

Python을 활용한 데이터 수집 II

1. 목표

- Python 기본 문법 실습
- 요청과 응답에 대한 이해
- 데이터 구조에 대한 분석과 이해

2. 준비사항

A. 사용 데이터

- i. 영화진흥위원회 오픈 API(<http://www.kobis.or.kr/kobisopenapi>)
- ii. 영화인 정보 API 서비스
- iii. 영화 정보 API 서비스
- iv. 주간/주말 박스오피스 API 서비스

B. 개발언어/프로그램

- i. Python 3.7 이상

C. 필수 라이브러리

- i. requests

3. 요구사항

커뮤니티 서비스 개발을 위한 데이터 수집 단계로, 전체 데이터 중 필요한 데이터를 크롤링 하는 과정을 진행합니다. 아래 기술된 사항은 필수적으로 구현해야 하는 내용입니다.

A. 필모리스트 카운트 기능 구현

영화인명과 필모리스트를 이용하여 영화인을 검색하고 몇 작품을 출연, 제작등을 하였는지 계산합니다. 완성된 기능은 향후 커뮤니티 서비스에서 제공되는 기능으로 사용됩니다.

i. 요청조건

1. 제공되는 kobis.py를 이용하여 영화인목록 조회 URL을 생성합니다.
2. 영화인명과 필모리스트를 이용하여 데이터를 요청합니다.

ii. 결과

1. 받아온 데이터에서 필모리스트를 가져옵니다.
2. 중복되는 데이터가 있다면 제일 첫번째 영화인 정보를 사용합니다.
3. 가져온 데이터를 | (파이프) 기호 기준으로 개수를 계산합니다.
4. 계산한 정보를 반환하는 함수 filmo_count를 완성합니다.

B. 영화 검색 기능 구현

영화명과 감독명을 이용하여 영화진흥위원회 API의 영화코드 정보를 가져오는 기능을 완성합니다. 완성된 함수는 다음 문제의 기본 기능으로 사용됩니다.

i. 요청조건

1. 제공되는 kobis.py를 이용하여 영화목록 조회 URL을 생성합니다.
2. 영화명과 감독명을 이용하여 데이터를 요청합니다.

ii. 결과

1. 받아온 데이터에서 영화코드 정보를 가져옵니다.
2. 검색결과가 없다면 0을 반환하고, 있다면 해당영화의 영화코드를 반환하는 함수 get_movie_cd를 완성합니다.

C. 영화 코드를 이용한 상세 정보 수집

이전 단계에서 완성한 영화 검색 기능을 이용하여 상세정보를 출력하는 함수를 완성합니다. 해당 기능은 향후 커뮤니티 서비스에서 기본으로 제공되는 영화 정보로 사용됩니다.

i. 요청조건

1. 제공되는 kobis.py를 이용하여 영화상세목록 조회 URL을 생성합니다.

2. 영화코드를 이용하여 데이터를 요청합니다.
3. 영화 코드는 problem_b.py의 함수 get_movie_cd를 이용하여 가져와 사용합니다.

ii. 결과

1. 받아온 데이터에서 다음과 같은 내용을 저장합니다.
영화명(국문), 개봉연도, 장르명, 배우명
2. 장르명과 배우명은 아래의 예시와 같이 저장합니다.
3. 배우가 1명 이상이면 첫번째 배우만 저장합니다.
4. 배우가 0명 이면 noActor를 저장합니다.
5. 영화 상세정보를 반환하는 함수 movie_info를 완성합니다.

iii. 예시

1. 출력

```
{
  'movieNm': '기생충',
  'openDt': '20190530',
  'genres': ['드라마'],
  'actors': '송강호'
}
```

D. 추가정보 수집

영화 데이터를 제공하는 다른 API를 사용하여 요청을 보내고 추가적인 정보를 수집하는 함수를 완성합니다.

i. 데이터

1. KMDB(<https://www.kmdb.or.kr/info/api/apiDetail/6>)
2. TMDB(<https://www.themoviedb.org/documentation/api?language=ko>)
3. 네이버 영화검색 API(<https://developers.naver.com/docs/search/movie/>)

4. 결과

반드시 활용하였던 데이터 정보 정리 및 저장한 파일에 대한 설명과 학습 내용을 README.md에 기록하여 제출합니다.

위에 명시된 사항은 최소 조건이며 추가적인 정보를 수정할 수 있습니다.

Gitlab에 제출하는 파일/폴더의 구조는 아래와 같습니다.

pjt02/

kobis.py

README.md

problem_a.py

problem_b.py

problem_c.py

problem_d.py