

# 08 - REST API

## 1. 목표

- API 요청에 대한 이해
- RESTful API 서버 구축
- API 문서화

## 2. 준비 사항

### 1. (필수) Python Web Framework

- Django 2.2.x
- Python 3.7.x

## 3. 요구 사항

### 1. 데이터베이스 설계

- `db.sqlite3` 에서 테이블 간의 관계는 아래와 같습니다.
- `movies_movies`

필드명	자료형	설명
<b>id</b>	Interger	Primary Key
<b>title</b>	String	영화명
<b>audience</b>	Integer	누적 관객수
<b>poster_url</b>	String	포스터 이미지 URL
<b>description</b>	Text	영화 소개
<b>genre_id</b>	Integer	Genre의 Primary Key(id 값)

- `movies_genres`

필드명	자료형	설명
<b>id</b>	Interger	Primary Key
<b>name</b>	String	장르 구분

- `movies_reviews`

필드명	자료형	설명
id	Integer	Primary Key
content	String	한줄평(평가 내용)
score	Integer	평점
movie_id	Integer	Movie의 Primary Key(id 값)
user_id	Integer	User의 Primary Key(id 값)

## 2. Seed Data 반영

1. 주어진 `movie.json` 과 `genre.json` 을 `movies/fixtures/` 디렉토리로 옮깁니다.
2. 아래의 명령어를 통해 반영합니다.

```
$ python manage.py loaddata genre.json
Installed 11 object(s) from 1 fixture(s)
$ python manage.py loaddata movie.json
Installed 10 object(s) from 1 fixture(s)
```

3. `admin.py` 에 `Genre` 와 `Movie` 클래스를 등록한 후, `/admin` 을 통해 실제로 데이터베이스에 반영되었는지 확인해봅시다.

## 3. movies API

- 아래와 같은 API 요청을 보낼 수 있는 서버를 구축해야 합니다.
- 허용된 HTTP 요청을 제외하고는 모두 405 Method Not Allowed를 응답합니다.

api		▼
GET	/api/v1/genres/	
GET	/api/v1/genres/{genre_pk}/	
GET	/api/v1/movies/	
GET	/api/v1/movies/{movie_pk}/	
POST	/api/v1/movies/{movie_pk}/scores/	
GET	/api/v1/scores/{score_pk}	
PUT	/api/v1/scores/{score_pk}	
DELETE	/api/v1/scores/{score_pk}	

### 1. GET /api/v1/genres/

```
curl -X GET "http://localhost:8000/api/v1/genres/" -H "accept: application/json"
```

장르의 목록을 요청 받아서 다음과 같은 결과를 응답합니다.

```
[
  {
```

```

    "id": 1,
    "name": "가족"
  },
  {
    "id": 2,
    "name": "공포(호러)"
  },
  {
    "id": 3,
    "name": "드라마"
  },
  ...
]

```

## 2. GET /api/v1/genres/{genre\_pk}/

특정 장르의 결과를 응답합니다.

```

{
  "id": 7,
  "movies": [
    {
      "id": 3,
      "title": "조커",
      "audience": 4960167,
      "poster_url": "https://movie-
phinf.pstatic.net/20190906_128/1567761736426S6Fje_JPEG/movie_image.jpg
?type=n77_110_2",

```

"description": ""내 인생이 비극인줄 알았는데, 코미디였어"\n고담시의 광대 아서 플렉은 코미디언을 꿈꾸는 남자. 하지만 모두가 미쳐가는 코미디 같은 세상에서 맨 정신으로는 그가 설 자리가 없음을 깨닫게 되는데... 이제껏 본 적 없는 진짜 '조커'를 만나라",

```

    "genre": 7
  }
],
"name": "스릴러"
}

```

만약, 없는 경로 변수(genre\_pk)로 접근하는 경우 404 Not Found 에러를 응답합니다.

```

# 예)
curl -X GET "http://localhost:8000/api/v1/genres/11/" -H "accept:
application/json"
# => 404 Not Found

```

## 3. GET /api/v1/movies/

영화 목록을 요청 받아서 다음과 같은 결과를 응답합니다.

```
[
  {
    "id": 1,
    "title": "82년생 김지영",
    "audience": 1120468,
    "poster_url": "https://movie-
phinf.pstatic.net/20191024_215/1571900079078PNazL_JPEG/movie_image.jpg
",
    "description": "1982년 봄에 태어나\n누군가의 딸이자 아내, 동료이자 엄마로\n2019년 오늘을 살아가는 '지영'(정유미).\n때론 어딘가 갇힌 듯 답답하기도 하지만\n남편 '대현'(공유)과 사랑스러운 딸\n그리고 자주 만나지 못해도 항상 든든한 가족들이 '지영'에겐
큰 힘이다. \n하지만 언젠가부터 마치 다른 사람이 된 것처럼 말하는 '지영'.\n'대현'은 아내
가 상처 입을까 두려워 그 사실을 털어놓지 못하고\n'지영'은 이런 '대현'에게 언제나 "괜찮
다"라며 웃어 보이지만 하는데...\n모두가 알지만 아무도 몰랐던\n당신과 나의 이야기",
    "genre": 3
  },
  {
    "id": 2,
    "title": "말레피센트 2",
    "audience": 1125713,
    "poster_url": "https://movie-
phinf.pstatic.net/20191004_160/1570150829240zmLfa_JPEG/movie_image.jpg
",
    "description": "동화는 끝났다! 두 세계의 운명을 건 가장 사악한 전쟁이 시작된다!
강력한 어둠의 지배자이자 무어스 숲의 수호자 '말레피센트'는 딸처럼 돌봐온 '오로라'와 '필립
왕자'의 결혼 약속으로 인간 세계의 '잉그리스 왕비'와 대립하게 된다. 이에 요정과 인간의 오랜
연합이 깨지고 숨겨진 요정 종족 다크페이의 리더 '코널'까지 등장하면서 두 세계는 피할 수 없는
거대한 전쟁에 휘말리게 되는데...",
    "genre": 10
  },
  ...
]
```

#### 4. GET /api/v1/movies/{movie\_pk}/

특정 영화의 결과를 응답합니다.

```
{
  "id": 4,
  "title": "가장 보통의 연애",
  "audience": 2793371,
  "poster_url": "https://movie-
phinf.pstatic.net/20190911_29/1568167420523xeKY5_JPEG/movie_image.jpg"
,
  "description": "전 여친에 상처받은 '재훈'(김래원).\n 여느 때처럼 숙취로 시작한 아
침,\n 모르는 번호의 누군가와 밤새 2시간이나 통화한 기록을 발견하게 되고\n 그 상대가 바로!
통성명한 지 24시간도 채 되지 않은 직장 동료 '선영'임을 알게 된다.\n \n 남친과 뒤끝 있는
이별 중인 '선영'(공효진).\n 새로운 회사로 출근한 첫날, 할 말 못 할 말 쏟아내며 남친과 헤
어지던 현장에서\n 하필이면! 같은 직장의 '재훈'을 마주친다.\n \n 만난 지 하루 만에 일보다
서로의 연애사를 더 잘 알게 된 두 사람.\n 하지만 미묘한 긴장과 어색함도 잠시\n '한심하다',
'어이없다' 부딪히면서도 마음이 쓰이는 건 왜 그럴까?",
  "genre": 4
}
```

만약, 없는 경로 변수(movie\_pk)로 접근하는 경우 404 Not Found 에러를 응답합니다.

```
# 예)
curl -X GET "http://localhost:8000/api/v1/movies/1000/" -H "accept:
application/json"
# => 404 Not Found
```

##### 5. POST /api/v1/movies/{movie\_pk}/reviews/

특정 영화에 평점을 등록합니다. 성공시 아래와 같이 응답합니다.

```
{
  "message": "작성되었습니다."
}
```

만약, 없는 경로 변수(movie\_pk)로 접근하는 경우 404 Not Found 에러를 응답합니다.

필수 필드가 누락된 경우 400 Bad Request 에러를 응답합니다.

##### 6. PUT /api/v1/reviews/{review\_pk}/

특정 평점을 수정합니다. 성공시 아래와 같이 응답합니다.

```
{
  "message": "수정되었습니다."
}
```

만약, 없는 경로 변수(review\_pk)로 접근하는 경우 404 Not Found 에러를 응답합니다.

필수 필드가 누락된 경우 400 Bad Request 에러를 응답합니다.

##### 7. DELETE /api/v1/reviews/{review\_pk}

특정 평점을 삭제합니다. 성공시 아래와 같이 응답합니다.

```
{  
  "message": "삭제되었습니다."  
}
```

만약, 없는 경로 변수(review\_pk)로 접근하는 경우 404 Not Found 에러를 응답합니다.

```
# 예)  
curl -X DELETE "http://localhost:8000/api/v1/reviews/19/" -H "accept:  
application/json"
```

## 4. API documents

---

작성한 API 서버의 가능한 요청 사항과 메서드/경로/경로변수를 테스트 할 수 있는 documents를 구성하여 스크린 샷을 제출합니다.

## 5. 결과물 제출

---

Python Web Framework를 활용해 작성한 모든 파일을 project08 디렉토리에 위치하도록 합니다.

결과물은 반드시 `README.md` 으로 활용 하였던 내용을 정리 해주세요.

```
project08/  
  ...  
  ...  
  ...  
  README.md
```