[07 – ROS2 PKG Service / Action]

한국폴리텍대학교 성남캠퍼스

Homework Review

0

Homework Review

❖ Missions

1. 사용자가 주고 받을 메시지 만들기
   1. 랜덤주행 시작, 정지
   2. 랜덤주행 영역지정 좌표(4개)
2. 2개의 노드(node1, node2)를 생성하고 node1은 turtlesim_node로 랜덤주행 Topic을 발행하며 node2로 부터 사용자가 정의한 메시지를 수신하여 랜덤주행 정보에 적용
3. Node2는 사용자가 값을 설정하고 사용자가 정의한 메시지를 이용하여 node1으로 Topic 발행

1 Service Definition

1 Service definition

❖ Service 정의하기

1. Service data type definition

2. Make a Service Server

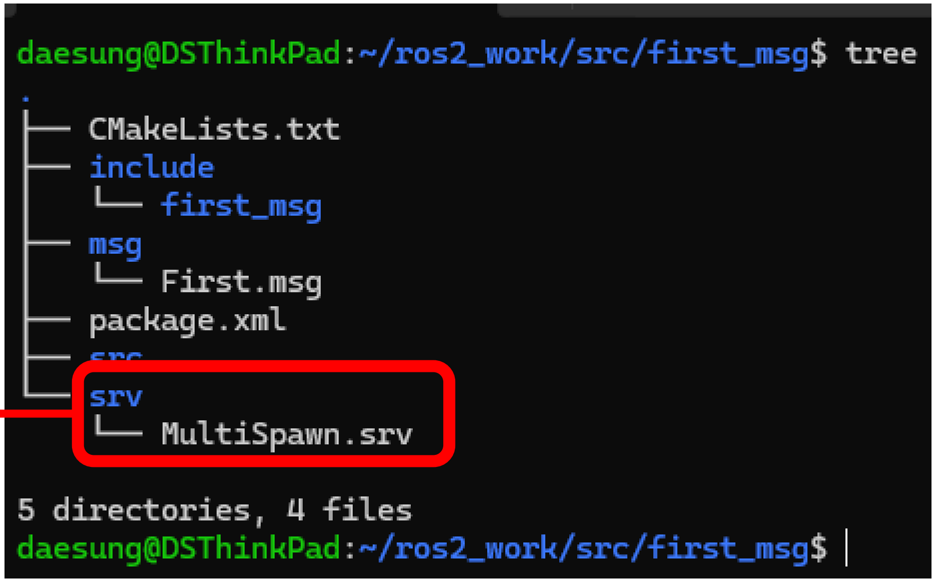
3. Make a Service Client

4. Multi-Spawn turtles Service

1

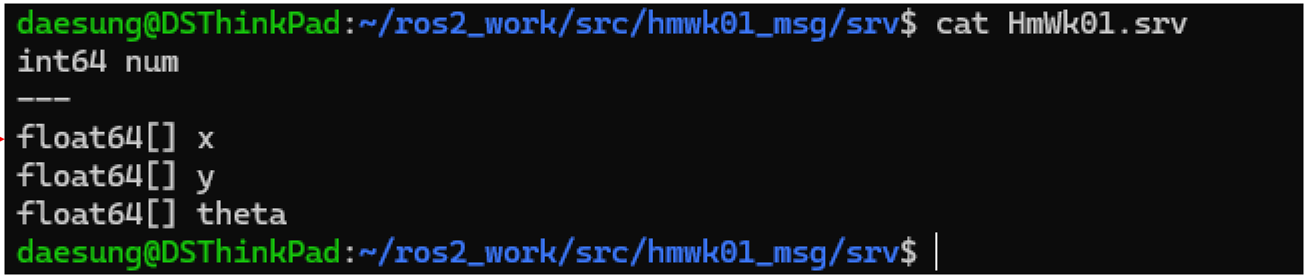
Service definition

❖ Service Data Type definition



```
daesung@DSThinkPad:~/ros2_work/src/first_msg$ tree
.
├── CMakeLists.txt
├── include
│   └── first_msg
├── msg
│   └── First.msg
├── package.xml
├── src
└── srv
    └── MultiSpawn.srv

5 directories, 4 files
daesung@DSThinkPad:~/ros2_work/src/first_msg$
```

- Service Request / Response Data Type을 정의하기 위해 위와 같이 srv 폴더를 생성하고 ~~~.srv 파일 생성

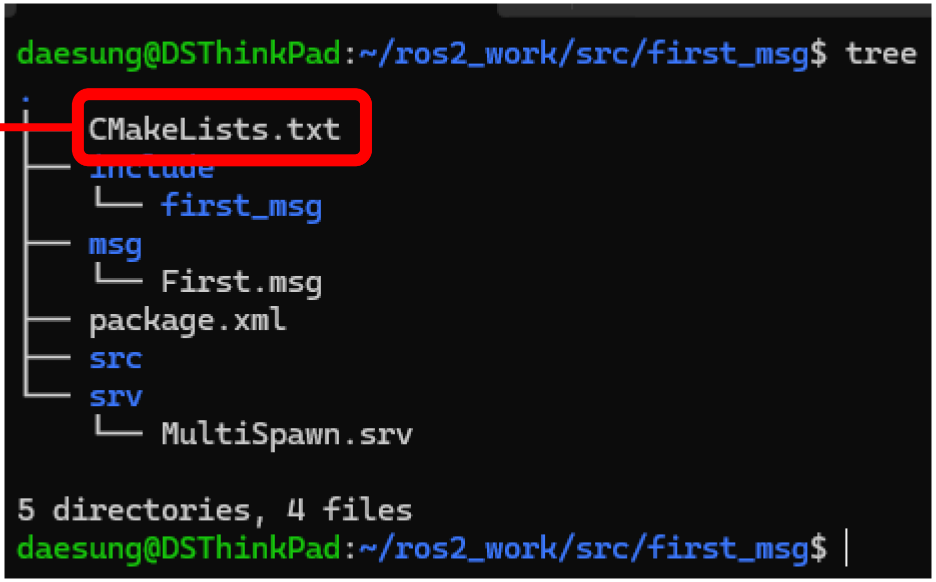


```
daesung@DSThinkPad:~/ros2_work/src/hmwk01_msg/srv$ cat HmWk01.srv
int64 num
---
float64[] x
float64[] y
float64[] theta
daesung@DSThinkPad:~/ros2_work/src/hmwk01_msg/srv$
```

:::ROS

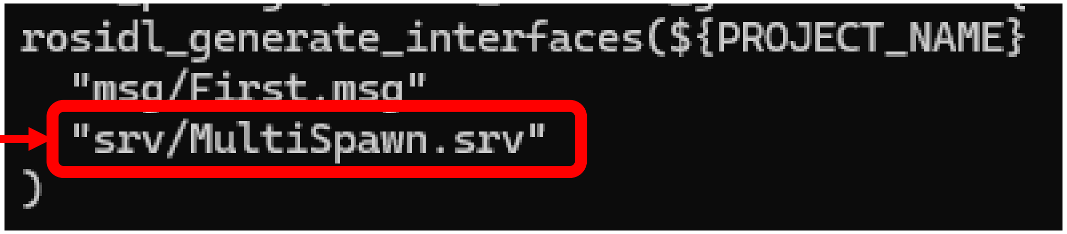
1 Service definition

❖ Service Data Type definition



```
daesung@DSThinkPad:~/ros2_work/src/first_msg$ tree
.
├── CMakeLists.txt
├── include
│   └── first_msg
├── msg
│   └── First.msg
├── package.xml
├── src
└── srv
    └── MultiSpawn.srv

5 directories, 4 files
daesung@DSThinkPad:~/ros2_work/src/first_msg$
```

- package.xml 파일은 변경사항 없음
- CMakeLists.txt 파일은 다음과 같이 새로 생성된 Service Data 파일의 경로 추가

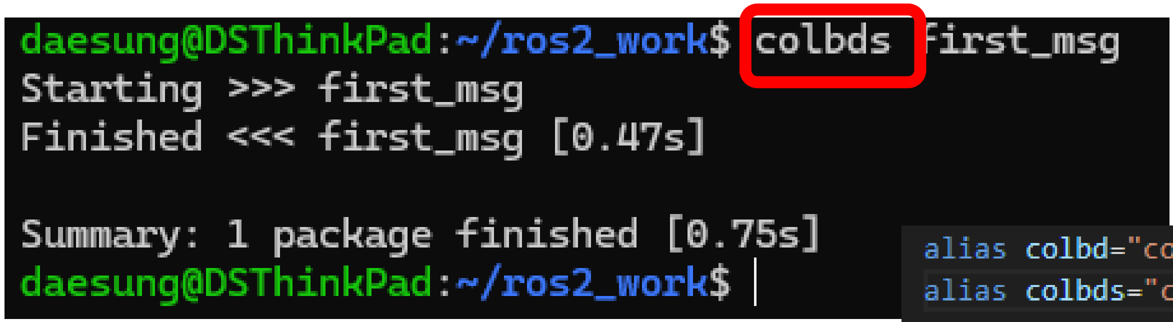


```
rosidl_generate_interfaces(${PROJECT_NAME}
  "msg/First.msg"
  "srv/MultiSpawn.srv"
)
```

1 Service definition

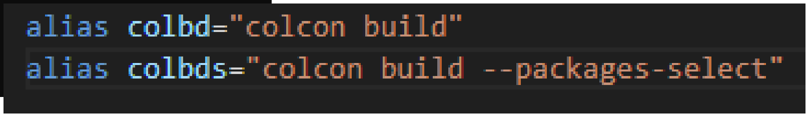
❖ Service Data Type definition

- Workspace 위치로 이동하여 colcon build 실행



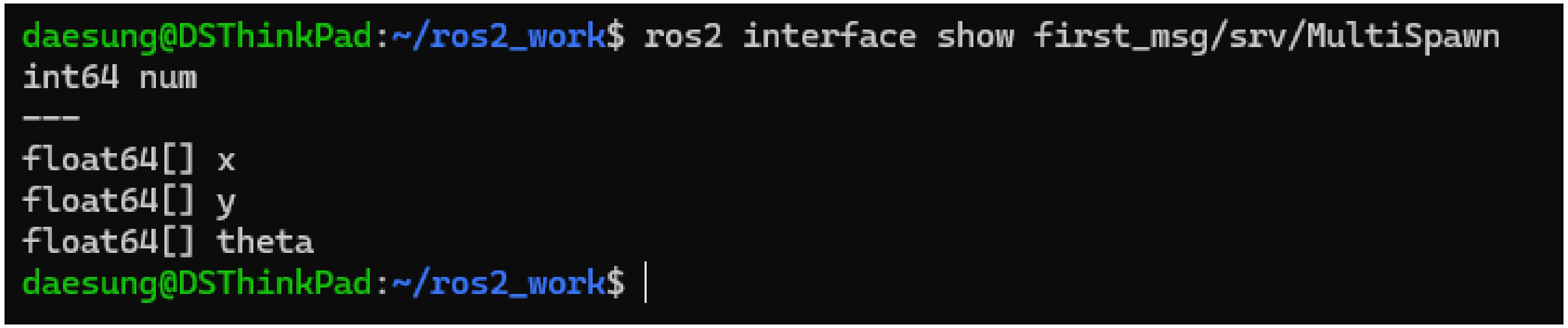
```
daesung@DSThinkPad:~/ros2_work$ colbds first_msg
Starting >>> first_msg
Finished <<< first_msg [0.47s]

Summary: 1 package finished [0.75s]
daesung@DSThinkPad:~/ros2_work$ 
```

alias로 ~/.bashrc에 추가



```
alias colbd="colcon build"
alias colbds="colcon build --packages-select"
```

- ros2 interface show 명령어를 통해 Service data type이 등록되었는지 확인

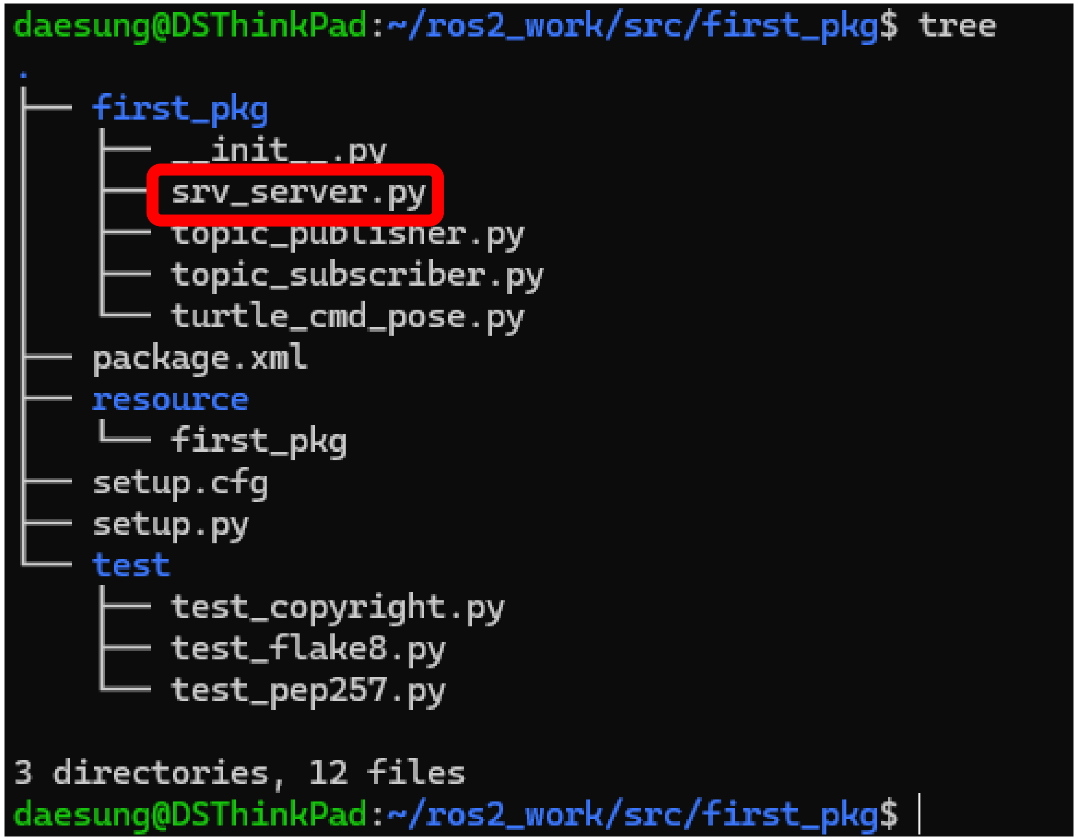


```
daesung@DSThinkPad:~/ros2_work$ ros2 interface show first_msg/srv/MultiSpawn
int64 num
---
float64[] x
float64[] y
float64[] theta
daesung@DSThinkPad:~/ros2_work$ 
```

Service definition

Make a service server

- Service Server는 앞서 사용했던 pkg폴더에 Python파일을 추가로 생성



```
daesung@DSThinkPad:~/ros2_work/src/first_pkg$ tree
.
├── first_pkg
│   ├── __init__.py
│   ├── srv_server.py
│   ├── topic_publisher.py
│   ├── topic_subscriber.py
│   └── turtle_cmd_pose.py
├── package.xml
├── resource
│   └── first_pkg
├── setup.cfg
├── setup.py
└── test
    ├── test_copyright.py
    ├── test_flake8.py
    └── test_pep257.py

3 directories, 12 files
daesung@DSThinkPad:~/ros2_work/src/first_pkg$
```

1 Service definition

❖ Make a service server

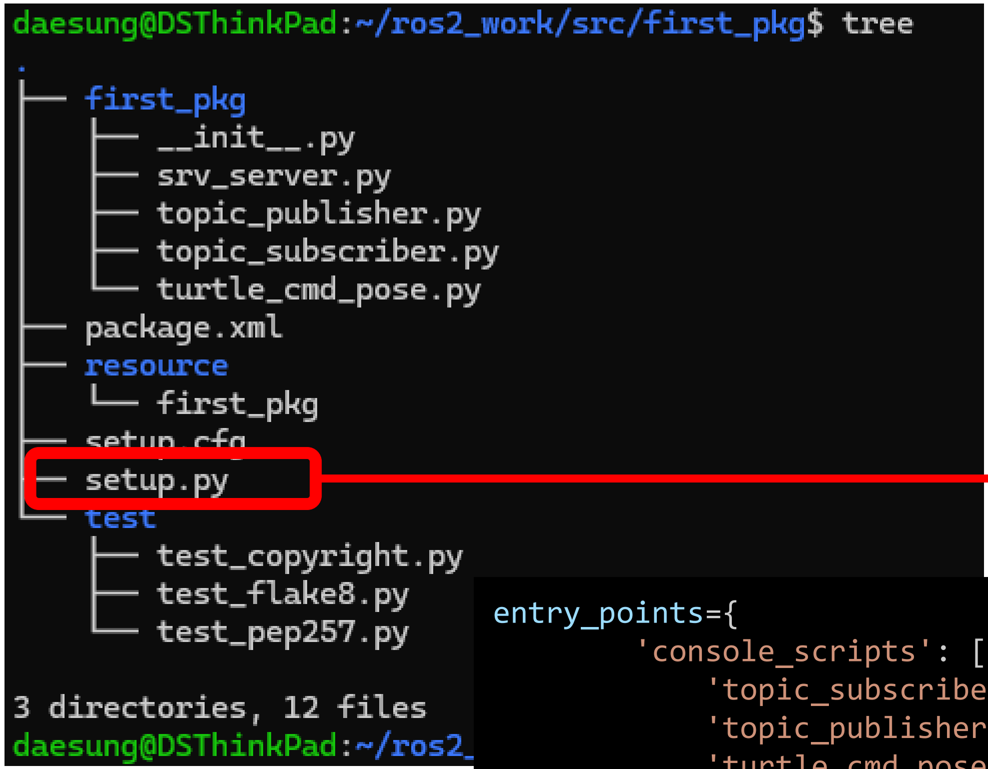
- srv_server.py

```python
import rclpy as rp
from rclpy.node import Node
from first_msg.srv import MultiSpawn

class Multi_spawn(Node):
    def __init__(self):
        super().__init__('Multi_spawn_node')
        self.server = self.create_service(MultiSpawn, 'Multi_spawn_node', self.callback_srv_server)

    def callback_srv_server(self, req, res):
        print('Req : ', req)
        res.x = [1., 2., 3.]
        res.y = [10., 20.]
        res.theta = [100., 200., 300.]
        return res

def main(args=None):
    rp.init(args=args)
    mlt_spn = Multi_spawn()
    rp.spin(mlt_spn)
    rp.shutdown()

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

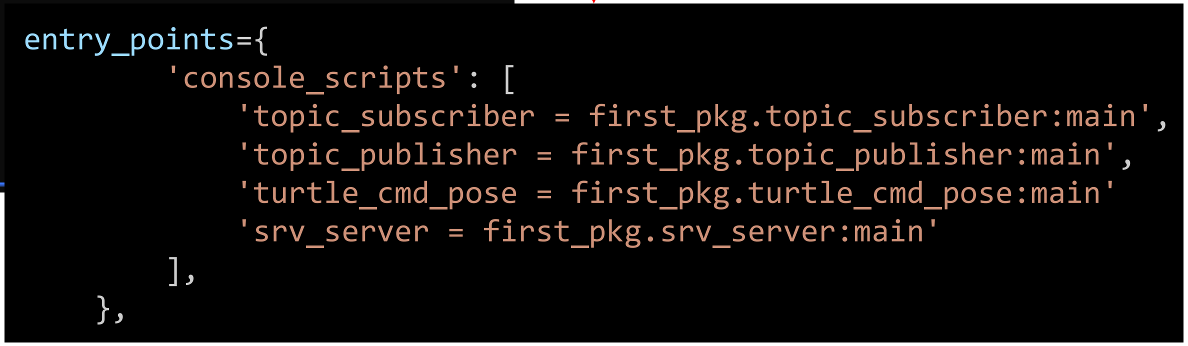
1 Service definition

❖ Make a service server

- 다음과 같이 setpup.py에 작성한 파일의 메인 함수 추가



```
daesung@DSThinkPad:~/ros2_work/src/first_pkg$ tree
.
├── first_pkg
│   ├── __init__.py
│   ├── srv_server.py
│   ├── topic_publisher.py
│   ├── topic_subscriber.py
│   └── turtle_cmd_pose.py
├── package.xml
├── resource
│   └── first_pkg
├── setup.cfg
├── setup.py
└── test
    ├── test_copyright.py
    ├── test_flake8.py
    └── test_pep257.py

3 directories, 12 files
daesung@DSThinkPad:~/ros2_
```

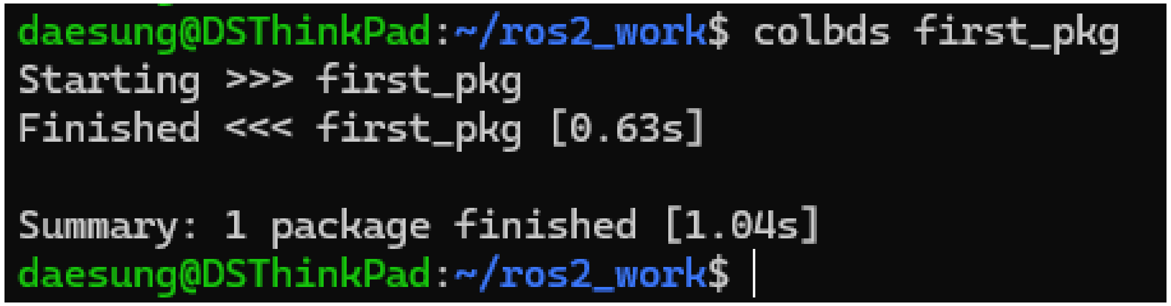


```
entry_points={
        'console_scripts': [
            'topic_subscriber = first_pkg.topic_subscriber:main',
            'topic_publisher = first_pkg.topic_publisher:main',
            'turtle_cmd_pose = first_pkg.turtle_cmd_pose:main'
            'srv_server = first_pkg.srv_server:main'
        ],
    },
```

1 Service definition

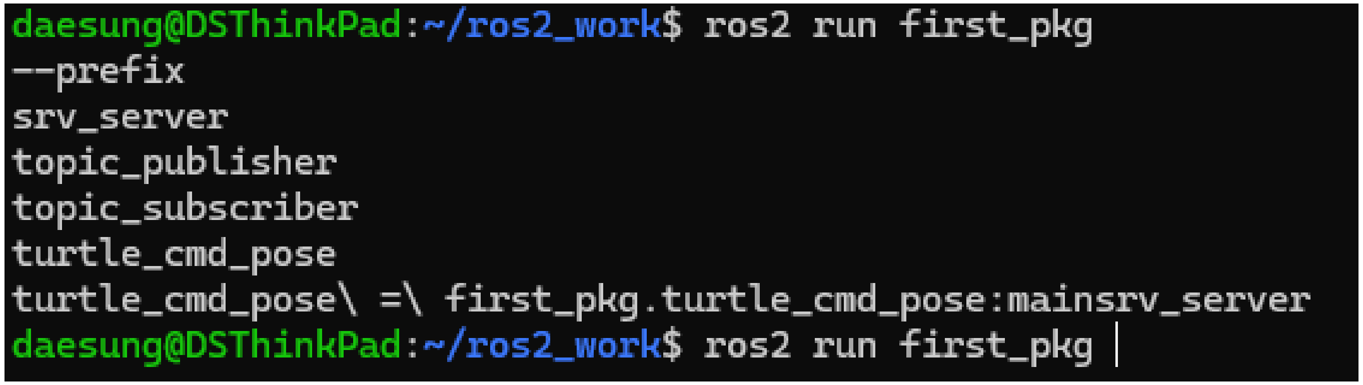
❖ Make a service server

- Package 빌드 후 서비스가 나타나는지 확인



```
daesung@DSThinkPad:~/ros2_work$ colbds first_pkg
Starting >>> first_pkg
Finished <<< first_pkg [0.63s]

Summary: 1 package finished [1.04s]
daesung@DSThinkPad:~/ros2_work$
```

- 서비스 실행 전 다음과 같이 패키지의 노드 확인

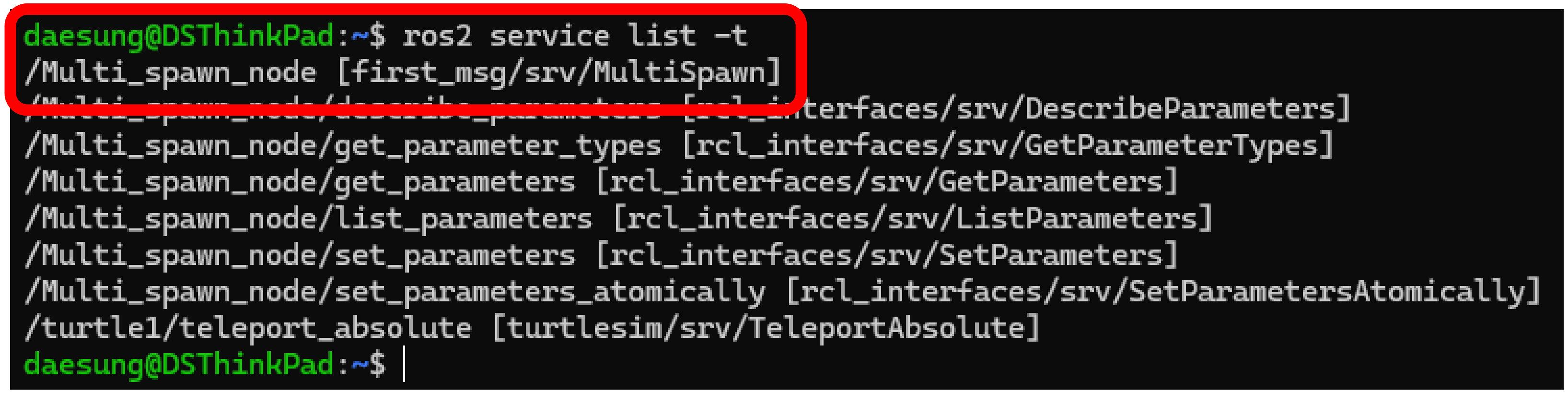


```
daesung@DSThinkPad:~/ros2_work$ ros2 run first_pkg
--prefix
srv_server
topic_publisher
topic_subscriber
turtle_cmd_pose
turtle_cmd_pose\ =\ first_pkg.turtle_cmd_pose:mainsrv_server
daesung@DSThinkPad:~/ros2_work$ ros2 run first_pkg
```

1 Service definition

Make a service server

- Service server 실행 후 Service list에 나타나는지 확인



```
daesung@DSThinkPad:~$ ros2 service list -t
/Multi_spawn_node [first_msg/srv/MultiSpawn]
/Multi_spawn_node/describe_parameters [rcl_interfaces/srv/DescribeParameters]
/Multi_spawn_node/get_parameter_types [rcl_interfaces/srv/GetParameterTypes]
/Multi_spawn_node/get_parameters [rcl_interfaces/srv/GetParameters]
/Multi_spawn_node/list_parameters [rcl_interfaces/srv/ListParameters]
/Multi_spawn_node/set_parameters [rcl_interfaces/srv/SetParameters]
/Multi_spawn_node/set_parameters_atomically [rcl_interfaces/srv/SetParametersAtomically]
/turtle1/teleport_absolute [turtlesim/srv/TeleportAbsolute]
daesung@DSThinkPad:~$
```

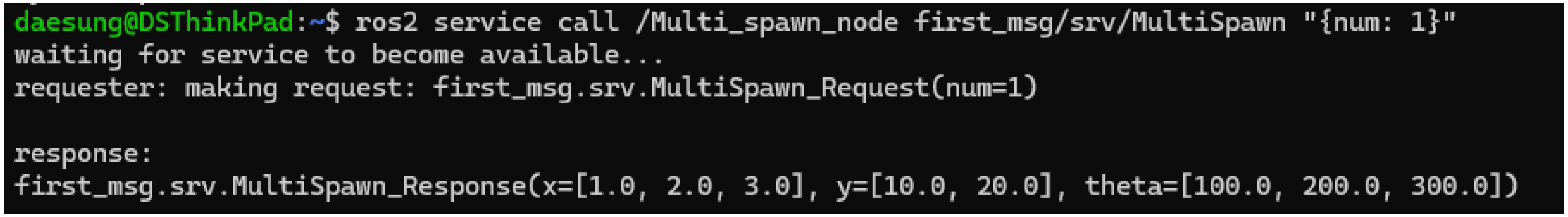
1 Service definition

Make a service server

- 다음과 같은 service call 호출

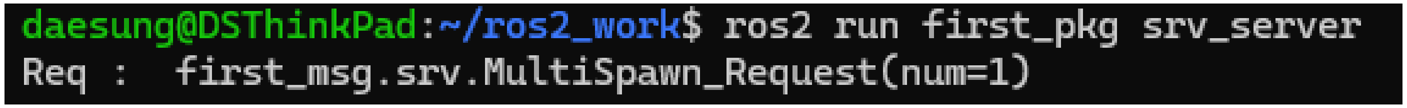
```
ros2 service call /Multi_spawn_node first_msg/srv/MultiSpawn "{num: 1}"
```

Service client



```
daesung@DSThinkPad:~$ ros2 service call /Multi_spawn_node first_msg/srv/MultiSpawn "{num: 1}"
waiting for service to become available...
requester: making request: first_msg.srv.MultiSpawn_Request(num=1)

response:
first_msg.srv.MultiSpawn_Response(x=[1.0, 2.0, 3.0], y=[10.0, 20.0], theta=[100.0, 200.0, 300.0])
```

srv_server



```
daesung@DSThinkPad:~/ros2_work$ ros2 run first_pkg srv_server
Req :  first_msg.srv.MultiSpawn_Request(num=1)
```

1

Service definition

❖ Make a service client

- srv_server.py를 다음과 같이 수정

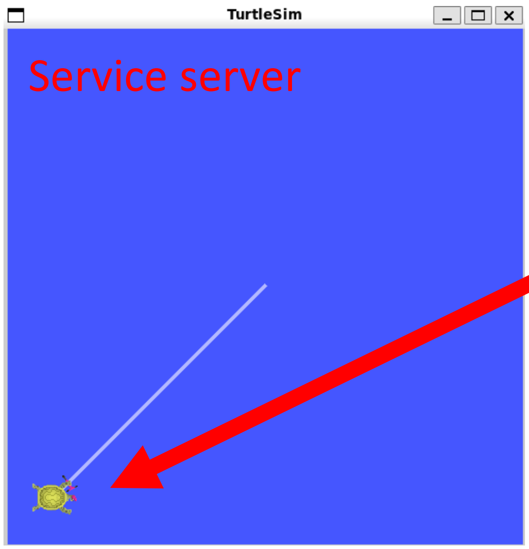
```python
import rclpy as rp
from rclpy.node import Node
from first_msg.srv import MultiSpawn
from turtlesim.srv import TeleportAbsolute

class Multi_spawn(Node):
    def __init__(self):
        super().__init__('Multi_spawn_node')
        self.server = self.create_service(MultiSpawn, 'Multi_spawn_node', self.callback_srv_server)
        self.teleport = self.create_client(TeleportAbsolute, '/turtle1/teleport_absolute')
        self.req_teleabs = TeleportAbsolute.Request()

    def callback_srv_server(self, req, res):
        self.req_teleabs.x = 1.
        self.req_teleabs.y = 1.
        self.req_teleabs.theta = 0.
        self.teleport.call_async(self.req_teleabs)
        return res

def main(args=None):
    rp.init(args=args)
    mlt_spn = Multi_spawn()
    rp.spin(mlt_spn)
    rp.shutdown()

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

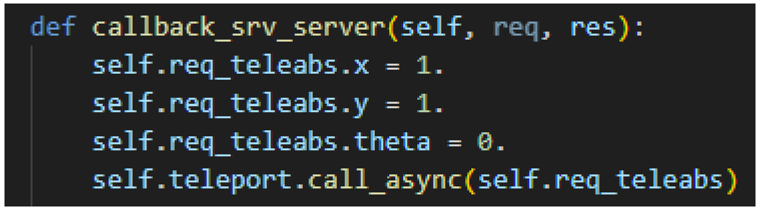
1

Service definition

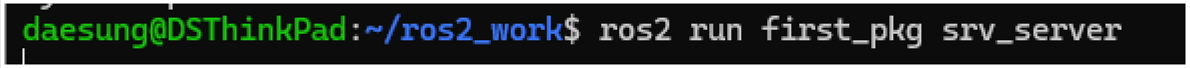
❖ Make a service client

- Package 빌드 후 실행 확인





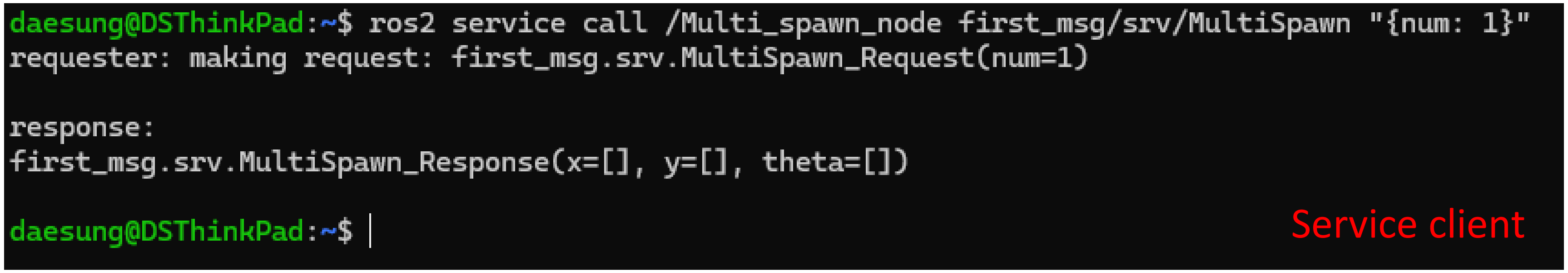
```
def callback_srv_server(self, req, res):
    self.req_teleabs.x = 1.
    self.req_teleabs.y = 1.
    self.req_teleabs.theta = 0.
    self.teleport.call_async(self.req_teleabs)
```



```
daesung@DSThinkPad:~/ros2_work$ ros2 run first_pkg srv_server
```

Service client

Service server



```
daesung@DSThinkPad:~$ ros2 service call /Multi_spawn_node first_msg/srv/MultiSpawn "{num: 1}"
requester: making request: first_msg.srv.MultiSpawn_Request(num=1)

response:
first_msg.srv.MultiSpawn_Response(x=[], y=[], theta=[])

daesung@DSThinkPad:~$ 
```

Service client

1 Service definition

❖ Mission

- 원하는 위치에 원하는 만큼 거북이를 spawn할 수 있는 코드를 작성 하시오.

1 Service definition

- Hint

```python
import rclpy as rp
from rclpy.node import Node
from first_msg.srv import MultiSpawn
from turtlesim.srv import TeleportAbsolute

class Multi_spawn(Node):
    def __init__(self):
        super().__init__('Multi_spawn_node')
        self.server = self.create_service(MultiSpawn, 'Multi_spawn_node', self.callback_srv_server)
        self.teleport = self.create_client(TeleportAbsolute, '/turtle1/teleport_absolute')
        self.req_teleabs = TeleportAbsolute.Request()

    def callback_srv_server(self, req, res):
        self.num = req.num
        print(self.num)
        res.x.append(3.)
        res.y.append(4.)
        res.theta.append(0.)
        self.req_teleabs.x = 1.
        self.req_teleabs.y = 1.
        self.req_teleabs.theta = 0.
        self.teleport.call_async(self.req_teleabs)
        return res

def main(args=None):
    rp.init(args=args)
    mlt_spn = Multi_spawn()
    rp.spin(mlt_spn)
    rp.shutdown()

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

Q & A