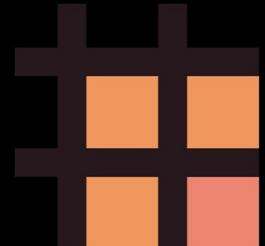


# 블로그 API 만들기 - DRF

21.5 Django seminar 03



서울대학교 컴퓨터공학부  
웹/앱 개발동아리  
와플스튜디오

# 오늘의 목표

- <https://github.com/wafflestudio/seminar-2023-django-assignment/tree/main/03>
- 지난번 구현했던 Blog Website를 Rest API 형태로 구현해보아요.

# Rest Framework가 뭘까?

- Django? Rest? Django가 서버 프레임워크 아니었나?
- Django는 원래 Full-stack framework.
  - Web Page를 띄우기 위한 프레임워크 (예시: 지난 번 과제)
- 현대의 대부분의 클라이언트 (웹, 모바일 등) Json 형태로 소통한다.
- 실험: <https://naver.com> 에서 개발자 도구 열어보기
- DO: Rest Framework를 설치.

# Form vs Serializer

- 사실은 Template vs Json API에 대응되는 개념
- 각자 Template에서의 Validation, JSON API에서의 Validation 영역을 담당한다.
- Django Form은 html의 <form />에 대응되고, Serializer는 Json validation에 대응된다.
- Form과 Serializer에서 대응되지 않는 건 어떻게 처리할까?
  - 예시: blog post 생성 시 created\_by
- DO: 지난 번 블로그 포스트 작성 / 댓글 작성은 DRF로 변경.
- Rest URL은 왜 저렇게 관리할까? 꼭 읽어봤으면 하는 글:  
[그런 Rest API로 괜찮은가?](#)

# Authentication

- Authentication: 특정 Request를 User에 대응시키기
- 기존 Django Template만 사용했을 때 Cookie라는 방식으로 잘 이루어지고 있었다.
- Django REST는 REST API에서 사용하는 각종 Authentication 방법들을 제공한다.
  - <https://www.django-rest-framework.org/api-guide/authentication/>
  - BasicAuthentication, SessionAuthentication, TokenAuthentication
  - TokenAuthentication을 사용해보자. (request header 조작)
- DO: TokenAuthentication 적용해보고, request.user 디버깅

# Authorization & Permission

- 로그인이 가능해졌으면, 특정 유저에게만 어떤 action을 허용하고 싶어질 수 있다.
- 예: 글을 읽는 것은 누구나 가능했지만, 글을 수정하는 것은 해당 글을 작성한 유저에게만 허용할 수 있음
- DRF에서는 Class-based Permission을 제공해 구현할 수 있다.
  - <https://www.djangoproject.org/api-guide/permissions/>
  - Has\_permission, has\_object\_permission을 override해 구현하곤 함.
- DO: 글 수정, 코멘트 수정 Permission 구현해보기.

# Mockings

- 테스트를 위해서 테스트 데이터를 매번 만드는 건 곤혹스러운 일.
- Django 자체에서 제공해주지는 않지만, Model이 OOP 스타일로 꽤 잘 구현되어있기 때문에 이를 Mocking해주는 라이브러리가 많다.
- Model-mommy를 사용해 Post와 Comment를 많이 만들어보자.
- [https://model-mommy.readthedocs.io/en/latest/basic\\_usage.html](https://model-mommy.readthedocs.io/en/latest/basic_usage.html)
- DO: Model-mommy 설치, comment, post 100개씩 생성.

# Pagination

- 소통할 데이터가 많아지게 되면, 한 번에 소통할 수 있는 데이터에 제한을 두고 싶어질 수 있다.
- 이는 실제로 꽤 유용한 방법 (각종 커뮤니티의 페이지를 생각해보자)
- 이런 방법을 Pagination이라고 하고, DRF에서 다양한 방식의 Pagination을 제공한다.
  - <https://www.djangoproject.org/api-guide/pagination/>
  - PageNumberPagination, LimitOffsetPagination, CursorPagination 등
  - CursorPagination은 각종 SNS의 무한로딩에 대응되는 꽤 유용한 방법.
- DO: 기존 Comment, Post View에 CursorPagination 구현해보기.

# Many-to-one Relationships, Many-to-Many Relationships

- DRF의 부분은 아니지만, RDB를 사용하는데 있어 꼭 사용할 수 있는 장점.
- RDB는 이름처럼 Relational Database: 관계형 데이터를 다루는데 큰 장점이 있다
  - ACID 원칙 참고: <https://en.wikipedia.org/wiki/ACID>
  - 특정 Column들에 Index를 걸거나, constraints를 거는 데 특화되어 있다
- Foreign Key와 Many To Many 관계는 사실 Table constraints / index에 대응된다.
  - ManyToMany Relationship은 ForeignKey를 두 번 쓴 것과 같다
- DO: Tag 구현 ([ERD](#))

# AWS RDS, ElasticBeanstalk Free-tier 사용하기

- 인스턴스 고르는 곳에 유의해주세요!
- 각자 Free Tier를 사용하실 수 있습니다.
- Billing을 한 번씩 확인해주시고, 과하게 나가고 있지 않은지 트래킹해주세요.
- DO: Django 배포 with RDS, ElasticBeanstalk -> Free tier

감사합니다 🎉  
다음 주 Survivor 회식 오세요!