

DongriGo Seeding Policy

Phase 1 Baseline

Version	[e.g., v1.0]
Date	2025-12-28
Author	Donghun Kim
Repository	https://github.com/lincoln-cmd/DongriGo
Production URL	https://dongrigo.onrender.com/
Related Commands	seed_prod, check_integrity, ops_check

Table of Contents

Table of Contents	2
1. 목적	3
2. 입력(Inputs)	3
3. 기본 원칙	3
4. seed_prod 동작 정의	3
4.1 기본 실행	3
4.2 --force	3
4.3 --wipe --force	4
5. 권장 운영 시나리오	4
5.1 최초 구축(Initial Setup)	4
5.2 배포 후 점검(Post-deploy)	4
6. 산출물/로그 보관	4
7. 트러블슈팅	5

1. 목적

운영(Render Postgres)에서 초기 데이터(country, post, postimage)를 안정적으로 주입하고, 반복 실행 시 데이터 중복, 덮어쓰기 사고를 방지한다.

2. 입력(Inputs)

- Fixture 파일
 - fixtures/prod_seed.json
 - 포함 대상: Country, Post, PostImage
- 데이터베이스
 - 로컬: SQLite
 - 운영: Render PostgreSQL
- 환경변수(운영, 로컬 공통)
 - config.settings
 - DEBUG
 - SECRET_KEY
 - ALLOWED_HOSTS
 - DATABASE_URL
 - USE_CLOUDINARY
 - CLOUDINARY_URL
- 관련 커맨드 및 도구
 - python manage.py migrate
 - python manage.py seed_prod
 - python manage.py check_integrity -fix -json
 - python manage.py ops_check -json

3. 기본 원칙

- 운영 DB에는 이미 데이터가 있는 상태에서 seed를 무심코 재실행하지 않는다.
- seed는 기본적으로 멍등(idempotent)하게 동작해야 한다.
- 동일 fixture를 다시 적용할 필요가 없으면 스킵한다. (SeedMeta hash 기반)

4. seed_prod 동작 정의

4.1 기본 실행

- python manage.py seed_prod
 - DB 상태와 SeedMeta를 참고해 스킵 또는 진행을 결정한다.
 - 동일 fixture 해시가 이미 적용되었다면 스킵 (hash match).

4.2 --force

- python manage.py seed_prod -force

- 운영에서 정말로 **fixture**를 다시 적용해야 하는 경우에만 1회 사용을 권장한다.
- 사용 전 **DB** 상태와 중복 가능성을 확인한다.

4.3 --wipe --force

- `python manage.py seed_prod --wipe --force`
 - 관련 테이블을 삭제(초기화)하고 **fixture**를 다시 넣는다.
 - 운영에서 사용하면 데이터가 소실되므로, 원칙적으로 초기 환경 구성이나 테스트 DB에서만 사용한다.

5. 권장 운영 시나리오

5.1 최초 구축(Initial Setup)

- 1) `python manage.py migrate`
- 2) `python manage.py seed_prod --force` (필요시 1회만)
- 3) `python manage.py check_integrity --fix --json`
- 4) `python manage.py seed_prod` (보통 스킵. Hash match 되어야 정상)

5.2 배포 후 점검(Post-deploy)

- 1) `python manage.py migrate`
- 2) `python manage.py ops_check --json`
- 3) `python manage.py seed_prod` (대부분 스킵되는 것이 정상)
- 4) `python manage.py check_integrity --fix --json`

6. 산출물/로그 보관

- seed 실행 로그 (콘솔 출력)
- `check_integrity --json` 결과 JSON
- 운영 배포 시점별 커밋 해시 (Release note)

7. 트러블슈팅

(A) seed가 'Skipped'로만 나오고 데이터가 안 들어가는 것처럼 보일 때

- 원인: 이미 동일 fixture가 적용되어 SeedMeta hash match로 스킵
- 조치:
 - 정말 다시 넣어야 한다면 `seed_prod -force`
 - 초기화가 목적이면 `seed_prod -wipe -force` (운영에서는 매우 주의)

(B) Fixture 로드 중 UnicodeDecodeError

- 원인: fixture가 UTF-8이 아닌 인코딩
- 조치: UTF-8로 다시 dump해서 교체 (Windows는 PYTHONUTF8=1 권장)

(C) Iso 길이 관련 오류 (value too long for type character varying(2) 등)

- 원인: iso_a2에 2글자 초과 값 존재
- 조치: fixture, 시딩 정규화 규칙에 맞게 수정 후 재시도

(D) 운영에서 DB 연결 실패

- 원인: DATABASE_URL 미설정 또는 잘못된 값
- 조치: Render Environment Variables에 DATABASE_URL 확인 후 재배포, 재시도

(E) 운영에서 media 이미지가 깨짐

- 원인: Cloudinary 환경변수 누락 또는 USE_CLOUDINARY 스위치 불일치
- 조치: CLOUDINARY_URL (또는 3종 키) + USE_CLOUDINARY=1 확인

(F) seed 이후 화면에 국가, 게시물이 안 보임

- 원인: 템플릿, 쿼리에서 필터 조건(발행 여부, 카테고리) 때문에 숨겨짐
- 조치: Admin에서 Post is_published 확인, 목록 뷰 필터 조건 점검