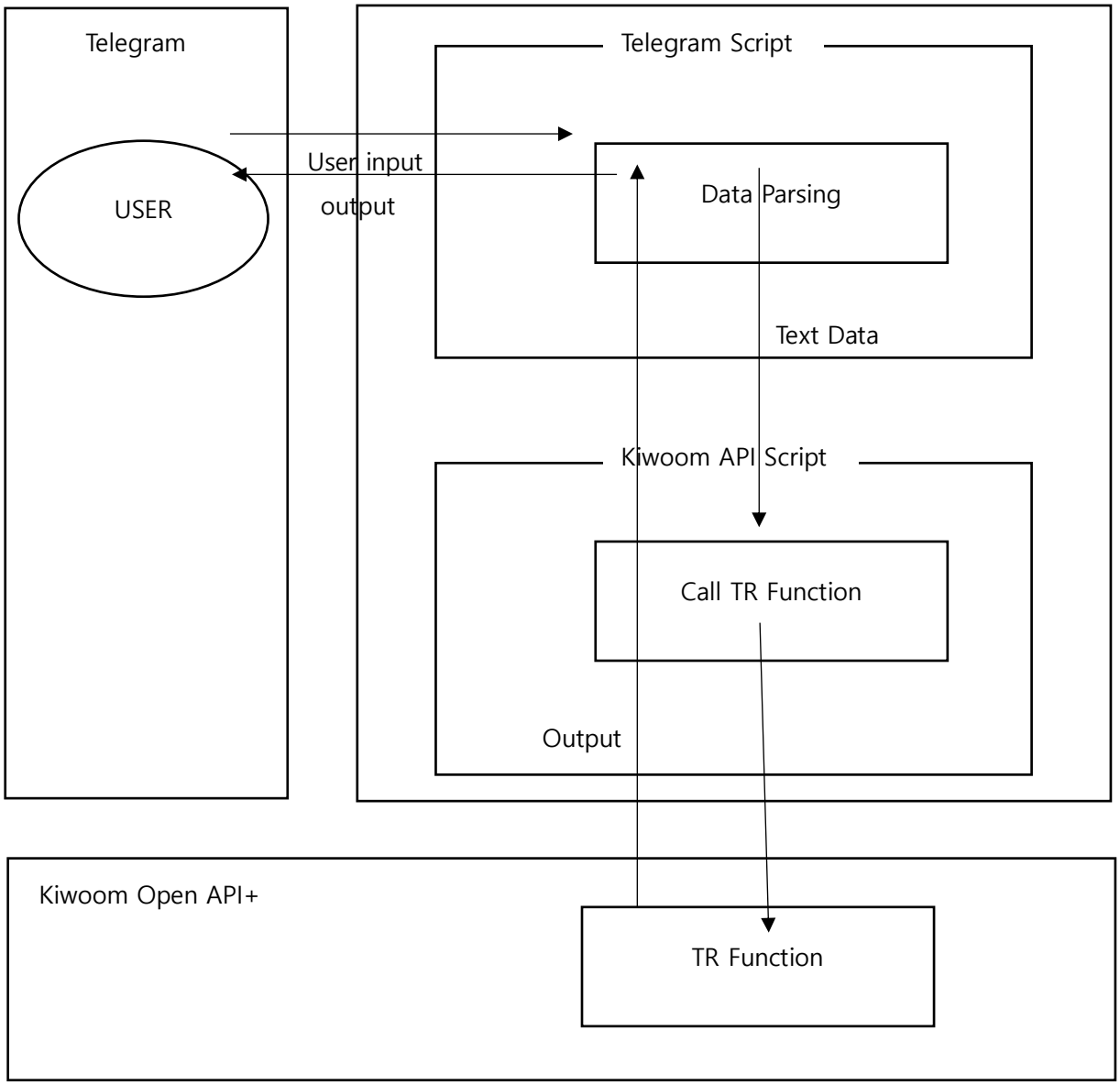
오픈소스 SW Architecture Design

소프트웨어학과 32173154 이승현

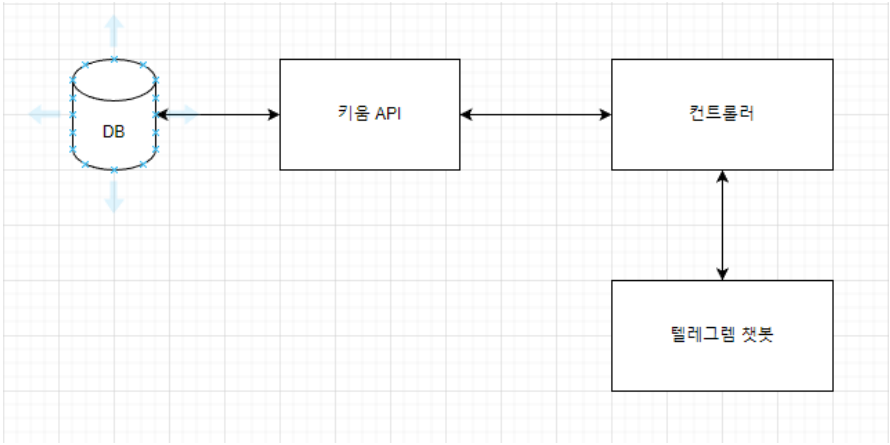
소프트웨어학과 32172086 석홍준

[전체적인 아키텍쳐 디자인]



[사용 기술]

- **웹 애플리케이션 구조**



-DB: 주식장의 데이터 베이스

-키움 API:데이터 베이스로부터 원하는 종목의 데이터를 추출하는 API

-컨트롤러: 키움 API로부터 받은 데이터를 가공하고 텔레그램 챗봇에 전송 및 텔레그램 으로부터 request 수신.

-텔레그램 챗봇: 사용자의 요청을 받고 이를 컨트롤러에 송신 및 이에 적합한 Response 수신

- **키움 Open API+ 사용**
  - **기능요청->이벤트 호출->데이터 획득**
  - **키움에서 제공하는 API 사용설명서**

5.2 OpenAPI 식별자

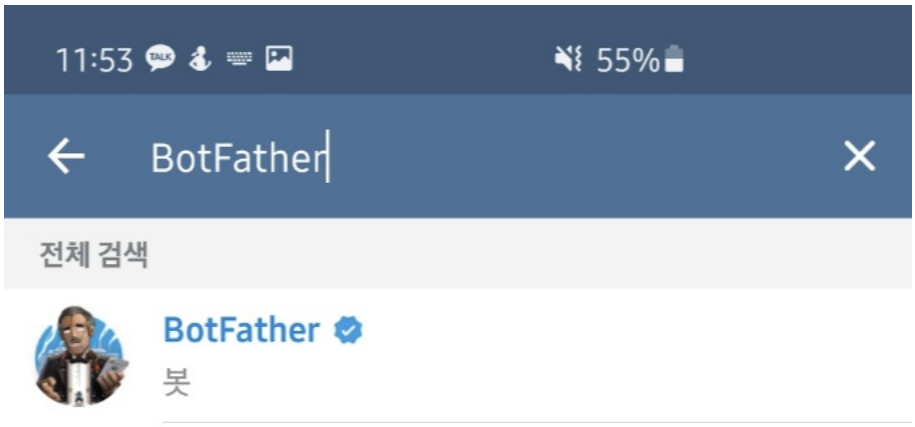
- OpenAPI 컨트롤의 인터페이스 식별자(Interface ID : GUID)는 유일성(Unique)을 지니며 이중으로 등록 되어서는 안된다.
- OpenAPI 컨트롤의 생성 및 데이터 상호 교환을 위한 컨트롤 인터페이스 식별자는 다음과 같다.

구 분	식별자	내용
Control	A1574A0D-6BFA-4BD7-9020-DED88711818D	컨트롤 클래스
Dispatch	CF20FBB6-EDD4-4BE5-A473-FEF91977DEB6	프로퍼티/메소드 제어
Event	7335F12D-8973-4BD5-B7F0-12DF03D175B7	이벤트 제어
Module	6D8C2B4D-EF41-4750-8AD4-C299033833FB	컨트롤 모듈

5.3 OpenAPI 메소드

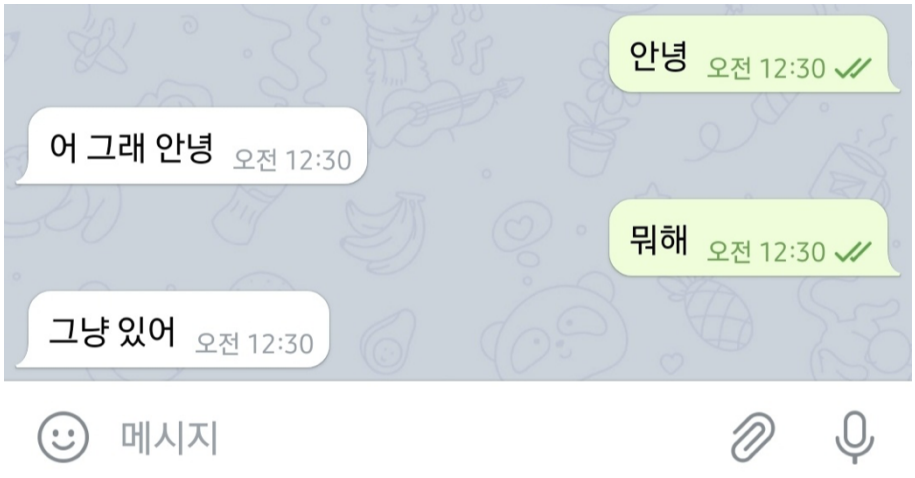
- OpenAPI 제어 및 편집을 위한 인터페이스 메소드(Interface Method) 목록은 다름과 같다.

ID	타입	이름	설명
1	LONG	CommConnect	로그인 윈도우를 실행한다.
2	void	CommTerminate	**더 이상 지원하지 않는 함수**
3	LONG	CommRqData	통신 데이터를 송신한다.
4	BSTR	GetLoginInfo	로그인 정보를 반환한다.
5	LONG	SendOrder	주식주문 Tran을 송신한다.
6	LONG	SendOrderCredit	주식 신용주문 Tran을 송신한다.
7	void	SetInputValue	
8	LONG	SetOutputFID	

- **텔레그램 챗봇 사용**
  - **Father Bot 을 통한 기본 챗봇 생성**
    - 

    - **생성하고자 하는 봇의 이름을 입력하고, 생성되는 토큰을 통하여 스크립트로 제어가능**
    - ```
      bot = telegram.Bot(token)
      bot.sendMessage(chat_id=id, text="테스트 중입니다.")

      # updater
      updater = Updater(token=token, use_context=True)
      dispatcher = updater.dispatcher
      updater.start_polling()

      # 사용자가 보낸 메세지를 읽어들이고, 답장을 보내줍니다.
      # 아래 함수만 입맛에 맞게 수정해주면 됩니다. 다른 것은 건들 필요없어요.
      def handler(update, context):
          user_text = update.message.text # 사용자가 보낸 메세지를 user_text 변수에 저장합니다.
          if user_text == "안녕": # 사용자가 보낸 메세지가 "안녕"이면?
              bot.send_message(chat_id=id, text="어 그래 안녕") # 답장 보내기
          elif user_text == "뭐해": # 사용자가 보낸 메세지가 "뭐해"면?
              bot.send_message(chat_id=id, text="그냥 있어") # 답장 보내기
      ```
    - 

  - ■
```
{'message_id': 3, 'date': 1615563765, 'chat': {'id': 1618277280, 'type': 'private', 'first_name': '원준'}, 'text': '대답해!', 'entities': [], 'c
{'message_id': 4, 'date': 1615563787, 'chat': {'id': 1618277280, 'type': 'private', 'first_name': '원준'}, 'text': '안녕!!!!!!', 'entities': [],
```

채팅시 Bot 에서 넘어오는 Json Request 를 가공하여 사용할 계획
```

실제 수행 시나리오

1. 사용자가 Kiwoom_bot 에 원하는 커맨드를 입력한다.

2. input 을 polling 하고 있던 updater 가 입력값을 인지하면 데이터를 받아오고, 이를 파싱하여 알맞는 함수를 호출한다.

ex) /info : BSTR GetCommRealData(LPCTSTR strCode, long nFid)

(실시간 시세데이터 반환, 옵션을 입력했을 경우 내부에서 로직 처리)

/news : 셀레니움을 이용한 네이버 주식 크롤링을 통해 뉴스 데이터를 가져옴

/search : BSTR GetConditionNameList()

조건 검색의 조건 리스트를 받아온다.

BOOL SendCondition(LPCTSTR strScrNo, LPCTSTR strConditionName, int nIndex, int nSearch)

조건검색 종목조회 TR 을 송신한다.

에러처리) 알맞는 명령어가 아닌 경우, 명령어 리스트를 다시 띄워주며 사용자에게 알려준다.

3. 처리한 데이터를 chatting bot script 를 통해 사용자에게 전달한다.

[진행 상황]

-Koa Studio 설치를 통한 api 참고

-Kiwoom bot 기초 명령어 테스트

```
import telegram
from telegram.ext import Updater
from telegram.ext import MessageHandler, Filters

commands = ['/info', '/alarm', '/news', '/search']
token = '6068822535:AAEdX6RZ9_7N85NtDsyYornSU5LeoH9tEdI'

bot = telegram.Bot(token=token) # 토큰으로 봇 생성
updater = Updater(token=token, use_context=True) # 업데이터 생성
dispatcher = updater.dispatcher # 디스패처 객체 생성

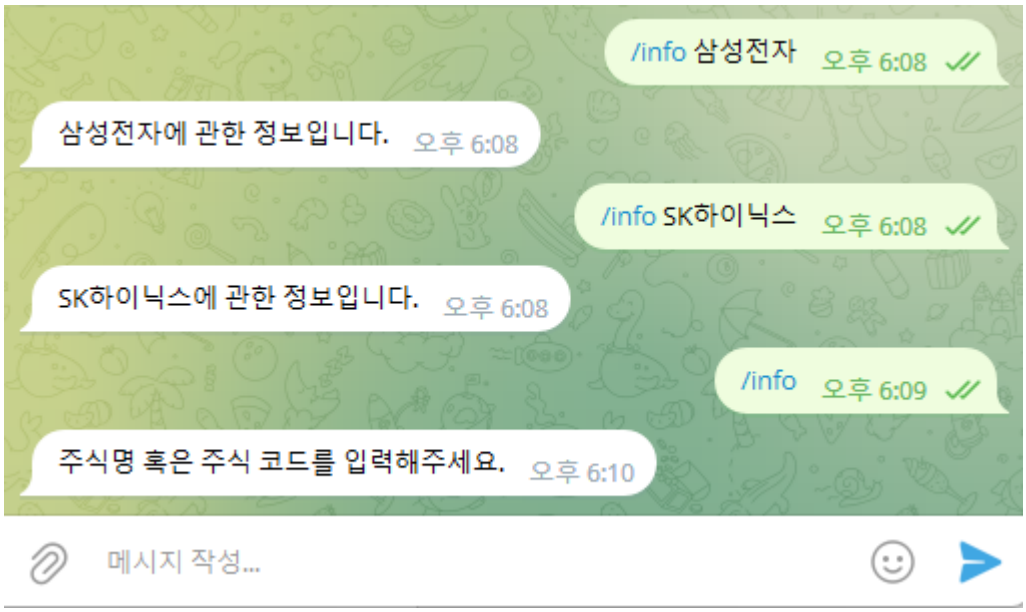
updater.start_polling() # 지속적으로 값 받아옴(polling 시작)


def handler(update, context):
    user_text = list(update.message.text.split()) # 데이터가 입력되면 메세지에서 text 값을 파싱하여 가져옴
    user_id = update.message.chat.id # 마찬가지로 id값 가져옴
    print(user_text)
    if user_text[0] == '/info': # 명령어 검증
        if len(user_text) == 1:
            bot.sendMessage(chat_id=user_id, text='주식명 혹은 주식 코드를 입력해주세요.')
        else:
            bot.sendMessage(chat_id=user_id, text=user_text[1] + '에 관한 정보입니다.')
    else:
        bot.sendMessage(chat_id=user_id, text="유효한 명령어가 아닙니다.\n 명령어 리스트: '/info', '/alarm', '/news', '/search'")


echo_handler = MessageHandler(Filters.text, handler)
dispatcher.add_handler(echo_handler) # 디스패쳐에 핸들러 드록
```

파더봇을 통함 Kiwoom bot 을생성한 후 스크립트를 통한 명령어 출력 테스트

-텔레그램 톡 화면



[테스트 시간이 장 이후 시간이라 다른 응답으로 대체]