

ANNA 챗봇 공부

* cmd창에서 명령어 : `python D:\project\WANNA\source\WANNA_chatbot.py`

*구글클라우드sdk사용하기 위한 세팅

1)GCP키만들기 (*중요* 꼭 해당 프로젝트 키를 만들 것)

2)환경변수설정

`set GOOGLE_APPLICATION_CREDENTIALS=D:\project\WANNA\MyFirstProject-cc3c252f2a86.json`

3)프로젝트 세팅해주기

4)GOOGLE Cloud SDK에서 `gcloud auth activate-service-account --key-file=D:\project\WANNA\MyFirstProject-cc3c252f2a86.json` 해주기

_05.07__

*구글 스피치 성공

*스피치 > text로 답안 수신까지 성공

- 원래는 List<>형식으로 문자열 여러개를 받아서 for문으로 각각 처리
- 스피치를 사용하여 한 문장씩 하기 위해 for문을 뺄

*CountDownLatch

- 생성 시 1이상의 카운트를 인자값으로 받음
- 특정 메소드 내부에서 await()메소드로 대기 상태를 만들어줌
- 설정한 카운트가 0이 되기를 기다렸다가 0이 되는 순간 대기상태 해제

_05.10__

* fulfilltent 대략적인 학습

- firebase기본

- url로 연동해서 json 주고 받기 >> 나중에 이걸 활용해서 상황에 맞는 답안 줄 것

* 파이썬에서 entity 받음 `//response.query_result.parameters['program']`

* 파이썬에서 외부 프로그램 실행해봄

- os모듈, subprocess.call 같은 방법이 있지만 메모리를 많이 잡아먹음
- ShellExecuteA를 사용 //import ctypes

*dialogflow에서 입력해놓은 말이 아니라 트레이닝이 필요한 말은 시간이 오래걸림

05.28

*Entity 추가할 때 자동확장 기능을 체크해야 intent 자연스럽게 이해함

*python에서 입력 > dialogflow > webhook에서 처리, 응답 이런 식으로 해야 할 듯

*python에서 웹서버

: python -m http.server 8080

06.01

*webhook연결

- URL : localhost는 안됨 > aws로 해결
- aws는 새 포트 쓰려면 인바운드 규칙에 추가해야함 //5000추가함

*webhook python

- response 요소 빼뜨리면 안됨. (fulfillmentText, fulfillmentMessage)
- 각 intent 구분해봐야함

*파이썬

- 파이썬 입력은 input('입력하세요')로 받음
- webbrowser.open(url)로 웹브라우저 오픈가능
//https://zetawiki.com/wiki/%ED%8C%8C%EC%9D%B4%EC%8D%AC_%EB%B8%8C%EB%9D%BC%EC%9A%B0%EC%A0%80_%EC%8B%A4%ED%96%89
- 무한루프는 while True :
- 윈도우 종료 명령어 os.system("shutdown -s -t 초")

06.16

*도커 명령어들

docker exec awesome_docker -it /bin/bash

(docker exec -i -t anna_docker /bin/bash)

docker cp img2.jpg awesome_docker:/user_img1.jpg

docker exec awesome_docker ls

docker exec awesome_docker sh -c 'rm -rf ./*.jpg'

sudo docker exec awesome_docker python ./openface/demos/compare.py img1.jpg
img2.jpg

sudo docker exec awesome_docker python ./openface/demos/compare.py

```
/images/{lennon*,clapton*}
```

```
os.system('sudo docker start awesome_docker')
os.system('sudo docker exec awesome_docker python ./openface/demos/compare.py
img1.jpg img2.jpg')
os.system('sudo docker stop awesome_docker')
*얼굴인식
-도커로 openface실행
```

08.07

*안드로이드 전화&문자 수신

1)매니페스트에 권한 설정

```
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

2)리시버도 추가

```
<receiver android:name=".클래스명">
    <intent-filter>
        <action android:name="android.intent.action.PHONE_STATE"/>
        //문자도
    </intent-filter>
</receiver>
```

3)클래스 생성 후 브로드캐스트 상속

4)on receive() 메소드 처리

5)어플이 실행되지 않은 상태에서도 백그라운드 작업하고 싶으면 service추가

08.13

*윈도우 토스트 알림메세지

1)파이썬 win10toast 라이브러리 사용 //C#도 가능한 듯

```
from win10toast import ToastNotifier
toaster = ToastNotifier()
toaster.show_toast("title", "content", icon_path="custom.ico", duration=2)
//타이틀,내용,아이콘경로(None),길이
```

2)jQuery

- 참고 :

https://zetawiki.com/wiki/JQuery_%ED%94%8C%EB%9F%AC%EA%B7%B8%EC%9D%B8_toast_%EC%8B%A4%EC%8A%B5

*파이썬 소켓통신

-Server

```
import socket
```

```

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind(('127.0.0.1', 5001))
server_socket.listen(0)
client_socket, addr = server_socket.accept()
data = client_socket.recv(65535)
print(data)

```

- Client

```

import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(('127.0.0.1', 5001))
sock.send('Hello')

```

08.29

*안드로이드 백그라운드 실행

- 안드로이드 O버전 부터 백그라운드 실행이 제한됨 > startService 제대로 동작 x
- startForegroundService 사용하여 해결
- 참고 링크

<https://medium.com/til-kotlin-ko/android-o%EC%97%90%EC%84%9C%EC%9D%98-%EB%B0%B1%EA%B7%B8%EB%9D%BC%EC%9A%B4%EB%93%9C-%EC%B2%98%EB%A6%AC%EB%A5%BC-%EC%9C%84%ED%95%9C-jobintentservice-250af2f7783c>

09.15

*PyQt5 사용법

- 1) Designer로 디자인 후 .ui파일 생성
- 2) python -m PyQt5.uic.pyuic 파일명.ui -o 생성파일명.py -x || uiExec2.py 파일
- 3) 생성된 파일 실행 시 ui 실행됨

- PyQt5 위에 타이틀바? 없애기

```
self.ui.setWindowFlags(QtCore.Qt.FramelessWindowHint)
```

- PyQt5 슬롯(이벤트같은거) 연결

09.16

*PyQt5 Widget/Dialog/window 차이 : <http://kit2013.tistory.com/242>

09.18

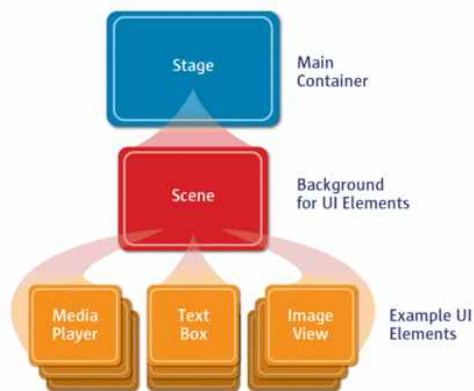
***JavaFx 설치방법** (<https://code.makery.ch/kr/library/javafx-tutorial/part1/>)

- 1) Help>Install New..>¹⁾JavaFx주소입력 > e(fx)clipse-install 선택 후 설치
- 2) SceneBuilder 설치 > 자바 Preferences 에서 환경설정 하고 연결해주기

***JavaFx 사용법** //자세한건 위 링크에서 예제보기

- 1) File/New/Other/JavaFX Project 생성 //자동으로 생성되는 application패키지 지우기
- 2) MVC나누어 패키지 생성
- 3) View 패키지에 FXML Document 생성 > 우클릭 후 "Open with Scene Builder"
- 5) Control 패키지에 MainApp.class 생성 후 스테이지와 씬 호출
- 6) Model 패키지엔 데이터 관련 클래스 추가 / View 패키지엔 view와 관련된 컨트롤러 추가
- 7) 컨트롤러와 Main연결
- 8) 뷰에 컨트롤러 설정 : SceneBuilder/Controller에서 관련 클래스 설정 > 각 위젯 Code에서 fx:id설정

***JavaFx 구조**



***JavaFX 기타**

- setCellValueFactory() : 객체의 어떤 필드를 각 열에 사용할지 결정
- ex. `firstNameColumn.setCellValueFactory(cellData -> cellData.getValue().firstNameProperty());` //StringProperty가 아니면 .asObject() 추가

09.21

***Jfoenix 요소 사용법**

- 1) 씬 빌더에서 Jfoenix.jar추가
- 2) 자바에서 lib 추가
- 3) 사용

1) <http://download.eclipse.org/efxclipse/updates-released/3.3.0/site/>

09.23

*상세검색 명령어

1)번역

- 파파고 사이트 실행, sk=입력한 문장의 언어, tk=번역할 언어, st=대상 문장
- 한국어로 번역해 달라하면 sk=auto, tk=ko
- translation = "https://papago.naver.com/?sk=ko&tk=en&st=메롱"

2)내 주변 맛집 || 특정 지역 맛집

- location변수 추가 하여 주변(잘 안됨)/특정 지역으로 바꿔 검색
- 뒷 키워드는 식당보단 맛집으로 검색하는게 더 유용
- restaurant = "https://www.google.co.kr/search?q=주변+맛집&tbm=lcl"

3)대중교통 정보(길찾기) "~에서~까지 어떻게 가?, 대중교통정보.."

- 명령어 분석에 따라 출발지/목적지만 적절히 바꿔주기
- transport = "https://www.google.co.kr/maps/dir/신도림역/영등포역"

*EC2 python3 설정

- yum install python36
- yum install python36-devel?
- /usr/bin/pip 심볼릭 링크 만들기 (실워치는 /usr/local/bin/pip3.6)
- 파일 실행시 python3로 실행

10.19

*python to java

- 자이썬이란 걸 찾아봤지만 라이브러리들 때문에 실패
- 결국 cmd명령어를 사용해서 야매로 실행하는 법으로 연동 성공

*구글 캘린더 연동

- 구글 API를 사용하기위해 gradle설치
- gradle 설치방법 :
 - 1) https://gradle.org/gradle-download/ 사이트에서 Binary only distribution 클릭하여 다운로드
 - 2) 압축 풀고 환경 변수 설정
 - GRADLE_HOME = 압축풀 경로
 - Path에 %GRADLE_HOME%\bin; 추가
 - 3)이클립스>Help>이클립스 마켓플레이스에서 그라들 설치
 - 4)나머지는 내가 세팅해봤으니까 아마 안건들여도 되지 않을까!?

10.25

***java 트레이아이콘 & UI숨기기**

1) 트레이아이콘

- trayIcon 옵션 설정 후 (이미지 등..)
- `SystemTray.getSystemTray().add(trayIcon);`
- 자세한 내용 : <https://blog.silentsoft.org/archives/6>

2) UI숨기기

- `Stage.hide()` / `Stage.show()` 사용
- 그냥 hide만 사용 시 자동종료 되어버림..

3) 자동종료 안되게

- `start()`에 `Platform.setImplicitExit(false);` 추가
//true면 마지막창 닫으면 자동종료, false면 호출 전까지 정상실행
- `show()`부분 수정

```
Platform.runLater(new Runnable() {  
    @Override  
    public void run() {  
        primaryStage.show();  
    }  
});
```

- `hide()`부분 수정

```
Platform.runLater(new Runnable() {  
    @Override  
    public void run() {  
        if(SystemTray.isSupported()) {  
            primaryStage.hide();  
        }else {  
            System.exit(0);  
        }  
    }  
});
```

10.27

*java to exe

1) Runnable Jar 파일 만들기

- 원래대로라면 프로젝트 export하면 되지만 메이븐 때문인지 되지 않았음
- 그래서 maven build로 생성함
- maven에 plugin 3개 추가 + mainclass지정
- run config..에서 clean install로 Run하면 jar파일 생성

2) 해당 jar파일 launch4j를 사용해 exe파일로 변경

- classpath도 설정해줌

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-shade-plugin</artifactId>
  <version>2.3</version>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-jar-plugin</artifactId>
  <configuration>
    <archive>
</manifest>
<addClasspath>true</addClasspath>
<classpathPrefix>lib/</classpathPrefix>
<mainClass>controller.MainApp</mainClass>
</manifest>
</archive>
</configuration>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-dependency-plugin</artifactId>
  <executions>
    <execution>
      <id>copy</id>
      <phase>install</phase>
      <goals>
        <goal>copy-dependencies</goal>
      </goals>
      <configuration>
        <outputDirectory>
          ${project.build.directory}/lib
        </outputDirectory>
      </configuration>
    </execution>
  </executions>
</plugin>
```


11.14

*PyQt5 빌드

1)ui파일 로드

```
import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

form_class = uic.loadUiType("D:\\project\\ANNA\\source\\UI\\mainUI.ui")[0]

#메인 페이지로드
class MainWindow(QMainWindow, form_class):
    def __init__(self):
        super().__init__()
        self.setupUi(self)

if __name__ == "__main__":
    app = QApplication(sys.argv)
    myWindow = MainWindow()
    myWindow.show()
    app.exec_()
```

2)리소스빌드

pyrcc5 D:\project\ANNA\source\UI\resource.qrc -o resource_rc.py

*pyQt designer tip

- 스타일 시트 속성

1) 탭바 선택메뉴 색상 QTabBar::tab:selected {background: rgb(245, 114, 119)}

2) 텍스트 색상 color: rgb(255, 255, 255);

*pyQt 버튼이벤트 & 화면전환

1) qtdesigner에서 signal&slot등록

- 시그널 : 발생한 이벤트

- 슬롯 : 이벤트 핸들러

2) 버튼이벤트 등록 (위에서 지정한 슬롯 이름으로)

```
def handleSignin(self): #로그인 검사 후 메인으로
    self.newWindow = MainWindow()
    self.newWindow.show()
    self.hide()
```

- 3) 화면전환을 위해 class 새로 생성 후 show() & 현재화면 hide()
 - self 꼭 써주기

*python에서 구글캘린더 사용

- 1) 설치 : pip install --upgrade google-api-python-client oauth2client
 2) 예제 사용

11.17

*pyqt5 calendarwidget 선택한 숫자 가져오기

- slot등록 해놓고

```
date = self.calendarWidget.selectedDate()
str_date = str(date.year())+"-"+str(date.month())+"-"+str(date.day())
self.lblSelectedDate.setText(str_date)
```

- 그냥 date.toString()으로 출력하면 요일 일 월 년 순으로 출력
 - date.year(), date.month(), date.day() 사용하여 년월일 출력
 - date.shortDayName(date.dayOfWeek()) 으로 요일 출력가능
 - 날짜 설정은 setSelectedDate(const QDate &date)

*calendarwidget 원하는 요일 노란색으로 표시

```
fm = QTextCharFormat()
fm.setForeground(Qt.red)
fm.setBackground(Qt.yellow)

for dday in holidays:
    dday2 = QDate.fromString(dday, "yyyyMMdd")
    self.cal.setDateTextFormat(dday2, fm)
```

11.18

*qDateTime >> python datetime

- QDateTimeEdit에서 선택된 시간 가져와서 파이썬 datetime으로 변경
 startT = self.edit_startTime.dateTime().toPyDateTime()

- 이걸RFC3339 형식으로 바꾸려면
 startT.isoformat()

*python self



- 메소드를 호출하면 자동으로 호출한 객체가 매개변수로 들어감
- 그래서 항상 self를 써주는 듯

11.20

*python to exe

- pyinstall 설치
- 환경변수 설정(python, python%scripts)
- 필요한 옵션 넣어서 실행
- 자세한 내용 : <http://mainia.tistory.com/5610>

[실행1:실패]

- 문제1 : PyQt5Wbin을 Path에 추가해줘야함
- 문제2 : WARNING: lib not found: api-ms-win-core-*.dll 에러발생
> Window SDK 설치하고 dll 파일이 있는 폴더를 pyinstaller 옵션 -p로 지정해주면 됨
> SDK : <https://developer.microsoft.com/en-us/windows/downloads/windows-10-sdk>
- 문제3 : ModuleNotFoundError: No module named 'PyQt5.sip'
> --hidden-import 옵션 써서 sip 있다고 알려주면 됨
- 참고사이트 : <https://winterj.me/pyinstaller/>
- 참고사이트 : <http://tina0430.tistory.com/34>

[실행2]

- 최종명령어 :

11.28

*폰연동

- Server : AWS
- Client : ANNA PC / ANNA APP
- pc에서 접속 시 pc소켓 따로 저장 후 스레드 >> 후에 스레드로 얼굴인식 등 작업할 것
- phone에서 알람 전송 시 저장해둔 pc소켓에 알람 보내줌
- 나중에 pc-phone 일치 여부 추가하기
- 참고 사이트 : <http://qkqhxl1.tistory.com/189> [파이썬 소켓통신]

11.30

*TTS

- 1) google text to speech api 사용
 - [문제] : mp3파일로만 저장되고 읽지 못함.
 - vlc 라이브러리를 사용해 보았지만 실행되지 않음
- 2) pyttsx3 라이브러리 사용
 - pip install pyttsx3
 - [문제] : pywintypes.com_error: (-2147352573, '구성원이 없습니다.', None, None)
 - [해결] : D:\program\Python\Python36-32\Scripts\pywin32_postinstall.py 실행

12.12

*캘린더 사용자 별로 구분

```
def __init__(self,USERID):  
    store = file.Storage('token_'+USERID+'.json')  
    creds = store.get()  
    if not creds or creds.invalid:
```

- 토큰파일에 사용자 id를 붙여 사용자 별로 토큰파일을 따로 만들어 해결