

# 상세 설계서

## 오픈소스 기여

-모듈 관계도

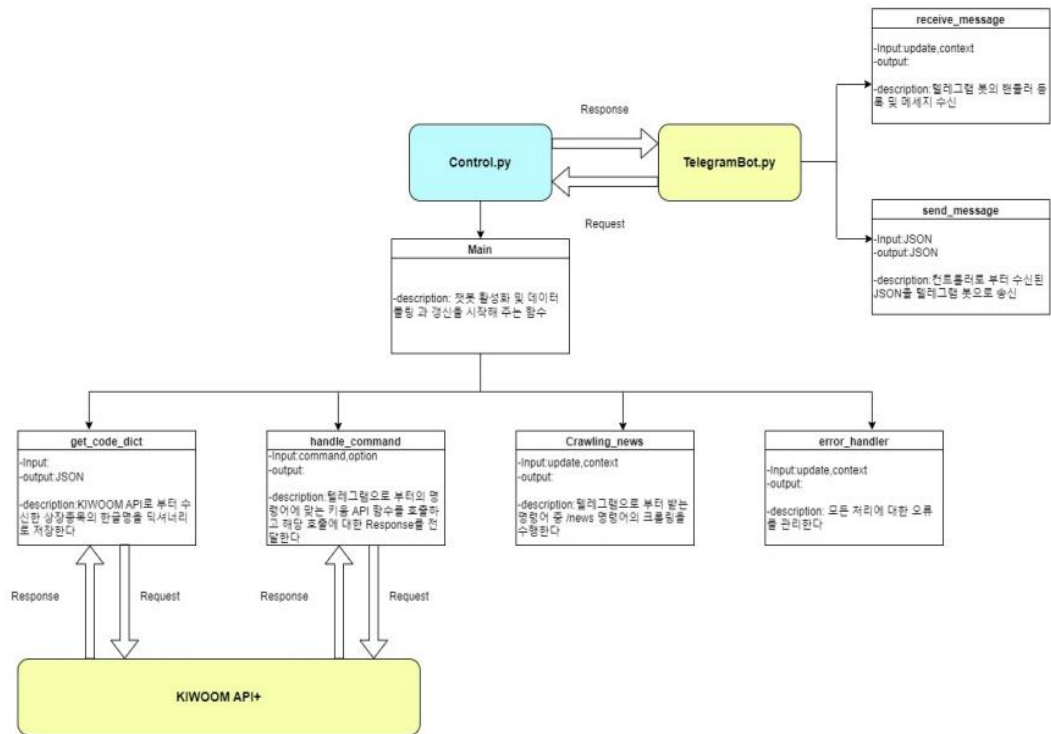
-API 상세 설계

32173154 이승현

32172086 석홍준



## 모듈 관계도



# -KIWOOM API+ 모듈

Command) /info : 실시간 시세 데이터 반환

## GetCommRealData

함수 원형	BSTR GetCommRealData(LPCTSTR strCode, long nFid)
설명	실시간 시세 데이터 반환
입력값	strCode - 종목 코드 nFid - 실시간 아이템
반환값	수신 데이터
비고	현재가 출력 - openApi.GetCommRealData("039490", 10) strCode는 OnReceiveRealData 첫번째 매개변수를 사용

## OnReceiveRealData

함수 원형	void OnReceiveRealData(LPCTSTR sJongmokCode, LPCTSTR sRealType, LPCTSTR sRealData)
설명	실시간데이터를 받은 시점을 반환
입력값	sJongmokCode - 종목코드 sRealType - 리얼타입 sRealData - 실시간 데이터전문
반환값	X
비고	X

## Command) /search : 조건 검색의 조건 리스트 받아오기

### GetConditionNameList

함수 원형	BSTR GetConditionNameList()
설명	조건검색 조건명 리스트를 받아옴
입력값	
반환값	조건명 리스트(인덱스^조건명)
비고	조건명 리스트를 구분(',') 사용하여 받아온다. Ex) 인덱스1^조건명;인덱스2^조건명;인덱스3^조건명...

### SendCondition

함수 원형	BOOL SendCondition(LPCTSTR strScrNo, LPCTSTR strConditionName, int nIndex, int nSearch)
설명	조건검색 종목조회TR송신
입력값	LPCTSTR strScrNo : 화면번호 LPCTSTR strConditionName : 조건명 int nIndex : 조건명인덱스 int nSearch : 조회구분(0:일반조회, 1:실시간조회, 2:연속조회)
반환값	성공 1, 실패 0
비고	단순 조건식에 맞는 종목을 조회하기 위해서는 조회구분을 0으로 하고, 실시간 조건검색을 하기 위해서는 조회구분을 1로 한다. OnReceiveTrCondition으로 결과값이 온다. 연속조회가 필요한 경우에는 응답받는 곳에서 연속조회 여부에 따라 연속조회를 송신하면된다.

### OnReceiveTrCondition

함수 원형	void OnReceiveTrCondition(LPCTSTR sScrNo, LPCTSTR strCodeList, LPCTSTR strConditionName, int nIndex, int nNext)
설명	조건검색 조회응답으로 종목리스트를 구분자(",")로 붙어서 받는 시점
입력값	LPCTSTR sScrNo : 종목코드 LPCTSTR strCodeList : 종목리스트(",")로 구분 LPCTSTR strConditionName : 조건명 int nIndex : 조건명 인덱스 int nNext : 연속조회(2:연속조회, 0:연속조회없음)
반환값	X
비고	X

**Command) /alarm : 사용자가 설정한 조건이 만족되었을 경우 알림을 전송**

#### SendCondition

함수 원형	BOOL SendCondition(LPCTSTR strScrNo, LPCTSTR strConditionName, int nIndex, int nSearch)
설명	조건검색 종목조회TR송신
입력값	LPCTSTR strScrNo : 화면번호 LPCTSTR strConditionName : 조건명 int nIndex : 조건명인덱스 int nSearch : 조회구분(0:일반조회, 1:실시간조회, 2:연속조회)
반환값	성공 1, 실패 0
비고	단순 조건식에 맞는 종목을 조회하기 위해서는 조회구분을 0으로 하고, 실시간 조건검색을 하기 위해서는 조회구분을 1로 한다. OnReceiveTrCondition으로 결과값이 온다. 연속조회가 필요한 경우에는 응답받는 곳에서 연속조회 여부에 따라 연속조회를 송신하면 된다.

#### SendConditionStop

함수 원형	Void SendConditionStop(LPCTSTR strScrNo, LPCTSTR strConditionName, int nIndex)
설명	조건검색 실시간 중지TR을 송신
입력값	LPCTSTR strScrNo : 화면번호 LPCTSTR strConditionName : 조건명 int nIndex : 조건명인덱스
반환값	X
비고	해당 조건명의 실시간 조건검색을 중지하거나, 다른 조건명으로 바꿀 때 이전 조건명으로 실시간 조건검색을 반드시 중지해야한다. 화면 종료시에도 실시간 조건검색을 한 조건명으로 전부 중지해줘야 한다.

- /search에도 동일하게 적용

## 에러처리

리턴 코드		설명
0	OP_ERR_NONE	정상처리
-10	OP_ERR_FAIL	실패
-11	OP_ERR_COND_NOTFOUND	조건번호 없음
-12	OP_ERR_COND_MISMATCH	조건번호와 조건식 틀림
-101	OP_ERR_CONNECT	서버접속 실패
-105	OP_ERR_INPUT	함수입력값 오류
-203	OP_ERR_NO_DATA	데이터 없음
-205	OP_ERR_DATA_RCV_FAIL	데이터 수신 실패
-300	OP_ERR_ORD_WRONG_INPUT	입력값 오류
-500	OP_ERR_ORD_SYMCODE_EMPTY	종목코드 없음

## 종목 코드와 종목 명 dict처리

### GetCodeListByMarket

함수 원형	BSTR GetCodeListByMarket(LPCTSTR sMarket)
설명	시장 구분에 따른 종목 코드 리스트 반환
입력값	sMarket – 시장구분
반환값	종목코드 리스트, 종목간 구분은 ‘;’이다.
비고	sMarket – 0:장내, 3:ELW, 4:뮤추얼펀드, 5:신주인수권, 6:리츠, 8:ETF, 9:하이일드펀드, 10:코스닥, 30:K-OTC, 50:코넥스(KONEX)

### GetMasterCodeName

함수 원형	BSTR GetMarsterCodeName(LPCTSTR strCode)
설명	종목 코드 한글명 반환
입력값	strCode – 종목코드
반환값	종목한글명
비고	장내외, 지수선물옵션, 주식선물옵션 검색 가능

# TelegramBot 모듈

## Receive\_message

함수 원형	def receive_message(Update,Context)
설명	Telegram 봇으로 부터 사용자의 메시지를 수신
입력값	Update:봇의 정보와 사용자의 정보 Context:사용자의 메시지
반환값	X
비고	정해진 명령어인 /info,/news,/search,/help 이외의 명령어는 예외 처리를 진행해 줘야하며 명령어에 맞는 모듈을 호출해 줘야한다.

## Send\_message

함수 원형	def send_message(marketname,info,uid)
설명	Control 모듈로 부터 받는 데이터의 메시지를 사용자에게 전달함
입력값	MarketName: 사용자가 요구한 시장의 한글명 Info:사용자가 요구한 명령어에 맞는 정보 Uid: 명령 요청 사용자의 Uid
반환값	X
비고	

# Control 모듈

## get\_code\_dict

함수 원형	def get_code_dict()
설명	Kiwoom 증권으로 부터 상장한 마켓의 종목코드 및 종목명을 받아 딕셔너리 처리
입력값	X
반환값	Dictionary_market
비고	Time 이벤트 스크립트를 통해 매일 9시에 함수가 실행 되도록 한다.

## Handle\_command

함수 원형	def handle_command(command,uid,option)
설명	Telegram 봇으로 부터 얻은 사용자의 명령어에 맞는 함수를 실행 및 관리
입력값	Command:사용자의 명령어 Uid:사용자의 Uid Option: 사용자의 명령어에 부수적인 옵션
반환값	MarketName: 사용자가 요청한 시장의 한글명 Uid: 사용자의 Uid Info: 사용자가 요청한 시장의 정보
비고	정해진 명령어인 /info,/news,/search,/help 이외의 명령어는 예외 처리를 진행해 줘야하며 명령어에 맞는 모듈을 호출해 줘야한다.



## Crawling\_news

함수 원형	def crawling_news(topic)
설명	/news 명령어를 수행하여 해당 토픽에 관련한 뉴스기사 크롤링
입력값	Topic: 검색하고자 하는 뉴스 토픽
반환값	<a href="#">News: 해당</a> 토픽에 관련한 제일 앞의 뉴스 글 Uri: 해당 토픽에 관련한 뉴스 글 검색 결과 Uri
비고	외부 모듈인 Selenium 과 BS4의 사용이 불가피

## 예시

```
from selenium import webdriver
```

```
from bs4 import BeautifulSoup
```

### chrome 브라우저를 사용한 데이터 크롤링

```
# 크롬 드라이버 경로 설정
```

```
driver_path = "/path/to/chromedriver"
```

```
# 크롬 드라이버 실행
```

```
driver = webdriver.Chrome(driver_path)
```

```
# 네이버 주식 뉴스 URL
```

```
url = "https://finance.naver.com/news/news_list.nhn?mode=RANK&date=20220418"
```

```
# 페이지 로드
```

```
driver.get(url)
```

```
# 페이지 소스코드 가져오기
```

```
html = driver.page_source
```

```
# BeautifulSoup 객체 생성
```

```
soup = BeautifulSoup(html, "html.parser")
```

```
# 뉴스 제목 추출
```

```
titles = soup.select("ul.realtimeNewsList dl dt a")
```