

Aufgaben(skills)

Realisiert wurde in diesem Projekt der "What is my next appointment" -Skill. Er gibt auf Nachfrage aus dem mi.Cloud Kalender den als nächsten eingetragenen Termin aus. Sind am aktuellen Tag ein Ganztages Event und ein einzelnes Event aufgeführt, wird das einzelne Event ausgegeben.

Außerdem kann man nach den Events eines bestimmten Tages oder nach allen kommenden Events fragen.

Skillentwicklung

Was wird benötigt

Für die Entwicklung eines Skills benötigt man einen Mycroft Client zum Testen und eine Python Entwicklungsumgebung. Um Befehlswörter zu bestimmen auf die reagiert werden kann, braucht es Files in denen die Wakewords festgelegt sind. Diese Files haben die Endung `.intent`. Um das Ergebnis auszugeben, wird dann ein Dialog File gebraucht. Also ein File mit der Endung `.dialog`. Typischer Aufbau der Ordner ist dann ein "vocab" Ordner für die Wakewords unterteilt in die jeweilige Sprache und ein "dialog" Ordner für die Antwortsätze. In diesen Dateien kann man mit geschweiften Klammern Variablen kennzeichnen. So z.B.:

"You have an appointment called {name} on {date}" in einer dialog-Datei. Auf diese Variablen kann im Code zugegriffen werden.

Wir haben uns beim Aufbau des Projekts an bereits vorhandenen Mycroft-Skills orientiert.

Wakewords/ intents

In unserem Fall bedeutet das, dass im Ordner `vocab/en-us` folgendes zu finden ist:

Für den "mehrere Events"- Skill sind folgende Schlüsselsätze in der Datei

"multiple.si.intent" zu finden:

- What is my schedule
- What are my upcoming events
- Schedule

Für den "nächstes Event"- Skill finden sich in der Datei "next.si.intent" die Schlüsselwörter:

- What is my next appointment
- When is my next event
- Schedule next

Für den "bestimmtes Event"- Skill kann man wie in der Datei "specific.si.intent" geschrieben folgendes sagen:

- What are my appointments (today/tomorrow/date)
- What is my agenda for (today/tomorrow/date)
- What are my appointments on (Monday/ Tuesday usw)

Welche Funktionen machen was

Funktionen die mit `@intent_file_handler` beginnen reagieren auf direkten Input des Nutzers. Sie werden also beim Erkennen der Wakewords aufgerufen. Für jeden Skill gibt es eine dieser Funktionen. Um die Antwort zu senden wird die Funktion

```
self.speak_dialog('dialog', data = {})
```

gerufen. In das Data Feld kann eingetragen werden, mit was die Variablen besetzt werden sollen.

In unserem Fall ist außerdem die für den Kalenderabruf wichtigste Funktion "fetch_events". Hier werden aus dem Cloud Kalender die Events der kommenden Tage und die relevanten Daten ausgelesen. Je nach Eventtyp (ganztägig oder genauer Termin) wird unterschieden. Die zurückgegebene Liste wird dann in den `@intent_file_handler` Funktionen sortiert und verarbeitet.

Vorgehen

In unserem Fall haben wir `intents` und `dialogs` bestimmt und sie erstellt. Dann haben wir `@intent_file_handler` Funktionen geschrieben und die Anbindung an den Kalender und Strukturierung der Einträge realisiert. Für die Verbindung mit MiCloud haben wir eine Config Datei mit den Accountdaten erstellt. Die Ausgewählten Informationen wurden dann an den `dialog` weiter gegeben. Zum Testen haben wir Kalendereinträge gemacht und sie versucht über Mycroft aufzurufen.

Besonderheiten

Config File

Für das Nutzen in Verbindung mit dem Kalender der Cloud, muss ein Passwort und Benutzername hinterlegt sein. Dies muss in der Config eingetragen werden.

relevante Webseiten zum Anlegen eines Mycroft Skills

- <https://mycroft-ai.gitbook.io/docs/>
- <https://github.com/lucas-goldner/si-calendar-skill-main>
- <https://github.com/MycroftAI/skill-reminder>
- <https://google.github.io/styleguide/pyguide.html>
- <https://www.python.org/about/gettingstarted/>
- <https://help.github.com/en/github/using-git>