

데이터베이스 설계를 통한 REST API 설계

1. 목표

- 데이터를 생성, 조회, 수정, 삭제할 수 있는 Web Application 제작
- DRF(Django Rest Framework)를 활용한 API Server 제작
- Database 1:N, M:N 관계의 이해와 데이터 관계 설정

2. 준비사항

A. 언어

- i. Python 3.9+
- ii. Django 3.2+

B. 도구

- i. Visual Studio Code
- ii. Chrome Browser
- iii. Postman

3. 요구사항

커뮤니티 서비스의 상세 기능 개발을 위한 단계로, 모델간의 관계 설정 후 데이터의 생성, 조회, 수정, 삭제할 수 있는 기능을 완성합니다. 데이터의 조회는 JSON 양식을 따릅니다.

A. 프로젝트 구조

pjt08/은 startproject 명령어로 생성되는 project 디렉토리입니다.

movies/는 startapp 명령어로 생성되는 application 디렉토리입니다.

아래의 폴더구조는 주요한 폴더와 파일만 명시되어 있습니다.

```
pjt08/  
  settings.py
```

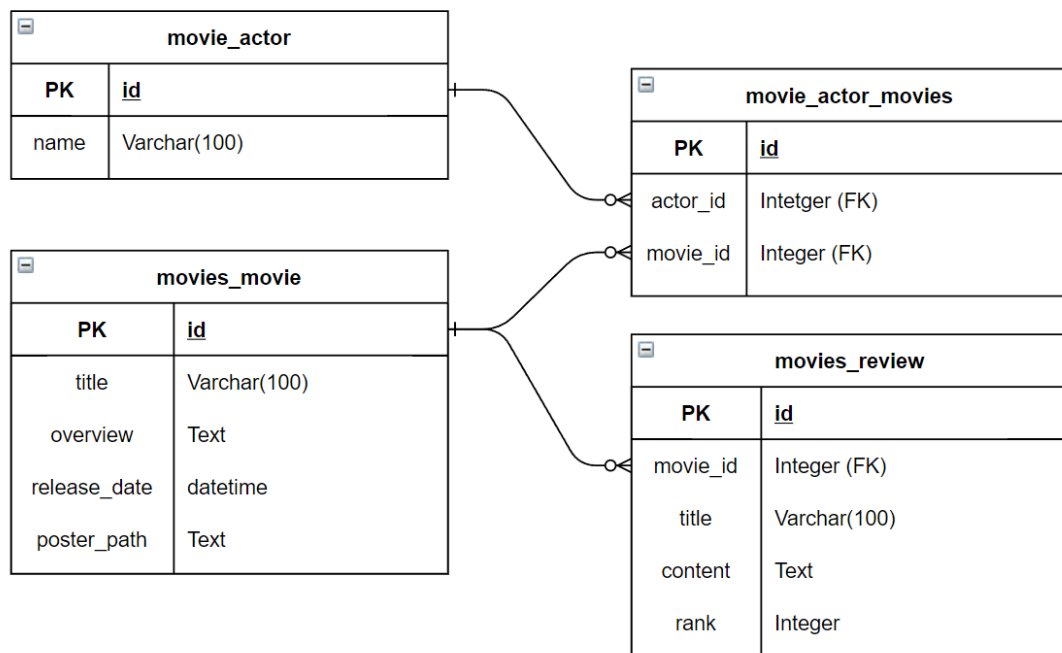
```

    urls.py
movies/
    migrations/
    models.py
serializers.py
    urls.py
    views.py
manage.py
...

```

B. Model

데이터베이스에서 모델의 ERD(Entity Relationship Diagram) 예시는 아래와 같습니다.



C. URL

TMDB(<https://developers.themoviedb.org/3/>) API 문서를 참고하여 페어와 의논 후 RESTful한 URL을 구성합니다.

D. Admin

- i. Actor, Movie, Review의 데이터 모두 admin 페이지에서 CRUD가 가능합니다.

E. Serializer & View

i. Serializer

- 1. Model과 URL경로 및 view 함수와 대응하는 적절한 Serializer를 생성합니다.

ii. Actor

- 1. 전체 배우 목록을 제공합니다.
- 2. 단일 배우 정보(출연 영화 포함)를 제공합니다.
- 3. 관리자 페이지에서 배우 데이터를 생성, 수정, 삭제합니다.

iii. Movie

- 1. 전체 영화 목록을 제공합니다.
- 2. 단일 영화 정보(출연 배우와 리뷰 데이터 목록 포함)를 제공합니다.
- 3. 관리자 페이지에서 영화 데이터를 생성, 수정, 삭제합니다.
- 4. (선택) 영화 데이터의 생성과 출연 배우 등록이 가능한 view 함수를 생성합니다.

iv. Review

- 1. 리뷰를 생성합니다.
- 2. 전체 리뷰 목록을 제공합니다.
- 3. 단일 리뷰 정보(영화 정보 포함)를 반환합니다.
- 4. 리뷰를 수정합니다.
- 5. 리뷰를 삭제합니다.

F. API test

1. Postman을 통해 API 테스트를 진행하고 결과를 README.md에 이미지로 첨부합니다.

4. 결과

- A. 제출 기한은 **21.10.29(금) 18:00**까지입니다. 제출기한을 엄수하여 주시기 바랍니다.
- B. 반드시 README.md 파일에 각 단계별 상세 구현 과정, 학습 내용 및 어려웠던 부분을 기록하여 제출해야 합니다.
- C. 위에 명시된 사항은 최소 조건이며 추가적인 정보를 수정할 수 있습니다.
- D. <https://lab.ssafy.com>에 pjt08 프로젝트를 생성하고 해당 프로젝트에 업로드하여 제출합니다.
- E. 각 반 담당 교수님을 Maintainer로 설정해야 합니다.