Woosh Robot SDK Interface

Version	Edit Time	Editor	Edit Content
v1.0-beta	2023-2-15	HuiMin	First draft
	2023-3-1		Add DMS interface
v1.0.5	2023-3-24		Add partial interface call instructions
v1.1.1	2023-6-20		Add robot occupancy interface Add simplified version of scene data Modify device status bit
v1.1.38	2024-4-23		Add runtime statistics related interfaces

Introduction

Omitted

Development and Runtime Environment

`SDK` is developed in `C++11`, `Demo` is developed in `C++17`.

`SDK` depends on `libzmq v4.3.4` and `libprotobuf v3.21`.

Currently, the following platforms are supported for secondary development:

1. Linux(GCC)

- Development Environment: *Ubuntu 20.4*
- Compiler: *gcc 9.4.0*

2. Windows(MinGW)

- Development Environment: *Windows 10*
- Compiler: *MinGW 8.1.0 32bit*

3. Windows(MSVC)

- Development Environment: *Windows 10*
- Compiler: *msvc 17.5*

4. Android(Clang)

- Development Environment: *NDK 21.4*
- Compiler: *NDK 21.4 Clang*

Main Framework

Omitted

SDK Description

File Description

```
├── CMakeLists.txt // demo CMakeLists
├── demo // woosh Robot SDK usage example source code directory
├── doc // Document directory
|   ├── woosh_robot_data_dictionary.pdf // woosh Robot data dictionary
|   └── woosh_robot_sdk_interface.pdf // woosh Robot SDK interface document (this
document)
├── include // SDK dependent header file directory
|   ├── google // Google ProtoBuf related header file directory
|   ├── woosh // Woosh data structure definition header file directory
|   ├── woosh_robot_def.h // SDK related type definition header file
|   └── woosh_robot.h // SDK interface header file
└── lib // SDK and related dependency library directory
├── android // Android platform SDK library file directory
    |   ├── libprotobuf.so
    |   └── libwoosh_robot.so
├── linux // Linux platform SDK library file directory
    |   ├── libprotobuf.so.32
    |   └── libwoosh_robot.so
└── windows // Windows platform SDK library file directory
        ├── mingw
        |   ├── libwoosh_robot.dll
        |   └── libwoosh_robot.dll.a
        └── msvc
            ├── libprotobuf.lib
            ├── libwoosh_commu.lib
            ├── libwoosh_proto.lib
            ├── libwoosh_robot.lib
            └── libzmq-mt-4_3_4.lib
```

Instructions

Woosh Robot SDK usage can refer to the examples in the `demo` directory. The following are simple usage instructions.

```
#include "woosh_robot.h"

// Communication connection settings
CommuSetting cs;
// Robot or DMS IP address
cs.addr = "172.20.12.88";
// Connection port settings
cs.port = 5410;
// Client identifier settings (customizable)
cs.identity = "woosdk-demo";

// Log printing callback function setting
cs.log_call_fun = [](const std::string &log) { printf("%s\n", log.c_str()); };

// Communication packet printing callback function setting
cs.print_pack_call_fun = [](const std::string &log) {
  printf("%s\n", log.c_str());
};

// Connection status callback function setting
cs.connect_status_call_fun = [&](const bool &is_connect) {
  printf("woosh robot %s:%d %s.\n", cs.addr.c_str(), cs.port,
         (is_connect ? "connected" : "disconnected"));
  std::lock_guard<std::mutex> lk(mutex_connect);
  connected = is_connect;
  cv.notify_one();
};

// Create robot connection instance
woosh::RobotPtr robot = Factory::newRobotInterface(cs);
// Run (call once)
robot->run();
```

Explanation: The SDK provides two modes: `connect to robot` and `connect to DMS`.

- Connect to robot mode

  ```
  // Create robot connection instance
  woosh::RobotPtr robot = Factory::newRobotInterface(cs);
  // Run (call once)
  robot->run();
  ```

- Connect to DMS mode

  ```
  // Create robot connection instance
  woosh::DispatchPtr dispatch = Factory::newDispatchInterface(cs);
  // Run (call once)
  dispatch->run();
  ```

Request example

```
// Request robot information
robot->robotInfoReq(
    robot_info,
    [&](const woosh::robot::RobotInfo &info, const bool &ok, const std::string
&msg) {
        if (ok) {
            std::cout << "Robot information request successful\n";
            std::cout << info.DebugString() << std::endl;
        } else {
            std::cout << "Robot information request failed, msg: " << msg <<
std::endl;
        }
    }, woosh::PPLB, woosh::PPLB);
```

Subscription example

```
// Subscribe to robot power information
robot->robotBatterySub(
    [&](const woosh::robot::Battery &info) {
        std::cout << "Power information updated, current power: " << info.power()
<< std::endl;
    }, woosh::PPLB);
```

Communication packet printing level

```
enum class PrintPackLevel {
  kDoNot = 0,  // Do not print
  kHead = 1,   // Print package header
  kBody = 3    // Print package header and body
};
```

The last two parameters of the request interface are used to set the print level of the communication `request packet` and `response packet`, defaulting to `kDoNot`.

The last parameter of the subscription interface is used to set the print level of the `subscription packet`, defaulting to `kDoNot`.

Interface Description

All interfaces are asynchronous calls and are mainly divided into two categories:

Request Interface

Interface format: **bool xxxReq(req_struct, rsp_callback_fun, req_print, rsp_print)**

Interface description:

- *req_struct*: Request structure, referred to as ***req*** in the following text, for details of the corresponding structure, please refer to the `data dictionary`.
- *rsp_callback_fun*: Response callback function.
- *req_print*: Set the print level of the request packet, default is not printing.
- *rsp_print*: Set the print level of the response packet, default is not printing.
- *bool*: Return value, `TRUE` for successful request.

Response callback function format: **void (rsp_struct, is_ok, msg)**

- *rsp_struct*: Response structure, referred to as ***rsp*** in the following text. Refer to the `data dictionary` for detailed structure.
- *is_ok*: `bool` type
  - `TRUE` indicates that the server successfully responded to the request and `rsp_struct` has a value.
  - `FALSE` indicates that the server failed to process the request and `rsp_struct` has no value.
- *msg*: Response message description, usually used to describe the reason when the request cannot be processed.

Subscription Interface

Interface format: **bool xxxSub(sub_callback_fun, sub_print)**

Interface description:

- *sub_callback_fun*: Subscription information update callback function.
- *sub_print*: Set the print level of the subscription update package, default is not printing.
- *bool*: Return value, `TRUE` for successful subscription.

Subscription information update callback function format: **void (sub_struct)**

- *sub_struct*: Subscription update data, referred to as ***sub*** in the following text. For detailed structure, refer to `data dictionary`.

Robot Information Related

The robot information related interface is shared by the `robot` and `DMS` modes.

Request All Data

Method name: **robotInfoReq**

- ***req***: `woosh::robot::RobotInfo`
  - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::RobotInfo`

Explanation:

1. This interface responds with all the data of the robot. The interface returns a large amount of data, so polling requests are not recommended.
2. It is recommended to request once to synchronize robot information when connecting for the first time or reconnecting after disconnection.

General Information Request

Method Name: **robotGeneralReq**

- ***req***: `woosh::robot::General`
  - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::General`

Explanation:

1. None

Configuration Information Request

Method Name: **robotSettingReq**

- ***req***: `woosh::robot::Setting`
  - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::Setting`

Explanation:

1. None

Configuration Information Subscription

Method Name: **robotSettingSub**

- ***sub***: `woosh::robot::Setting`

Explanation:

1. None

Robot State Request

Method Name: **robotStateReq**

- ***req***: `woosh::robot::RobotState`
  - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::RobotState`

Explanation:

1. None

Robot State Subscription

Method Name: **robotStateSub**

- ***sub***: `woosh::robot::RobotState`

Explanation:

1. None

Robot Mode Request

Method Name: **robotModeReq**

- ***req***: `woosh::robot::Mode`
  - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::Mode`

Explanation:

1. None

Robot Mode Subscription

Method Name: **robotModeSub**

- ***sub***: `woosh::robot::Mode`

Explanation:

1. None

Pose Velocity Request

Method Name: **robotPoseSpeedReq**

- ***req***: `woosh::robot::PoseSpeed`
  - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::PoseSpeed`

Explanation:

1. None

Pose Velocity Subscription

Method Name: **robotPoseSpeedSub**

- ***sub***: `woosh::robot::PoseSpeed`

Explanation:

1. None

Battery Information Request

Method Name: **robotBatteryReq**

- ***req***: `woosh::robot::Battery`
  - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::Battery`

Explanation:

1. None

Battery Information Subscription

Method Name: **robotBatterySub**

- ***sub***: `woosh::robot::Battery`

Explanation:

1. None

Network Information Request

Method Name: **robotNetworkReq**

- ***req***: `woosh::robot::Network`
  - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::Network`

Explanation:

1. None

Network Information Subscription

Method Name: **robotNetworkSub**

- ***sub***: `woosh::robot::Network`

Explanation:

1. None

Scene Information Request

Method Name: **robotSceneReq**

- ***req***: `woosh::robot::Scene`
  - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::Scene`

Explanation:

1. None

Scene Information Subscription

Method Name: **robotSceneSub**

- ***sub***: `woosh::robot::Scene`

Explanation:

1. None

Task Progress Information Request

Method Name: **robotTaskProcessReq**

- ***req***: `woosh::robot::TaskProc`
  - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::TaskProc`

Explanation:

1. None

Task Progress Information Subscription

Method Name: **robotTaskProcessSub**

- ***sub***: `woosh::robot::TaskProc`

Explanation:

1. None

Device Status Information Request

Method Name: **robotDeviceStateReq**

- ***req***: `woosh::robot::DeviceState`
  - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::DeviceState`

Explanation:

1. None

Device Status Information Subscription

Method Name: **robotDeviceStateSub**

- ***sub***: `woosh::robot::DeviceState`

Explanation:

1. None

Hardware Status Information Request

Method Name: **robotHardwareStateReq**

- ***req***: `woosh::robot::HardwareState`
    - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::HardwareState`

Explanation:

1. None

Hardware Status Information Subscription

Method Name: **robotHardwareStateSub**

- ***sub***: `woosh::robot::HardwareState`

Explanation:

1. None

Operation Status Information Request

Method Name: **robotOperationStateReq**

- ***req***: `woosh::robot::OperationState`
    - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::OperationState`

Explanation:

1. None

Operation Status Information Subscription

Method Name: **robotOperationStateSub**

- ***sub***: `woosh::robot::OperationState`

Explanation:

1. None

Robot Model Reque

Method Name: **robotModelReq**

- ***req***: `woosh::robot::Model`
    - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::Model`

Explanation:

1. None

Robot Model Subscription

Method Name: **robotModelSub**

- ***sub***: `woosh::robot::Model`

Explanation:

1. None

Robot Navigation Path Request

Method Name: **robotNavPathReq**

- ***req***: `woosh::robot::NavPath`
    - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::NavPath`

Explanation:

1. None

Robot Navigation Path Subscription

Method Name: `robotNavPathSub`

- ***sub***: `woosh::robot::NavPath`

Explanation:

1. None

Historical Task Request

Method Name: **robotTaskHistoryReq**

- ***req***: `woosh::robot::TaskHistory`
    - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::TaskHistory`

Explanation:

1. By default, it returns the latest 50 historical tasks.

Status Code Request

Method Name: **robotStatusCodesReq**

- ***req***: `woosh::robot::count::StatusCodes`
    - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::count::StatusCodes`

Explanation:

1. By default, it returns the latest 50 status codes.

Status Code Subscription

Method Name: **robotStatusCodeSub**

- ***sub***: `robot::count::StatusCode`

Explanation:

1. None

Unresolved Abnormal Code Request

Method Name: **robotAbnormalCodesReq**

- ***req***: `woosh::robot::count::AbnormalCodes`
    - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::count::AbnormalCodes`

Explanation:

1. None

Unresolved Abnormal Code Subscription

Method Name: **robotAbnormalCodesSub**

- ***sub***: `woosh::robot::count::AbnormalCodes`

Explanation:

1. None

Robot Configuration Related

The robot configuration related interfaces are shared by both the `Robot` and `DMS` modes.

Robot Identification Setting

Method Name: **setIdentity**

- ***req***: `woosh::robot::setting::Identity`
    - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::setting::Identity`

Explanation:

1. Response failed, `rsp` returns the current identifier.

Connect Server Configuration

Method name: **setServer**

- ***req***: `woosh::robot::setting::Server`
    - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::setting::Server`

Explanation:

1. Response failed, `rsp` returns the current connection configuration.

Switch Autonomous Recharge

Method name: **autoCharge**

- ***req***: `woosh::robot::setting::AutoCharge`
    - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::setting::AutoCharge`

Explanation:

1. Response failed, `rsp` returns the current autonomous recharge setting.

Switch Autonomous Parking

Method name: **autoPark**

- ***req***: `woosh::robot::setting::AutoPark`
    - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::setting::AutoPark`

Explanation:

1. Response failed, `rsp` returns the current autonomous parking setting.

Switch Goods Detection

Method name: **goodsCheck**

- ***req***: `woosh::robot::setting::GoodsCheck`
    - *robot_id*: Specify the robot ID when connecting to DMS.
- ***rsp***: `woosh::robot::setting::GoodsCheck`

Explanation:

1. Response failed, `rsp` returns the current goods detection setting.

Power Configuration

Method name: **configPower**

- ***req***: `woosh::robot::setting::RsPower`
    - *robot_id*: Specify the robot ID when connecting to DMS.

- ***rsp***: `woosh::robot::setting::RsPower`

Explanation:

1. Response failed, `rsp` returns the current power configuration.

Map Related

The map-related interfaces are shared by the 'Robot' and 'DMS' modes.

Scene List Request

Method name: **sceneListReq**

- ***req***: `woosh::map::SceneList`
- ***rsp***: `woosh::map::SceneList`

Explanation:

1. None

Scene Data Request

Method name: **sceneDataReq**

- ***req***: `woosh::map::SceneData`
  - *name*: If the scene name is not specified, the current scene data will be returned.
- ***rsp***: `woosh::map::SceneData`

Explanation:

1. None

Scene Data Request (Easy)

Method name: **sceneDataEasyReq**

- ***req***: `woosh::map::SceneDataEasy`
  - *name*: If the scene name is not specified, the current scene data will be returned.
- ***rsp***: `woosh::map::SceneDataEasy`

Explanation:

1. None

Download Map Request

Method name: **downloadMap**

- ***req***: `woosh::map::Download`
  - *name*: If the scene name is not specified, the current scene data will be returned.
- ***rsp***: `woosh::map::DownloadResponse`

Explanation:

1. None

Upload Map Request

Method name: **uploadMap**

- ***req***: `woosh::map::Upload`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. None

Rename Map or Scene Request

Method name: **renameMap**

- ***req***: `woosh::map::Rename`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. None

Delete Scene or Map Request

Method name: **deleteMap**

- ***req***: `woosh::map::Delete`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. None

Robot Request Related

The robot request-related interfaces are exclusive to the 'Robot' mode.

Initialize Robot

Method Name: **initRobotReq**

- ***req***: `woosh::robot::InitRobot`
  - *is_record*: If `TRUE`, initialize the robot with the recorded reset point.
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. This interface is only valid when the control mode is `automatic` and the robot status is `uninitialized`, `idle`, and `abnormal`.

Set Robot Pose

Method Name: **setRobotPoseReq**

- ***req***: `woosh::robot::SetRobotPose`
  - If empty, initialize to the map origin <0, 0, 0>.
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. None

Set Robot Occupancy

Method Name: **setOccupancyReq**

- ***req***: `woosh::robot::SetOccupancy`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. Cannot be occupied in non-pause state during the task, other robot states can be occupied.

Switch Control Mode

Method Name: **switchControlModeReq**

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

Explanation:

1. Only available for debugging, please use the physical knob for normal operation.

Switch Work Mode

Method Name: **switchWorkModeReq**

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

Explanation:

1. This interface is only valid when the control mode is `automatic` mode and the robot status is `uninitialized`, `idle`, `charging`, and `abnormal`.

Switch Map

Method Name: **switchMapReq**

- ***req***: `woosh::robot::SwitchMap`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. This interface is only valid when the control mode is `automatic` mode and the robot status is `uninitialized`, `idle`, `charging`, and `abnormal`.

Build Map Request

Method Name: **buildMapReq**

- ***req***: `woosh::robot::BuildMap`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. This interface is only valid when the control mode is `automatic` mode and the work mode is `deployment` mode.

Build Map Data Subscription

Method Name: **buildMapDataSub**

- ***sub***: `woosh::robot::BuildMapData`

Explanation:

1. Only data is available when mapping is enabled.

Deployment Request

Method Name: **deploymentReq**

- ***req***: `woosh::robot::Deployment`
- ***rsp***: `woosh::robot::DeploymentResponse`

Explanation:

1. None

Execute Predefined Tasks

Method Name: **execPreTaskReq**

- ***req***: `woosh::robot::ExecPreTask`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. This interface is only valid when the control mode is `automatic` and the robot status is `idle` or `charging`.

Execute Task Request

Method Name: **execTaskReq**

- ***req***: `woosh::robot::ExecTask`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. This interface is only valid when the control mode is `automatic` and the robot status is `idle` or `charging`.

Action Command Request

Method Name: **actionOrderReq**

- ***req***: `woosh::robot::ActionOrder`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. This interface is only valid when the control mode is `automatic` and the robot status is `task in progress`.

Plan Navigation Path

Method Name: **planNavPathReq**

- ***req***: `woosh::robot::PlanNavPath`
- ***rsp***: `woosh::robot::NavPath`

Explanation:

1. None

Change Navigation Path

Method Name: **changeNavPathReq**

- ***req***: `woosh::robot::ChangeNavPath`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. This interface is only valid when the control mode is `automatic` and the robot status is `task in progress`.

Change Navigation Mode

Method Name: **changeNavModeReq**

- ***req***: `woosh::robot::ChangeNavMode`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. This interface is only valid when the control mode is `automatic` and the robot status is `task in progress`.

Voice Broadcast

Method Name: **speakReq**

- ***req***: `woosh::robot::Speak`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. None

Speed Control

Method Name: **twistReq**

- ***req***: `woosh::robot::Twist`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. This interface is valid when the control mode is set to `automatic`, and the robot status is `uninitialized` and `idle`.

Follow

Method name: **followReq**

- ***req***: `woosh::robot::Follow`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. None

WiFi Information

Method name: **robotWiFiReq**

- ***req***: `woosh::robot::RobotWiFi`
- ***rsp***: `woosh::robot::RobotWiFi`

Explanation:

1. None

Runtime Statistics Data Request

Method name: **countDataReq**

- ***req***: `woosh::robot::CountData`
- ***rsp***: `woosh::robot::CountDataResponse`

Explanation:

1. None

Runtime Statistics Operation Subscription

Method name: **robotCountOperationSub**

- ***sub***: `woosh::robot::count::Operation`

Explanation:

1. None

Runtime Statistics Task Subscription

Method name: **robotCountTaskSub**

- ***sub***: `woosh::robot::count::Task`

Explanation:

1. None

Runtime Statistics Status Subscription

Method name: **robotCountStatusSub**

- ***sub***: `woosh::robot::count::Status`

Explanation:

1. None

Radar Point Cloud Data Request

Method name: **scannerDataReq**

- ***req***: `woosh::robot::ScannerData`
- ***rsp***: `woosh::robot::ScannerData`

Explanation:

1. Return the latest frame of radar point cloud data

Radar Point Cloud Data Subscription

Method name: **scannerDataSub**

- ***sub***: `woosh::robot::ScannerData`

Explanation:

1. This interface has a large amount of data, it is recommended not to subscribe for a long time to avoid network congestion.

Dispatch-related Requests

The dispatch-related request interface is unique to the `dispatch` mode.

Robot Dispatch Information Request

Method name: **dispatchRobotReq**

- ***req***: `woosh::dispatch::robot::Robot`
- ***rsp***: `woosh::dispatch::robot::Robot`

Explanation:

1. None

Robot Dispatch Information Subscription

Method name: **dispatchRobotSub**

- ***sub***: `woosh::dispatch::robot::Robot`

Explanation:

1. None

Dispatch Robot List Request

Method name: **dispatchRobotsReq**

- ***req***: `woosh::dispatch::robot::Robots`
- ***rsp***: `woosh::dispatch::robot::Robots`

Explanation:

1. None

Switch Scene Request

Method name: **switchSceneReq**

- ***req***: `woosh::dispatch::system::SwitchScene`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. None

Get Current Scene Request

Method name: **sceneSettingsReq**

- ***req***: `woosh::dispatch::system::SceneSettings`
- ***rsp***: `woosh::dispatch::system::SceneSettings`

Explanation:

1. None

Scene Configuration Update Subscription

Method name: **sceneSettingsSub**

- ***sub***: `woosh::dispatch::system::SceneSettings`

Explanation:

1. None

Specify Robot Charging Request

Method name: **gotoChargeReq**

- ***req***: `woosh::dispatch::system::GotoCharge`
- ***rsp***: `google::protobuf::Empty`

Explanation:

1. None

Find Task Request

Method Name: **findTaskReq**

- ***req***: `woosh::dispatch::task::FindTask`
- ***rsp***: `woosh::task::TaskSetList`

Explanation:

1. None

Task Instruction Request

Method Name: **taskOrderReq**

- ***req*** : `woosh::dispatch::task::TaskOrder`
- ***rsp*** : `google::protobuf::Empty`

Explanation:

1. None

Top Task Request

Method Name: **stickTaskReq**

- ***req*** : `woosh::dispatch::task::StickTask`
- ***rsp*** : `google::protobuf::Empty`

Explanation:

1. None

Add WOOSH Task Request

Method Name: **addTaskReq**

- ***req*** : `woosh::task::WooshTaskSet`
- ***rsp*** : `woosh::task::WooshTaskSet`

Explanation:

1. None

Task Set State Subscription

Method Name: **taskSetStateSub**

- ***sub*** : `woosh::dispatch::task::TaskSetState`

Explanation:

1. None