

Pinterest

- Python의 대표 웹 프레임워크인 django를 통해 pinterest 특유의 카드형 레이아웃을 벤치마킹한 웹서비스를 만들었습니다.
- 디자인에 소요되는 시간을 단축시키고, 백엔드 기능 구현 및 리팩토링에 중점을 둔 프로젝트입니다.



기간

2021.07 ~ 2021.08



역할

- API 설계
- DB 설계 및 관리

기능 구조 - API

Accounts App

- 회원가입
- MyPage (paginate_by = 5)
- 비밀번호 변경 (login_required, account_ownership_required)
- 회원탈퇴 (login_required, account_ownership_required)
- 로그인, 로그아웃

Article App

- 게시글 생성 (login_required)
- 게시글 뷰
- 게시글 변경 (article_ownership_required)
- 게시글 삭제 (article_ownership_required)
- 모든 게시글의 목록 뷰 (paginate_by = 5)

Project App

- 프로젝트 생성 (login_required)
- 프로젝트 상세 뷰 (paginate_by = 25)
- 프로젝트 리스트 뷰 (paginate_by = 25)

Comment App

- 댓글 생성
- 댓글 삭제 (comment_ownership_required)

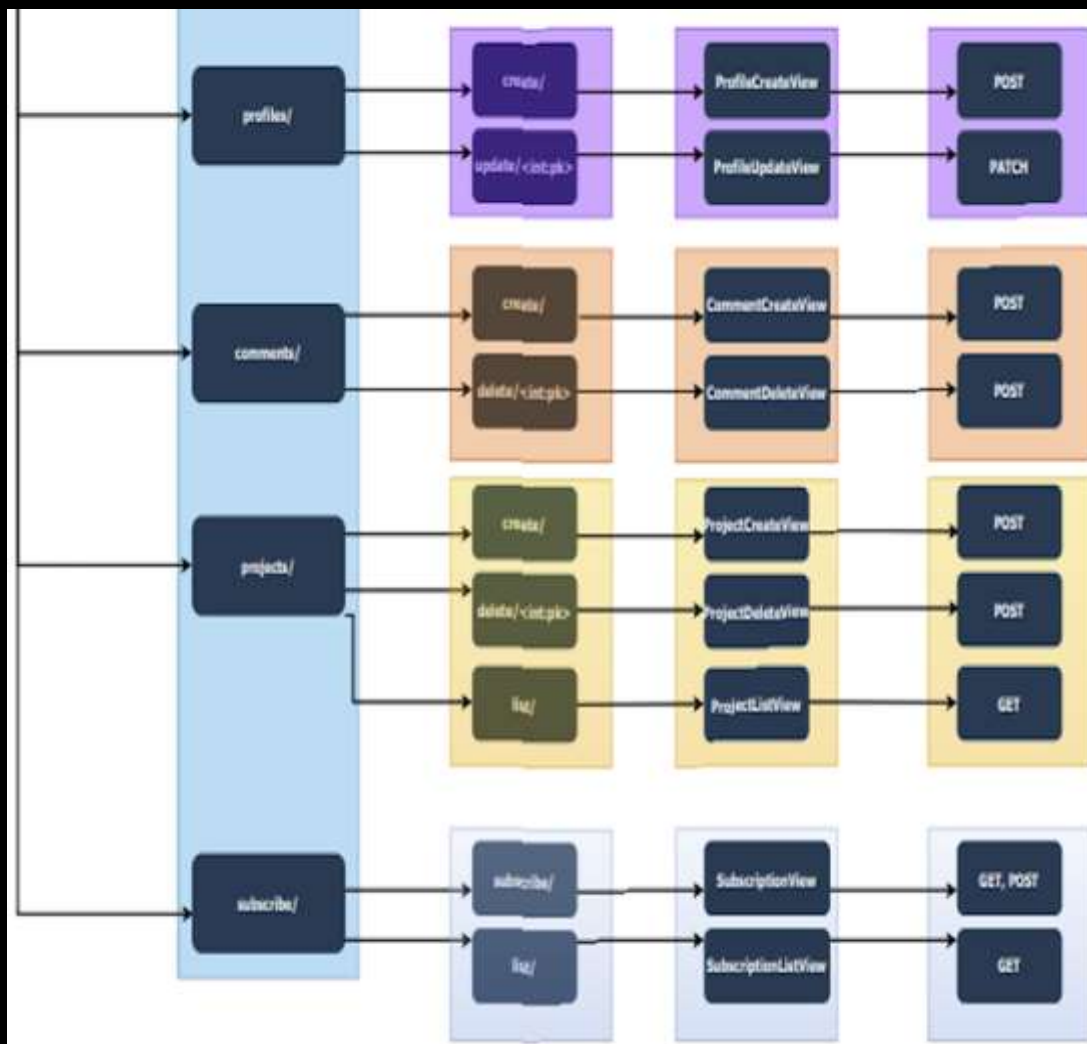
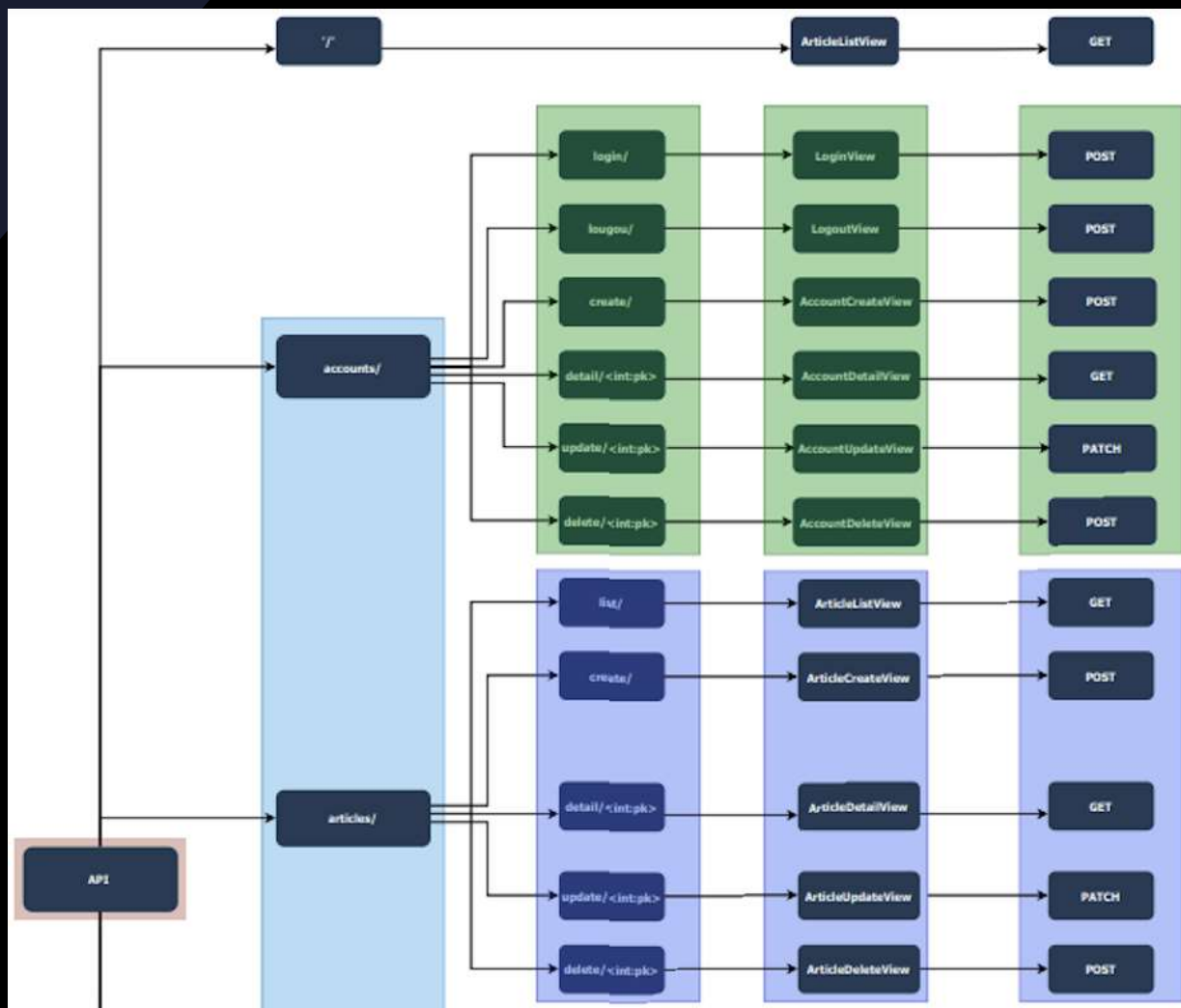
Profile App

- 프로필 생성
- 프로필 변경 (profile_ownership_required)

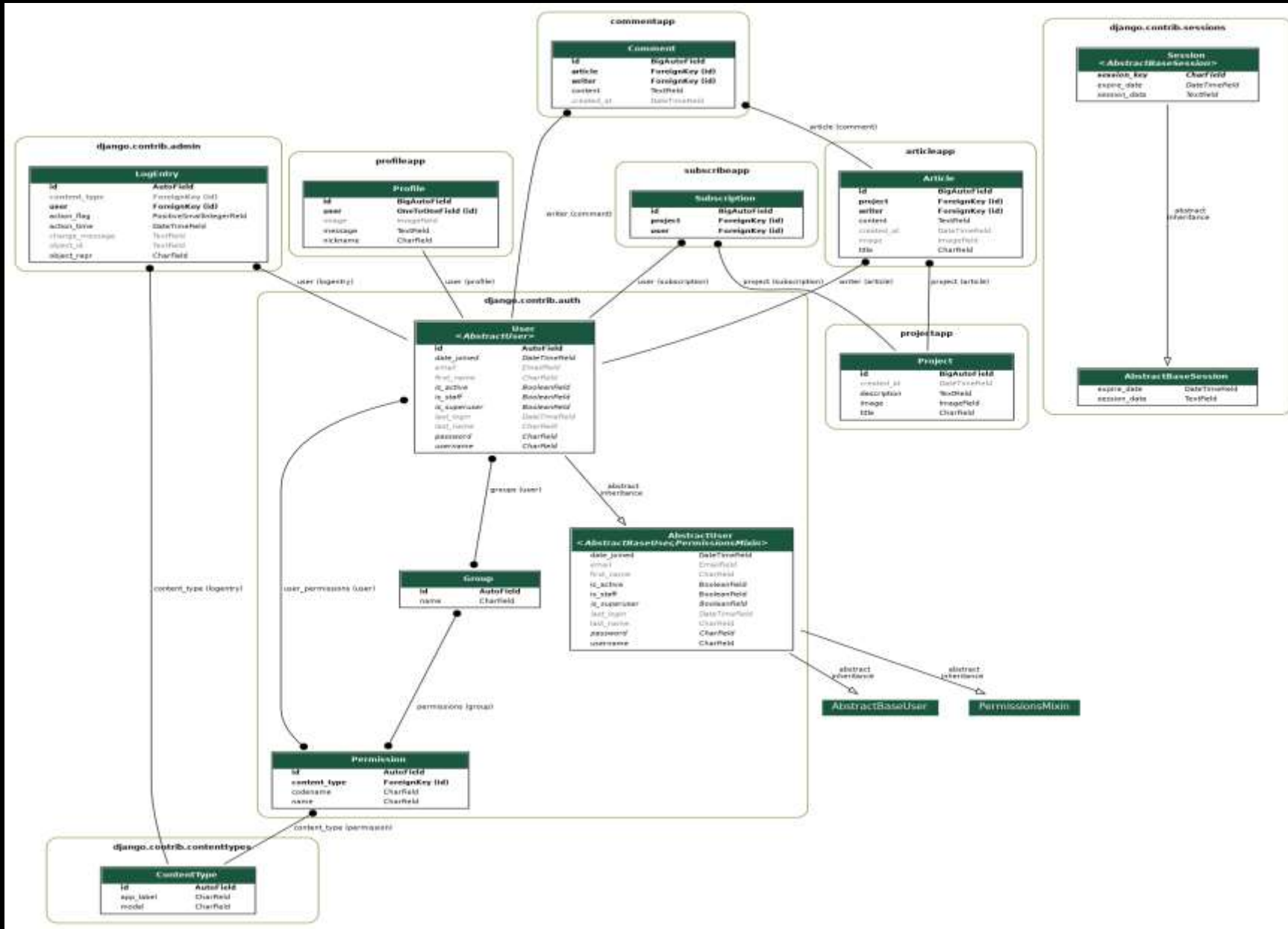
Subscribe App

- * 구독 기능 구현 (login_required)
- * 구독한 프로젝트의 모든 게시글을 보여주는 기능 구현 (login_required) (paginate_by = 5)

기능 구조 - API 2



ERD



Deploy

- 개발 환경과 배포 환경을 분리하였습니다.
- AWS EC2 에 Docker 를 설치했습니다.
- Docker 안에는 django, nginx, gunicorn 을 컨테이너로 만들어 적용 했습니다.
- Docker 는 GUI 프로그램인 Portainer 를 사용했습니다.
- Docker Swarm 에 SECRET_KEY 사용하여 보안을 적용 했습니다.
- docker-compose.yml 로 손쉽게 배포 적용 했습니다.
- AWS Route53 설정하고 도메인 및 EC2 와 연결하여 도메인 이름을 설정하였습니다.
- AWS Load Balancer 사용하여 https 적용하여 보안을 높였습니다.
- AWS Elastic IP 적용 하였습니다.