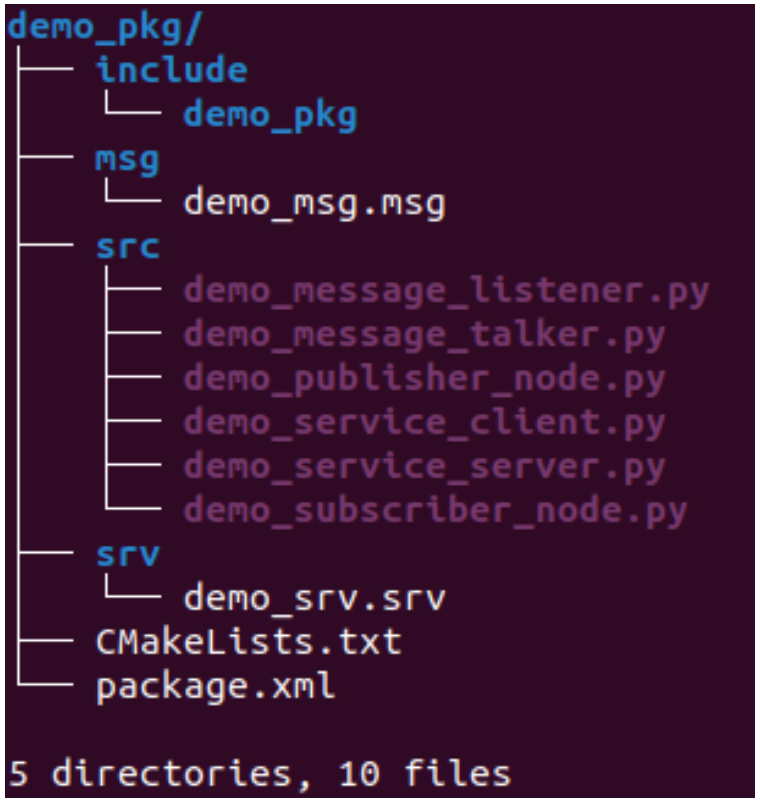
Robot Operating System

Dive into ROS

ROS demo_package

- Node
- Message
- Service



Creating a publisher node

Step 1:

Create node

Create executable **demo_publisher_node.py** file

Creating a publisher node

Step 2:

Import necessary modules

```
import rospy
from std_msgs.msg import String
```

Creating a publisher node

Step 3:

Create a publisher function

```
def publisher_fun():
   rospy.init_node('node_name')
   pub = rospy.Publisher('topic_name', topic_data_class, queue_size)
   message = 'some string'
   rate = rospy.Rate(10)
   rospy.loginfo('something')
   while not rospy.is_shutdown():
       pub.publish(message)
       rate.sleep()
```

Creating a publisher node

Step 4:

Create final python module

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

Creating a subscriber node

Step 1:

Create node

Create executable **demo_subscriber_node.py** file

Creating a subscriber node

Step 2:

Import necessary modules

```
import rospy
from std_msgs.msg import String
```

Creating a subscriber node

Step 3:

Create a subscriber function

```
def subscriber_fun():
    rospy.init_node('node_name')
    subs = rospy.Subscriber('topic_name', topic_data_class, callback)
    rospy.spine()
```

Creating a subscriber node

Step 4:

Create a callback function

```
def callback_fun(msg):
    Rospy.loginfo(‘Message received-->’ + msg.data)
```

Creating a subscriber node

Step 5:

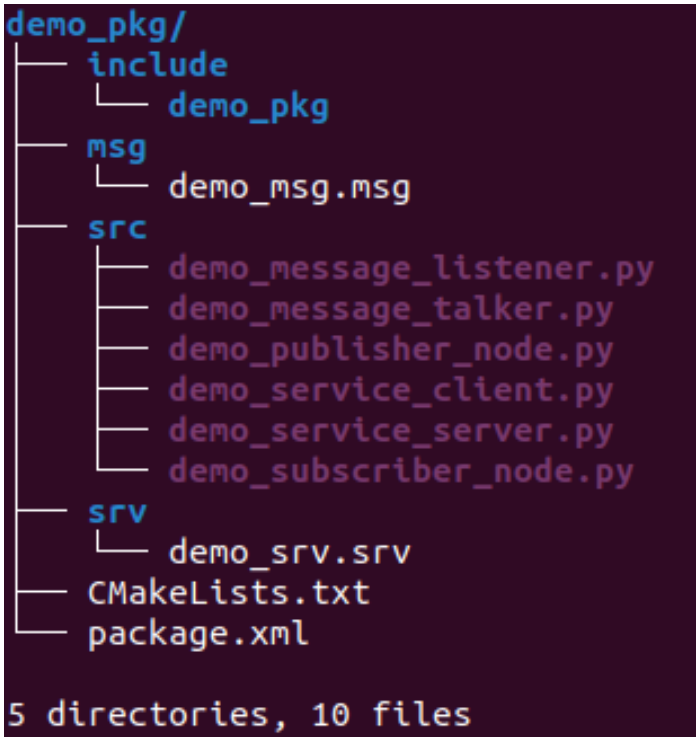
Create final python module

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

Creating custom Message

Step 1:

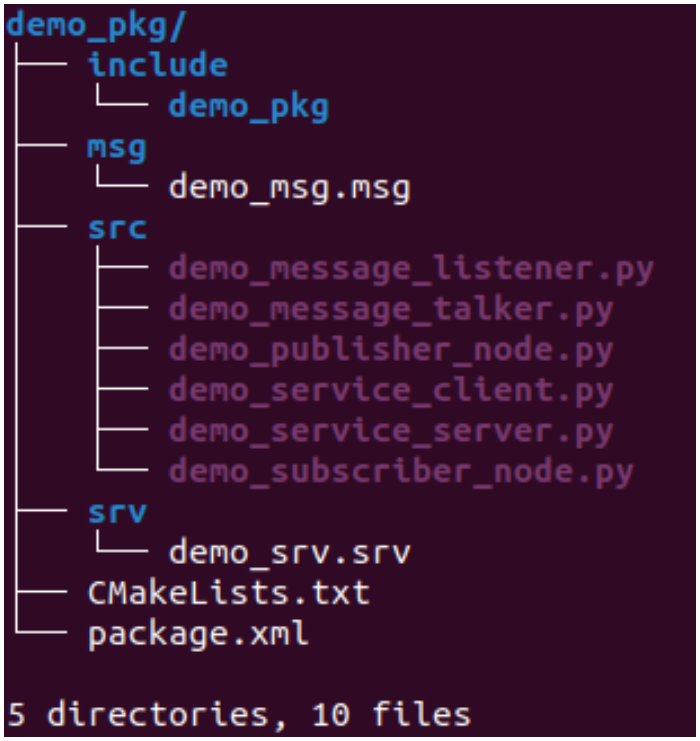
Create msg/ directory in package folder



Creating custom Message

Step 2:

Create **demo_msg.msg** file



Creating custom Message

Step 3:

Specify message type and name in demo_msg.msg file

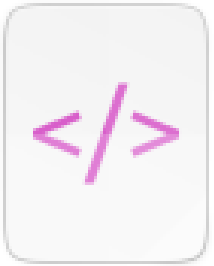
```
# msg type/msg name

string greeting

int32 number
```

Creating custom Message

Step 4:

Modify CMakeLists.txt and package.xml

CMakeLists.txt

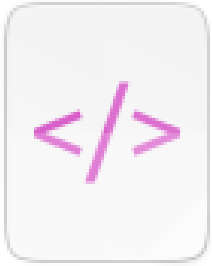


package.xml

Creating custom Message

package.xml

Add these item at the end of package.xml file

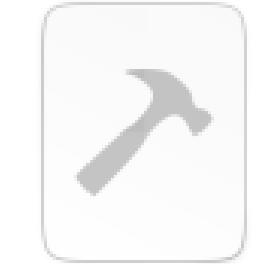


```
<build_depend>message_generation</build_depend>
<exec_depend>message_runtime</exec_depend>
```

```
50   <!--   <doc_depend>doxygen</doc_depend> -->
51   <buildtool_depend>catkin</buildtool_depend>
52   <build_depend>roscpp</build_depend>
53   <build_depend>rospy</build_depend>
54   <build_depend>std_msgs</build_depend>
55   <build_export_depend>roscpp</build_export_depend>
56   <build_export_depend>rospy</build_export_depend>
57   <build_export_depend>std_msgs</build_export_depend>
58   <exec_depend>roscpp</exec_depend>
59   <exec_depend>rospy</exec_depend>
60   <exec_depend>std_msgs</exec_depend>
61   <build_depend>message_generation</build_depend>
62   <exec_depend>message_runtime</exec_depend>
63
64
65   <!-- The export tag contains other, unspecified, tags -->
66   <export>
67     <!-- Other tools can request additional information be placed here -->
68
69   </export>
70 </package>
```

Creating custom Message

CMakeLists.txt



- Add message_generation in find package section:

```
find_package(catkin REQUIRED COMPONENTS
  roscpp
  rospy
  std_msgs
  message_generation
)
```

- Uncomment the following line and add the custom message file:

```
add_message_files(
   FILES
   demo_msg.msg
#   Message2.msg
)

generate_messages(
   DEPENDENCIES
   std_msgs
 )
```

```
$ cd ~/catkin_ws/
$ catkin_make
```

Creating custom Message

Check is the message is compiled cirrectly

```
(base) elnaz@Eli-Ubuntu:~$ rosmsg show demo_pkg/demo_msg
string greeting
int32 number
```

Creating publisher using demo_msg

Step 1:

Create node

```
Create executable demo_message_talker.py file
```

Creating publisher using demo_msg

Step 2:

Import necessary modules

```
import rospy
from demo_pckg.msg import demo_msg
```

Creating publisher using demo_msg

Step 3:

Creating send message function

```
def send_msg():
    rospy.init_node('demo_message_talker')
    greeting_msg = 'Hello ROS developer!'
    pub = rospy.Publisher('greeting_number', demo_msg, queue_size=10)
    rate = rospy.Rate(10)
    my_msg = demo_msg()
    while not rospy.is_shutdown():
        my_msg.number = 2021
        my_msg.greeting = greeting_msg
        rospy.loginfo('Node is sending message...')
        pub.publish(my_msg)
        rate.sleep()
```

Creating publisher using demo_msg

Step 4:

Create final python module

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

Creating subscriber using demo_msg

Step 1:

Create node

```
Create executable demo_message_listener.py file
```

Creating subscriber using demo_msg

Step 2:

Import necessary modules

```
import rospy
from demo_pckg.msg import demo_msg
```

Creating subscriber using demo_msg

Step 3:

Create a subscriber function

```
def listener():
    rospy.init_node('demo_message_listener')
    subs = rospy.Subscriber('greeting_number', demo_msg, callback=callback_fun)
```

Creating subscriber using demo_msg

Step 4:

Create a callback function

```
def callback_fun(msg):
    number_msg = msg.number
    greeting_msg = msg.greeting
    rospy.loginfo(f'{greeting_msg} , the recieved number is : {number_msg}')
```

Creating subscriber using demo_msg

Step 5:

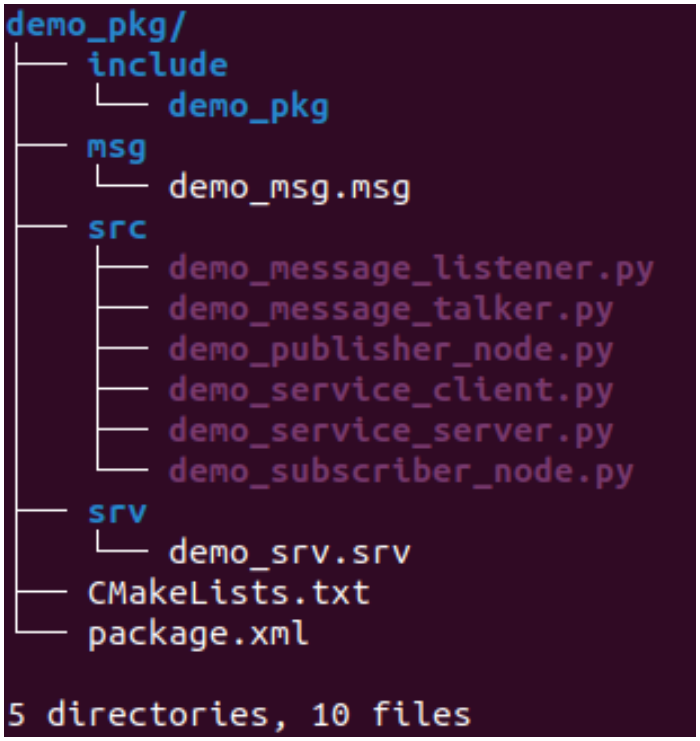
Create final python module

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

Creating custom Service

Step 1:

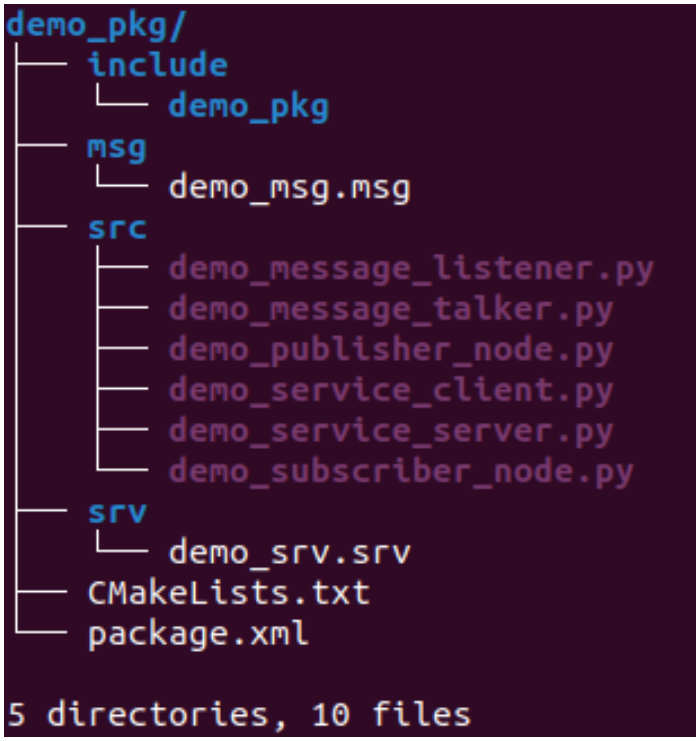
Create srv/ directory in package folder



Creating custom Service

Step 2:

Create **demo_srv.srv** file



Creating custom Service

Step 3:

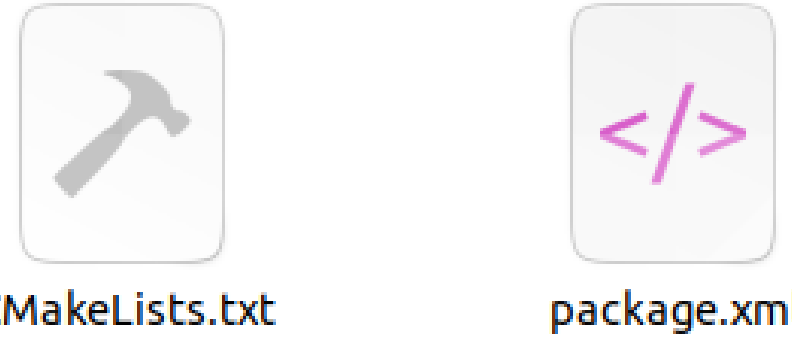
Specify Request and Response message type and name in demo_srv.srv file

```
# Request msg type
string in_name
---
# Response msg type
string out_greeting
```

Creating custom Service

Step 4:

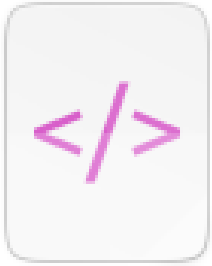
Modify CMakeLists.txt and package.xml



Creating custom Service

package.xml

Add these item at the end of package.xml file



```
<build_depend>message_generation</build_depend>
<exec_depend>message_runtime</exec_depend>
```

```
50  <!--   <doc_depend>doxygen</doc_depend> -->
51  <buildtool_depend>catkin</buildtool_depend>
52  <build_depend>roscpp</build_depend>
53  <build_depend>rospy</build_depend>
54  <build_depend>std_msgs</build_depend>
55  <build_export_depend>roscpp</build_export_depend>
56  <build_export_depend>rospy</build_export_depend>
57  <build_export_depend>std_msgs</build_export_depend>
58  <exec_depend>roscpp</exec_depend>
59  <exec_depend>rospy</exec_depend>
60  <exec_depend>std_msgs</exec_depend>
61  <build_depend>message_generation</build_depend>
62  <exec_depend>message_runtime</exec_depend>
63
64
65  <!-- The export tag contains other, unspecified, tags -->
66  <export>
67    <!-- Other tools can request additional information be placed here -->
68
69  </export>
70 </package>
```

Creating custom Service

CMakeLists.txt

- Add message_runtime in catkin_package() :

```
  CATKIN_DEPENDS roscpp rospy std_msgs
  message_runtime
#  DEPENDS system_lib
)
```

- Uncomment the following line and add the custom service file:

```
 add_service_files(
   FILES
   demo_srv.srv
#   Service2.srv
)
```

```
$ cd ~/catkin_ws/
$ catkin_make
```

Creating custom Service

Check is the service is compiled cirrectly

```
(base) elnaz@Eli-Ubuntu:~$ rossrv show demo_pkg/demo_srv
string in_name
---
string out_greeting
```

Creating demo_srv Server

Step 1:

Create node

```
Create executable demo_service_server.py file
```

Creating demo_srv Server

Step 2:

Import necessary modules

```
import rospy
from demo_pckg.srv import demo_srv
```

Creating demo_srv Server

Step 3:

Creating server function

```
def say_hello_server():
    rospy.init_node("demo_service_server")
    server = rospy.Service('greeting_service', demo_srv, handler=say_hello)
    print("Server is ready to say greeting :)")
    rospy.spin()
```

Creating demo_srv Server

Step 4:

Create handler function

```
def say_hello(req):
    resp = 'Hello dear ' + '"' + req.in_name + '"'
    print(resp)
    return (resp)
```

Creating demo_srv Server

Step 5:

Create final python module

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

Creating demo_srv Client

Step 1:

Create node

```
Create executable demo_service_client.py file
```

Creating demo_srv Client

Step 2:

Import necessary modules

```
import rospy
import sys
from demo_pckg.srv import demo_srv
```

Creating demo_srv Client

Step 3:

Creating send request function

```
def send_name(in_name):
    rospy.wait_for_service('greeting_service')
    client_greeting = rospy.ServiceProxy('greeting_service', demo_srv)
    response = client_greeting(in_name)
    print(f"Server responded ---> {response.out_greeting}")
```

Creating demo_srv Client

Step 5:

Create final python module and define input

```
if __name__ == "__main__":
    in_name = str(sys.argv[1])
    send_name(in_name)
```


ROS
NOETIC
NINJEMYS