字段名	数据类型	数据描述
val	Int	停留时间，单位为秒

JSON 数据响应，如下：

```
{
    "isError": false,      请求是否成功
    "errInfo": ""          请求错误信息
}
```

5.2.3 设置循环/单次导航

1. 设置导航模式为循环导航（循环执行导航队列）

```
http://192.168.31.200:8080/updateGoalState?key=mode&val=loop
```

2. 设置导航模式为单次导航（只执行完导航队列一次就停止）

```
http://192.168.31.200:8080/updateGoalState?key=mode&val=order
```

5.2.4 设置目标点

1. 添加目标点到别名队列

比如：机器人当前在门口 1 位置，设置名称为 door1，则请求 URL，如下：

```
将目标点添加进“别名队列”，只有别名队列中存在的点，才能够加入到导航队列中

http://192.168.31.200:8080/addPoseGoalAliases?key=door1&val=1.0_0.0_0.0_0.0_0.0_0.0_1.0
```

请求参数说明，如下：

字段名	数据类型	数据描述
key	String	自定义的目标点名称
val	String	pose 属性值拼接的字符串

JSON 数据响应，如下：

```
{
    "isError": false,      请求是否成功
    "errInfo": ""          请求错误信息
}
```

2. 添加目标点到导航队列

将别名队列中已存在的目标点添加到导航队列请求 URL 如下：

```
http://192.168.31.200:8080/addGoalQueue?key=GoalQueueA&val=door1
```

请求参数说明，如下：

字段名	数据类型	数据描述
val	String	目标点名称

JSON 数据响应，如下：

```
{
    "isError": false,      请求是否成功
    "errInfo": ""          请求错误信息
}
```

5.2.5 设置当前导航点

当停止一个任务，去新的目标点时，将当前导航点变更为新的目标点

```
http://192.168.31.200:8080/updateGoalState?key=currentGoalA&val=door2
```

请求参数说明，如下：

字段名	数据类型	数据描述
val	String	目标点名称

5.2.6 删除目标点

1. 删除某个目标点

以 mygoalA 为例，则请求 URL，如下

```
http://192.168.31.200:8080/delGoalQueue?key=GoalQueueA&val=door1 从导航队列中删除目标点
val:要删除的点的名称
http://192.168.31.200:8080/delGoalAliases?key=door1   从别名队列中删除目标点
key: 要删除的点的名称
```

2. 删除所有目标点

请求 URL，如下：

```
http://192.168.31.200:8080/delGoalQueue?key=GoalQueueA     删除导航队列中所有的点
http://192.168.31.200:8080/delGoalAliases    删除别名队列中所有的点
```

JSON 数据响应，如下：

```
{
    "isError": false,      请求是否成功
    "errInfo": ""          请求错误信息
}
```

5.2.7 开始/停止导航

1. 开始/继续导航

请求 URL，如下：

```
http://192.168.31.200:8080/updateGoalState?key=currentState&val=running
```

JSON 数据响应，如下：

```
{
    "isError": false,      请求是否成功
    "errInfo": ""          请求错误信息
}
```

2. 停止/暂停导航

请求 URL，如下：

```
http://192.168.31.200:8080/updateGoalState?key=currentState&val=stopped
```

JSON 数据响应，如下：

```
{
    "isError": false,      请求是否成功
    "errInfo": ""          请求错误信息
}
```

3. 设置手动触发导航*

设置下一个目标点导航控制为手动触发，则请求 URL，如下：

```
http://192.168.31.200:8080/updateGoalState?key=loopWay&val=manual
```

在手动触发导航状态下，执行导航到下一个目标点，则请求 URL，如下：

```
http://192.168.31.200:8080/updateGoalState?key=isNavNext&val=1
```

JSON 数据响应，如下：

```
{
   "isError": false,          请求是否成功
   "errInfo": ""              请求错误信息
}
```

4. 自动触发导航*（默认）

设置下一个目标点导航控制为自动触发，则请求 URL，如下：

```
http://192.168.31.200:8080/updateGoalState?key=loopWay&val=auto
```

JSON 数据响应，如下：

```
{
   "isError": false,          请求是否成功
   "errInfo": ""              请求错误信息
}
```

5.2.8 查询导航状态

请求 URL 如下：

```
http://192.168.31.200:8080/getGoalState
```

JSON 数据响应如下：

```
{
    "result":"[\"priorGoalQueue\",\"GoalQueueB\",\"currentGoal\",\"name4\",\"intervalTime
\",\"3\",\"goalQueue\",\"GoalQueueA\",\"loopWay\",\"auto\",\"currentQueue\",\"GoalQueueA\",
\"successNum\",\"24\",\"mode\",\"loop\",\"failedNum\",\"0\",\"isNavNext\",\"0\",\"currentSt
ate\",\"running\"]",                    //导航状态
    "res":true,                         //请求是否成功
    "req":"getGoalStates",              //请求功能类型
    "err":""                            //请求失败的错误信息
}
```

其中 result 为导航具体状态：

currentGoal：当前正在前往的目标点

intervalTime：到达某个目标点后停留的时间，单位秒

loopWay：从一个点到下一个点是自动触发(auto)还是手动触发(manual)

successNum：成功到达目标点的次数

mode：导航是循环导航(looper)还是单次导航(order)

failedNum：未能成功到达目标点而被放弃的次数

isNavNext：手动触发式导航是否正在前往下一个点(0：否 -1：是)

currentState：当前是否正在导航（running 正在导航 stoped 导航停止）

5.2.9 查询目标点

1. **查询别名队列中的目标点**

请求 URL 如下：

```
http://192.168.31.200:8080/getGoalAliases

//可以用来查看别名队列中是否有你要添加的目标点，如果没有，则无法添加进导航队列
```

2. 查询导航队列中的目标点

请求 URL 如下：

```
http://192.168.31.200:8080/getGoalQueue?key=GoalQueueA

//可以用来查看要导航的点是否存在于导航队列中，如果没有，则无法导航到该点
```

第6章　修订历史

日期	内容
2017-10-12	新版初稿
2018-04-16	增加控制接口和数据接口、修改结构