

# Odom 服务接口

获取里程计数据

- 简介
- 消息结构
- 主要字段说明
- 回调函数实例
- 测试方法
- 注意事项

## 获取里程计数据

简介

里程计服务接口用于订阅机器人的位置、方向、线速度、角速度等信息的 `leg_odom` topic。该 topic 使用 `nav_msgs::msg::Odometry` 消息类型，包含机器人底盘所发布的里程计相关信息。

Topic 名称	Topic类型	角色
<code>leg_odom</code>	<code>nav_msgs::msg::Odometry</code>	订阅方

消息结构

nav\_msgs::msg::Odometry 消息结构 Bash

1

std\_msgs/Header header

2

string child\_frame\_id

3

geometry\_msgs/PoseWithCovariance pose

4

geometry\_msgs/TwistWithCovariance twist

主要字段说明

`header` : 包含时间戳和坐标系的标准 ROS 消息头。

- `stamp` : 时间戳, 指示数据采集的时间。
- `frame_id` : 坐标系ID, 指示里程计pose数据所参考的坐标系。
- `child_frame_id` : 表示里程计twist数据所参考的机器人坐标系。

`pose` : 机器人的位姿信息, 通常包括位置 ( `position` ) 和方向 ( `orientation` ) 。

- `pose.pose.position` : 通过三维数组存储的位置信息, 数组三个元素分别表示机身几何中心在世界坐标系下的x、y、z坐标值 (  $m$  )
- `pose.pose.orientation` : 通过四维数组存储的方向四元数信息, 用于描述机器人坐标系相对于odom坐标系的旋转状态。

`twist` : 机器人的线速度和角速度信息, 用于表示机器人的运动状态。

- `twist.twist.linear` : 机器人线速度, 包括 x、y 和 z 分量。
- `twist.twist.angular` : 机器人角速度, 表示机身绕机器人坐标系z轴的旋转角速度 (  $rad/s$  )

## 回调函数实例

```

1 void odometryCallback(nav_msgs::msg::Odometry::SharedPtr odom_data) {
2     const auto& position = odom_data->pose.pose.position;
3     if (isOdomInvalid(Eigen::Vector3d{position.x, position.y, position.z}))
4     {
5         SPDLOG_WARN("odometry data is invalid !!!");
6         return;
7     }
8     geometry_msgs::msg::TransformStamped transform;
9     // get current_pose_
10    transform.transform.rotation = odom_data->pose.pose.orientation;
11    transform.transform.translation.x = position.x;
12    transform.transform.translation.y = position.y;
13    transform.transform.translation.z = position.z;
14    transform.header = odom_data->header;
15    transform.child_frame_id = odom_data->child_frame_id;
16    tf_broadcaster->sendTransform(transform);
17    // get current_vel_
18    current_vel_[0] = odom_data->twist.twist.linear.x;
19    current_vel_[1] = odom_data->twist.twist.linear.y;
20    current_vel_[2] = odom_data->twist.twist.linear.z;
21    current_vel_[3] = odom_data->twist.twist.angular.x;
22    current_vel_[4] = odom_data->twist.twist.angular.y;
23    current_vel_[5] = odom_data->twist.twist.angular.z;
24 }

```

## 测试方法

- 订阅里程计消息：

```
1 ros2 topic echo /leg_odom
```

## 注意事项

- 获取里程计数据时，应先确保里程计 ROS2 节点已启动并正在发布 `leg_odom` topic。