





Let's Have Fun with Robot's APIs

JavaOne 2015



About Us

Erwan Pinault
SW Engineer



Thalía Cruz
SW Engineer



Objective:
To develop a “living
robot” with a Java
application.



NAOqi and Java

First steps with Java NAOqi SDK

Let's Code!

Using a virtual robot

Demo time!

Test your code on a real robot



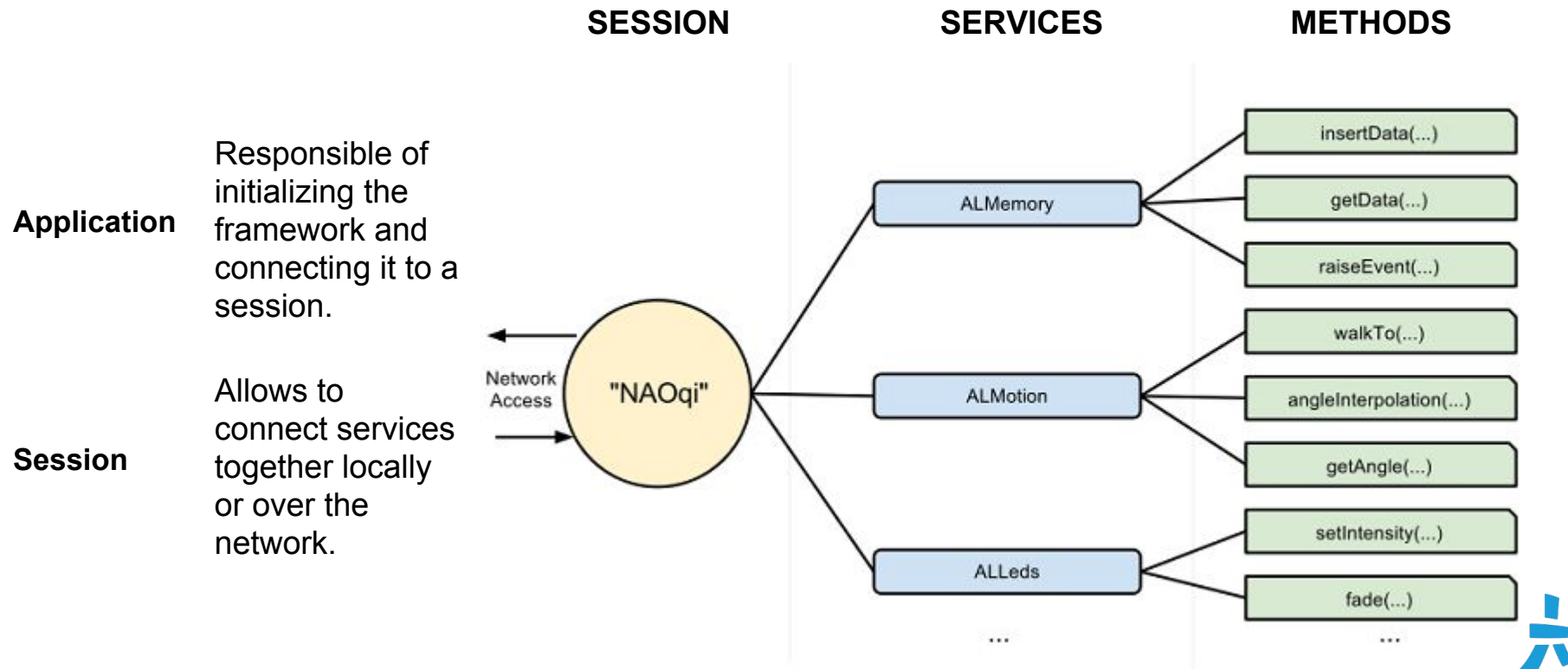
NAOqi and Java

First steps with Java NAOqi SDK

SoftBank Group



NAOqi



Libqi

Our homemade middleware that provides cross-platform and cross-language support.

Python

```
#python  
textToSpeechProxy.say("let's dance")
```

Let's dance

C++

```
//cpp  
textToSpeechProxy->say("let's dance");
```





Java NAOqi SDK

Java
NAOqi
SDK = Libqi
 java + Dedicated
 wrapper
 classes

- Use Aldebaran's API.
- Create your own services.
- Run your services from a remote machine or directly in the robot.*



*No JRE is present in the robot's system image by default.

What can our robots do?

Feel

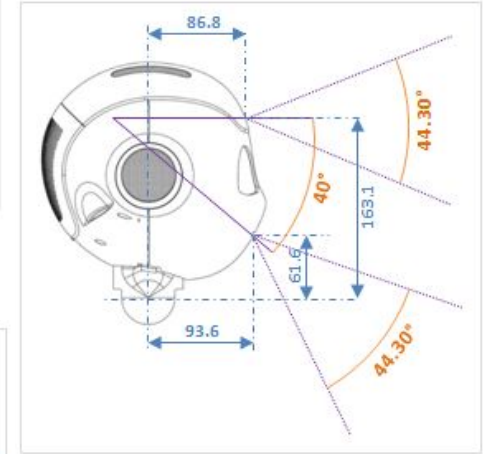
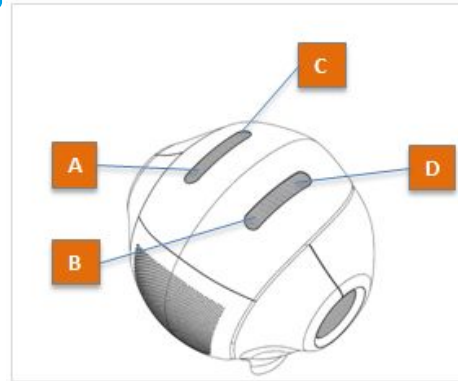
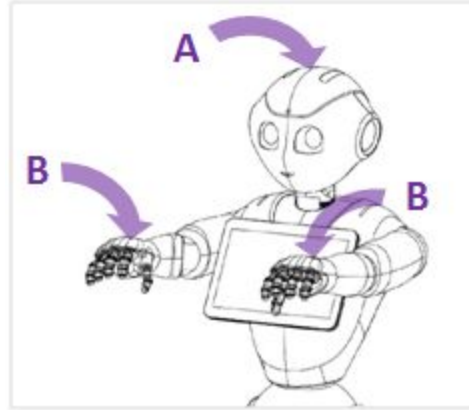
Through their tactile
sensors

See

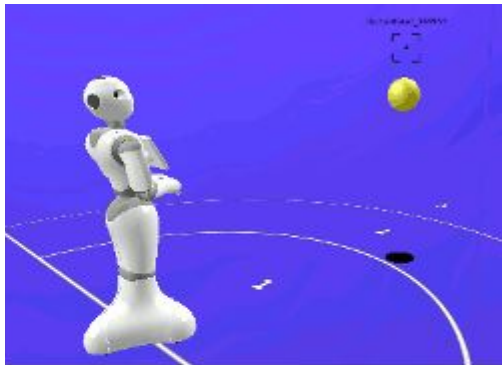
Through their cameras

Hear

Through their microphones



What can our robots do?



Talk

Through their loudspeakers

Move

Thanks to their motors
and joints





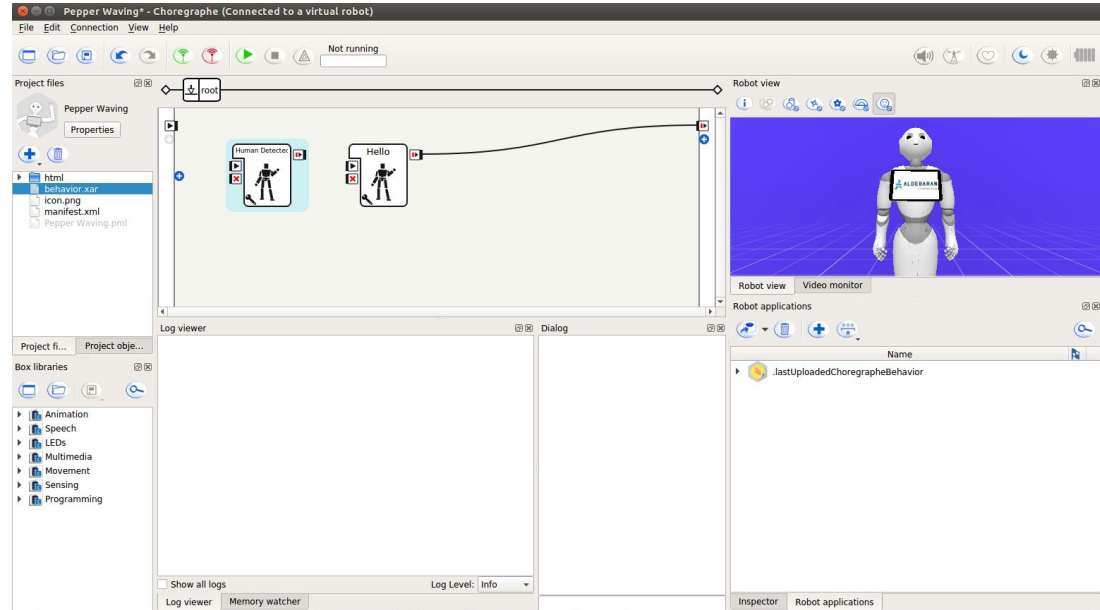
Let's code!

Using a virtual robot

Choregraphe

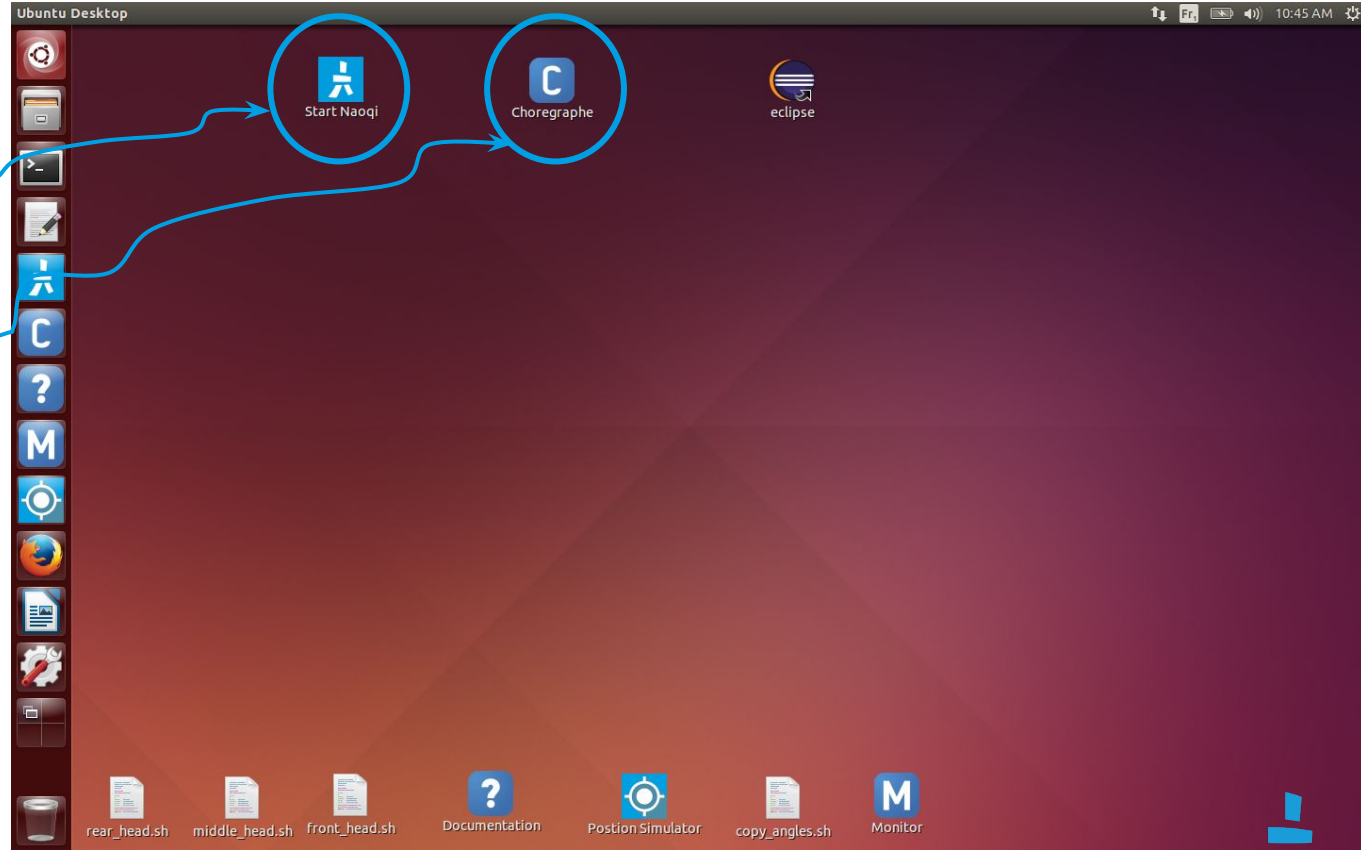
A multi-platform desktop application that allows to:

- Create behaviors.
- Run and test them on a virtual robot, or directly on a real one.
- Monitor and control your robot.



Setup

1. Launch a virtual robot
2. Open Choregraphe
3. Connect Choregraphe to your virtual robot





#1: Talk



#1: Hello Java

```
public class Main {  
    public static void main(String[] args) {  
        Application application = new Application(args, "tcp://127.0.0.1:44392");  
        try {  
            application.start();  
            Session session = application.session();  
            ALTextToSpeech tts = new ALTextToSpeech(session);  
            tts.setLanguage("English");  
            tts.say("Hello Java!");  
        }  
        catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Create a new application.

Start the application. This will create a session and connect it to the robot.

Retrieve the created session.

Get the ALTextToSpeech service from the current session.

When everything is done, the application will end.





#2: Feel



#2: Reacting to events

...

```
ALMemory memory = new ALMemory(session);
```

Get the ALMemory service in the current session.

```
memory.subscribeToEvent("MiddleTactilTouched", (touch) -> {
```

Subscribe to the event.

```
    if ((float) touch == 1.0) {  
        System.out.println("Touched");  
    }  
});
```

Do something when the middle tactile head sensor is touched.

```
});
```

```
application.run();
```

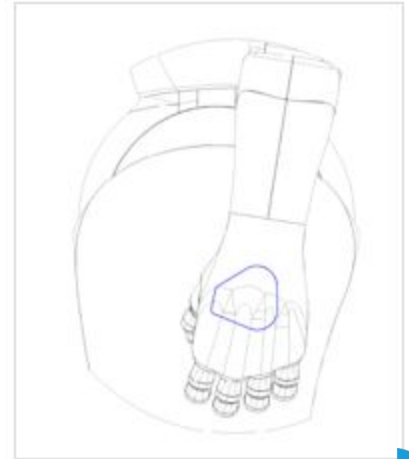
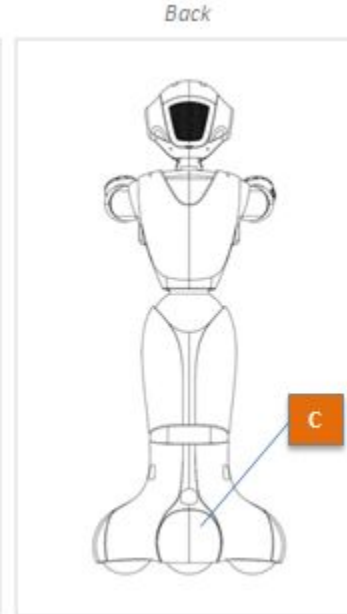
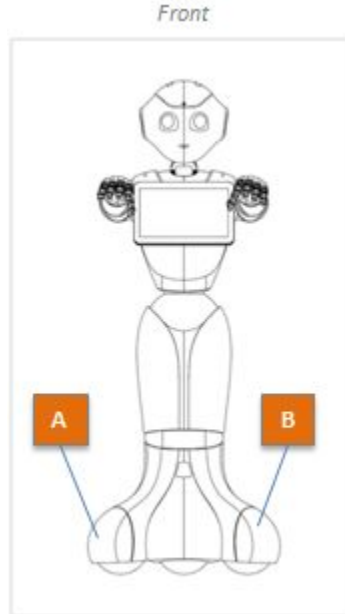
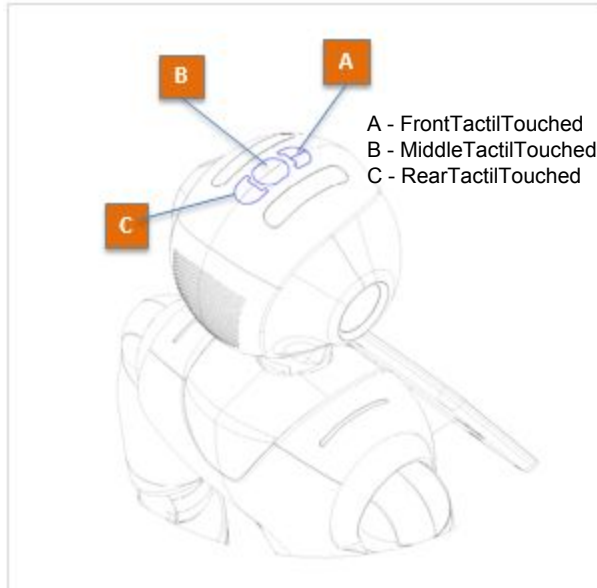
Keep the application running, so it can listen to the event.



Feel → Talk

Goal: Make the robot say something when he's touched (head, bumpers or hands).

See [ALTouch](#) for more details.





#3: Hear



#3: Understanding words 1/2

...

```
ArrayList<String> listOfWords = new ArrayList<String>();  
listOfWords.add("Hello");  
listOfWords.add("Pepper");
```

Create a list of words to be recognized.

```
ALSpeechRecognition speechRecognition = null;
```

```
try {  
    speechRecognition = new ALSpeechRecognition(session);  
  
    speechRecognition.setVocabulary(listOfWords, false);  
  
    speechRecognition.subscribe(APP_NAME);  
}
```

Get the ALSpeechRecognition service.

Add the words to the robot's vocabulary.

Subscribe to the ALSpeechRecognition service.

```
catch (Exception exception) {  
    exception.printStackTrace();  
    System.out.println("Running on virtual robot");  
}
```

There is no ALSpeechRecognition service on a virtual robot.



#3: Understanding words 2/2

```
ALMemory memory = new ALMemory(session);
```

```
memory.subscribeToEvent("WordRecognized", (value) -> {  
    List<String> words = (List<String>) value;  
    String word = words.get(0);  
    System.out.println(word);  
});
```

Do something when one of the words is recognized.

```
application.run();
```

Keep the application running, so it can listen to the event.

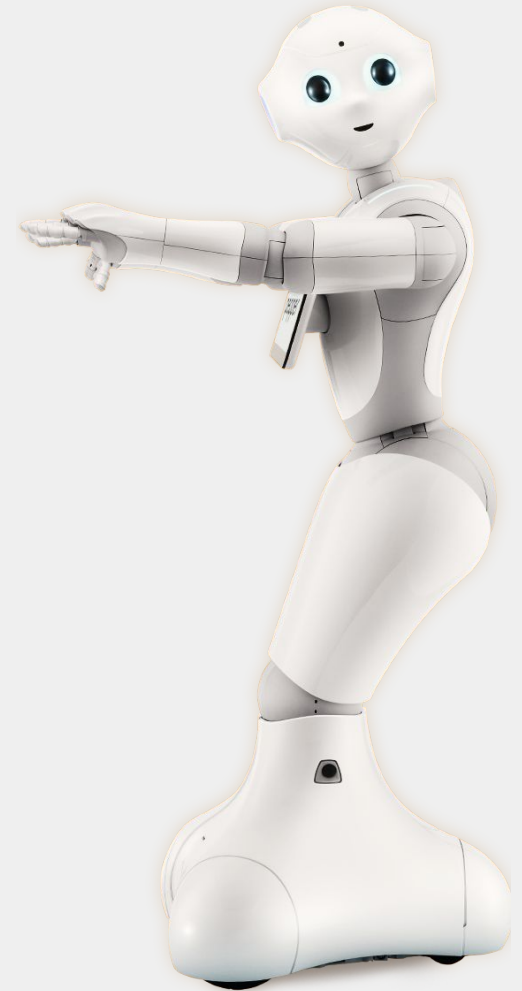
```
if (speechRecognition != null)  
    speechRecognition.unsubscribe(APP_NAME);
```

Once the application has been stopped, unsubscribe from the service.





#4: Move



#4: Strike a pose 1/2

...

```
ALMotion motion = new ALMotion(session);
```

Get the ALMotion service from the current session.

```
List<String> bodyParts = motion.getBodyNames("Body");
```

Get the whole list of the body joints.

```
ArrayList<Float> pose = new ArrayList<Float>(Arrays.asList(-0.015708f, -0.381962f, -1.213f,  
0.12f, -0.87f, -0.28f, 0.26f, 0.6942f, -0.000575502f, -0.0149823f, -0.0604201f, -1.01578f,  
-0.12f, 0.87f, 0.28f, -0.26f, 0.02f, 0f, 0f, 0f)) ;
```

Create a list of angles (in radians)

```
motion.angleInterpolationWithSpeed(bodyParts, pose, 0.3f);
```

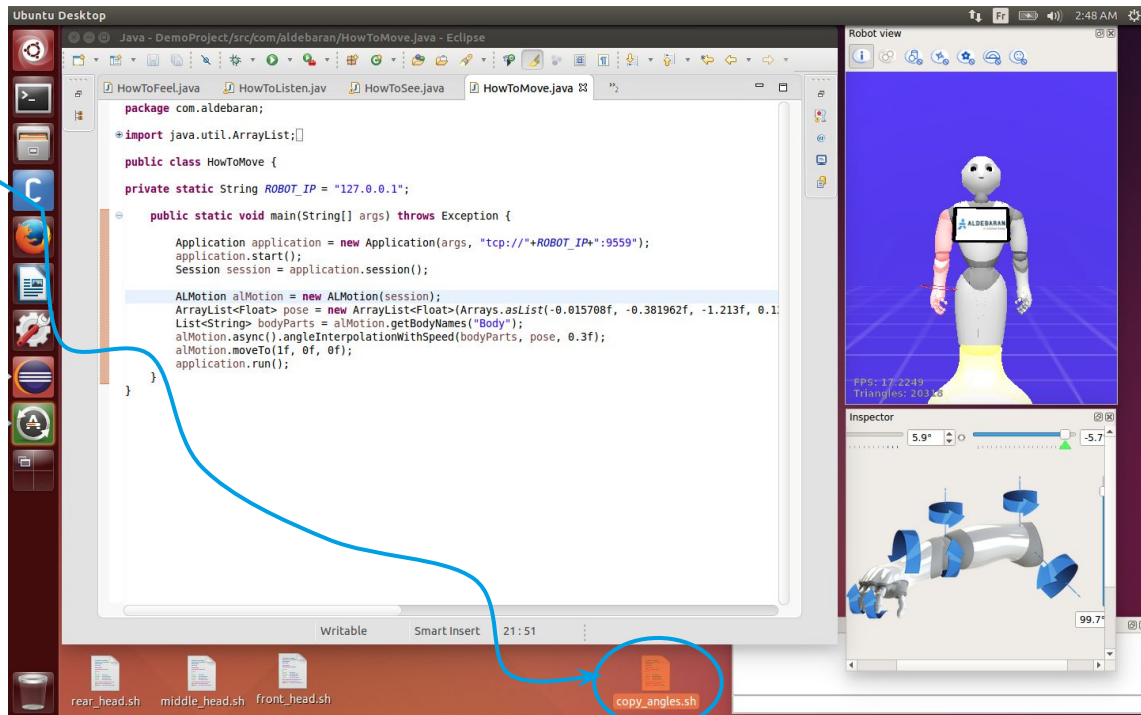
Apply the angles to the corresponding body joint.



#4: Strike a pose 2/2

How to get the list of angles in radians easily?

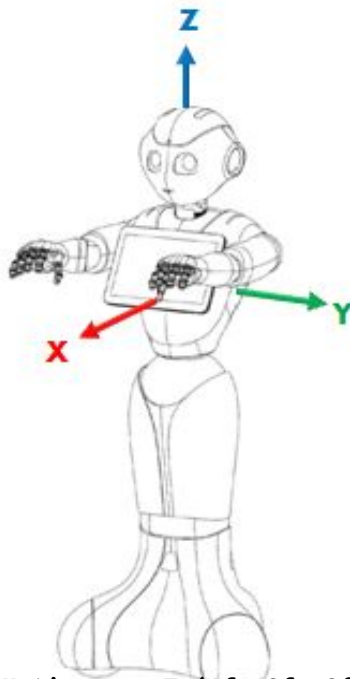
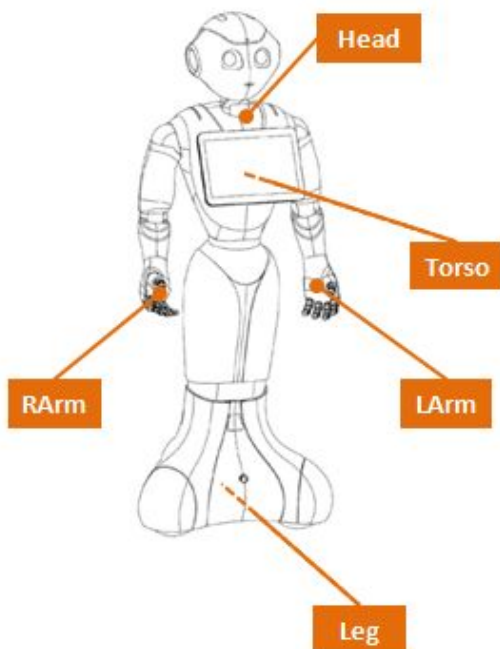
Try using
`copy_angles.sh`



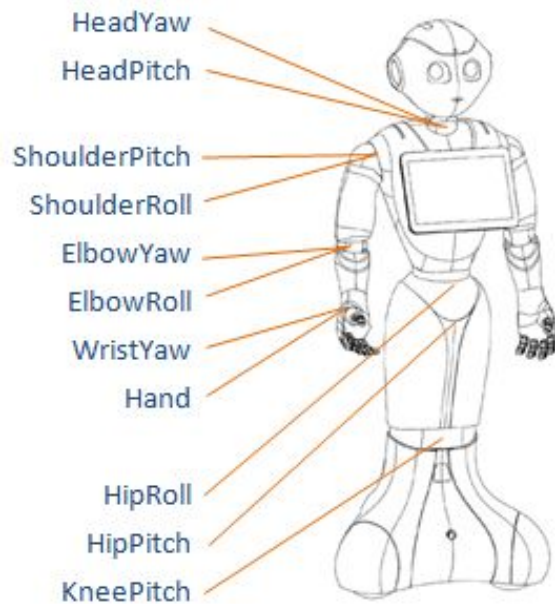
Hear → Move

Goal: Make the robot understand your orders and move his body (or a part of it).

See [ALMotion](#) for more details.



```
ALMotion.moveTo(1f, 0f, 0f);
```



A living robot



Action



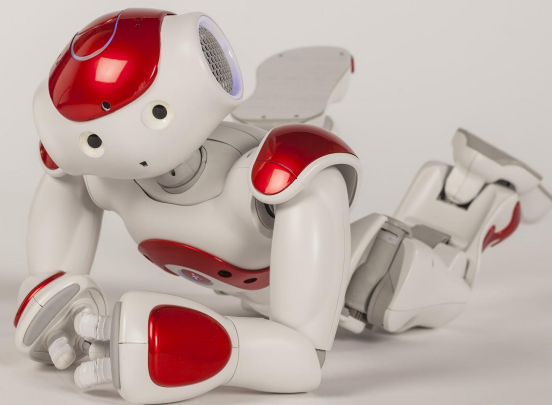
Reaction



What else can you do?

Some ideas:

- Say “move like jagger”. → The robot shows his special moves.
- Say “go to sleep”. → The robot goes to rest.
- Pet your robot. → The robot makes cute sounds.
- Hold your robot’s hand. → The robot says “I’ll always be with you”.
- The robot sees someone. → The robot tracks the person.
(ADVANCED)



It's up to your imagination...






Demo time!

Test your code on a real robot



Let's continue having fun!



CREATE ROBOTICS
APPLICATIONS

JOIN THE
DEVELOPER
PROGRAM

<http://community.aldebaran.com>

