

ORM Lab assignment

새로운 세상에 오신 여러분들을 환영합니다. 이번 과제는 우리가 이때까지 배운 모든 것을 한번에 쏟아내는 스파게티 같은 과제입니다(토주의). 파이썬 때 배운 OOP와 크롤링 부터 시작해서, CSV 그리고 DB와 ORM을 모두 엮어서 하나의 과제를 수행한다. 엄청 재밌을거 같다. 어서 빨리 과제에 도전해보자

Purpose

이번 과제의 목적은 ORM을 이해하는 것이다. 이를 위해 아래 task를 수행한다.

- 본 과제 목적에 맞게 DB를 설계한다.
- 설계된 DB에 맞도록 ORM 코드를 작성한다.
- 웹에서 특정 데이터를 스크래핑해서 수집한다.
- 수집된 데이터를 ORM 코드를 이용해서 DB에 저장한다.
- 저장된 정보를 Query할 수 있는 모듈을 개발한다.

Specification

데이터의 수집

본 과제는 먼저 미국의 주식 데이터 정보를 모아, 이를 DB화하는데 목적이 있다. 우리가 데이터를 가져올 곳은 미국의 구글 금융 서버입니다. 역시 구글 신은 다양한 데이터를 제공하고 있습니다(우리에게 제공할 속제 데이터도 관리하면서요^^.) 데이터는 CSV로 제공됩니다. 간단히 아래의 URL을 여러분의 웹 브라우저에 입력봅시다.

<https://finance.google.co.uk/bctzjpsun/historical?q=KRX:005930&startdate=Jan+01+2000&enddate=Mar+16+2018&output=csv>

위 주소의 메인 서버 주소는 <https://finance.google.co.uk/bctzjpsun/historical?> 이며 그 뒤로 나오는 q, startdate, enddate 등의 값에 입력에 따라 다양한 주식 정보 데이터를 다운로드 받을 수 있습니다. 설정할 수 있는 값은 아래와 같습니다.

키	설명	값	비고
q	종목(심볼)	KRX:005930	한국 Samsung Electronics Co. Ltd.
startdate	시작 월-일-년	Jan+01+2000	
enddate	종료 월-일-년	Mar+16+2018	
output	다운로드 데이터 Type	csv	csv 타입 데이터만 지원

위의 값을 자유롭게 바꾸면 다양한 주식 정보를 구할 수 있습니다. 예를 들면, q=KRX:005380 를 입력한다면 현대 자동차의 주식정보를 다운로드 받을 수 있습니다. 참고로 앞에 KRX는 한국 거래소를 뜻하는 약자입니다.

기업들은 아래와 같이 다양한 기업들을 선택할 수 있고, 기업코드만 안다면 우리나라 전체 기업의 주식 정보를 기간을 정해서 다운로드 받을 수 있습니다.

종목 심볼	종목명	종목 심볼	종목명
005930	삼성전자	032830	삼성생명
005380	현대차	051910	LG화학
005490	POSCO	009540	현대중공업
012330	현대모비스	017670	SK텔레콤
000660	SK하이닉스	105560	KB금융

035420	NAVER	096770	SK이노베이션
005935	삼성전자우	023530	롯데쇼핑
015760	한국전력	086790	하나금융지주
055550	신한지주	000810	삼성화재
000270	기아차	066570	LG전자

우리는 우리나라 기업들의 증권 종목코드를 먼저 알아야합니다. 이를 위해서는 종목 코드를 가져와야 하는데 종목코드를 가져오는 건 아래 두 사이트를 참고하셔서 가져오시기 바랍니다. 참고로 엑셀을 다운로드 받아서 **pandas**로 코드를 처리해도 무방합니다.

- <http://bigdata-trader.com/itemcodehelp.jsp>
- <http://tradingidea.tistory.com/entry/%EC%A3%BC%EC%8B%9D-%EC%BD%94%EB%93%9C-%EA%B0%80%EC%A0%B8%EC%98%A4%EA%B8%B0>

DB의 구성

데이터를 다운로드 받을 준비를 마쳤다면 **DB**를 구성하기 위해 **DB** 설계를 해야 합니다. 꼭 아래 처럼 하실 필요는 없지만 아래 블로그를 참고하시기 바랍니다.

<http://devse.tistory.com/entry/%EC%BD%94%EC%8A%A4%ED%94%BC-%EC%A3%BC%EA%B0%80-%EB%8D%B0%EC%9D%B4%ED%84%B0-17%EB%85%84%EC%B9%98-20000104-%EB%B6%80%ED%84%B0>

프로그램의 구성

본 프로그램은 아래와 같은 모듈들로 구성되어 있습니다. 여기는 구성해야할 최소 모듈을 제시한 것으로 필요시 추가적으로 만들어 개발할 수 있습니다.

models.py

- 생성될 **Table**에 관한 정보를 **ORM class**로 저장하는 모듈입니다.
- 저장될 **Table**은 주식이름 및 종목 코드 저장용 테이블 과 종목별 일일 주식정보 테이블 이며, 테이블 이름은 자유롭게 정한다.
- 테이블에 들어갈 **column**은 아래 요구사항에 맞춰 작성한다.

create_db.py

- **database**를 생성하는 모듈입니다. `python create_db.py`를 실행하면 아래와 같은 이벤트가 발생합니다.
- **pymysql**을 사용하여 **DB**에 접속합니다. 이 때, **DB** 접속과 관련된 정보들은 `config.json` 파일에 저장되어 실행시 **parsing**되어야 한다.
- **korean_stock** **DB**의 존재 여부를 확인하고 존재하지 않을 경우, **korean_stock**를 생성합니다.
- 처음 **database**를 생성하는 것이라면 생성된 주식이름 및 종목 코드 저장용 테이블 에 기업 종목 코드와 기업명에 정보를 업데이트 한다.

참고

- `config.json`에 포함된 정보
 - `host` - DBMS host 주소
 - `db_id` - DBMS 접속 유저 ID
 - `db_password` - DBMS 접속 유저 password 정보
 - `db_name` - 생성될 DB의 이름

stock_query.py

- 주식 정보를 가져오는 모듈입니다. 다음과 같은 함수를 제공합니다.

함수명	interface	내용
-----	-----------	----

query_all	(stock_name)	해당 주식명(예 - 삼성전자)의 2001년부터 현재까지의 일별 주식정보를 반환한다. 반환되는 형태는 pandas dataframe 으로 반환해준다
query_by_date	(stock_name, statedate, enddate)	해당 주식명(예 - 삼성전자)의 지정된 기간의 date 를 반환된다. date 형식은 다음과 같이 지정한다. 예시: Mar-16-2018 . 반환되는 형태는 pandas dataframe 으로 반환해준다
update_stock	(stock_name)	해당 주식 이름을 가진 주식의 정보를 최신(프로그램이 실행한 날자까지)으로 업데이트 한다

- 위 데이터를 반환하는 모듈은 **pandas** 모듈을 사용하여 반환한다.
- **query**가 되는 시점에서 해당 데이터들은 **DB**에 저장될 수 있도록 한다.
- 본 프로그램은 **jupyter notebook** 에서 **import** 하여 사용된다고 가정한다.

```
import stock_query
stock_query.query_all("삼성전자")
```

- 주식명이 완전히 맞지 않을 때는 사용자에게 **이름을 확인하세요** 라고 출력해 준다.
- 본 프로그램은 **jupyter notebook** 에서 **import** 하여 사용된다고 가정한다.

제출

본 lab assignment의 제출은 코드들을 zip으로 압축하여 **cyber campus**를 통해 제출한다.