

# 10/05 (월) Django



## | Background

- ✓ Web Framework
- ✓ Django

## | Goal

- ✓ DRF(Django REST Framework)에 대한 이해

## | Problem

- ❖ DRF를 활용하여 게시글 관련 REST API 서버를 구축하시오.
- ❖ 프로젝트는 api, 앱은 articles로 만들어 진행한다.
- ❖ DRF 내장 Form을 활용하여 요청을 테스트한다.

## 1. Model & Admin

- Article
  - title, content, created\_at, updated\_at 컬럼이 존재한다.
  - 관리자 페이지를 활용하여 5개의 게시글을 생성한다.

## 2. serializers

- ArticleListSerializer
  - 모든 게시글 정보를 반환하기 위한 Serializer
  - id, title 필드 정의
- ArticleSerializer
  - 상세 게시글 정보를 반환 및 생성하기 위한 Serializer
  - id, title, content, created\_at, updated\_at 필드 정의



### 3. url & views

- **GET** api/v1/articles/
    - 모든 게시글의 id와 title 컬럼을 JSON으로 응답한다.
  
  - **POST** api/v1/articles/
    - 검증에 성공하는 경우 새로운 게시글의 정보를 DB에 저장하고 저장된 게시글의 정보를 응답한다.
    - 검증에 실패하는 경우 400 Bad Request 오류를 발생시킨다.
  
  - **GET** api/v1/articles/<article\_pk>/
    - 특정 게시글의 모든 컬럼을 JSON으로 응답한다.
  
  - **PUT & DELETE** api/v1/articles/<article\_pk>/
    - PUT 요청인 경우 특정 게시글의 정보를 수정한다.
      - 검증에 성공하는 경우 수정된 게시글의 정보를 DB에 저장한다.
      - 검증에 실패할 경우 400 Bad Request 오류를 발생시킨다.
      - 수정이 완료되면 수정한 게시글의 정보를 응답한다.
    - DELETE 요청인 경우 특정 게시글의 정보를 삭제한다.
      - 삭제가 완료되면 삭제한 게시글의 id를 응답한다.
- ❖ 각 요청에 맞는 JSON 응답 결과와 **views.py**, **serializers.py** 코드를 마크다운에 작성하여 제출하시오.