Service client tutorial

Goal

The goal of this tutorial is to learn how to create a service client in ROS 2 in C++. This service client will send a request to the service server */controller_manager/switch_controller* to switch between two controllers for the arm. In particular there will be one running controller called *arm_controller* and another controller called *custom_controller* will be launched with the following command launch file already available in the package

```
ros2 launch agimus_change_controllers_tutorial controllers.launch.py
```

ROS 2 tutorial: link

Clone the tutorial

Navigate the source folder of your ROS 2 workspace and clone the following package to get started:

```
git clone
https://agimus-user:frpTR_--SSsbKWRJkK5V@gitlab.com/pal-robotics/agimus_winte
r_school/tutorials/agimus_change_controllers_tutorial.git
```

Create a class

1.

First, in the header file, change_controllers.hpp, add the necessary dependencies, given below:

```
#include <rclcpp/rclcpp.hpp>
#include "controller_manager_msgs/srv/switch_controller.hpp"
```

2.

Create a class, *ChangeControllers*, that inherits from the *rclcpp::Node* class. Use both the files *change_controllers.cpp* and *change_controllers.hpp* file. Create a simple constructor for this class initialised in the header and defined in the source file.

3.
Register the class as a component node as done in the ROS 2 tutorial. The advantage of using component nodes is that the node does not require a main function to be started.

4.
Add the following to CmakeLists.txt following the public ROS 2 tutorial. Instead of *ament_cmake*, use *ament_cmake_auto*, this simplifies the structure of CmakeLists.txt.

- Create a library that contains *change_controllers.cpp*.
- Register the node as a component node in the previously created library.

6.
Add the required dependencies to the *package.xml*:

- rclcpp
- rclcpp_actions
- rclcpp_components
- controller_manager_msgs
- controller_manager

Create functions

1.
Following the structure of the public ROS 2 tutorial, create the following three functions. For this tutorial no main function is required.

```
void SwtichCtrl();
void handleSwitchControllersResponse(
rclcpp::Client<controller_manager_msgs::srv::SwitchController>::SharedFuture);
```

2.
To get more information about the service interface run the following command in a terminal, this shows the list of the running controllers:

```
ros2 service list
ros2 service call controller_manager/list_controllers
```

3.
In the constructor of *ChangeController*, declare **two** ROS parameters, called *controller_activated* and *controller_deactivated*, and get the value of these parameters. Store the name of the controllers that you want to switch between as member variables of the class.

4.
Ensure that the service called in your function is able to switch between the arm_controller and the custom_controller.

Create launch file

Create a launch file, named *change_controllers.launch.py*, that launches the previously created service node. This launch file does the following:

1. Declare two launch arguments, *controller_activated* and *controller_deactivated*.
2. Create the node that runs the service client from this tutorial. Add as parameters the value of the launch arguments.
3. Add both the launch arguments and the node to the launch description.

To test the launch file, start a simulation of tiago. In another terminal, run the following:

```
ros2 launch agimus_change_controllers_tutorial change_controllers.launch.py
controller_activated:=custom_controller controller_deactivated:=arm_controller
```