

Python을 활용한 데이터 수집 II

1. 목표

- Python 기초 문법 실습
- 데이터 구조에 대한 분석과 이해
- 요청과 응답에 대한 이해
- API의 활용과 문서 분석

2. 준비사항

A. 사용 데이터

- i. TMDb API(<https://developers.themoviedb.org/3>)
- ii. 영화 정보 및 API 서비스

B. 개발언어/프로그램

- i. Python 3.9 이상

C. 필수 라이브러리

- i. requests

3. 요구사항

커뮤니티 서비스 개발을 위한 데이터 수집 단계로, 전체 데이터 중 필요한 영화 데이터를 수집하는 과정입니다. 아래 기술된 사항은 필수적으로 구현해야 하는 내용입니다. 완성된 기능들은, 향후 커뮤니티 서비스에서 활용할 수 있습니다.

A. 인기 영화 조회

인기 영화의 개수를 출력합니다.

i. 요청

1. requests를 이용하여 인기 영화 정보(Get Popular)에 요청을 보냅니다.

ii. 결과

1. popular를 기준으로 받아온 데이터에서 영화 리스트의 개수를 계산합니다.
2. 계산한 정보를 반환하는 함수 popular_count를 완성합니다.

B. 특정 조건에 맞는 인기 영화 조회 I

popular를 기준으로 가져온 영화 목록 중 평점이 8 이상인 영화들의 목록을 출력합니다.

i. 요청

1. requests를 이용하여 인기 영화 정보(Get Popular)에 요청을 보냅니다.

ii. 결과

1. 받아온 데이터에서 vote_average를 기준으로 점수가 8 이상인 영화들의 목록을 리스트로 반환하는 함수 vote_average_movies를 완성합니다.

C. 특정 조건에 맞는 인기 영화 조회 II

영화목록을 평점순으로 출력하는 함수를 완성합니다. 해당 기능은 향후 커뮤니티 서비스에서 기본으로 제공되는 영화 정보로 사용됩니다.

i. 요청

1. requests를 이용하여 인기 영화 정보(Get Popular)에 요청을 보냅니다.

ii. 결과

1. 받아온 데이터 중 평점이 높은 영화 다섯개의 정보를 리스트로 반환하는 함수 ranking을 완성합니다.

D. 특정 영화 추천 영화 조회

제공된 영화 제목을 기준으로 추천영화 목록을 출력합니다.

i. 요청

1. requests를 이용하여 영화 검색(Search Movies) 요청을 보냅니다.
2. 응답 받은 결과를 바탕으로 id를 찾아 추천영화 목록 조회 (Get Recommendations) URL을 생성합니다.
3. requests를 이용하여 URL에 요청을 보냅니다.

ii. 결과

1. TMDB에서 추천받은 영화 리스트에서 제목을 리스트에 저장합니다.
2. 저장된 리스트를 반환하는 함수 recommendation을 완성합니다.
3. 올바르게 않은 영화 제목으로 id가 없는 경우 None을 반환합니다.
4. id값은 있지만 추천영화가 없는 경우 빈 리스트를 반환합니다.

E. 배우, 감독 리스트 출력

영화에 출연한 배우들과 감독의 정보가 저장된 딕셔너리를 출력합니다.

i. 요청

1. requests를 이용하여 영화 검색(Search Movies) 요청을 보냅니다.
2. 응답 받은 결과를 바탕으로 id를 찾아 크레딧 조회 (Get Credits) URL을 생성합니다.
3. requests를 이용하여 URL에 요청을 보냅니다.

ii. 결과

1. cast_id 값이 10보다 작은 배우의 이름을 리스트에 저장합니다.
2. department 값이 Directing인 감독의 이름을 리스트에 저장합니다.
3. 반환되는 딕셔너리는 cast, crew 두개의 key를 가지고 각각 배우리스트와 감독리스트를 value로 갖습니다.
4. 완성된 딕셔너리를 반환하는 함수 credits를 완성합니다.

F. 특정 영화 배우, 감독 리스트 조회 (선택)

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. 영진위(<https://www.kobis.or.kr/kobisopenapi/homepg/main/main.do>)
3. 네이버 영화검색 API(<https://developers.naver.com/docs/search/movie/>)