

02 PJT

Python을 활용한 데이터 수집 2

INDEX

- Python을 활용한 데이터 수집 2
 - 목표
 - 준비사항
 - 요구사항
 - 제출

목표

프로젝트 목표

- Python 기본 문법 습득
- 데이터 구조에 대한 분석과 이해
- 요청과 응답에 대한 이해
- API의 활용과 API 문서 숙지

준비사항

| 개발도구 및 라이브러리

- 개발도구
 - Visual Studio Code
 - Python 3.9+
- 필수 라이브러리
 - [Requests](#)
- API
 - [TMDB API](#) - 영화 정보 및 API 서비스

요구사항

| 공통 요구사항

- 커뮤니티 서비스 개발을 위한 데이터 수집 단계로, 필요한 영화 데이터를 수집하는 과정입니다.
- 완성된 기능들은 향후 커뮤니티 서비스에서 활용할 수 있습니다.
- API 요청 시 언어 및 지역 설정 데이터는 한국을 기준으로 합니다.
- 반드시 제공된 examples/ 폴더의 예시 파일을 먼저 참고합니다.
 - 예시 파일에는 이번 프로젝트 해결을 위해 알아야 하는 혹은 직접적인 도움이 될 수 있는 코드가 작성되어 있습니다.

A. 인기 영화 조회 (problem_a)

- 인기 영화 목록을 응답 받아 개수를 출력합니다.

1. 풀이

- requests 라이브러리를 사용하여 TMDB에서 현재 인기있는 영화 목록 ([Get Popular](#)) 데이터를 요청합니다.
- 응답 받은 데이터의 영화 개수를 반환하는 함수 popular_count를 작성합니다.

A. 인기 영화 조회 (problem_a)

2. 결과

- problem_a.py 실행 예시

```
20
```

B. 특정 조건에 맞는 인기 영화 조회 1 (problem_b)

- 인기 영화 목록 중 평점이 8점 이상인 영화 목록을 출력합니다.

1. 풀이

- TMDB에서 현재 인기있는 영화 목록([Get Popular](#)) 데이터를 요청합니다.
- 응답 받은 데이터 중 평점(vote_average)이 8점 이상인 영화 목록을 반환하는 함수 vote_average_movies를 작성합니다.

B. 특정 조건에 맞는 인기 영화 조회 1 (problem_b)

2. 결과

- problem_b.py 실행 예시

```
[{'adult': False,
  'backdrop_path': '/ocUp7DJBic8VJgLEw1prcyK1dYv.jpg',
  'genre_ids': [28, 12, 878],
  'id': 634649,
  'original_language': 'en',
  'original_title': 'Spider-Man: No Way Home',
  'overview': '미스터리오의 계약으로 세상에 정체가 탄로난 스파이더맨 피터 파커는 하루 아침에 평범한 일상을 잃게 된다. 문제를 '
    '해결하기 위해 닥터 스트레인을 찾아가 도움을 청하지만 뜻하지 않게 멀티버스가 열리면서 각기 다른 차원의 '
    '불청객들이 나타난다. 닥터 옥토퍼스를 비롯해 스파이더맨에게 깊은 원한을 가진 숙적들의 강력한 공격에 피터 파커는 '
    '사상 최악의 위기를 맞게 되는데...',
  'popularity': 1842.592,
  'poster_path': '/voddFVdjUoAtfoZZp2RUmuZILDI.jpg',
  'release_date': '2021-12-15',
  'title': '스파이더맨: 노 웨이 홈',
  'video': False,
  'vote_average': 8.1,
  'vote_count': 13954},
  # 생략
]
```

주의) TMDb 'Get Popular' 응답 데이터는 시기에 따라 예시 출력과 다를 수 있습니다.

C. 특정 조건에 맞는 인기 영화 조회 2 (problem_c)

- 인기 영화 목록을 평점이 높은 순으로 5개의 영화 데이터 목록 출력합니다.

1. 풀이

- TMDB에서 현재 인기있는 영화 목록([Get Popular](#)) 데이터를 요청합니다.
- 응답 받은 데이터 중 평점(vote_average)을 기준으로 평점이 높은 영화 5개 정보를 리스트로 반환하는 함수 ranking을 작성합니다.
- [sort 메서드](#) 혹은 [sorted 함수](#)의 특정 파라미터를 이용합니다.

C. 특정 조건에 맞는 인기 영화 조회 2 (problem_c)

2. 결과

- problem_c.py 실행 예시

```
[{'adult': False,
  'backdrop_path': '/odJ4hx6g6vBt4lBWKFD1tI8WS4x.jpg',
  'genre_ids': [28, 18],
  'id': 361743,
  'original_language': 'en',
  'original_title': 'Top Gun: Maverick',
  'overview': '최고의 파일럿이자 전설적인 인물 매버릭은 자신이 졸업한 훈련학교 교관으로 발탁된다. 그의 명성을 모르던 팀원들은 '
              '매버릭의 지시를 무시하지만 실전을 방불케 하는 상공 훈련에서 눈으로 봐도 믿기 힘든 전설적인 조종 실력에 모두가 '
              '압도된다. 매버릭의 지휘 아래 견고한 팀워크를 쌓아가던 팀원들에게 국경을 뛰어넘는 위험한 임무가 주어지자 매버릭은 '
              '자신이 가르친 동료들과 함께 마지막이 될지 모를 하늘 위 비행에 나서는데...',
  'popularity': 911.817,
  'poster_path': '/jMLiTGCo0vXJuwMzZGoNOUPfuj7.jpg',
  'release_date': '2022-06-22',
  'title': '탑건: 매버릭',
  'video': False,
  'vote_average': 8.4,
  'vote_count': 1463},
  # 생략
]
```

주의) TMDb 'Get Popular' 응답 데이터는 시기에 따라 예시 출력과 다를 수 있습니다.

D. 특정 추천 영화 조회 (problem_d)

- 제공된 영화 제목('기생충', '그래비티', '검색할 수 없는 영화')을 검색하여 추천 영화 목록을 출력합니다.

1. 풀이

- 제공된 영화 제목으로 TMDB에서 영화를 검색([Search Movies](#))합니다.
- 응답 받은 결과 중 첫번째 영화의 id 값을 찾아 해당 영화에 대한 추천 영화 목록 ([Get Recommendations](#))을 가져옵니다.
- 추천 영화 목록 중 첫번째 영화만 출력하는 함수 recommendation을 작성합니다.

D. 특정 추천 영화 조회 (problem_d)

2. 결과

- 추천 영화가 없을 경우 []를 반환합니다.
- 검색한 영화 정보가 없다면 None을 반환합니다.
- problem_d.py 실행 예시

```
['조커',  
'1917',  
'조조 래빗',  
'원스 어폰 어 타임 인... 할리우드',  
'결혼 이야기',  
'나이트스 아웃',  
'아이리시맨',  
'포드 V 페라리',  
'작은 아씨들',  
'라이트하우스',  
'미드소마',  
'더 플랫폼',  
'Assassination',  
'언컷 젬스',  
'센과 치히로의 행방불명',  
'내 여자친구를 소개합니다',  
'그린 북',  
'스타워즈: 라이즈 오브 스카이워커',  
'두 교황',  
'살인의 추억',  
'펄프 픽션']  
[]  
None
```

주의) TMDB 'Get Recommendations' 응답 데이터는 시기에 따라 예시 출력과 다를 수 있습니다.

E. 출연진, 연출진 데이터 조회 (problem_e)

- 제공된 영화 제목('기생충', '검색할 수 없는 영화')을 검색하여 해당 영화의 출연진(cast) 그리고 스태프(crew) 중 연출진 목록만을 출력합니다.

1. 풀이

- 제공된 영화 제목으로 TMDB에서 영화를 검색([Search Movies](#))합니다.
- 응답 받은 결과 중 첫번째 영화의 id 값을 찾아 해당 영화에 대한 출연진과 스태프 목록([Get Credits](#))을 가져옵니다.
- 출연진은 cast_id 값이 10 미만인 출연진만 추출하며, 연출진은 스태프 부서가 Directing인 데이터만 추출합니다.
- 위 조건을 만족해서 답을 반환하는 함수 credits를 작성합니다.

E. 출연진, 연출진 데이터 조회 (problem_e)

2. 결과

- 검색한 영화 정보가 없다면 None을 반환합니다.
- problem_e.py 실행 예시

```
{'cast': ['Song Kang-ho',  
         'Lee Sun-kyun',  
         'Cho Yeo-jeong',  
         'Choi Woo-shik',  
         'Park So-dam',  
         'Lee Jung-eun',  
         'Jang Hye-jin'],  
 'directing': ['Bong Joon-ho',  
              'Park Hyun-cheol',  
              'Han Jin-won',  
              'Kim Seong-sik',  
              'Lee Jung-hoon',  
              'Yoon Young-woo']}
```

None

F. 선택 과제

- 영화 데이터를 제공하는 다른 API를 사용하여 내가 원하는 데이터를 추출하고 만들어봅니다.
- API 목록
 - KMDB
(<https://www.kmdb.or.kr/info/api/apiDetail/6>)
 - 영화진흥위원회
(<https://www.kobis.or.kr/kobisopenapi/homepg/main/main.do>)
 - 네이버 영화검색 API
(<https://developers.naver.com/docs/serviceapi/search/movie/movie.md>)

제출

제출 시 주의사항

- 제출기한은 금일 18시까지 입니다. 제출기한을 지켜 주시기 바랍니다.
- 반드시 README.md 파일에 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점을 등을 상세히 기록하여 제출합니다.
 - 단순히 완성된 코드만을 나열하지 않습니다.
- 위에 명시된 요구사항은 최소 조건이며, 추가 개발을 자유롭게 진행할 수 있습니다.
- <https://lab.ssafy.com/>에 프로젝트를 생성하고 제출합니다.
 - 프로젝트 이름은 '프로젝트 번호 + pjt'로 지정합니다. (ex. **01_pjt**)
- 반드시 각 반 담당 교수님을 Maintainer로 설정해야 합니다.