

패키지 생성

package

```
$ cd ~/Workspaces/ros2_ws/src
$ ros2 pkg create simple_service_pkg --build-type
ament_python --dependencies rclpy std_msgs
```

서비스 인터페이스 std_msgs/srv/SetBool

SetBool.srv

```
# Request
bool data # e.g. for hardware enabling / disabling
---
# Response
bool success # indicate successful run of triggered
service
string message # informational, e.g. for error messages
```

서비스 서버

service_server.py

src/simple_service_pkg/simple_service_pkg/service_server.py

```
import rclpy
from rclpy.node import Node
from std_srvs.srv import SetBool
```

class PowerServer(Node):

```
def __init__(self):
    super().__init__('service_server_node')
    self.srv = self.create_service(SetBool, 'power_service',
self.power_cb)
    self.get_logger().info('Service Server Running...')
```

def power_cb(self, request, response):

```
self.get_logger().info('Request Received... ')
```

```
if request.data:
    response.success = True
    response.message = 'Power On'
elif not request.data:
    response.success = True
    response.message = 'Power Off'
else:
    response.success = False
    response.message = 'Error'
print(request)
print(response)
return response
```

def main(args=None):

```
rclpy.init(args=args)
node = PowerServer()
```

```
try:
    rclpy.spin(node)
except KeyboardInterrupt:
    node.get_logger().info('Keyboard Interrupt')
finally:
    node.destroy_node()
    rclpy.shutdown()
```

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

서비스 클라이언트

service_client.py

src/simple_service_pkg/simple_service_pkg/service_client.py

```
import rclpy
from rclpy.node import Node
from std_srvs.srv import SetBool
```

class PowerClient(Node):

def __init__(self):

```
    super().__init__('service_client_node')
    self.client = self.create_client(SetBool,
'power_service')
    self.req = SetBool.Request()
    self.get_logger().info('Service Client Start')
```

def send_request(self, user_input):

```
self.req.data = (user_input.lower() == 'on')
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 = PowerClient()
```

```
try:
```

```
    #pass
```

```
    user_input = input('Enter an power "on" or "off" : ')
    res = node.send_request(user_input)
    node.get_logger().info('Server returned: ' +
```

```
res.message)
```

```
except KeyboardInterrupt:
```

```
    node.get_logger().info('Keyboard Interrupt')
```

```
finally:
```

```
    node.destroy_node()
    rclpy.shutdown()
```

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

setup.py

```
'console_scripts': [
```

```
    'service_server = simple_service_pkg.service_server:main',
    'service_client = simple_service_pkg.service_client:main'
```

```
],
```

빌드

build & run

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

노드 실행(ros2 run)

```
$ ros2 pkg executables simple_service_pkg
$ ros2 run simple_service_pkg service_server
$ ros2 run simple_service_pkg service_client
```