ROS2 Odd Even Service Programming (sohi edit 2023.08.)

https://github.com/sohicode/ros2_basics/

SOHI CODE

Package

서비스 패키지 만들기

```
  $ cd ~/Workspaces/ros2_ws/src
  $ ros2 pkg create oe_service_pkg --build-type
ament_python
      --dependencies rclpy std_srvs interface_pkg
```

OddEvenCheck.srv

서비스 인터페이스
src/interface_pkg/srv/OddEvenCheck.srv

```
# Request
int64 number
---
# Response
string decision
```

odd_even_server.py

서비스 서버 - 짝홀 판단값
src/oe_service_pkg/oe_service_pkg/odd_even_server.py

```
import rclpy
from rclpy.node import Node
from interface_pkg.srv import OddEvenCheck

class  OddEvenCheckServer(Node):

    def __init__(self):
        super().__init__('odd_even_server_node')
        self.srv = self.create_service(OddEvenCheck,
'odd_even_check', self.odd_even_cb)
        self.get_logger().info('Odd Even Check Service Server Running...')

    def odd_even_cb(self, request, response):
        self.get_logger().info('Request Received... ')
        if request.number % 2 == 0:
            response.decision = 'Even'
        elif request.number % 2 == 1:
            response.decision = 'Odd'
        else:
            response.decision = 'Error'

        print(request)
        print(response)
        return response


def main(args=None):
    rclpy.init(args=args)
    node = OddEvenCheckServer()
    try:
        rclpy.spin(node)
    except KeyboardInterrupt:
        node.get_logger().info('Keyboard Interrupt')
    finally:
        node.destroy_node()
        rclpy.shutdown()

if __name__ == '__main__':
    main()
```

odd_even_client.py

서비스 클라이언트
숫자 보내고 짝홀 판단값을 받음
src/oe_service_pkg/oe_service_pkg/odd_even_client.py

```
import rclpy
from rclpy.node import Node
from service_pkg.srv import OddEvenCheck

class OddEvenCheckClient(Node):
    def __init__(self):
        super().__init__('odd_even_client_node')
        self.client = self.create_client(OddEvenCheck,
'odd_even_check')
        self.req = OddEvenCheck.Request()
        self.get_logger().info('Service Client Start')

    def send_request(self, num):
        self.req.number = int(num)
        self.client.wait_for_service()
        self.future = self.client.call_async(self.req)
        rclpy.spin_until_future_complete(self, self.future)
        self.result = self.future.result()
        return self.result

def main(args=None):
    rclpy.init(args=args)
    node = OddEvenCheckClient()

    try:
        #pass
        user_input = input('Enter an Integer: ')
        res = node.send_request(user_input)
        node.get_logger().info('Server returned: ' +
res.decision)
    except KeyboardInterrupt:
        node.get_logger().info('Keyboard Interrupt')
    finally:
        node.destroy_node()
        rclpy.shutdown()

if __name__ == '__main__':
    main()
```

setup.py

```
'console_scripts': [
    'oe_server = oe_service_pkg.odd_even_server:main',
    'oe_client = oe_service_pkg.odd_even_client:main'
 ],
```

build & run

빌드

```
  $ cd ~/Workspaces/ros2_ws
  $ colcon build --symlink-install --packages-select
oe_service_pkg
  $ source ./install/setup.bash
```

노드 실행(ros2 run)

```
  $ ros2 pkg executables oe_service_pkg
  $ ros2 run oe_service_pkg oe_server
  $ ros2 run oe_service_pkg oe_client
```