

Documentation

This documentation describes the use of Mycroft.ai. Mycroft is an Open-Source voice assistant which can be used on Linux, Android and especially the Raspberry Pi.

The official documentation can be found on <https://mycroft.ai/documentation/>

Setup

Linux

On Linux the configuration of Mycroft is pretty easy. There are two ways to install it. On many Linux distributions it is possible to install Mycroft using the built in package manager. The other way is to download it directly from GitHub and install it manually (see <https://github.com/MycroftAI/mycroft-core>).

Raspberry Pi

To use a Pi as a voice assistant, you need a Raspberry Pi 3 or better, an USB or 3.5mm speaker or monitor with speakers and an USB-microphone. For the setup you might also need an keyboard and a monitor. Picroft, the Mycroft version for Raspberry Pi, can be downloaded on <https://github.com/MycroftAI/enclosure-picroft/> and can simply be flashed on a SD-card using a flashing-software like Etcher.

When starting the first time, you need to follow the setup, configure the internet connection and speakers as well as the microphone. Configuring the microphone can be a little bit more complicated if you are not using the recommended microphones. If this is the case you need to use a tool called "alsamixer" or "pulseaudio" to choose the right device.

After setting up the Pi, the Mycroft client will be started for the first time and you can continue with pairing it to your Mycroft Home account.

When using a Pi you need to check that your power adapter has enough ampere. Mycroft recommends using a power adapter with 2.5A, but 2A was also tested successfully. If you don't have enough amps the recognition of voice commands might get worse and some devices like the speakers or microphone might not work at all.

Recommended order of steps for setting up the Pi:

1. Flash the Picroft on the SD-card and start the Pi
2. go through the setup and connect it to the internet
3. update Picroft ("sudo apt-get update" and "sudo apt-get upgrade")
4. configure the microphone using alsamixer or install pulseaudio (sudo apt-get install pulseaudio)
5. restart the pi
6. connect Picroft to your Mycroft Home account

Android

To use Mycroft on Android you need an already set up Mycroft device for example a Raspberry Pi.

1. install Android Studio
2. clone GitHub repository (<https://github.com/MycroftAI/Mycroft-Android>)
3. install the app on your Android device with Android Studio and start it
4. go to general and use the IP address of your Mycroft device ("Hey Mycroft, what is you IP address")
5. use Mycroft on your Android device

Pairing

After installing Mycroft you need to connect it to your Mycroft Home account. In this way Mycroft can manage API-Keys better and can offer additional services such as their marketplace. Theoretically Mycroft can be used without a Home account and with your own API-Keys.

The pairing is done using a six digit long authentication key which will be output by Mycroft after its first start. This key has to be added online to your account. After that the complete use of Mycroft is possible.

Installation von Skills

After pairing your device, skills can be installed easily using the Mycroft Marketplace. Despite this you can also use the Mycroft skill manager which allows the manual download and installation of skills via the command line. Skills can also be placed directly into `/opt/mycroft/skills`.

Use your own skills on a Pi:

1. enable SSH on the Pi (sudo raspi-config → Interfacing Options → SSH)
2. copy your skill to the Pi: `scp -r path_to_skill/skill_name/ pi@ip_address:/opt/mycroft/skills/skill_name.mycroftai/`
3. restart the Pi

Other operating systems

For now there are no Mycroft versions for Windows or MacOS. There is a version for Android which will work for SDK24 and later. The installation needs to be done via Android Studio as there is no ready to use APK.

Development of skills

Skills for Mycroft are based on Python, which is why you need a basic understanding for it in order to develop skills.

There are three terms which are really important for the understanding of Mycroft skill:

- **dialog:** A dialog is a phrase which is spoken by Mycroft, for example as an answer for a question.
- **utterance:** Utterances are sentences which are spoken by users after they used the wake word ("Hey Mycroft").
- **intent:** An intent is an identifiable call of a skill. Utterances are compared to existing intents and the matched skill is activated.

For skills there is a repository on GitHub (<https://github.com/MycroftAI/mycroft-skills>) where you can find many community skills and skills by Mycroft as well as a template for new skills (https://github.com/MycroftAI/mycroft-skills/tree/19.02/00__skill_template).

The structure for the skill repositories is relatively strict:

- **dialog-directory:** All dialog-files for various languages will go here. Dialog-Files end with `.dialog`
- **vocab-directory:** The vocab-directory contains all intent-files in different languages that the skill shall react to. Intent-files end with `.intent` and can contain regular expressions. Furthermore there are also vocab-files which end with `.vod` and contain vocabs which can be used in skills for example synonyms.
- **regex-directory:** This directory is optional and contains regex-files with regular expressions that can occur in intents. Regex-files end with `.rx`.
- **init.py:** This python file contains the handling of all intents, processes them and returns the results to the user.

Inside the Python file, intents are mapped on functions using decorators. In front of a function that will process an intent has to be a decorator that defines which intent the function is mapped to.

Example:

```
@intent_file_handler("hello.world.intent") # decorator, hello.world.intent is a file
in vocab
def handle_hello_world(self):
    return self.speak_dialog("hello.world") # hello.world is a .dialog file
```

A more detailed explanation on the necessary Python packages and how to develop skills can be found on <https://mycroft.ai/documentation/skills/introduction-developing-skills/>. Existing skills in the skills repository can also be used as an orientation on how to develop skills.

Example skill

The health care skill is an example skill to show case some of the features that mycroft provides. It saves "patients" using their names in a data structure. Furthermore one can add heart rates to a patient and calculate the average heart rate.

Usage

say things like ...

to create patient:

- "create *name lastname*"
- "Add patient *name lastname*"

to delete patient:

- "remove *name lastname*"
- "delete patient *name lastname*"

to add heartrate:

- "add heartrate *heartrate* to *name lastname*"
- "create a heartrate of *heartrate* for patient *name lastname*"

to get average heartrate:

- "show the average heartrate of *name lastname*"

to delete the data structur:

- "delete data"

To use all the commands, you have to add a patient first. A proper name consists allways of a first and a last name. The skill will tell you if the naming was not successful. If you use the command to delete the data structure, the skill will will ask for your confirmation. You should wait some seconds before you answer. The CLI will allso show when Mycroft is ready. The only word you can confirm your choice with is "yes". Every other word will abort the process.