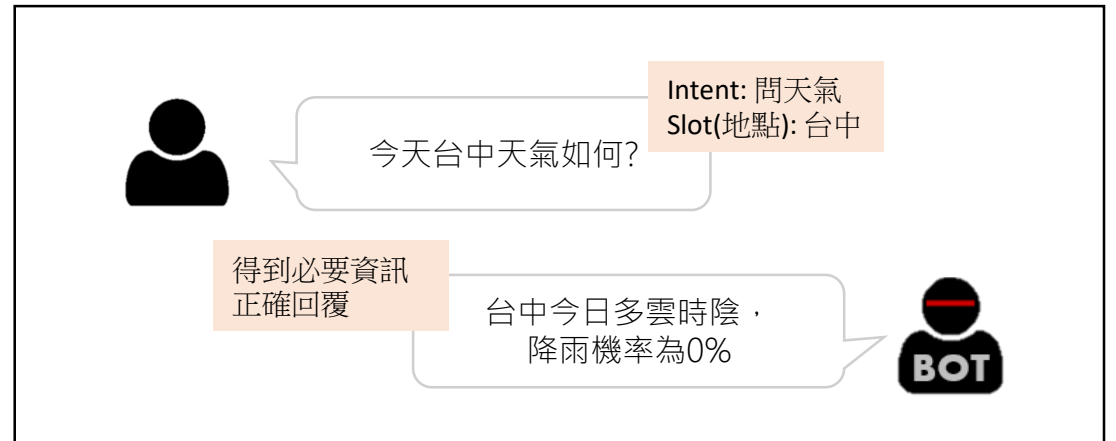
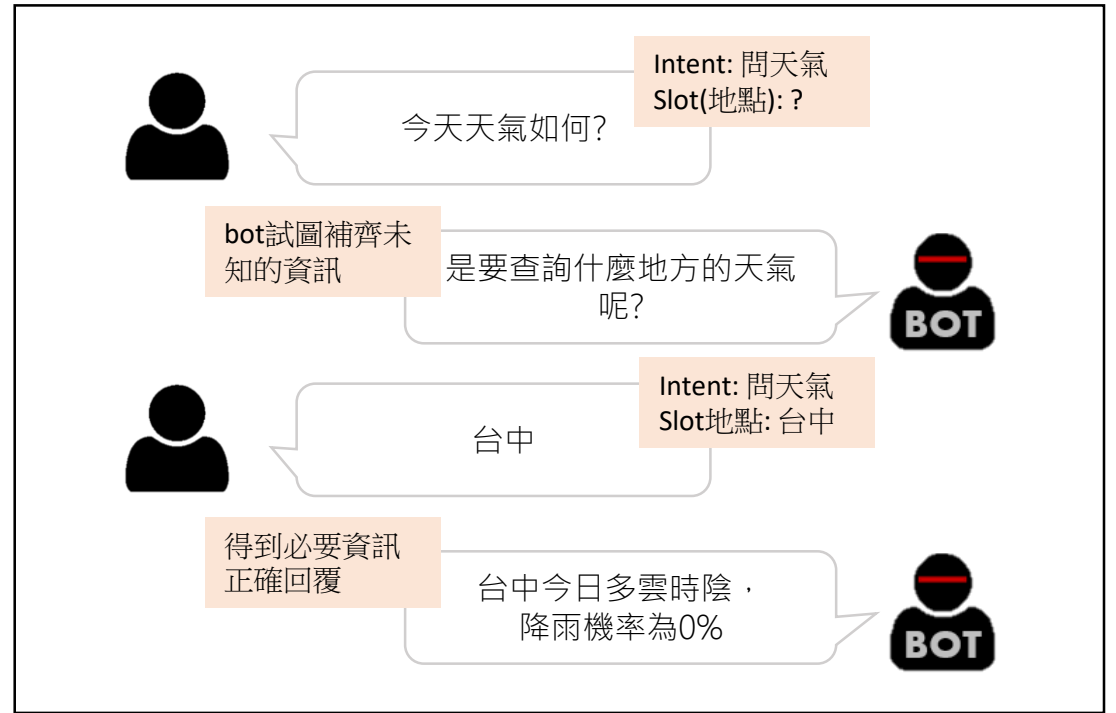


Rasa_入門2

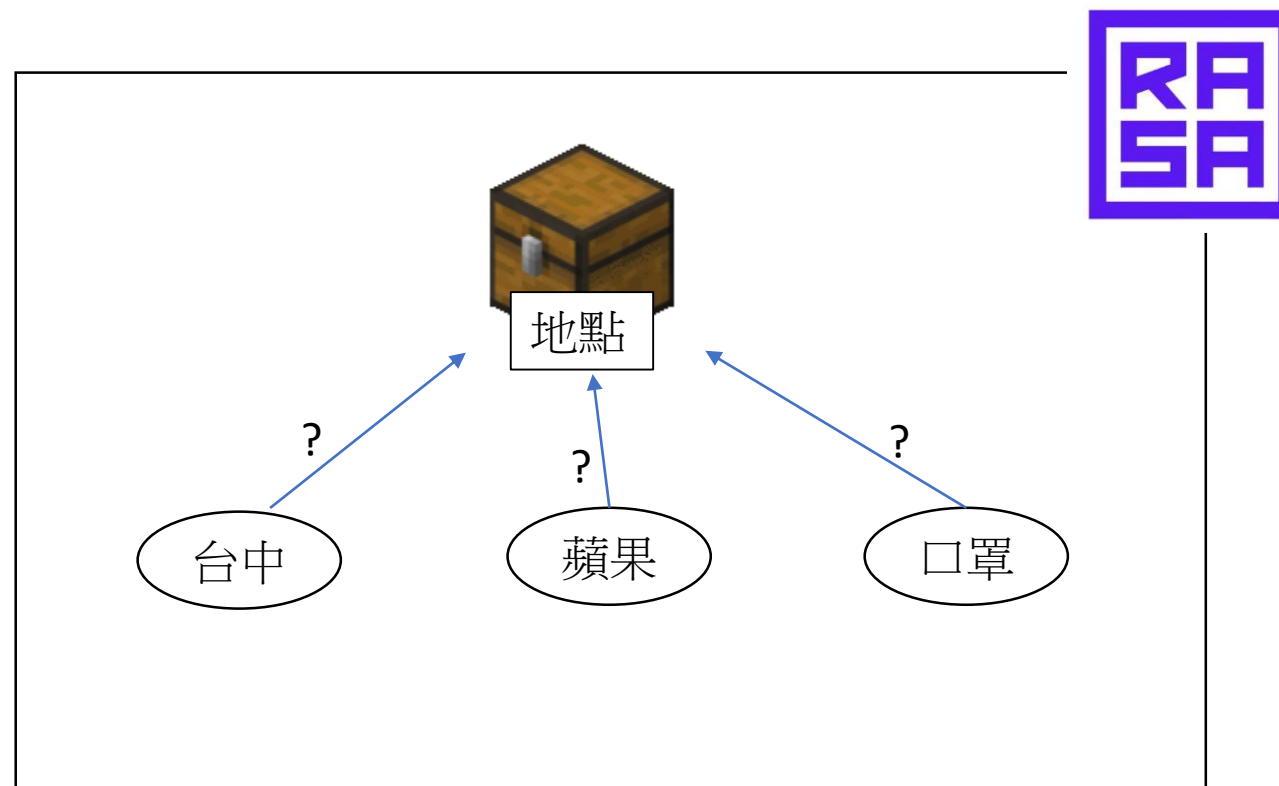
Slot & Entity(1)

- Slot會以key-value的形式去儲存使用者提供的外部訊息，可以在很大程度上維持對話的完整與順利，或是幫助我們蒐集資料。
- 以右為例地點就是我們的slot，整段對話會因為知不知道地點而有很大的差別，我們的目標除了正確辨識intent外，還必須將必要的slot正確的填滿。



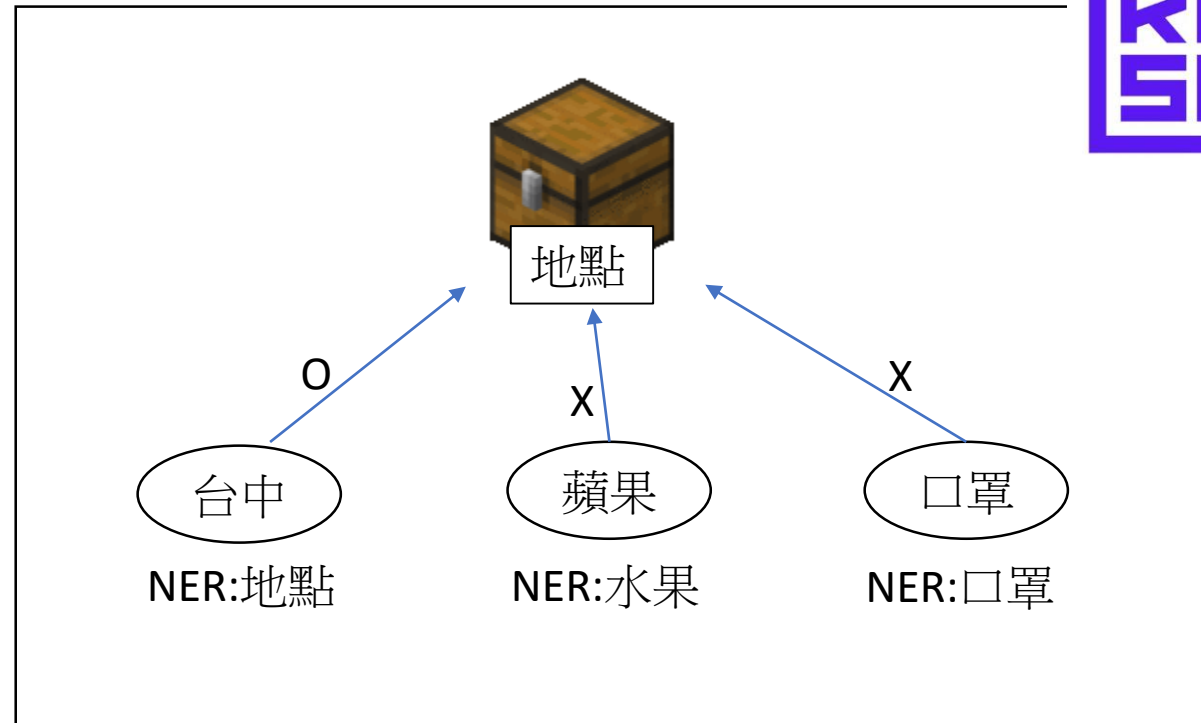
Slot & Entity(2)

- 對於Rasa來說Slot就是一個空位而已。就像是一個箱子上貼上標籤，箱子是空的就要填滿，但要如何將正確的東西放入箱中呢？



Slot & Entity(2)

- 這時候就輪到NER登場了，
- 命名實體識別(Named Entity Recognition)又稱作專名識別、命名實體，是指識別文本中具有特定意義的實體，主要包括人名、地名、機構名、專有名詞等，以及時間、數量、貨幣、比例數值等文字。
- 透過NER幫助RASA識別Entity，就能夠正確的填值。



Action

- Action是指機器人接下來要執行的操作，除了先前用過的response，還包括了RASA預設的default actions和custom actions。
- custom actions就是自定義的action，這是rasa的關鍵。透過編寫程式碼，你可以控制你的機器人開關電燈，也可以像response一樣拿來回覆使用者單純的訊息

default actions包含三個

action_listen，表示停止預測，等待用戶的輸入

action_restart，表示重置整個會話

action_default_fallback，撤消使用者訊息，並返回不理解該訊息

```
This is a simple example for a custom action which utters "Hello World!"

from typing import Any, Text, Dict, List

from rasa_sdk import Action, Tracker
from rasa_sdk.executor import CollectingDispatcher

class ActionHelloWorld(Action):

    def name(self) -> Text:
        return "action_hello_world"

    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:

        dispatcher.utter_message(text="Hello World!")

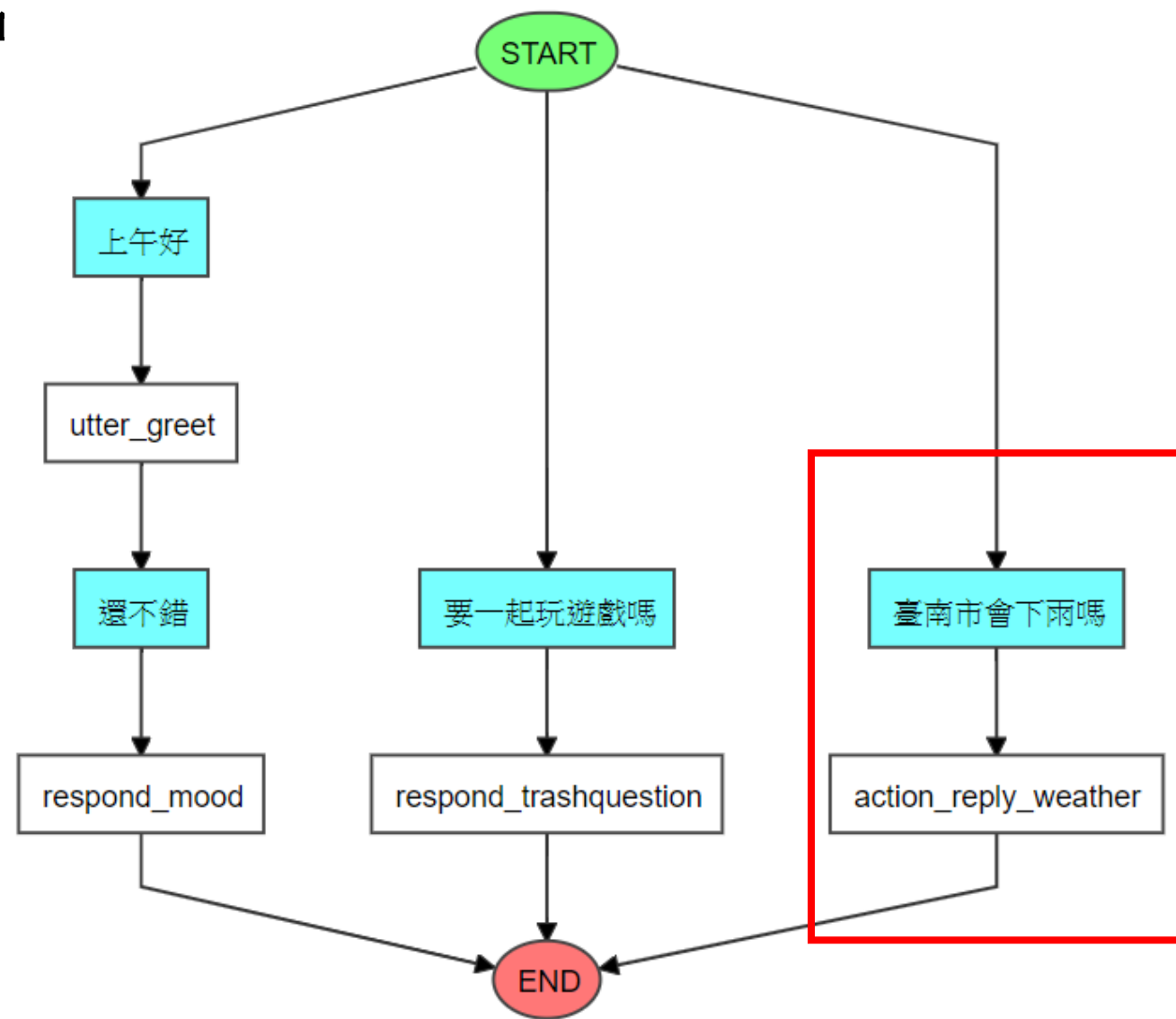
        return []
```

實作:Demo3 天氣機器

- 目標:

結合Slot、Entity、Custom Action為機器人添加一個能簡單查詢中央氣象局天氣的功能。

```
Your input -> 臺南市的天氣如何  
臺南市的天氣為晴時多雲,降雨機率0%,溫度為16°C至24°C  
Your input -> 台北市會下雨嗎  
臺北市的天氣為晴時多雲,降雨機率0%,溫度為15°C至22°C  
Your input ->
```



Step.1: Domain.yml

- 添加詢問天氣的intent。
- 添加用來儲存地名的slot及用來辨識地名的entity。
要注意的是slot的形態要這邊這宣告好。
- 稍後會編寫一個custom action，也需要在domain內添加該custom action的名稱。

有關slot:

Slot的型態除了text外還有Boolean、list等等

也可以透過initial_value預先設定好初值:

slots:

name:

type: text

initial_value: "Alice"

更多操作請詳閱

<https://rasa.com/docs/rasa/core/slots/>

```
intents:
- greet
- mood
- trashquestion
- ask_weather

slots:
  city:
    type: text

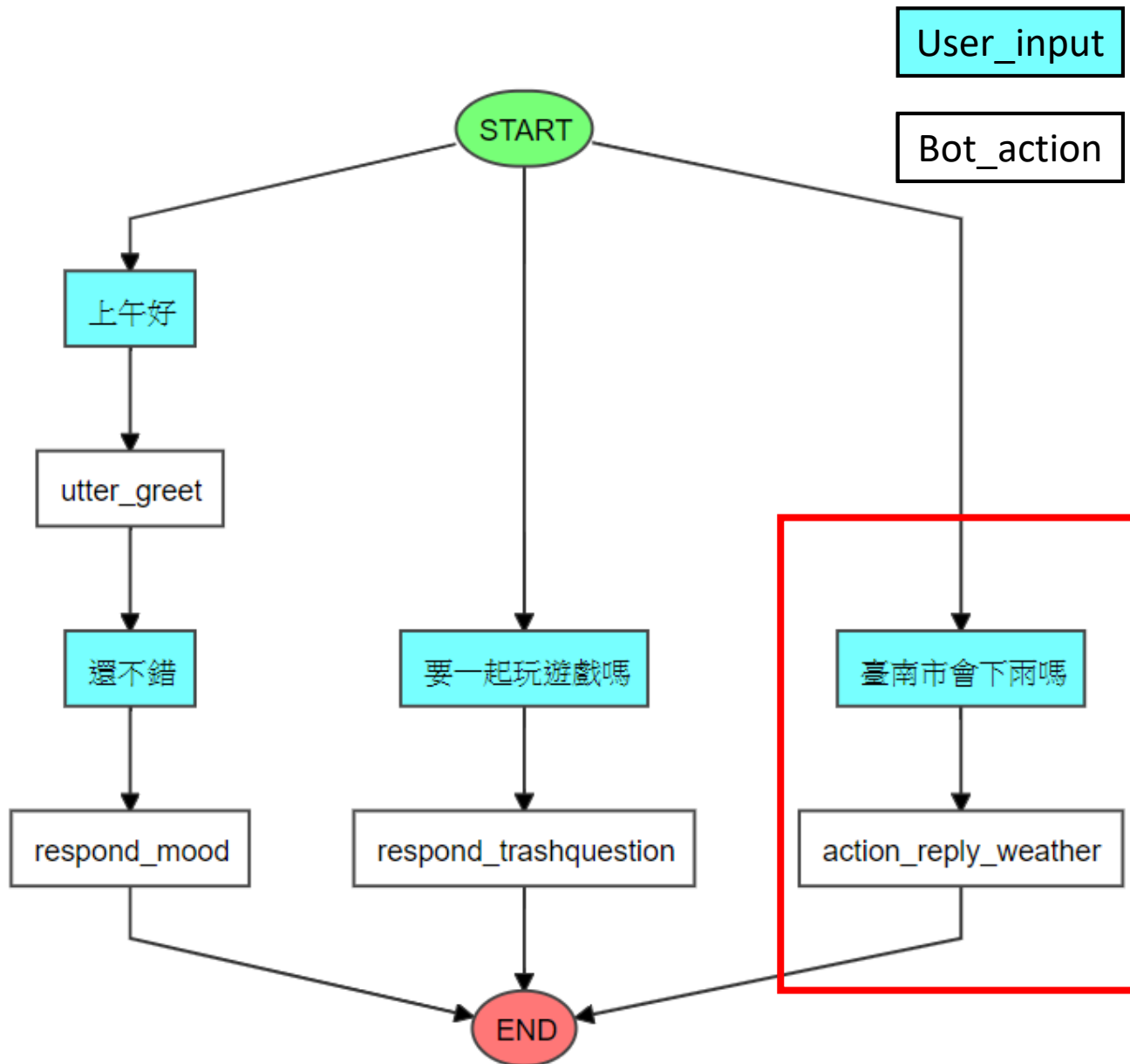
entities:
- city

actions:
- respond_trashquestion
- respond_mood
- action_reply_weather
```

Step.2: Stories.md

- 依情境添加story

```
## story_weather1  
* ask_weather  
  - action_reply_weather
```



Step.3: nlu.md & entity標註

- 添加語料並標註entity
- 標註格式為[value](entity)

[臺中市](city)就代表[臺中市]屬於(city)這個entity的value

需要注意的是entity的辨識較為困難，
所以建議再語料中標記盡可能多的entity，
以免辨識精度不夠造成功能有誤。

若有entity中同義字需要處理可以使用右邊方式
效果皆為將[台中市]或是[台中]辨識為[臺中市]

若使用同義字功能，
需要在config中添加EntitySynonymMapper

```
## intent: ask_weather
- [臺中市](city)天氣如何
- [臺北市](city)天氣怎麼樣
- [宜蘭縣](city)是晴天嗎
- [臺南市](city)會下雨嗎
- [嘉義縣](city)是陰天嗎
- [南投縣](city)在下雨嗎
```

```
- [臺中市](city)天氣如何
- [台中市](city:臺中市)天氣如何
- [台中](city:臺中市)天氣如何
```

```
## synonym:臺中市
- 台中市
- 台中
```

Step.4: config.yml

- 使用官方推薦的配置
- 需要特別注意的是若使用同義字功能，需要在config中添加EntitySynonymMapper

Pipeline選擇請參考

<https://rasa.com/docs/rasa/nlu/choosing-a-pipeline/>

詳細的compomemt及policies請查閱

<https://rasa.com/docs/rasa/nlu/components/>

<https://rasa.com/docs/rasa/core/policies/>

```
language: zh
pipeline:
  - name: JiebaTokenizer
    intent_tokenization_flag: True
    intent_split_symbol: "+"
  - name: LexicalSyntacticFeaturizer
  - name: CountVectorsFeaturizer
  - name: CountVectorsFeaturizer
    analyzer: "char_wb"
    min_ngram: 1
    max_ngram: 4
  - name: DIETClassifier
  - name: EntitySynonymMapper
  - name: ResponseSelector
    retrieval_intent: trashquestion
  - name: ResponseSelector
    retrieval_intent: mood

# Configuration for Rasa Core.
# https://rasa.com/docs/rasa/core/policies/
policies:
  - name: MemoizationPolicy
  - name: TEDPolicy
    max_history: 5
    epochs: 100
  - name: MappingPolicy
```

Step.5: custom action(1)

- 如先前所說，要透過custom action來實現查詢中央氣象局天氣的功能

- 首先至中央氣象局-氣象資料開放平台

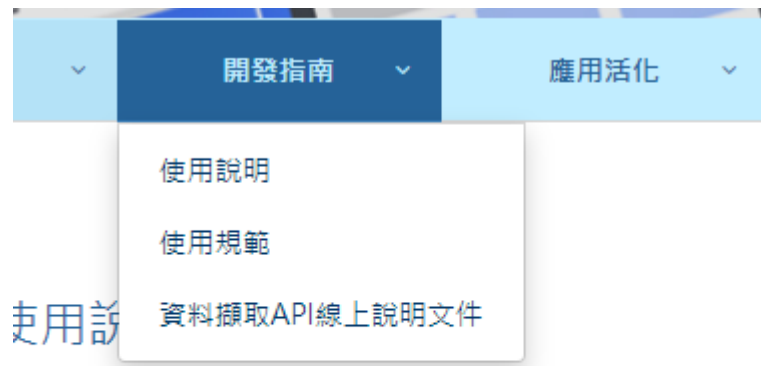
- <https://opendata.cwb.gov.tw/userLogin>

- 註冊會員並登入後可以得到API的授權碼



Step.5: custom action(2)

- 在開發指南中可以查看API使用說明及規範
- 基本上是簡單易用的RESTful API，回傳格式有XML、JSON等



中央氣象局開放資料平臺提供各式資料，為能優化開放資料
或透過facebook登入。民眾或機關可在登入本平臺後透過

※ URL： `https://opendata.cwb.gov.tw/fileapi/v1/opendataapi/{dataid}?Authorization={apikey}&format={format}`

`{dataid}` 為各資料集代碼（參照：資料清單） ex.F-A0012-001

`{apikey}` 為會員帳號對應之授權碼 ex.CWB-1234ABCD-78EF-GH90-12XY-IJKL12345678

`{format}` 為資料格式，請參照各資料集頁面確認可下載之檔案格式 ex.XML、CAP、JSON、ZIP、KMZ、GRIB2

※ 範例：`https://opendata.cwb.gov.tw/fileapi/v1/opendataapi/F-A0012-001?Authorization=CWB-1234ABCD-78EF-GH90-12XY-I`

並請加入快取功能，如上述所示。

Step.5: custom action(2)

- 知道怎麼使用API後在RASA專案跟目錄下新增一個python檔，編寫code時間氣象資料的抓取。
- 也可以直接寫在rasa專案的actions.py內，但這邊為了便免程式碼雜亂，所以先在外面寫好在import進去。

__pycache__	2020/4/15 下午 0...	檔案資料夾
data	2020/4/14 下午 0...	檔案資料夾
models	2020/4/15 下午 0...	檔案資料夾
results	2020/4/14 下午 0...	檔案資料夾
tests	2020/4/14 下午 0...	檔案資料夾
__init__.py	2020/4/6 下午 07...	PY 檔案
actions.py	2020/4/15 下午 0...	PY 檔案
config.yml	2020/4/15 下午 0...	YML 檔案
credentials.yml	2020/4/6 下午 07...	YML 檔案
domain.yml	2020/4/15 下午 0...	YML 檔案
endpoints.yml	2020/4/7 下午 09...	YML 檔案
get_weather.py	2020/4/15 下午 0...	PY 檔案
graph.html	2020/4/14 下午 0...	Chrome HTML

```
1 import requests,json
2
3 class Weather():
4     @staticmethod
5     def get_data(LocationName):
6         url=("https://opendata.cwb.gov.tw/api/v1/rest/datastore/"
7             "F-C0032-001?Authorization=CWB-0FC54884-FCEA-4D0C-98E0-DB0A495D752D"
8             "&locationName={locationName}&format=JSON").format(locationName=LocationName)
9         response=requests.get(url)
10        weatherJson = json.loads(response.text)
11        weatherData=[]
12        for i in range(0,5):
13            weatherData.append(
14                weatherJson["records"]["location"][0][0]["weatherElement"]
15                [i]["time"][0][0]["parameter"]['parameterName'])
16        return weatherData
17
18 print(Weather.get_data("臺中市"))
```

問題 輸出 偵錯主控台 終端機

1: Python

['晴時多雲', '0', '14', '寒冷至舒適', '24']

Step.5: custom action(3) – actions.py

- 進入actions.py
- 在def name中返回action的名字
action_reply_weather。要注意致邊的名字需要與先前Domain中的命名一樣。
- 在def run中編寫要實現的功能。
透過tracker.get_slot()取得slot的值，
之後透過dispatcher.utter_message()
讓機器人發出想要傳出的訊息。

更多custom actions請查閱

<https://rasa.com/docs/rasa/api/rasa-sdk/#rasa-sdk>

```
from typing import Any, Text, Dict, List
from rasa_sdk import Action, Tracker
from rasa_sdk.executor import CollectingDispatcher
from get_weather import Weather

class ActionReplyWeather(Action):
    def name(self) -> Text:
        return "action_reply_weather"
    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
        city = tracker.get_slot("city")

        # 由於氣象局API皆使用"臺"一字，為了避免user使用"台"字造成API查詢錯誤，在這邊更正
        # 也可以在語料階段直接將"台"匹配成"臺"，如[台北](city:臺北)
        # 這邊為了降低語料方面的工作量，直接透過custom actions用程式解決。
        if city[0]=="台":
            city = "臺"+city[1:]

        data=Weather.get_data(city)
        dispatcher.utter_message(
            "{city}的天氣為{wx},降雨機率{pop}%,溫度為{min}°C至{max}°C".format(
                city=city,wx=data[0],pop=data[1],min=data[2],max=data[4]))
        return []
```

Step.5: custom action(4) – endpoints.yml

- 編寫完成後，由於custom actions不是直接調用而是利用服務來調用的。請至endpoints.yml中定義action_endpoint

```
10  # Server which runs your custom actions.
11  # https://rasa.com/docs/rasa/core/actions/#custom-actions/
12
13  action_endpoint:
14    url: "http://localhost:5055/webhook"
15
16  # Tracker store which is used to store the conversations.
17  # By default the conversations are stored in memory
```

Step.6: train&run

rasa train

訓練完後開啟兩個終端分別執行

rasa run actions

rasa shell

```
C:\Windows\System32\cmd.exe - rasa run actions
Microsoft Windows [版本 10.0.17763.1098]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。
D:\User\Desktop\RASA\examples\demo3>rasa run actions
2020-04-15 21:03:49      rasa_sdk.endpoint - Starting action endpoint server...
2020-04-15 21:03:49      rasa_sdk.executor - Registered function for 'action_reply_weather'.
```

```
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input -> 台中市天氣如何
Building prefix dict from the default dictionary ...
Loading model from cache D:\AppData\Local\Temp\jieba.cache
Loading model cost 0.609 seconds.
Prefix dict has been built successfully.
臺中市的天氣為晴時多雲,降雨機率0%,溫度為14°C至24°C
Your input -> 嘉義縣天氣如何
嘉義縣的天氣為晴時多雲,降雨機率0%,溫度為14°C至24°C
Your input -> 嘉義市天氣如何
嘉義市的天氣為晴時多雲,降雨機率0%,溫度為15°C至25°C
```