



ROS 2™

آشنایی با ROS2

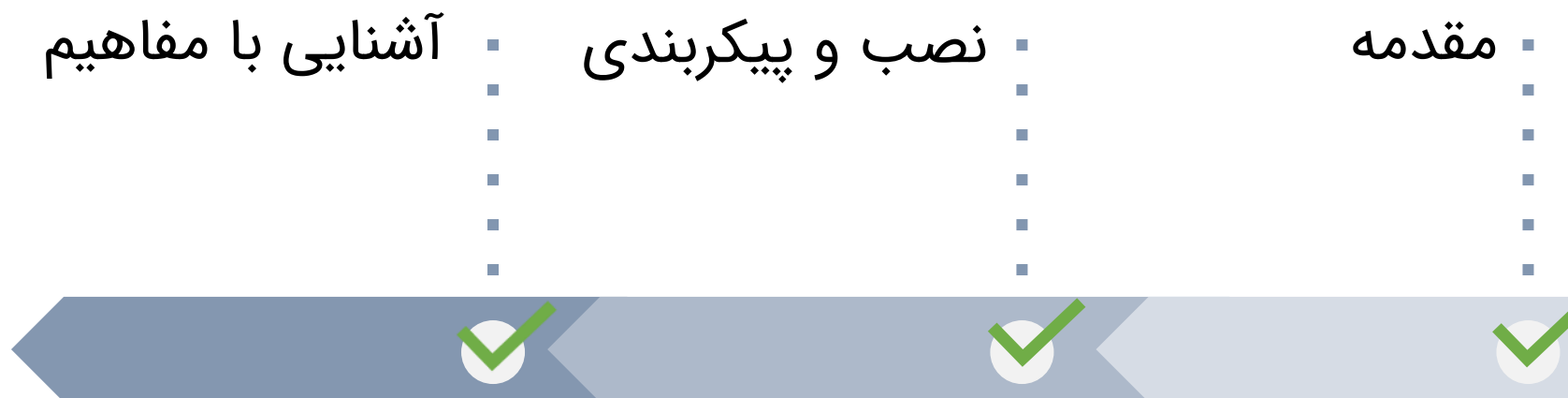


جلسه دوم : پیاده سازی و Customize کردن





آنچه تاکنون آموختیم:





آنچه امروز می‌آموزیم:

Custom msg

نوشتن اولین Node

پی‌کربندی WS

Sub-Pub

ساخت پکیج

Client-Server



پیکربندی WS



Setup env

```
$ source /opt/ros/iron/setup.bash
```



Make ws directory

```
$ mkdir -p ~/ros2_ws/src
```

```
$ cd ~/ros2_ws/src
```



ROS 2™

پیکربندی WS



Clone a repo

```
$ git clone https://github.com/ros/ros_tutorials.git -b iron
```



پیکربندی WS



Resolve dependencies

make sure that you are in ros2_ws director

```
$ cd ..
```

```
$ rosdep install -i --from-path src --rosdistro iron -y
```



ROS 2™

پیکربندی WS



Build the workspace with colcon

```
$ colcon build
```



پیکربندی WS



```
Build the workspace with colcon  
$ colcon build
```

```
ros@Ubuntu22:~/ros2_ws_tutorial$ tree -dx  
.  
├── build  
├── install  
├── log  
└── src  
  
4 directories
```




پیکربندی WS

Source the overlay

```
# underlay
$ source /opt/ros/iron/setup.bash

$ cd ~/ros2_ws

#overlay
$ source install/local_setup.bash
```



ساخت پکیج

پیکربندی WS

Create a package

```
$ cd ~/ros2_ws/src
```



ساخت پکیج

پیکربندی WS

Create a package

```
$ ros2 pkg create --build-type ament_python --license Apache-2.0 --node-name <hello_word_node> <package_name>
```



ساخت پکیج

پیکربندی WS



Build workspace

```
$ cd ~/ros2_ws
```

```
$ colcon build
```



ساخت پکیج



Build workspace

```
$ cd ~/ros2_ws
```

```
$ colcon build
```



Single package build

```
$ colcon build --packages-select <package_name>
```

پیکربندی WS



ساخت پکیج

پیکربندی WS



Source setup file

```
$ source install/setup.bash
```



ساخت پکیج

پیکربندی WS



Use package

```
$ ros2 run <package_name> <node_name>
```



نوشتن اولین Node

ROS 2™

```

Publisher Node

import rclpy
from rclpy.node import Node

from std_msgs.msg import String

class Robot(Node):

    def __init__(self):
        super().__init__('robot')
        self.publisher_ = self.create_publisher(String, 'joint', 10)
        timer_period = 0.5 # seconds
        self.timer = self.create_timer(timer_period, self.timer_callback)
        self.i = 0

    def timer_callback(self):
        msg = String()
        msg.data = 'Joint value is : %d' % self.i
        self.publisher_.publish(msg)
        self.get_logger().info('Publishing: "%s"' % msg.data)
        self.i += 1

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

    robot_joint_pub = Robot()

    rclpy.spin(robot_joint_pub)

    # Destroy the node explicitly
    # (optional - otherwise it will be done automatically
    # when the garbage collector destroys the node object)
    robot_joint_pub.destroy_node()
    rclpy.shutdown()

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

پیکربندی WS

ساخت پکیج

Publisher



نوشتن اولین Node

پیگیری WS

ساخت پکیج

Publisher



Add dependencies

```
<!-- ros2_ws/src/demp_pkg/package.xml -->  
<exec_depend>rclpy</exec_depend>  
<exec_depend>std_msgs</exec_depend>
```



نوشتن اولین Node

پی‌کربندی WS

ساخت پکیج

Publisher

```
Add an entry point

# in ros2_ws/src/demp_pkg/setup.py file

entry_points={
    'console_scripts': [
        'robot = demp_pkg.robot:main',
    ],
},
```



ROS 2™

نوشتن اولین Node

پی‌کرند WS

ساخت پکیج

Publisher



Build and run

```
# in ~/ros2_ws
```

```
$ rosdep install -i --from-path src --rosdistro iron -y
```



نوشتن اولین Node

پی‌کربندی WS

ساخت پکیج

Publisher



Build the package

```
# in ~/ros2_ws directory
```

```
$ colcon build --packages-select demo_pkg
```



ROS 2™

نوشتن اولین Node

پی‌کرند WS

ساخت پکیج

Publisher



Source setup file

```
# in ~/ros2_ws
```

```
$ source install/setup.bash
```



نوشتن اولین Node

ROS 2™

```
Subscriber

import rclpy
from rclpy.node import Node

from std_msgs.msg import String

class Controller(Node):

    def __init__(self):
        super().__init__('controller')
        self.subscription = self.create_subscription(
            String,
            'joint',
            self.listener_callback,
            10)
        self.subscription # prevent unused variable warning

    def listener_callback(self, msg):
        self.get_logger().info('The recieved joint value: "%s"' % msg.data)

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

    robot_controller = Controller()

    rclpy.spin(robot_controller)

    # Destroy the node explicitly
    # (optional - otherwise it will be done automatically
    # when the garbage collector destroys the node object)
    robot_controller.destroy_node()
    rclpy.shutdown()

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

پی‌کربندی WS

ساخت پکیج

Publisher

Subscriber



نوشتن اولین Node

و دوباره سه قدم برای تعریف node جدید

```
3 Steps - 1 setup.py file

# modify ~/ros2_ws/src/demo_pkg/setup.py file

entry_points={

    'console_scripts': [

        'my_node = demo_pkg.my_node:main',

        'robot = demo_pkg.robot:main',

        'controller = demo_pkg.controller:main'

    ],

},
```

پیگیری WS

ساخت پکیج

Publisher

Subscriber



نوشتن اولین Node

و دوباره سه قدم برای تعریف node جدید

پیگیری WS

ساخت پکیج

Publisher

Subscriber



3 Steps - 2 colcon build

```
# in ~/ros2_ws directory
```

```
$ colcon build --packages-select demo_pkg
```




نوشتن اولین Node

و دوباره سه قدم برای تعریف node جدید

پیگیری WS

ساخت پکیج

Publisher

Subscriber



3 Steps - 3 source setup.bash

```
# in ~/ros2_ws directory
```

```
$ source install/setup.bash
```



ROS 2™ نوشتن Node – Service Server

تعریف یک پکیج جدید برای استفاده از
سرویس‌های از پیش نوشته شده

پیگیری WS

ساخت پکیج

Pub - Sub

Server



Create a new package

```
# in ~/ros2_ws/src directory
```

```
$ ros2 pkg create --build-type ament_python --license Apache-2.0 py_srv_demo  
--dependencies rclpy example_interfaces
```



ROS 2™ نوشتن Node – Service Server

نمایش نوع سرویس انتخاب شده

پیگیری WS

ساخت پکیج

Pub - Sub

Server



Create a new package

```
$ ros2 interface show example_interfaces/srv/AddTwoInts
```



ROS 2 :: نوشتن Service Server Node

```
Create a new node for server

from example_interfaces.srv import AddTwoInts

import rclpy
from rclpy.node import Node

class MyService(Node):

    def __init__(self):
        super().__init__('minimal_service')
        self.srv = self.create_service(AddTwoInts, 'add_two_ints',
self.add_two_ints_callback)

    def add_two_ints_callback(self, request, response):
        response.sum = request.a + request.b
        self.get_logger().info('Incoming request\na: %d b: %d' % (request.a, request.b))

        return response

def main():
    rclpy.init()

    my_service = MyService()

    rclpy.spin(my_service)

    rclpy.shutdown()

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

پیکربندی WS

ساخت پکیج

Pub - Sub

Server



ROS 2™ نوشتن Service Server – Node

تعریف Node در setup.py

```
Define node in setup.py file

# in ~/ros2_ws_src/py_srv_demo/setup.py

entry_points={

    'console_scripts': [

        'demo_server = py_srv_demo.demo_server:main',

    ],

},
```

پیکربندی WS

ساخت پکیج

Pub - Sub

Server



ROS 2™ نوشتن Node – Service Server

چک کردن dependency ها

پیکربندی WS

ساخت پکیج

Pub - Sub

Server



check for missing dependencies

```
# in ~/ros2_ws
```

```
$ rosdep install -i --from-path src --rosdistro iron -y
```



ROS 2™ نوشتن Service Server – Node

پیکربندی WS

ساخت پکیج

Pub - Sub
Server



build package

```
# in ~/ros2_ws
```

```
$ colcon build --packages-select py_srv_demo
```



ROS 2™ نوشتن Service Server – Node

پیکربندی WS

ساخت پکیج

Pub - Sub
Server



Source setup.bash

```
# in ~/ros2_ws
```

```
$ source install/setup.bash
```




ROS 2 :: نوشتن Service Client – Node

```
Client Node

import sys

from example_interfaces.srv import AddTwoInts
import rclpy
from rclpy.node import Node

class MyClient(Node):

    def __init__(self):
        super().__init__('minimal_client')
        self.cli = self.create_client(AddTwoInts, 'add_two_ints')
        while not self.cli.wait_for_service(timeout_sec=1.0):
            self.get_logger().info('service not available, waiting again...')
        self.req = AddTwoInts.Request()

    def send_request(self, a, b):
        self.req.a = a
        self.req.b = b
        return self.cli.call_async(self.req)

def main():
    rclpy.init()

    minimal_client = MyClient()
    future = minimal_client.send_request(int(sys.argv[1]), int(sys.argv[2]))
    rclpy.spin_until_future_complete(minimal_client, future)
    response = future.result()
    minimal_client.get_logger().info(
        'Result of add_two_ints: for %d + %d = %d' %
        (int(sys.argv[1]), int(sys.argv[2]), response.sum))

    minimal_client.destroy_node()
    rclpy.shutdown()

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

پیکربندی WS

ساخت پکیج

Pub - Sub

Server - client



ROS 2 :: نوشتن Service Client – Node

```
build package

# in ~/ros2_ws

$ colcon build --packages-select py_srv_demo
```

پیکربندی WS

ساخت پکیج

Pub - Sub

Server - client

```
Source setup.bash

# in ~/ros2_ws

$ source install/setup.bash
```



نوشتن msg دلخواه



Create a package

```
# in ros2_ws/src/ directory
```

```
$ ros2 pkg create --build-type ament_cmake --license Apache-2.0  
tutorial_interfaces
```

پیگیری WS

ساخت پکیج

Pub - Sub

Server - client

Custom msg



نوشتن msg دلخواه



Create a msg directory

```
# in ros2_ws/src/tutorial_interfaces/ directory
```

```
$ mkdir msg
```

پیگر بندی WS

ساخت پکیج

Pub - Sub
Server - client

Custom msg



نوشتن msg دلخواه



Create a msg file

```
# in ros2_ws/src/tutorial_interfaces/msg directory
```

```
$ > Joints.msg
```

پیگر بندی WS

ساخت پکیج

Pub - Sub

Server - client

Custom msg



نوشتن msg دلخواه



msg structure

```
float32 joint1  
float32 joint2
```

پیگیری WS

ساخت پکیج

Pub - Sub

Server - client

Custom msg



نوشتن msg دلخواه

```
Modify CMakeLists.txt file

find_package(rosidl_default_generators REQUIRED)

rosidl_generate_interfaces(${PROJECT_NAME}
  "msg/Joints.msg"
)
```

پیگیری WS

ساخت پکیج

Pub - Sub

Server - client

Custom msg



نوشتن msg دلخواه

پیکربندی WS

ساخت پکیج

Pub - Sub

Server - client

Custom msg



Modify Package.xml file

```
<buildtool_depend>roscpp_default_generators</buildtool_depend>  
<exec_depend>roscpp_default_runtime</exec_depend>  
<member_of_group>roscpp_interface_packages</member_of_group>
```




نوشتن msg دلخواه

پیگیری WS

ساخت پکیج

Pub - Sub

Server - client

Custom msg



Build Package

```
$ colcon build --packages-select tutorial_interfaces
```



نوشتن msg دلخواه

پیگیری WS

ساخت پکیج

Pub - Sub

Server - client

Custom msg



Source setup.bash

```
# in ~/ros2_ws
```

```
$ source install/setup.bash
```



ROS 2TM

ممنون از توجه شما