

Servidores de acción con ROS.

• No son bloqueantes

- Sigue paradigma cliente/servidor, asíncrono
- Mensajes de acción
(lo correcto es colocarlos en el paquete del servidor de acción)

Carpeta:

action

↳ Mensaje.action

↳ # goal
int32 objetivo

result
int32 resultado

feedback
int32 realimentación

será necesario modificar package.xml y cmakeLists.txt
al compilarlos genera los archivos de mensaje

rostopic list

rosservice list

~~roaction list~~

¿Cómo ves las acciones disponibles?

rostopic list

↳ /action_server / cancel

/action_server / feedback

/action_server / goal

/action_server / status


/action_server / result



Si. ahora hacemos:

rosmg list | grep action_server

action-server / Mensaje Action
action-server / Mensaje Action Feedback
action-server / Mensaje Action Goal
action-server / Mensaje Action Result
action-server / Mensaje Feedback
action-server / Mensaje Goal
action-server / Mensaje Result



rosmmsg show
(para visualizar tipo de
datos y nombres campos)

¿Cómo se crea un cliente de
acción?

```
# /usr/bin/env python
```

```
import rospy
```

```
import actionlib as Ojo
```

```
from action-server.msg import package
```

```
from actionlib import (MensajeAction, MensajeGoal, MensajeResult,  
MensajeFeedback)
```

```
def mi_funcion (feedback):  
    print feedback.
```

rospy.init_node

cliente = actionlib.SimpleActionClient(

 '/nombre-accion', MensajeAction)

 ↑
 tipo de mensaje

cliente.wait_for_server()

goal = MensajeGoal()

goal.cycles = 10

cliente.send_goal(goal, done_cb =

 done_callback, feedback_cb = mi_funcion)

cliente.cancel_goal() ← para
 neces cancelaciones

status = cliente.get_state() ← anticipadas

cliente.wait_for_result() ← para saber
 estado de la
 acción

↑
Espera hasta
que la
acción termine,
devuelve true
cuando
sucede

0 → pending

1 → active

2 → Done

3 → Warning

4 → Error

Modificaciones en `MakeLists.txt`

I) `find-package catkin REQUIRED COMPONENTS`
 `rospy`
 `actionlib_msgs`
)

II) `add_action_files (`
 `FILES`
 `mi_mensaje.action`
)

III) `generate_messages (`
 `DEPENDENCIES`
 `std_msgs actionlib_msgs`
)

IV Comprobar:

```
catkin-package (  
  CATKIN-DEPENDS  
    rspy  
)
```

V Comprobar

```
include-directories (  
  $ } catkin-INCLUDE_DIRS {  
)
```

Modificaciones package.xml:

añadir:

```
<build-depends> actionlib </build-depends>
```

```
<build-depends> actionlib-msgs </build-depends>
```

```
<build-export-depends> actionlib </build-export-  
  depends>
```

```
<build-export-depends> actionlib-msgs </build-  
  export-  
  depends>
```

```
<exec- depend> action-lib </exec- depend>  
< exec- depend> actionlib-msgs </exec- depend>
```

(reacción servidor acción)

```
#!/usr/bin/env python
```

```
import rospy  
import actionlib
```

```
from server-action.msg import
```

```
Mensaje Action, Feedback, Result
```

```
class nombreacum-simple(object):
```

```
-- realimentacion = Mensaje Feedback()
```

```
-- resultado = Result()
```



```

def __init__(self):
    self.__act_serv =
        actionlib.SimpleActionServer("nombre", Action,
                                     self.goal_callback, False)
    self.__act_serv.start()

```

Mensaje Action
↓

```

def goal_callback(self, goal):
    <----->
    <----->
    objetivo = goal.goal
    while self.__act_serv.is_preempted():
        requested()
        self.__act_serv.set_preempted()
        break

```

Ojo nombre del campo en función declaración mensaje

requested() :

```

:
:
: self.--realimentacion.feedback = ---
: self.--act-serv.publish-feedback
: (self.--realimentacion)
:
:

```

```

:
: if success:
:     self.--resultado.result = ---
:     self.--act-serv.set-succeeded(
:                                     self.--resultado)
:

```

```

:
: else:
:     self.--act-serv.set-aborted( )

```

↗
 ojo, no se
 debe cancelar a
 menos si ya
 está preempted

```

# Módulo ppal
rospy.init_node('____')
accion-simple( )
rospy.spin( )

```

Pasos a seguir para crear
mensaje servidor acción:

acum-personalizado

↳ creamos carpeta "action"

↳ Mensaje.action

↳ #goal
int32 objetivo

#result
int32 resultado

#feedback
int32 realimentacion


Modificamos makeLists y package.xml

catkin_make

catkin_make --only-package - - -

source devel/setup.bash

rosmmsg list | grep MyAction
Message



MyAction {
Action
ActionFeedback
ActionGoal
ActionResult
Feedback
Goal
Result