1. 요구사항

가. 파이썬을 이용해서 REST API 서버를 만들어 보세요

- 영화 제목을 사용해서 영화결과 리스트를 출력해 주는 API서버를 개발.

- 영화 검색 API는 'https://www.kmdb.or.kr/info/api/apiDetail/6' 에서 발급한 API 조회키를 사용해서 검색함.

- Flask 를 이용해서 8080 포트를 통해 http://localhost:8080/movie?search=검색어 를 사용해서 검색을 하면

- Method 는 GET 방식으로 요청 가능하도록 함.

{

result:[

{movieId : '등록ID' title : '영화제목' , directorNm :....... }, // API에서 제공하는 검색 결과 전체

{movieId : '등록ID' title : '영화제목' , directorNm :....... }

{movieId : '등록ID' title : '영화제목' , directorNm :....... }....

]

}

나. 파이썬 API 서버를 Docker에 실행 해 보세요

- Docker Image로 만들때 Flask로 개발한 API 서버가 포함되도록 개발

- Docker Image안에 폴더 구조는 자유롭게 작성해도 됨.

2. 테스트 코드

가. 테스트 코드를 작성해 보세요.

나. Intellij 를 사용해서 Java 언어를 활용한 JUnit 코드를 작성해 보세요.

- http://localhost:8080/movie?search=검색어 를 호출하는 JUnit 코드를 작성해 보세요.

- 앞에서 만든 파이썬 REST 서버를 Docker로 실행 시킨후, JUnit 코드를 실행했을때, 결과를 출력하는 JUnit 코드를 작성해 보세요.

- TestCase 1 : 빈공백으로 검색한 경우 , 검색 결과가 없음으로 출력 되었는가?

- TestCase 2 : 한글로 타이타닉을 검색한 경우 , 검색 결과가 정상적으로 출력 되었는가?

- TestCase 3 : 한글로 타이 타닉을 검색한 경우 , 검색 결과가 없다고 나오는가?

3. Git

가. 위 2개의 프로젝트를 Git에 Commit & Push

from flask import Flask, jsonify, request

import requests

app = Flask(\_\_name\_\_)

@app.route('/movie')

def search\_movie():

search = request.args.get('search', '')

api\_url = f'https://www.kmdb.or.kr/info/api/apiDetail/6?key=6TM92DE2N8JUA38LAE38&title={search}'

response = requests.get(api\_url)

result = response.json()

return jsonify(result)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=8080)

from flask import Flask, jsonify, request

import requests

app = Flask(\_\_name\_\_)

app.config['JSON\_AS\_ASCII'] = False

@app.route('/movie')

def search\_movie():

search = request.args.get('search', '')

api\_url = f'http://api.koreafilm.or.kr/openapi-data2/wisenut/search\_api/search\_json2.jsp?ServiceKey=6TM92DE2N8JUA38LAE38&collection=kmdb\_new2&query={search}'

# api\_url = f'https://www.kmdb.or.kr/info/api/apiDetail/6?key=6TM92DE2N8JUA38LAE38&title={search}'

response = requests.get(api\_url)

response.encoding = 'utf8'

# print(response.text)

result = response.json()

movie\_list = result['Data'][0]['Result']

json = []

for movie in movie\_list:

print('#####################')

# 제작년도, 제목, 감독, 배우, 나라

print('#####################')

print(f'제작년도 : {movie["prodYear"]}')

print(f'제목 : {movie["title"]}')

print(f'감독 : {movie["directors"]["director"]}')

print(f'배우 : {movie["actors"]["actor"]}')

print(f'나라 : {movie["nation"]}')

print('#####################')

element = {}

element['prodYear'] = movie["prodYear"]

element['title'] = movie["title"]

element['actors'] = movie["actors"]["actor"]

element['directors'] = movie["directors"]["director"]

element['nation'] = movie["nation"]

json.append(element)

return json

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=8080)

##############################################

# 파이썬 라이브러리

##############################################

https://docs.python.org/ko/3/library/index.html

##############################################

# Flask 라이브러리

##############################################

https://flask-docs-kr.readthedocs.io/ko/latest/

##############################################

# Flask 샘플

##############################################

# Flask 를 인스톨 한후, 라이브러리를 Import 해 준다.

# pip install flask

from flask import Flask, escape, request

app = Flask(\_\_name\_\_)

# HTTP 메서드를 지정 가능 , methods=['GET', 'POST']

@app.route('/')

def hello():

name = request.args.get("name", "World")

##############################################

# f-string

##############################################

# >>> x = 1

# >>> y = 2

# >>> f"{x} + {y}는 {x + y}입니다."

##############################################

return f'Hello, {escape(name)}!'

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host="0.0.0.0", port=5000)

##############################################

# Request 다루기

##############################################

# Post Method

request.form['username']

# Post Method

request.args.get("name")

# 요청 데이터 변환 함수

request.get\_json()

request.get\_data()

##############################################

# URL 호출

##############################################

import requests, json

url = requests.get("https://jsonplaceholder.typicode.com/users")

text = url.text

data = json.loads(text)

params = {'code':'code'}

requests.post(url = url, data = json.dumps(params))

FROM python:3

WORKDIR /usr/src/app

# COPY requirements.txt ./

RUN pip install flask

COPY . .

CMD [ "python", "./test.py" ]

docker build -t mh-python ./