

05/11 (월) Django



| Background

- ✓ Web Framework
- ✓ Django

| Goal

- ✓ DRF(Django Rest Framework)에 대한 이해

| Problem

- ❖ DRF를 활용하여 음악 정보 관련 REST API 서버를 구축하시오.
- ❖ 프로젝트는 api, 앱은 musics로 만들어 진행한다.

1. Model & Admin

- 작성 할 모델 및 필드 정보
 - 관리자 페이지를 활용하여 가수, 가수별 노래, 노래별 감상평을 각각 2개씩 작성한다.
- Artist
 - name : 가수의 이름
- Music
 - Artist와 1:N 관계
 - title : 노래의 제목
- Comment
 - Music과 1:N 관계
 - content : 노래의 감상평

2. serializers

- ArtistSerializer
 - 모든 가수 정보를 반환하기 위한 Serializing
 - id, name 필드 정의
- ArtistDetailSerializer
 - 상세 가수 정보를 반환하기 위한 Serializing
 - id, name, musics_count 필드 정의
- MusicSerializer
 - 모든 음악정보를 반환하기 위한 Serializing
 - id, title 필드 정의

05/11 (월) Django



- MusicDetailSerializer
 - 상세 음악정보를 반환하기 위한 Serializing
 - id, title, comments 필드 정의
- CommentSerializer
 - 모든 음악 감상평 정보를 반환하기 위한 Serializing
 - Id, content 필드 정의

3. url & view

- Postman을 활용하여 각 HTTP method에 맞는 요청을 테스트하시오.
- **GET** api/v1/artists/
 - 모든 가수의 id와 name 컬럼을 json으로 반환한다.
- **GET** api/v1/artists/<artist_pk>/
 - 특정 가수의 모든 컬럼을 json으로 반환한다.
 - 해당 가수의 노래 정보와 노래의 개수 정보를 같이 반환한다.
- **GET** api/v1/musics/
 - 모든 음악의 id와 title 컬럼을 json으로 반환한다.
- **GET** api/v1/musics/<music_pk>/
 - 특정 음악의 모든 컬럼을 json으로 반환한다.
 - 해당 노래의 감상평 정보를 같이 반환한다.
- **POST** api/v1/musics/<music_pk>/comments/
 - 특정 음악의 감상평을 작성한다.
 - 검증에 성공하는 경우 새로운 음악 감상평을 DB에 저장하고 저장된 감상평을 반환한다.
 - 만약 검증에 실패하는 경우 400 Bad Request 오류를 발생 시킨다.
- **PUT & DELETE** api/v1/comments/<comment_pk>/
 - PUT 요청인 경우 특정 감상평을 수정한다.
 - 검증에 성공하는 경우 수정된 감상평을 DB에 저장한다.
 - 검증에 실패할 경우 400 Bad Request 오류를 발생 시킨다.
 - 수정이 완료된 이후에 **성공적으로 수정되었습니다.** 라는 메시지를 반환한다.
 - DELETE 요청일 경우 특정 감상평을 삭제한다.
 - 삭제가 완료된 이후에 **성공적으로 삭제 되었습니다.** 라는 메시지를 반환한다.

❖ 각 요청에 맞는 JSON 응답 결과의 캡처 사진과 **views.py, serializers.py** 코드를 마크다운에 작성해서 제출하시오.