AppStore

관리자 가이드

설치부터 SSO·권한·워크플로·운영까지

버전	v2.6.0
작성일	2026-09-12
문서	관리자 가이드

AppStore 관리자 가이드

대상 버전: **v2.6.0** · 실린 화면은 모두 이 버전을 실제로 띄워 찍은 것입니다.

화면을 쓰는 사람을 위한 설명은 사용자 가이드에 있습니다. 이 문서는 그 화면을 띄워 놓고 지키는 사람을 위한 것입니다.

---

1. 구성 요소

AppStore는 컨테이너 하나로 동작합니다. React SPA, REST API, OIDC 세션, AI SSE 프록시, MCP endpoint, PostgreSQL 마이그레이션과 초기 카탈로그가 모두 하나의 Go 바이너리 안에 들어 있습니다.

구성 요소	필수	역할	비고
`appstore:vX.Y.Z` 컨테이너	예	SPA + API + MCP + AI 프록시	Alpine 기반, non-root `appstore:appstore` (uid/gid 10001), read-only rootfs
PostgreSQL	예	유일한 영속 저장소	앱·사용자·역할·설정·암호화된 secret·감사 로그 전부
Keycloak (OIDC)	아니오	조직 SSO	없으면 Bootstrap 관리자만 로그인 가능
AI Provider	아니오	AI 스트리밍 응답	OpenAI 호환 / vLLM / Ollama / Custom
Reverse proxy	권장	HTTPS 종단, 외부 노출	컨테이너는 평문 8080만 listen

주고받는 것:

방향	상대	내용
AppStore → PostgreSQL	TCP, DSN의 `sslmode` 설정에 따름	마이그레이션(시작 시 advisory lock 아래 자동), 조회·변경
브라우저 → AppStore	HTTP 8080	SPA, `/api/v1`, `/mcp`, `/openapi.json`, `/docs`
AppStore → Keycloak	HTTPS	discovery 문서, token, userinfo, JWKS
AppStore → AI Provider	HTTP(S)	채팅 스트리밍 프록시
AppStore → 임의 이미지 주소	HTTP(S)	로고·파비콘 **주소에서 가져오기**를 쓸 때만

컨테이너에 영속 볼륨이 없습니다. 루트 파일 시스템은 읽기 전용이고 `/tmp` 만 64MB tmpfs입니다.

---

2. 설치

릴리스 자산 하나(`appstore-vX.Y.Z.tar.gz`)로 폐쇄망까지 반입할 수 있습니다. 아래는 v2.6.0 기준입니다.

2.1. 선행 조건

항목	요구
호스트	Linux/AMD64 + Docker Engine
PostgreSQL	스키마 변경 권한이 있는 데이터베이스. 마이그레이션이 확장 설치 없이 `gen_random_uuid()` 를 쓰므로 **PostgreSQL 13 이상**
네트워크	컨테이너에서 PostgreSQL을 이름으로 찾을 수 있어야 합니다
포트	컨테이너 `8080/tcp` 하나. 볼륨 없음
자원	서비스 자체는 가볍습니다. 부하 대부분은 PostgreSQL에 걸립니다
SSO를 쓸 경우	도달 가능한 Keycloak Issuer와 신뢰되는 TLS 체인

PostgreSQL 이미지는 릴리스에 포함되지 않습니다. 먼저 준비하세요.

2.2. 이미지 반입과 검증

```
sha256sum appstore-v2.6.0.tar.gz          # 릴리스 노트의 SHA-256과 대조
gzip -t appstore-v2.6.0.tar.gz
gzip -dc appstore-v2.6.0.tar.gz | docker load
docker image inspect appstore:v2.6.0 \
  --format '{{.RepoTags}} user={{.Config.User}} version={{index .Config.Labels "org.opencontainers.image.ve
```

`user=appstore:appstore` 와 archive 이름에 맞는 version label이 보여야 합니다. 하나라도 다르면 배포하지 마세요.

소스에서 직접 빌드할 수도 있습니다.

```
docker build \
  --build-arg VERSION=v2.6.0 \
  --build-arg COMMIT="$(git rev-parse HEAD)" \
  --build-arg BUILD_DATE="$(date -u +%Y-%m-%dT%H:%M:%SZ)" \
  -t appstore:v2.6.0 .
```

2.3. 환경 파일 준비

```
install -m 600 .env.example appstore.env
openssl rand -base64 32                    # ENCRYPTION_KEY 값 생성
```

```
# appstore.env
POSTGRES_DSN=postgres://appstore:ENCODED_PASSWORD@postgres.example.internal:5432/appstore?sslmode=require
BOOTSTRAP_ADMIN=admin
BOOTSTRAP_ADMIN_PASSWORD=replace-with-a-long-random-password
ENCRYPTION_KEY=replace-with-the-generated-32-byte-base64-value
```

값은 모두 예시입니다. 실제 비밀값을 저장소나 이미지에 넣지 마세요.

2.4. 실행

`docker run`:

```
docker run -d \
  --name appstore \
  --restart unless-stopped \
  --read-only \
  --tmpfs /tmp:size=64m,mode=1777 \
  --cap-drop ALL \
  --security-opt no-new-privileges:true \
  --env-file ./appstore.env \
  -p 127.0.0.1:8080:8080 \
  appstore:v2.6.0
```

Docker Compose (저장소의 `docker-compose.yml`):

```
APPSTORE_VERSION=v2.6.0 docker compose up -d --no-build
docker compose ps
docker compose logs --tail 100 appstore
```

`APPSTORE_VERSION` · `APPSTORE_COMMIT` · `APPSTORE_BUILD_DATE`는 Compose가 이미지 태그와 라벨을 고르는 호스트 쪽 치환 값이며 컨테이너 안으로 전달되지 않습니다.

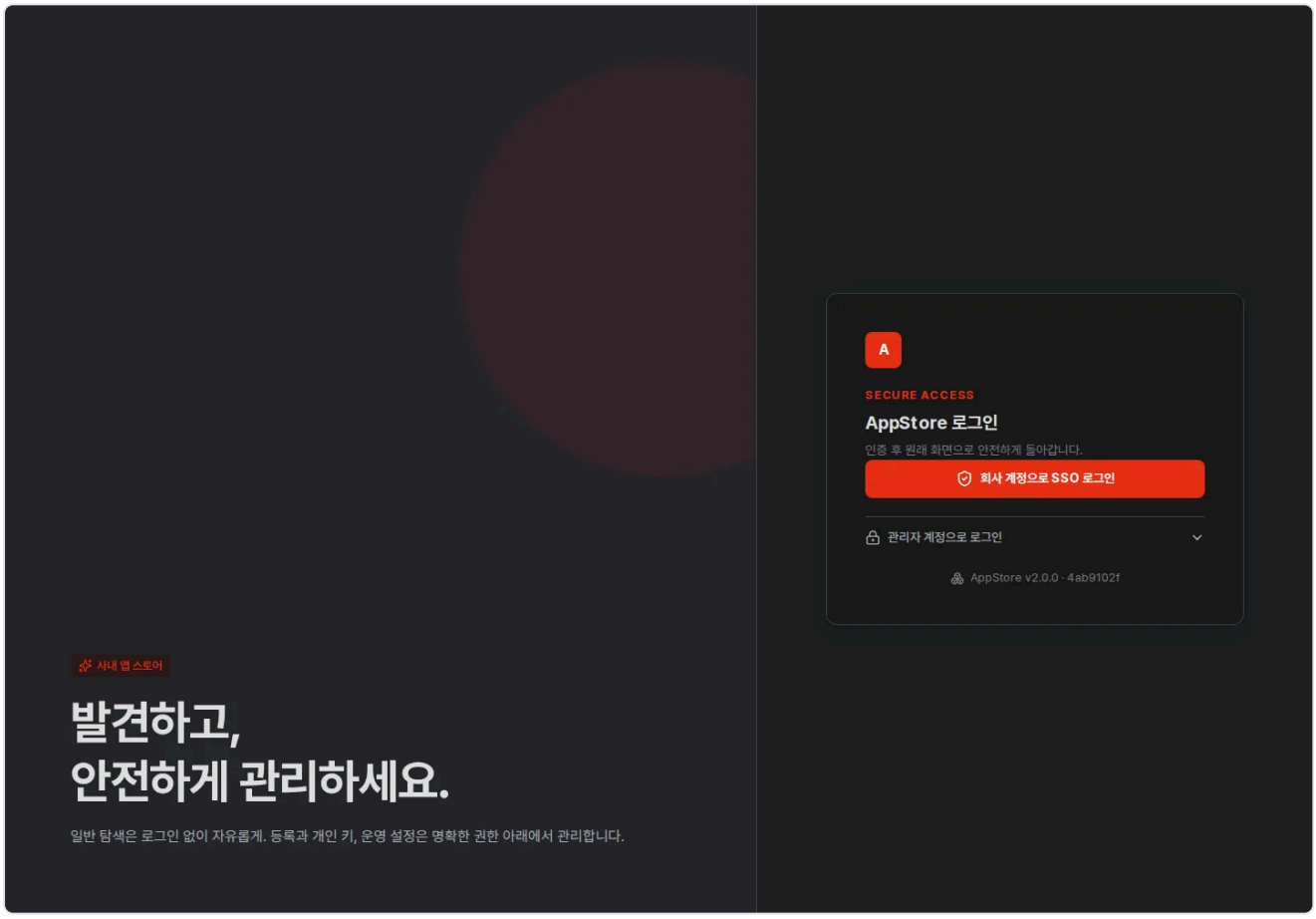
2.5. 기동 확인

```
curl --fail http://127.0.0.1:8080/health/live    # 프로세스 응답
curl --fail http://127.0.0.1:8080/health/ready   # DB 연결과 마이그레이션 완료
curl --fail http://127.0.0.1:8080/api/version    # version / commit / buildDate
```

첫 기동에서 서비스는 순서대로 마이그레이션 적용 → `ENCRYPTION_KEY` 지문 확인 → 초기 카탈로그 seed → Bootstrap 관리자 생성을 수행합니다. `/health/ready`가 200을 돌려주면 여기까지 끝난 상태입니다.

2.6. 최초 관리자 계정

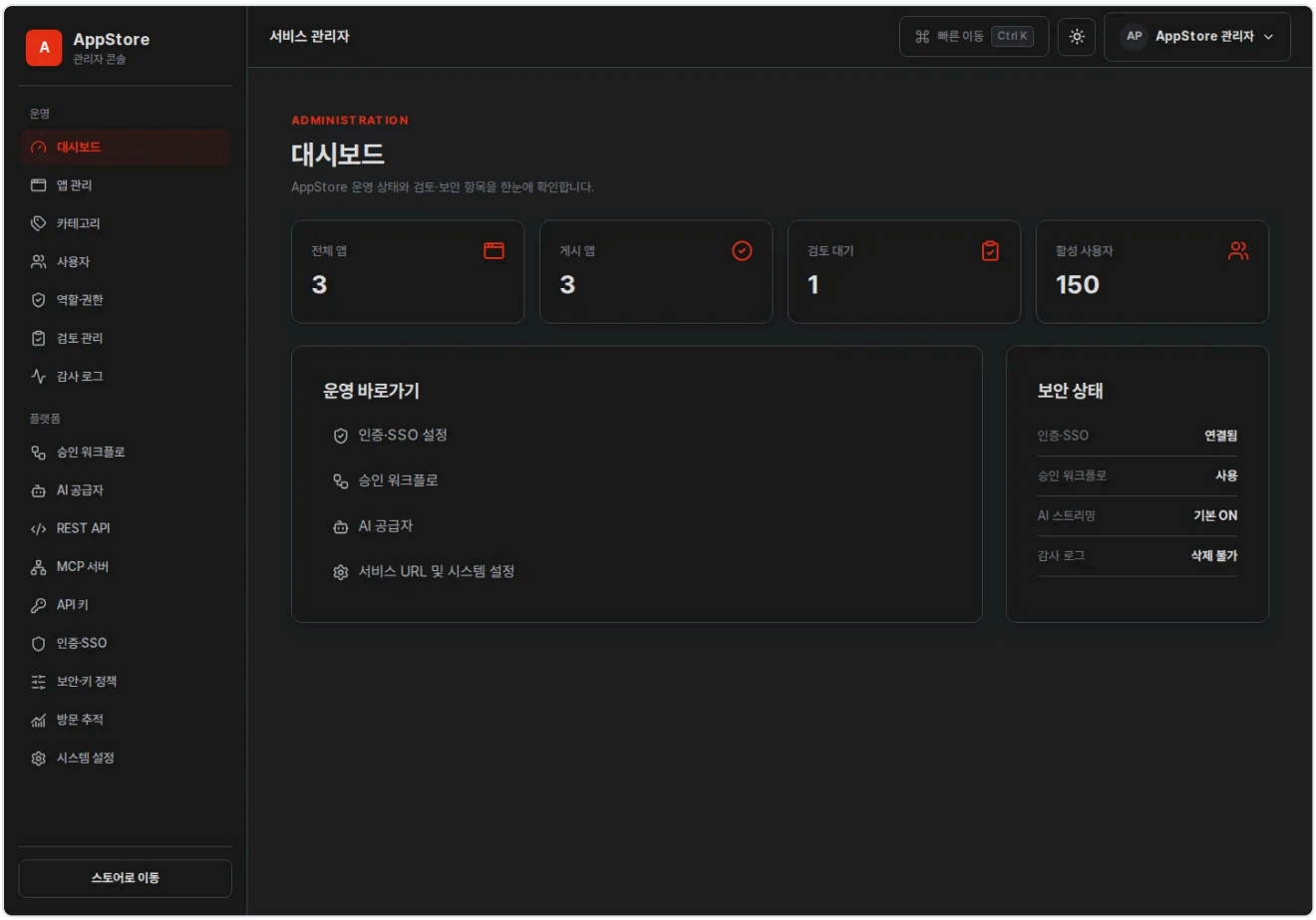
OIDC를 아직 설정하지 않은 상태에서는 `BOOTSTRAP_ADMIN` / `BOOTSTRAP_ADMIN_PASSWORD`로 만들어진 로컬 Super Admin으로 로그인합니다. 로그인 화면의 **관리자 계정으로 로그인**을 펼치거나 `/admin/bootstrap` 주소로 바로 들어갑니다.



최초 관리자 로그인 — SSO 설정 전에 쓰는 복구용 관리자 로그인 화면

화면 설명 그대로 *SSO를 사용할 수 없을 때를 위한 복구용 관리자 계정입니다.* 일상 계정으로 공유하지 마세요. SSO를 붙인 뒤에는 별도의 SSO Super Admin을 지정해 이 계정 의존도를 낮춥니다.

로그인하면 관리자 콘솔 대시보드가 열립니다.



관리자 대시보드 — 앱·사용자 운영 상태와 검토·보안 항목 요약

---

3. 설정

3.1. 환경 변수 (전수)

애플리케이션 코드가 읽는 환경 변수는 정확히 아래 네 개입니다(`internal/config/config.go`). 이 계약은 `scripts/check-env-contract.sh` 가 CI에서 강제합니다.

이름	기본값	필수	설명
`POSTGRES_DSN`	없음	예	PostgreSQL 연결 문자열. 비어 있으면 기동하지 않습니다. 비밀번호의 예약 문자는 URL 인코딩하세요
`BOOTSTRAP_ADMIN`	없음	예	최초·복구 Super Admin 식별자. 앞뒤 공백은 잘라 냅니다
`BOOTSTRAP_ADMIN_PASSWORD`	없음	예	Bootstrap 로그인 비밀번호. **12자 미만이면 기동 자체가 실패**합니다
`ENCRYPTION_KEY`	없음	예	DB에 저장되는 OIDC·AI secret의 암호화 키. 32 raw byte / 64자리 hex / 32 byte의 base64 표현

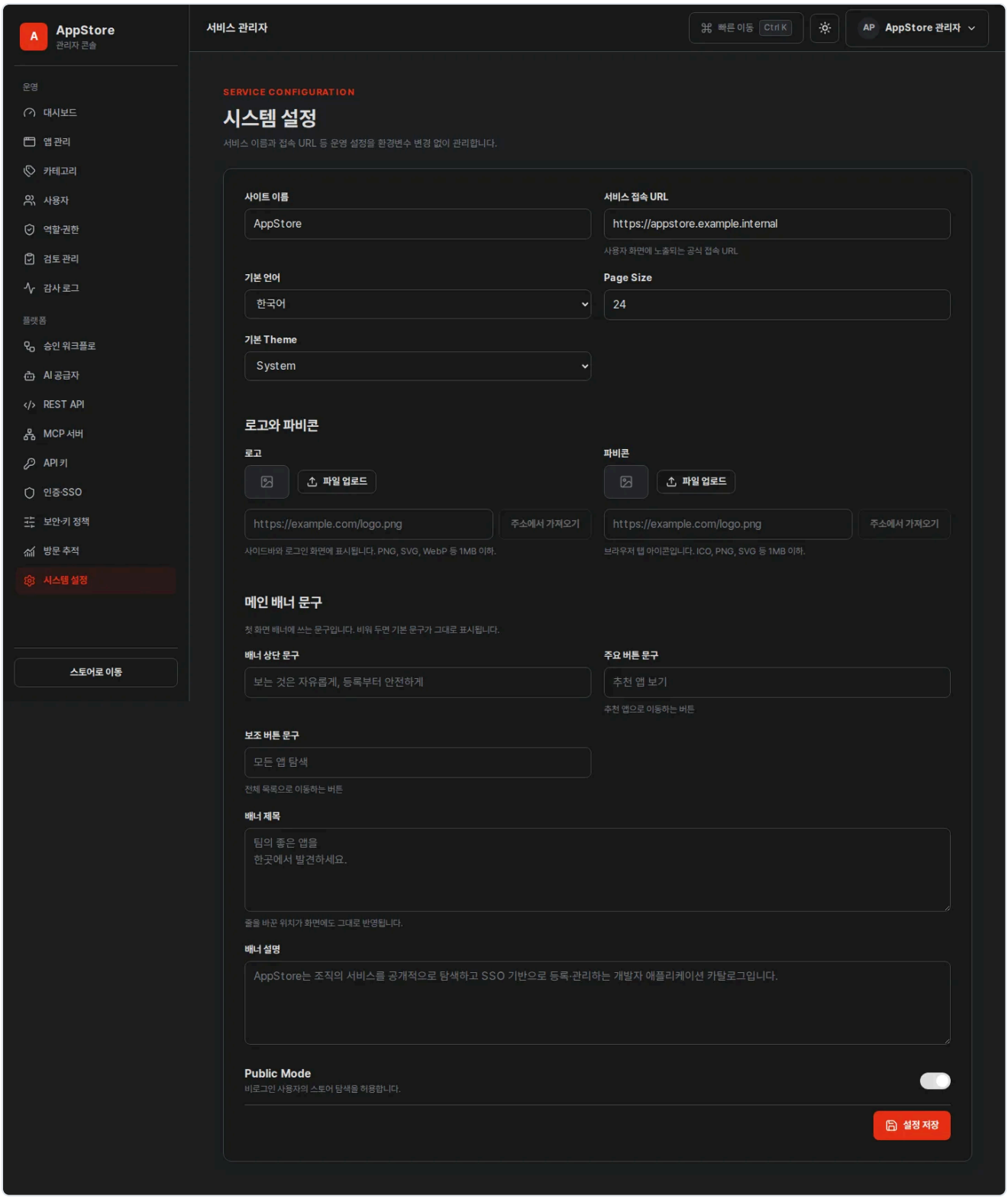
파생 값:

항목	값	바꿀 수 있나
listen 주소	`:8080`	아니오 — 코드에 고정(`config.ListenAddress`). 외부 포트는 `-p` 로 바꿉니다
요청 헤더 timeout	10초	아니오
요청 읽기 timeout	30초	아니오
idle timeout	2분	아니오
종료 유예	20초	아니오

그 밖의 운영 설정 — 서비스 URL, OIDC, 역할 매핑, 워크플로, 키 정책, AI, REST API, MCP, 브랜딩 — 은 **환경 변수가 아니라 관리자 화면과 PostgreSQL**에서 관리합니다. 컨테이너를 다시 만들지 않고 바꿀 수 있고, 변경은 감사 로그에 남습니다.

> `ENCRYPTION_KEY` 는 첫 기동에서 지문이 DB에 기록됩니다. 다른 키로 기동하면 `ENCRYPTION_KEY does not match the initialized database` 로 실패하며 서비스가 뜨지 않습니다. 키를 잃으면 암호화된 OIDC·AI secret은 복원할 수 없습니다.

3.2. 시스템 설정 — 서비스 URL과 배너

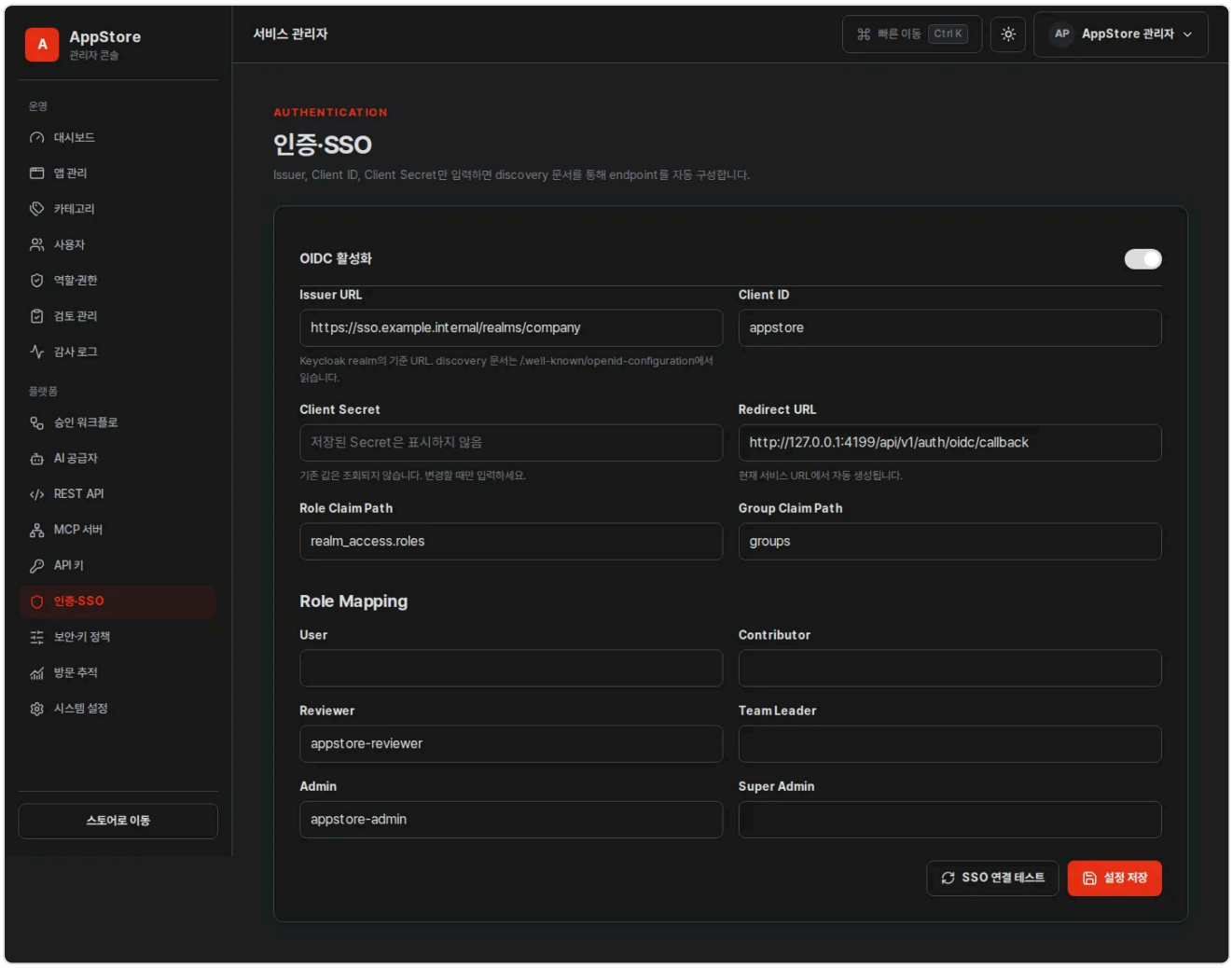


시스템 설정 — 사이트 이름, 서비스 접속 URL, 테마, 공개 모드와 배너 문구

항목	초기값	설명
사이트 이름	`Dev App Store`	화면 상단과 로그인 화면에 노출
서비스 접속 URL	빈 값	사용자에게 노출되는 공식 HTTPS origin. OIDC Redirect URL이 이 값에서 자동 생성됩니다
기본 Theme	`system`	light / dark / system
기본 언어	`ko`	
페이지 크기	`24`	목록 한 쪽에 보여 줄 앱 수
공개 모드	켬	끄면 비로그인 탐색이 막히고 익명 요청은 *익명 AppStore 탐색이 비활성화되어 있습니다.* 로 거부됩니다
배너 상단 문구 / 제목 / 설명 / 주요·보조 버튼 문구	빈 값	비워 두면 기본 문구가 그대로 나옵니다. 제목의 줄바꿈은 화면에 그대로 반영됩니다
로고 / 파비콘	없음	**파일 업로드** 또는 **주소에서 가져오기**. 1MB 이하 PNG·JPEG·WebP·GIF·SVG·ICO

로고와 파비콘은 파일 시스템이 아니라 **PostgreSQL에 저장**됩니다. 이미지를 주소로 가져와도 서버가 한 번 내려받아 저장한 뒤 이 서비스의 origin에서 제공하므로, 업그레이드로 컨테이너를 교체해도 남고 원격 이미지를 막는 콘텐츠 보안 정책에도 걸리지 않습니다. 로고는 로그인 전 화면에도 보이므로 비공개 정보가 담기지 않았는지 확인하세요.

3.3. Keycloak OIDC



인증·SSO — Issuer, Client ID, 가려진 Secret, 역할 매핑과 연결 테스트

Keycloak 쪽 준비:

1. 표준 Authorization Code Flow를 허용하는 confidential client를 만듭니다.
2. 화면의 **Redirect URL**(서비스 접속 URL + `/api/v1/auth/oidc/callback`)을 Valid Redirect URIs에 그대로 등록합니다.
3. Web Origin과 post-logout redirect를 조직 정책에 맞게 제한합니다.

AppStore 쪽 입력:

칸	초기값	설명
OIDC 활성화	꺼짐	연결 테스트가 성공한 뒤에 켭니다
Issuer URL	빈 값	realm까지 포함한 기준 URL(예: `https://sso.example.internal/realms/company`). `/.well-known/...` 은 붙이지 않습니다
Client ID / Client Secret	빈 값	저장한 Secret은 다시 조회할 수 없습니다. 화면에는 *저장됨 Secret은 표시하지 않음* 만 보입니다
Role Claim Path	`realm_access.roles`	
Group Claim Path	`groups`	
Role Mapping	`appstore-user` → user 등 6개 기본 매핑	아래 4장 참조

연결 테스트는 저장 전 화면에 입력한 Issuer를 그대로 시험합니다. discovery 문서 URL, issuer 일치, authorization·token·userinfo·logout·JWKS endpoint와 PKCE 지원 여부를 확인합니다. 실패 메시지는 원인을 그대로 돌려줍니다.

실패 메시지	원인과 조치
`...에 연결하지 못했습니다`	폐쇄망에서 컨테이너가 Keycloak에 닿지 못하거나 TLS를 신뢰하지 못합니다
`... 요청이 HTTP 404를 반환했습니다`	realm 경로 오타. Issuer에 realm이 포함돼야 합니다
`discovery 문서에 필수 항목이 없습니다: ...`	응답이 discovery 문서가 아닙니다. reverse proxy가 로그인·오류 페이지를 대신 주고 있는지 확인
`discovery 문서의 issuer(...)가 입력한 Issuer URL(...)과 다릅니다`	Keycloak이 내부 hostname으로 issuer를 발행합니다. `KC_HOSTNAME` 을 사용자에게 노출되는 주소로 맞춥니다

Client Secret과 refresh token은 브라우저에 저장되지 않습니다. 세션 쿠키는 서버가 Secure·HttpOnly로 관리합니다.

---

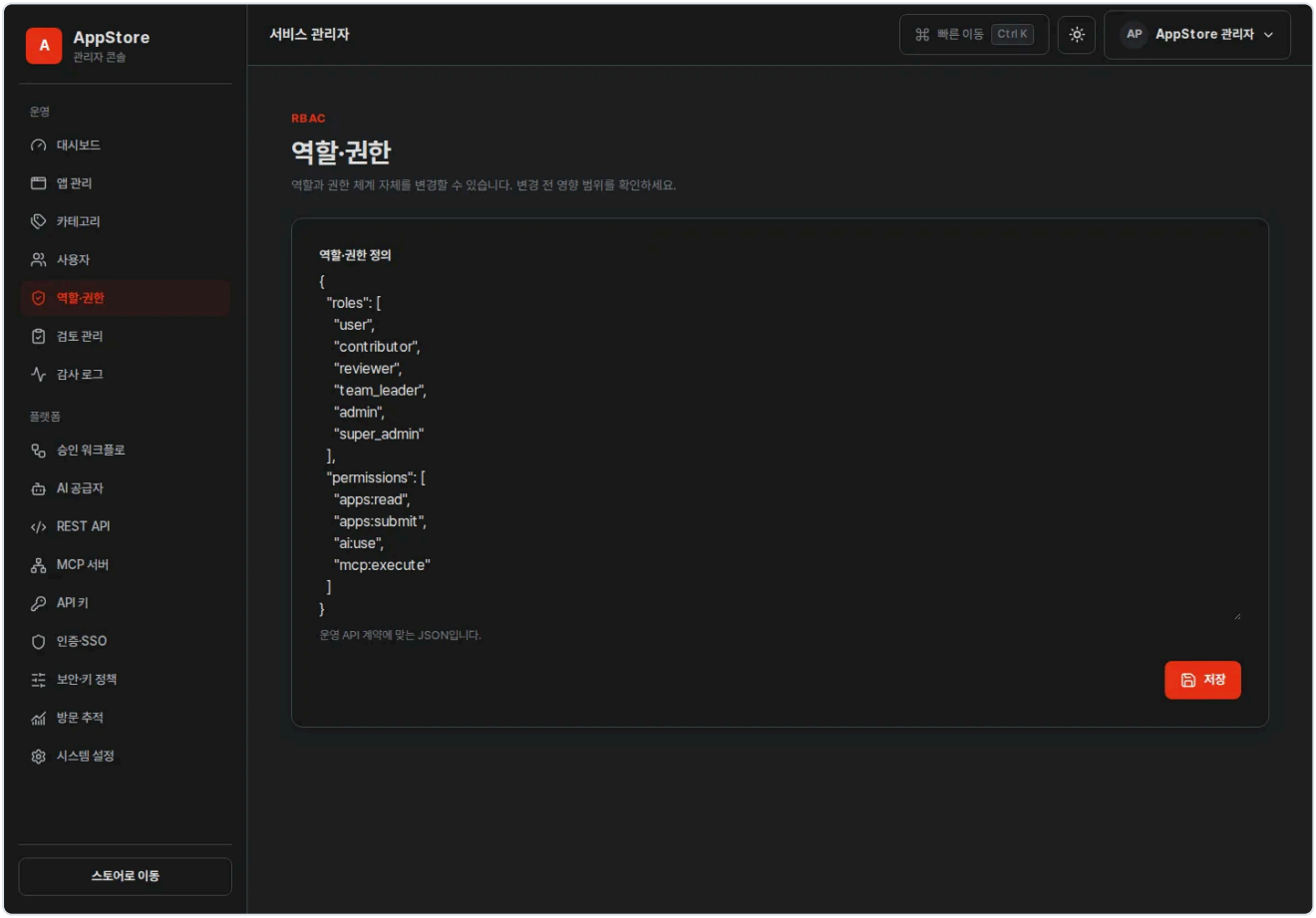
4. 계정과 권한

4.1. 역할

기본 역할 6개와 각 역할이 받는 권한입니다(`migrations/000001_init.sql` 의 시드 값).

역할	key	기본 권한
User	`user`	`apps:read`, `favorites:read`, `favorites:write`, `keys:manage`, `mcp:read`
Contributor	`contributor`	User + `apps:write`, `apps:update`, `apps:delete`, `apps:submit`, `ai:use`, `mcp:execute`
Reviewer	`reviewer`	`apps:read`, `reviews:read`, `reviews:decide`, `mcp:read`
Team Leader	`team_leader`	Reviewer와 동일. 다만 자기 팀의 검토 건만 보고 처리합니다
Admin	`admin`	전체 권한
Super Admin	`super_admin`	전체 권한. Bootstrap·복구 관리자

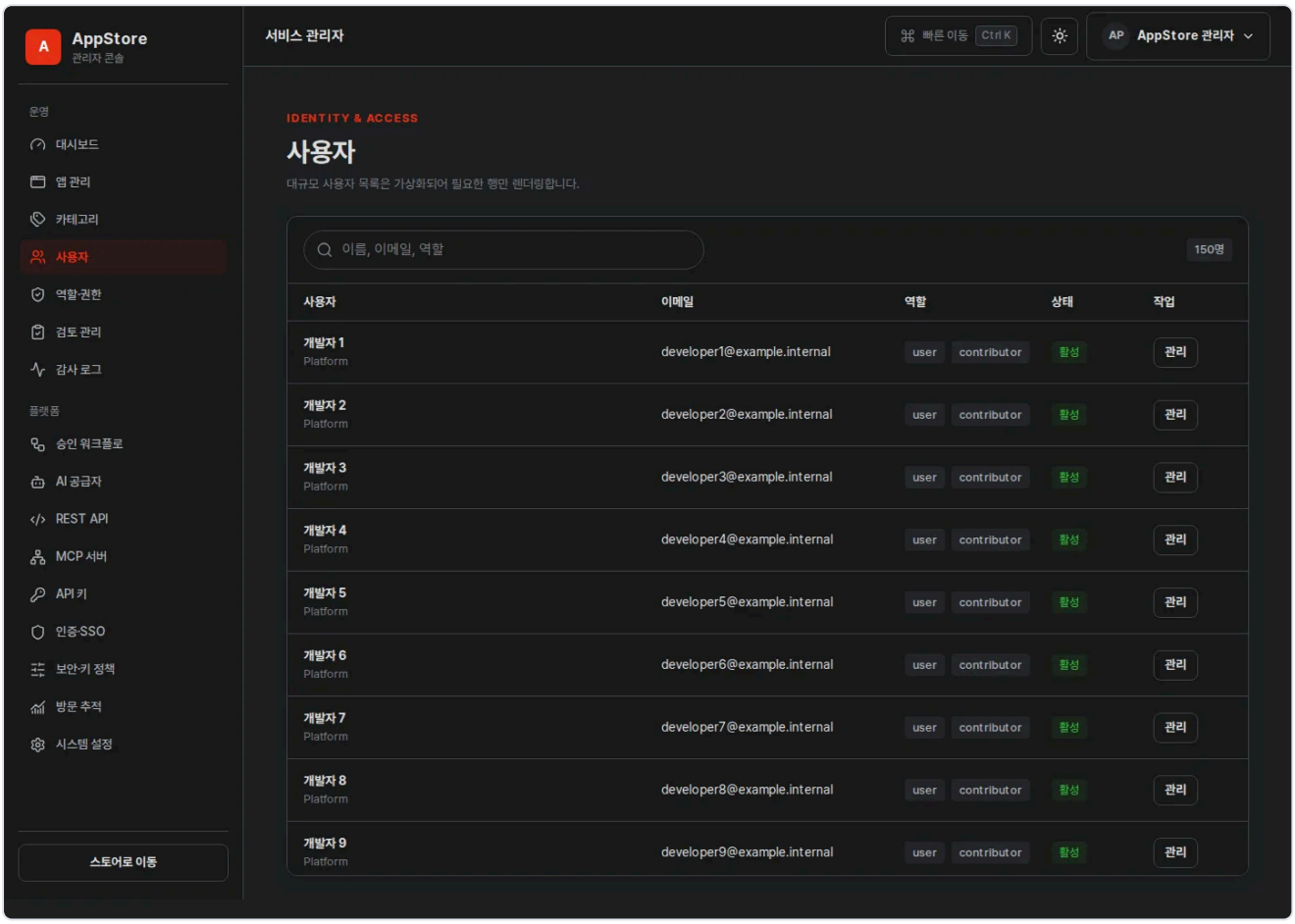
권한 key 전체: `apps:read` · `apps:write` · `apps:update` · `apps:delete` · `apps:submit` · `apps:manage` · `favorites:read` · `favorites:write` · `reviews:read` · `reviews:decide` · `keys:manage` · `ai:use` · `mcp:read` · `mcp:execute` · `users:manage` · `roles:manage` · `settings:read` · `settings:write` · `audit:read`.



역할·권한 — 역할과 권한 체계 자체를 편집하는 화면

역할과 권한 정의는 화면에서 JSON으로 편집할 수 있습니다. 기본 시스템 역할은 **key를 바꿀 수 없고 삭제할 수 없습니다**(`기본 시스템 역할의 key는 변경할 수 없습니다.` / `기본 시스템 역할은 삭제할 수 없습니다.`). 바꾸기 전에 영향 범위를 확인하세요.

4.2. 사용자



사용자 — 프로필, 활성 상태와 AppStore 역할을 변경하는 화면

이름·이메일·역할로 검색하고 역할을 바꿉니다. 대량 목록은 가상화되어 필요한 행만 렌더링합니다. 안전장치가 있습니다 — *현재 로그인한 계정은 비활성화할 수 없습니다.*, *현재 로그인한 계정은 삭제할 수 없습니다.*, *Bootstrap 관리자는 삭제할 수 없습니다.*

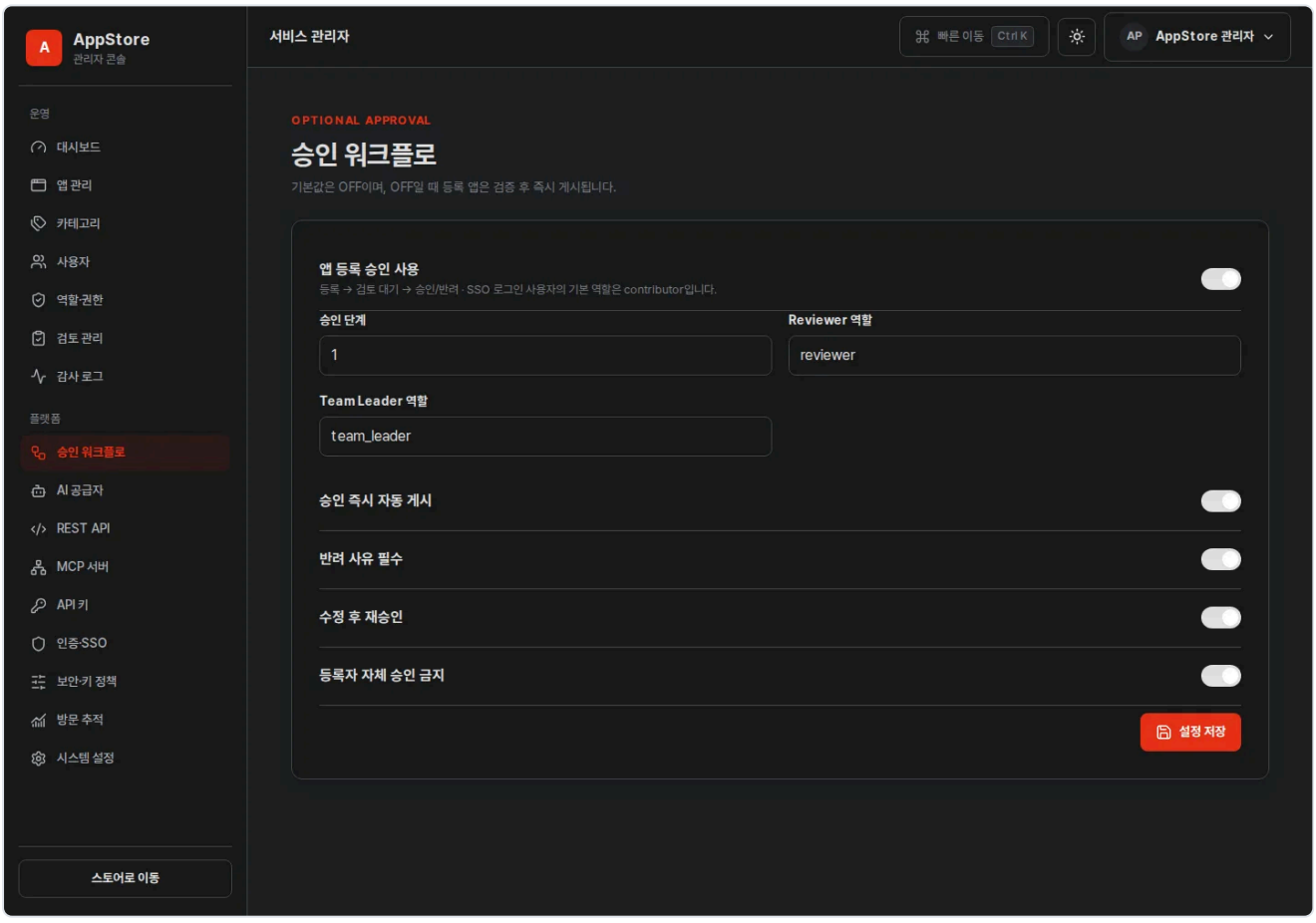
4.3. SSO 역할 매핑

Keycloak 역할 이름은 코드에 고정돼 있지 않습니다. `인증 · SSO` 화면의 Role Mapping에서 외부 값을 AppStore 역할에 붙입니다. 초기값은 다음과 같습니다.

AppStore 역할	기본 외부 역할
User	`appstore-user`
Contributor	`appstore-contributor`
Reviewer	`appstore-reviewer`
Team Leader	`appstore-manager`
Admin	`appstore-admin`
Super Admin	`appstore-super-admin`

매핑을 저장한 뒤 각 역할의 시험 계정으로 `/submit`, `/review`, `/admin`을 실제로 열어 확인하세요. 401(로그인 필요)과 403(권한 거부)을 구분해 기록합니다.

4.4. 승인 워크플로

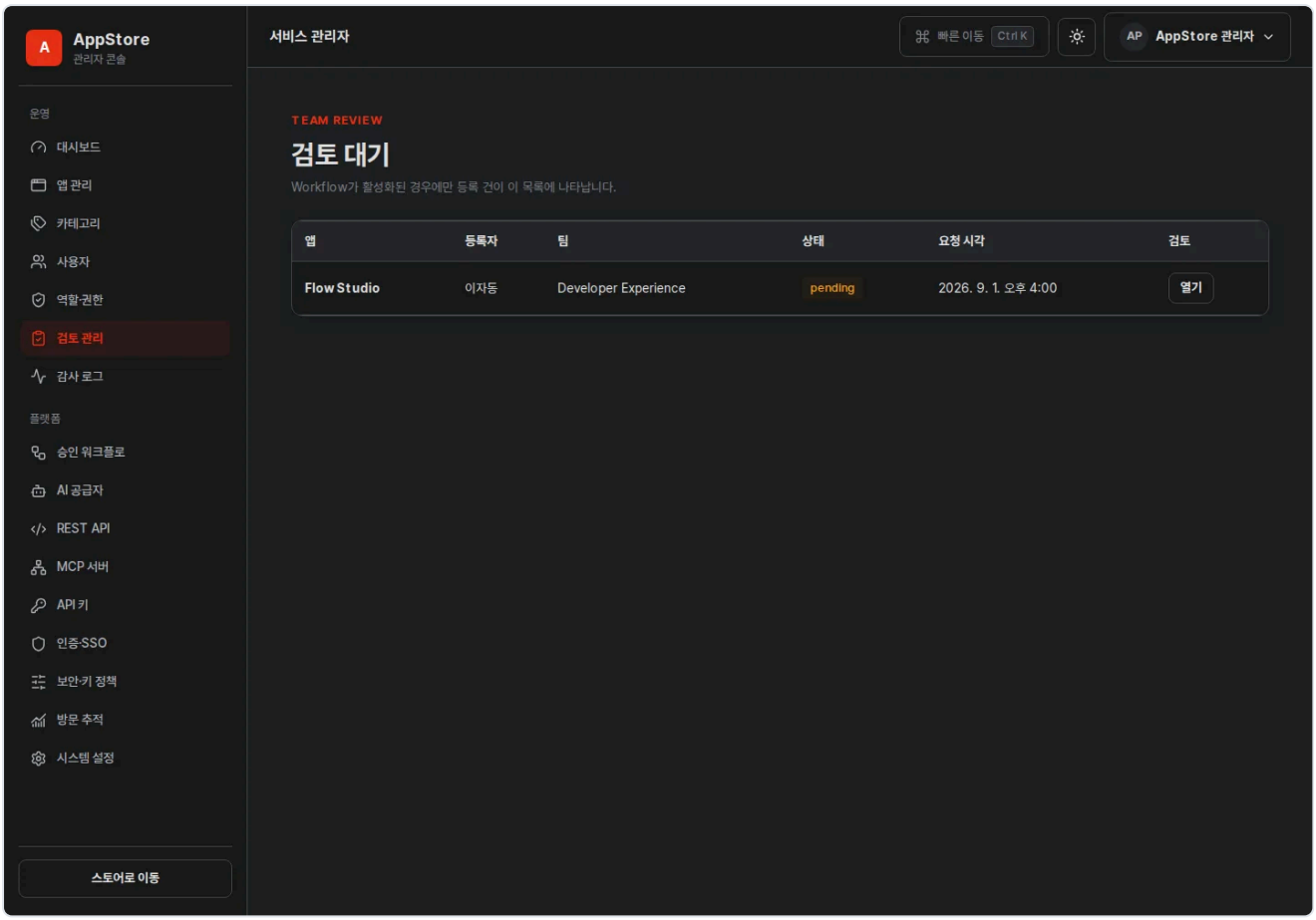


승인 워크플로 — 검토 단계, 자동 게시, 반려 사유 필수 등 승인 정책

설정	초기값	뜻
앱 등록 승인 사용	**꺼짐**	꺼져 있으면 등록이 곧 게시입니다
승인 단계	1 (1~10)	단계마다 검토자가 한 번씩 승인해야 합니다
검토자 역할	`["reviewer"]`	
팀장 역할	`["team_leader"]`	팀장 검토에는 사용자 팀 정보가 필요합니다
승인 즉시 자동 게시	켬	마지막 승인 직후 게시할지
반려 사유 필수	켬	등록자가 무엇을 고쳐야 하는지 알 수 있게 합니다
수정 후 재승인	켬	게시된 앱을 고치면 다시 검토를 받습니다
등록자 자체 승인 금지	켬	등록자와 승인자를 분리합니다

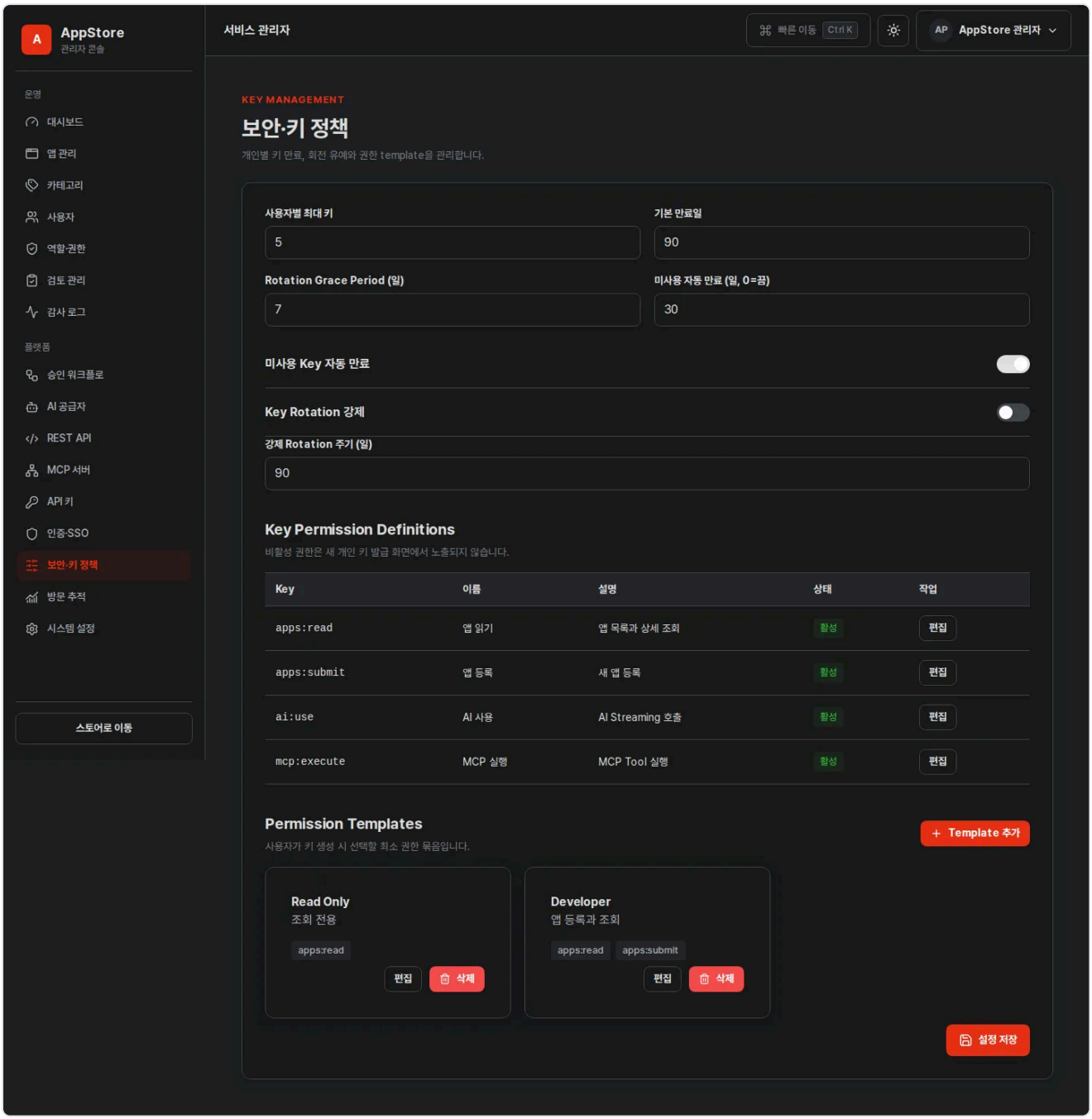
기본 역할이 함께 움직입니다 — 승인을 **켜면** SSO 로그인 사용자의 기본 역할이 `contributor` 가 되어 누구나 제출할 수 있되 게시 전에 검토를 거칩니다. **꺼져 있으면** 제출이 곧 게시이므로 기본 역할은 조회 위주의 `user` 로 유지되고 등록 권한은 역할 매핑으로만 부여됩니다. 기존 사용자에게는 다음 SSO 로그인부터 반영됩니다.

승인이 켜져 있으면 반려된 앱은 소유자가 내용을 고쳐 저장하는 것만으로 다시 1단계 검토 대기로 돌아갑니다. 별도 재제출 조작이 없습니다.



검토 관리 — 서비스 전체의 앱 검토 현황

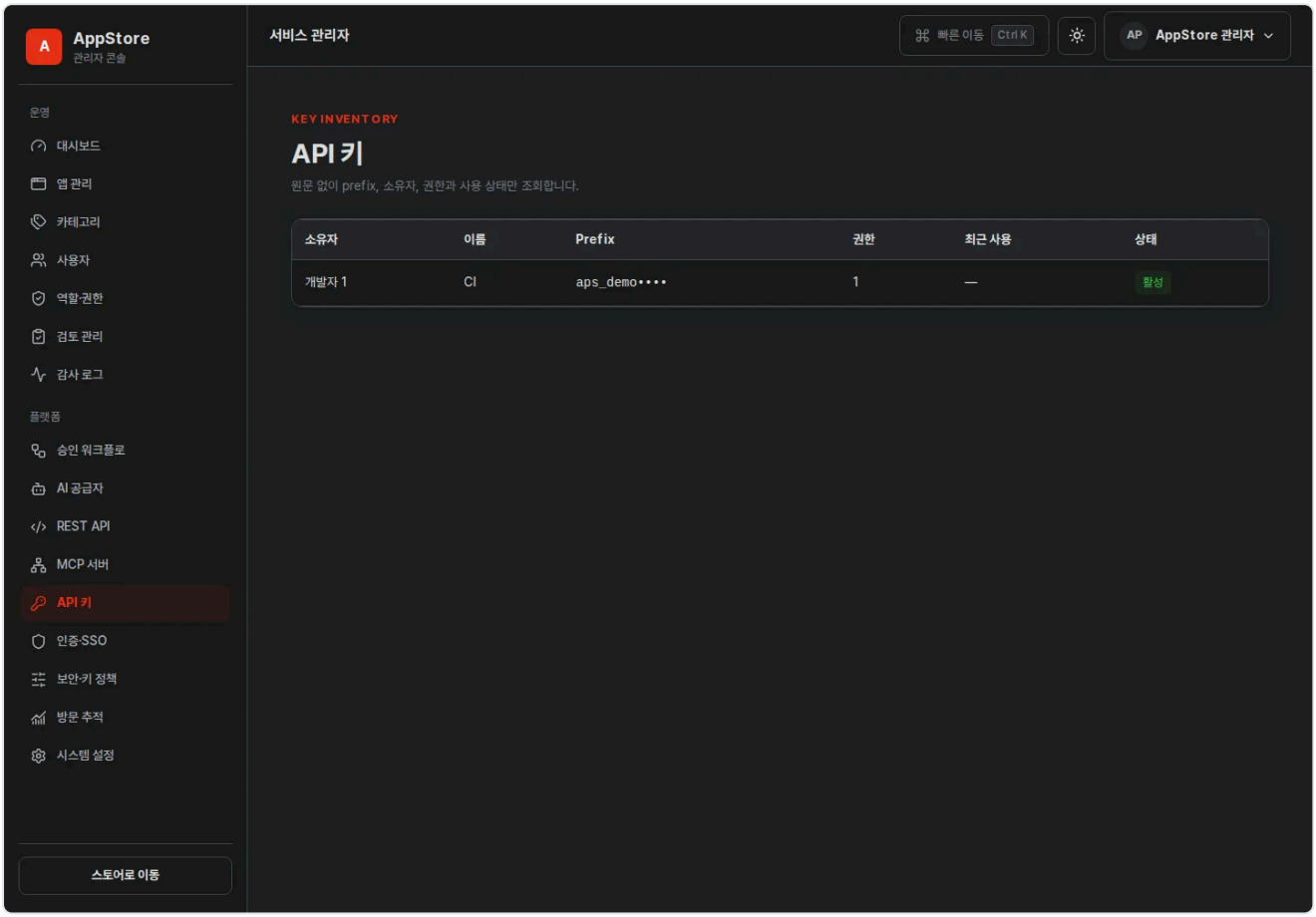
4.5. 개인 키 정책



보안·키 정책 — 사용자별 최대 키, 만료, 회전 유예와 권한 템플릿

화면의 칸	초기값	허용 범위
사용자별 최대 키	5	1~100
기본 만료일	90	1~3650
Rotation Grace Period (일)	7	0~365
미사용 Key 자동 만료 / 미사용 자동 만료 (일, 0=끔)	꺼짐 / 90	1~3650
Key Rotation 강제 / 강제 Rotation 주기 (일)	꺼짐 / 90	1~3650

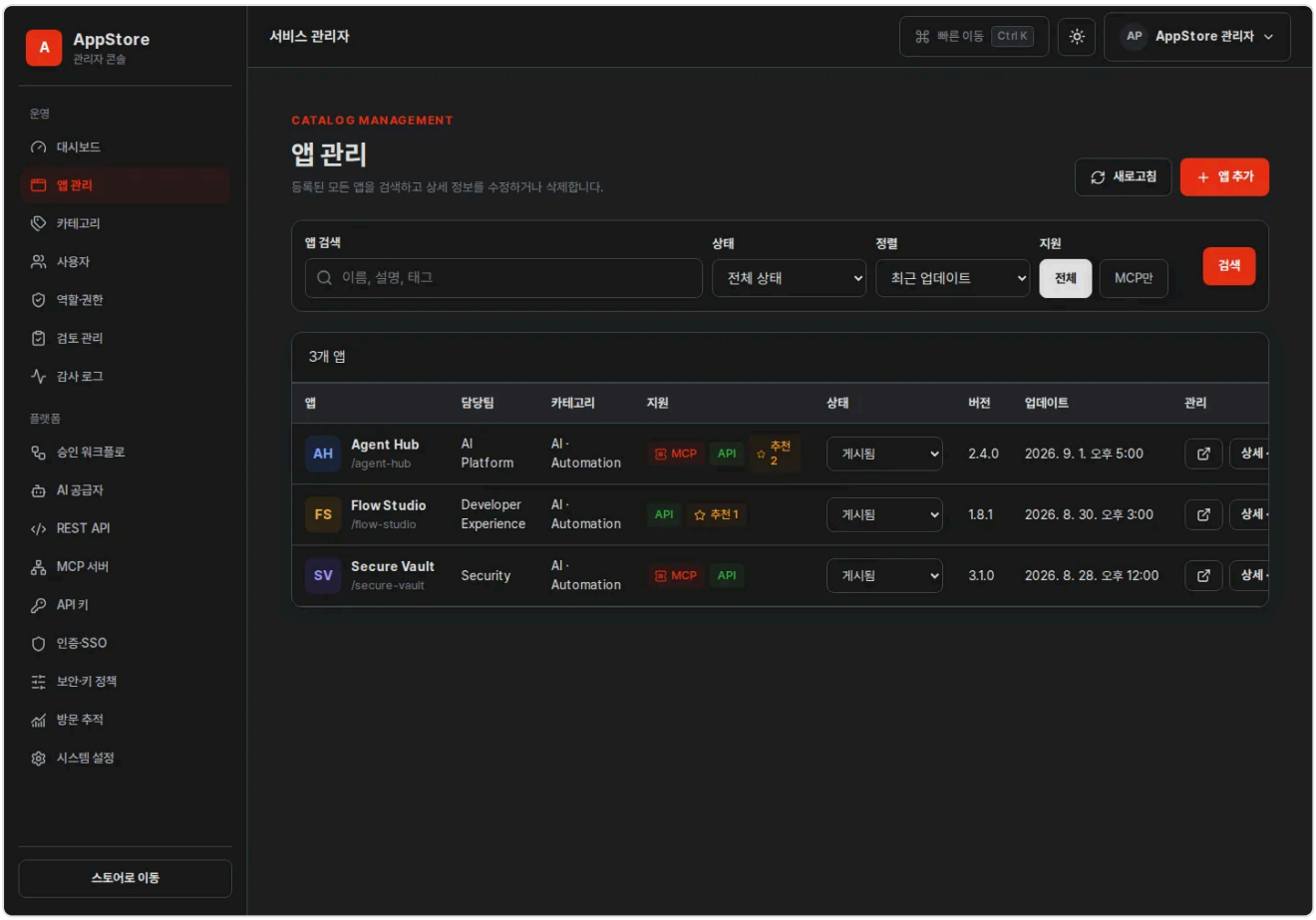
같은 화면에서 Key 권한 정의(사용자가 키에 붙일 수 있는 권한)와 권한 템플릿을 관리합니다. 비활성화한 권한은 새 키 발급 화면에 나타나지 않고, 템플릿에는 활성 권한만 넣을 수 있습니다(*활성화된 Key 권한만 Template에 포함할 수 있습니다.*). 초기 템플릿은 Read Only · Developer · AI Client · MCP Client · Full Access입니다.



API 키 — 서비스 전체 키의 prefix, 소유자, 권한과 사용 상태

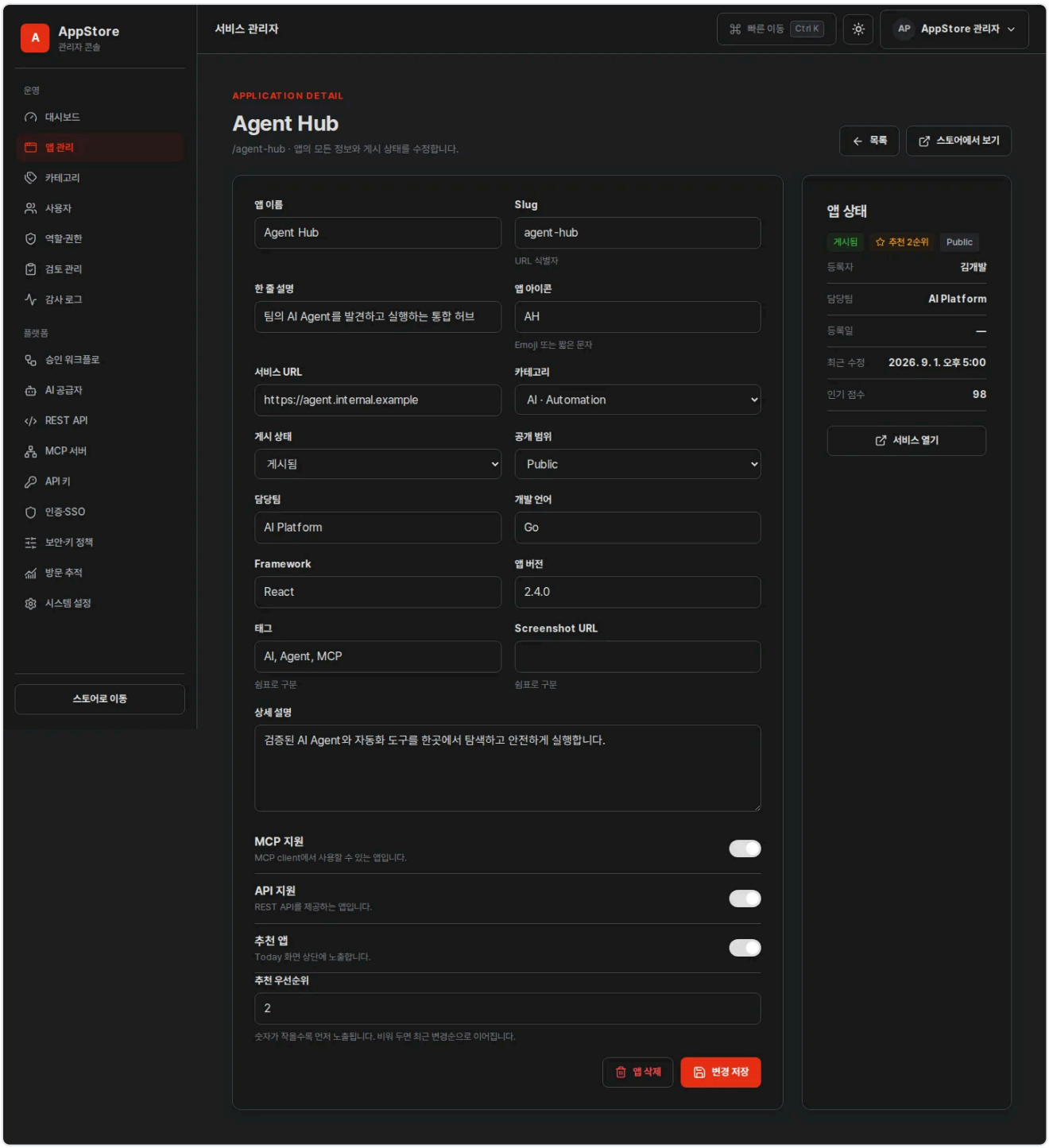
관리자 화면에서도 키 원문은 보이지 않습니다. *원문 없이 prefix, 소유자, 권한과 사용 상태만 조회합니다.* DB에는 prefix와 keyed hash만 저장됩니다.

4.6. 카탈로그 관리



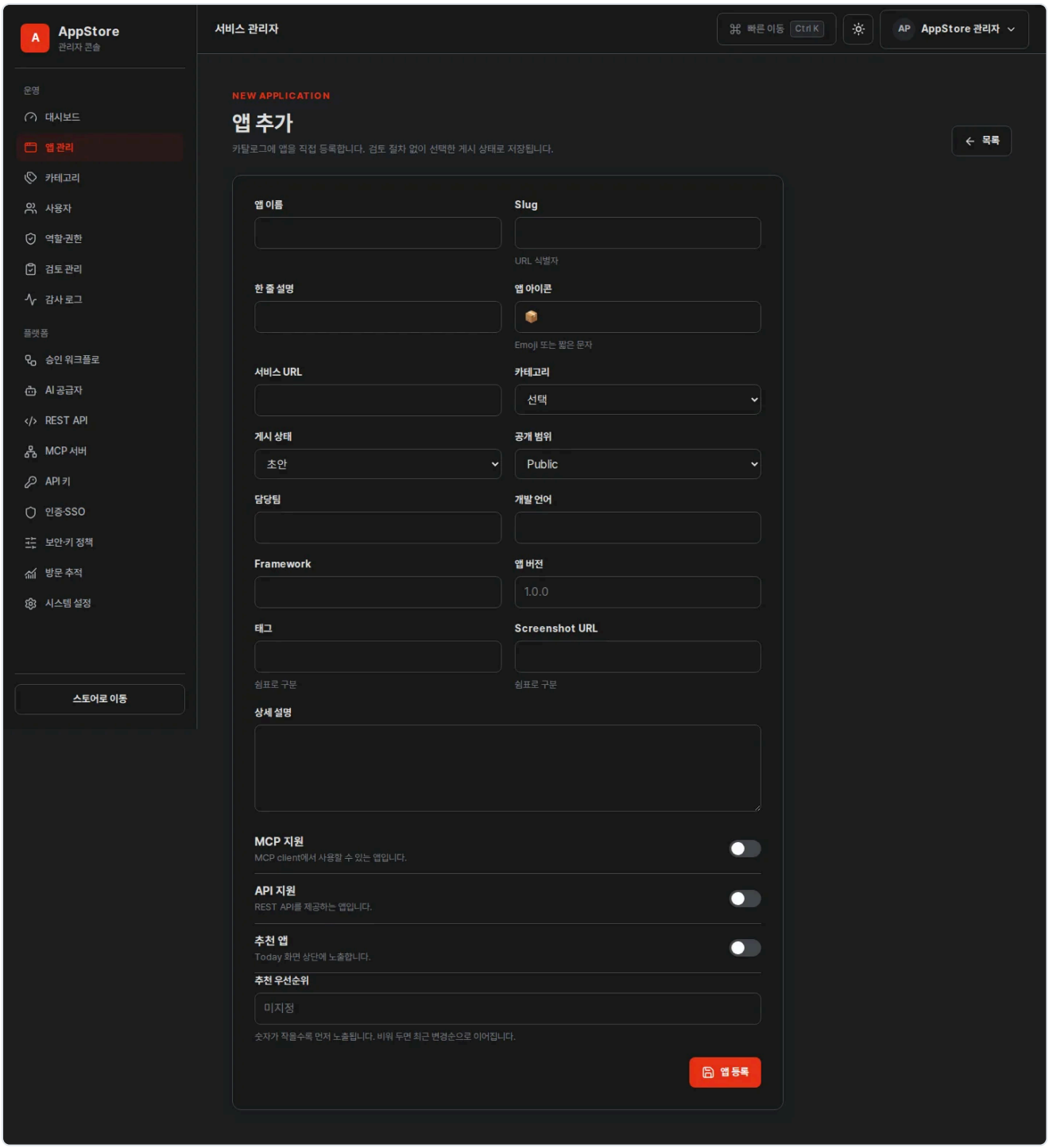
앱 관리 — 등록된 모든 앱을 검색하고 상태를 바꾸는 화면

이름·설명·태그 검색, 게시 상태, 지원 기능, 정렬로 좁힙니다. 조건은 URL에 남습니다. 목록의 상태 select는 게시 상태만 즉시 바꾸고, 나머지 정보는 **상세 · 수정**에서 편집합니다.



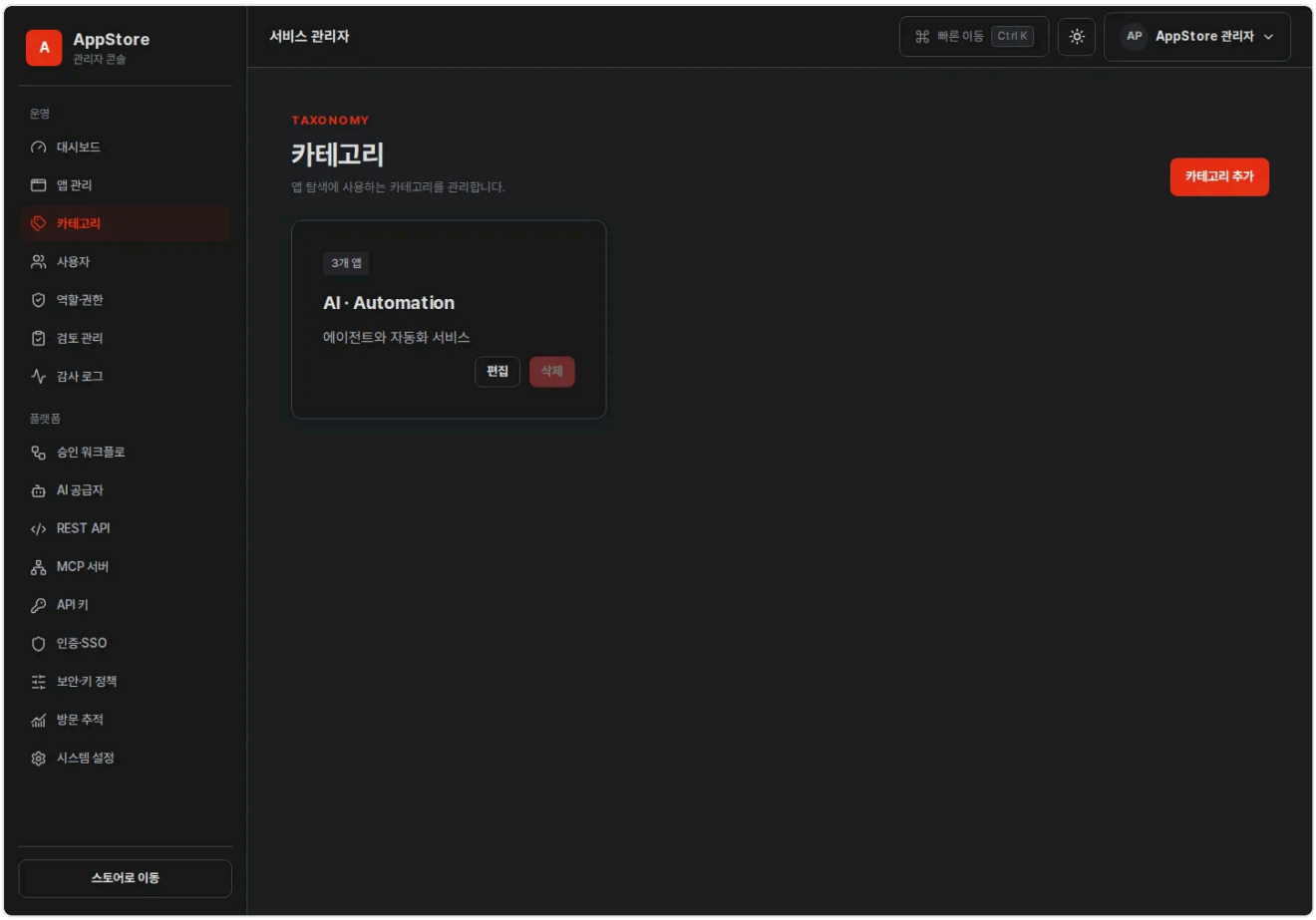
앱 상세 — 앱의 모든 정보와 게시 상태를 수정하는 관리자 화면

- **추천 우선순위** — 숫자가 작을수록 투데이의 추천 진열에서 먼저 나옵니다. 비워 두면 최근 변경순으로 이어집니다.
- 게시를 멈추기만 하려면 상태를 **보관됨**으로 바꿉니다. 카탈로그에서 내려가지만 기록은 남습니다.
- **앱 삭제**는 영구 삭제입니다. *앱과 함께 즐겨찾기, 검토 이력, 버전 기록이 모두 삭제되며 되돌릴 수 없습니다.*



관리자 앱 등록 — 검토 절차 없이 카탈로그에 앱을 직접 등록

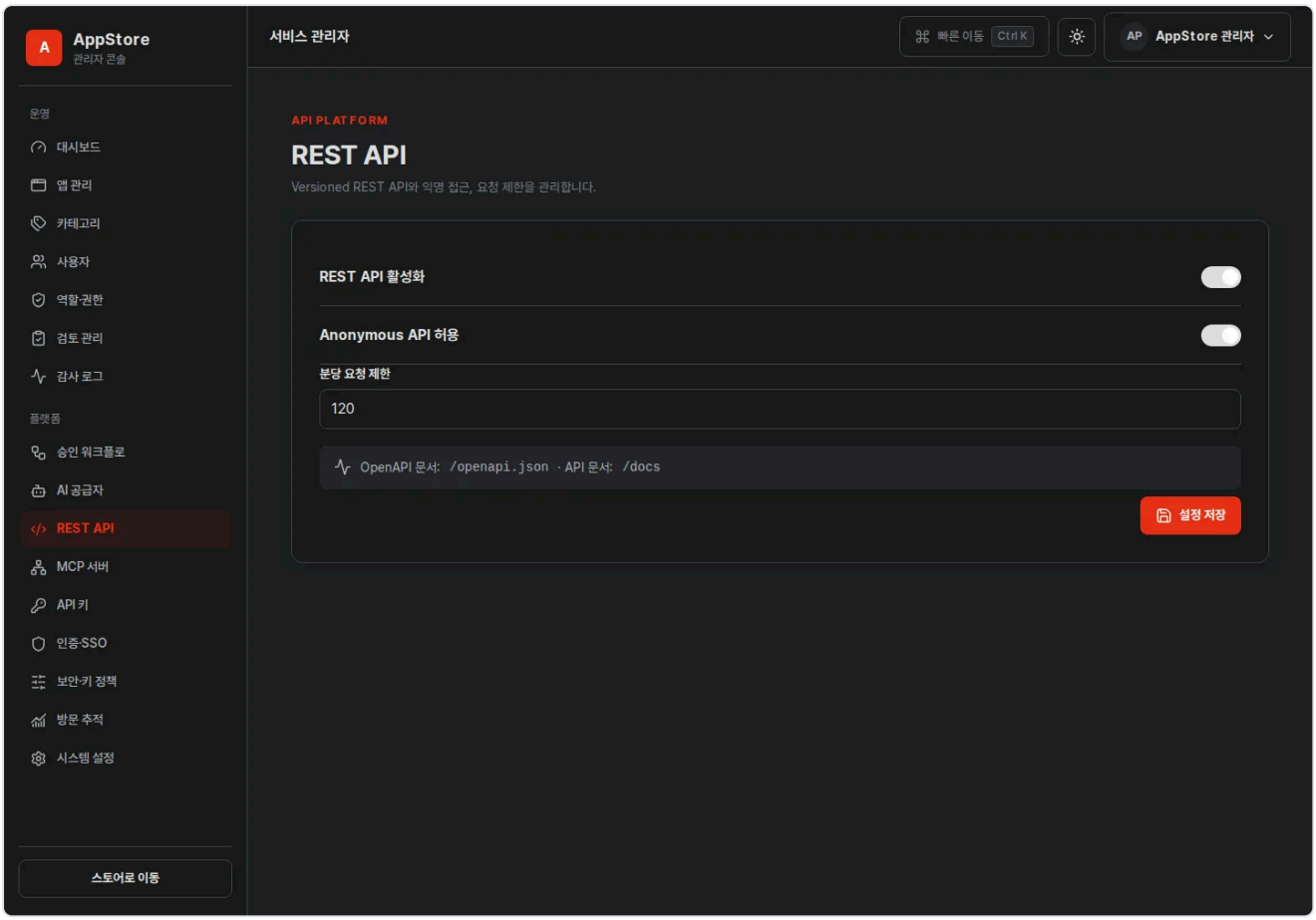
관리자는 검토 절차를 거치지 않고 선택한 게시 상태로 바로 등록할 수 있습니다.



카테고리 — 앱 탐색에 쓰는 카테고리 관리

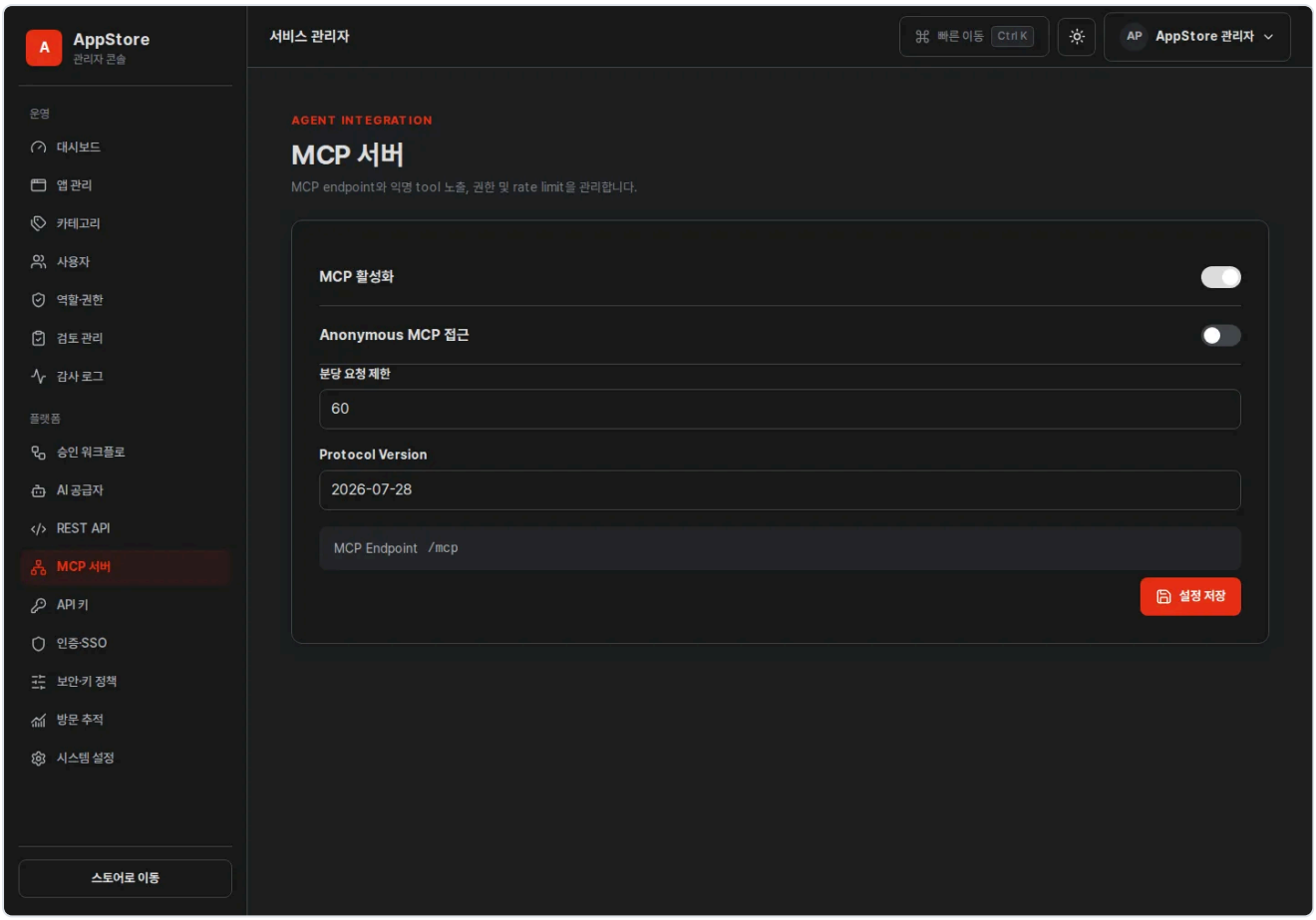
카테고리는 식별자(slug), 이름, 아이콘, 설명, 정렬 순서를 받습니다. 식별자와 이름은 비워 둘 수 없고, 아이콘을 비우면 📦 가 기본값으로 들어갑니다. *사용 중인 카테고리는 삭제할 수 없습니다.*

4.7. REST API · MCP · AI



REST API — 공개 범위와 분당 요청 제한

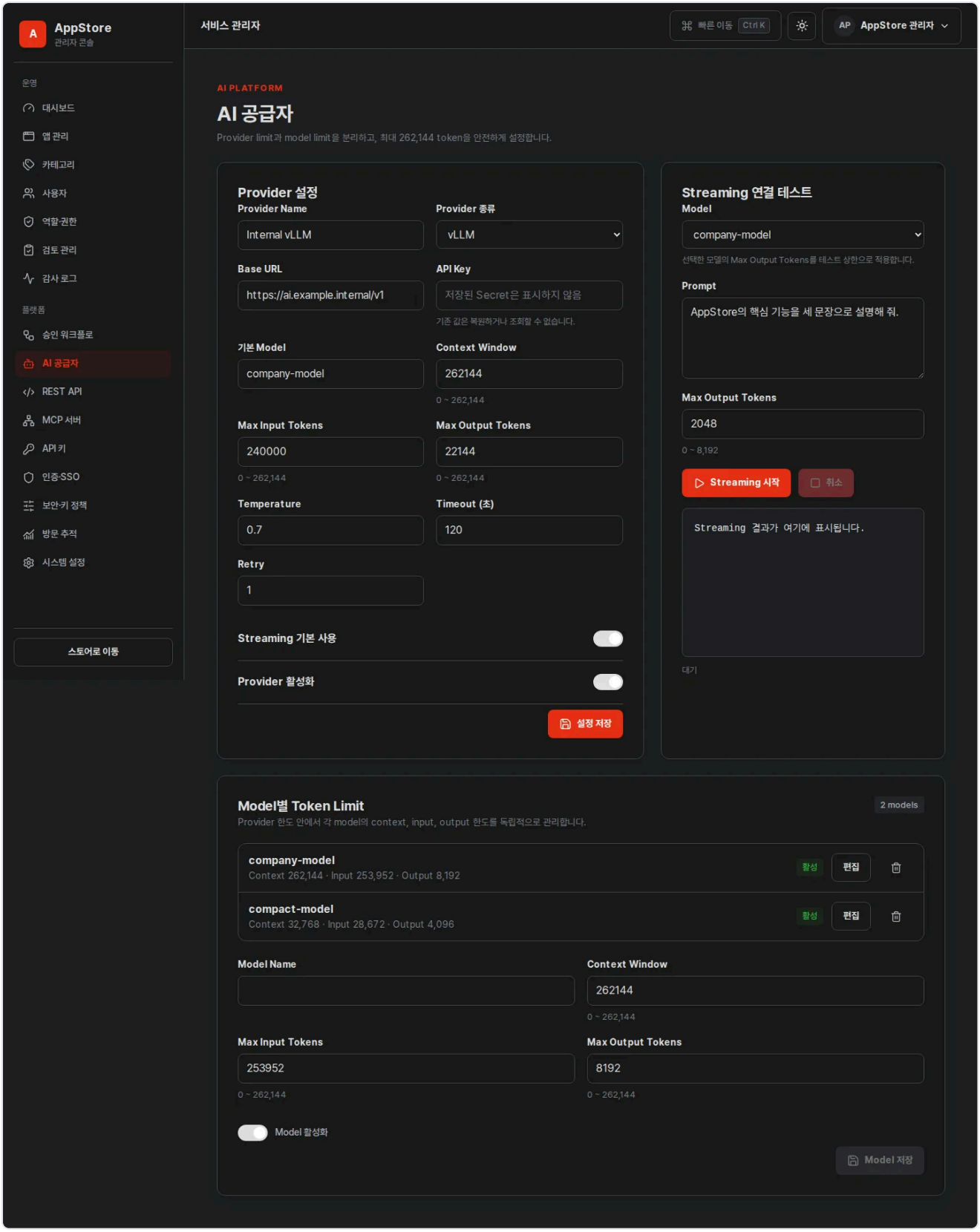
설정	초기값
API 사용	켬
익명 조회 허용	켬
분당 요청 제한	120



MCP 서버 — endpoint 사용 여부, 익명 tool 노출과 rate limit

설정	초기값
MCP 사용	켬
익명 조회 허용	켬
분당 요청 제한	60
Protocol Version	`2026-07-28`

MCP tool은 호출자의 권한에 따라 다르게 보입니다. 익명·인증 사용자는 조회 tool(`apps_list`, `apps_search`, `app_get`, `categories_list`, `featured_apps`, `trending_apps`, `mcp_apps`)을, 인증 사용자는 자기 앱 tool(`my_apps`, `app_submit`, `app_update`)을, 관리자는 운영 tool(`apps_manage`, `settings_get`, `workflow_get`)을 함께 봅니다.

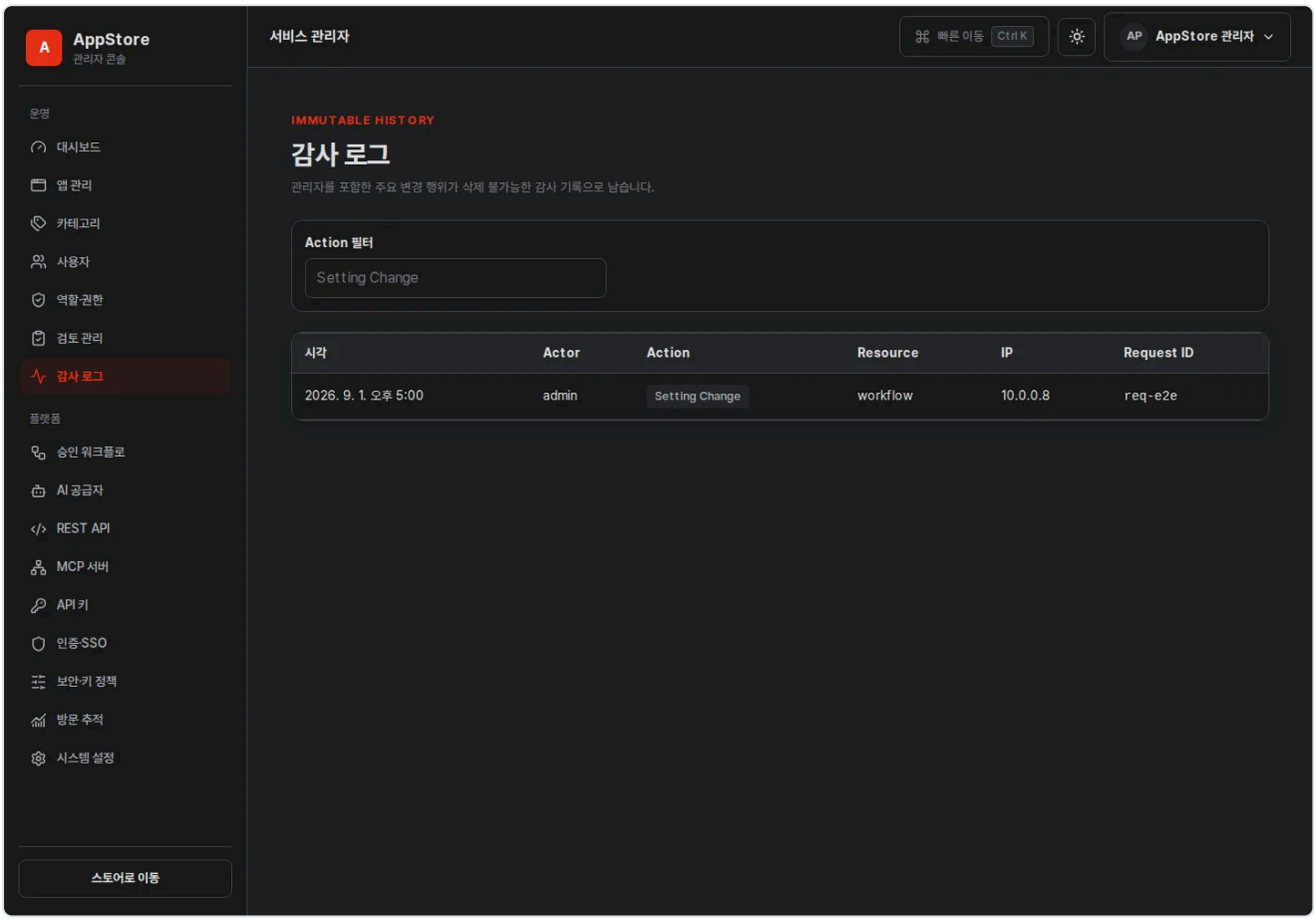


AI 공급자 — provider와 모델별 token 한도, streaming 설정

Provider의 Base URL·API Key·기본 모델과, 모델별 Context Window / Max Input Tokens / Max Output Tokens를 따로 관리합니다. 최대 262,144 token까지 설정할 수 있고, *Model의 최대 입력·출력 token 합은 context window를 넘을 수 없습니다.* API Key는 `ENCRYPTION_KEY` 로 암호화되어 저장되고 화면에는 마스크만 보입니다.

Provider 한도와 모델 한도를 분리해 두어야 upstream이 지원하지 않는 값을 강제로 보내지 않습니다.

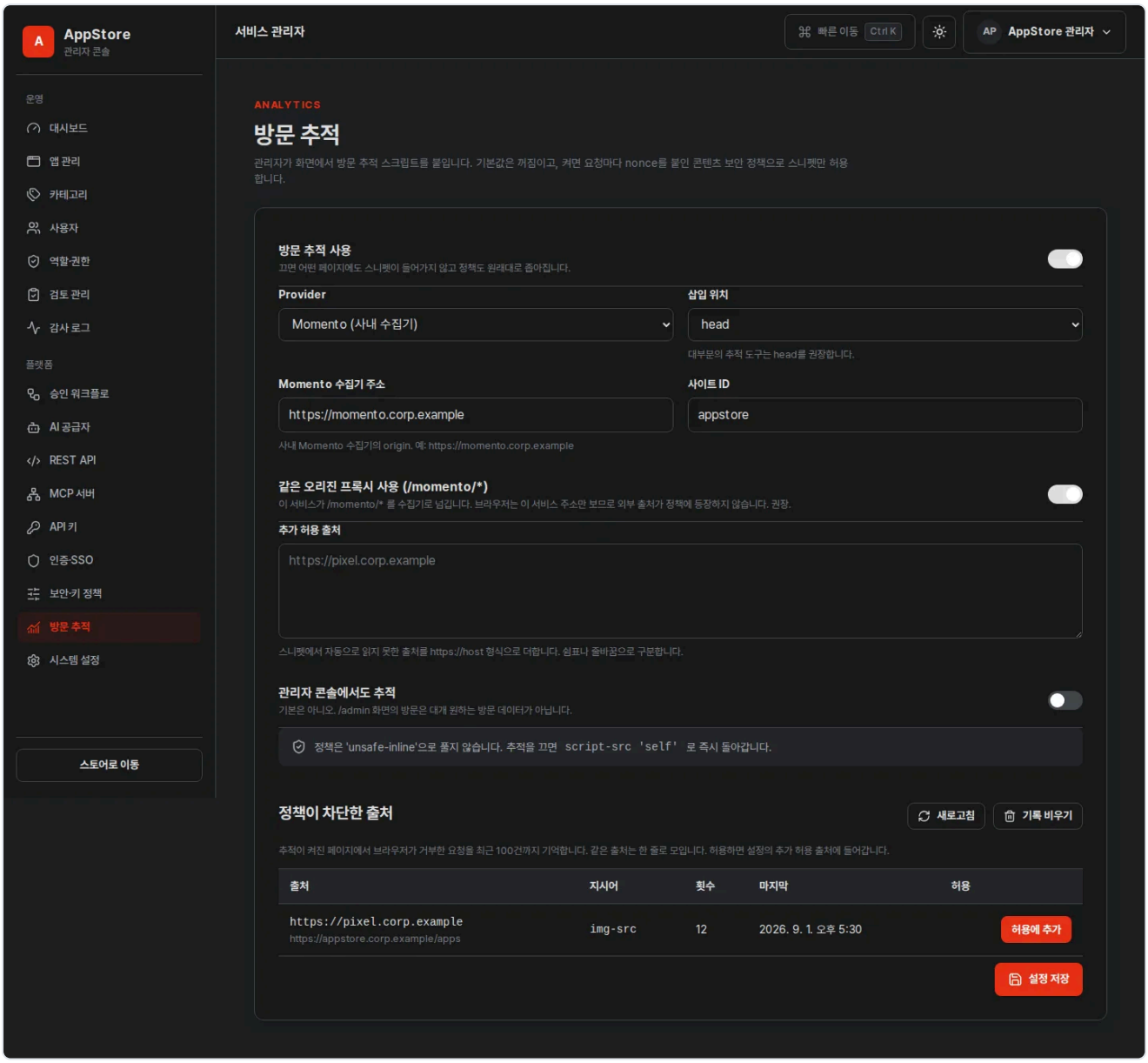
4.8. 감사 로그



감사 로그 — 삭제할 수 없는 관리자 행위 기록

로그인, 앱 변경, 승인·반려, 키 수명주기, 역할·설정 변경이 actor·대상·시각·요청 ID와 함께 남습니다. DB 트리거(`audit_logs_immutable`)가 UPDATE와 DELETE를 막으므로 **관리자에게도 삭제 기능이 없습니다.**

4.9. 방문 추적과 콘텐츠 보안 정책



방문 추적 — provider, Momento 프록시, 허용 출처와 정책이 차단한 출처

관리자 → **방문 추적**에서 방문 추적 스크립트를 화면에 붙입니다. 설정은 환경 변수가 아니라 PostgreSQL(`system_settings.analytics`)에 저장되므로 수집기 주소가 바뀌어도 재배포 없이 바꿀 수 있고, 변경은 감사 로그에 `analytics.setting.update` 로 남습니다. **기본값은 꺼짐**입니다. 새로 설치한 곳은 이 화면을 손대기 전까지 아무것도 달라지지 않습니다.

칸	초기값	설명
방문 추적 사용	꺼짐	끄면 어떤 페이지에도 스니펫이 들어가지 않고 정책도 원래대로 좁아집니다
Provider	사용 안 함	`Momento` · `GA4` · `GTM` · `Matomo` · `직접 붙여 넣기`
Momento 수집기 주소 / 사이트 ID	빈 값	사내 Momento 수집기의 origin과 사이트 id
같은 오리진 프록시 사용	켬	이 서비스가 `/momento/*` 를 수집기로 넘깁니다. 브라우저는 이 서비스 주소만 보므로 외부 출처가 정책에 등장하지 않습니다
측정 ID / Matomo 주소 · 사이트 ID	빈 값	GA4·GTM·Matomo용. 영문·숫자· `_ . : -` 64자 이내
추적 스니펫	빈 값	직접 붙여 넣기용 `<script>` 태그. **8KB**를 넘으면 저장되지 않습니다
추가 허용 출처	빈 값	스니펫에서 자동으로 읽지 못한 `https://host` 를 쉼표·줄바꿈으로 나열
관리자 콘솔에서도 추적	꺼짐	`/admin` 화면은 기본으로 추적하지 않습니다
삽입 위치	head	`head` 또는 `body`

Momento를 먼저 씁니다. Momento는 사내 자체 호스팅 수집기라 방문 데이터가 밖으로 나가지 않는 유일한 선택지입니다. 같은 오리진 프록시를 켜 두면 스니펫은 `/momento/tracker.js` 를 읽고 이벤트를 `/momento` 로 보내며, 서비스가 그 요청을 수집기 주소로 전달합니다. 전달할 때 이 서비스의 세션 쿠키와 `Authorization` 헤더는 떼어 냅니다 — 수집기는 이벤트만 받고 방문자의 세션은 받지 않습니다. 추적이 꺼져 있거나 provider가 Momento가 아니면 `/momento/*` 는 404입니다. GA4·GTM은 데이터가 Google로 나가므로 폐쇄망에서는 동작하지 않습니다.

콘텐츠 보안 정책. 모든 화면은 `script-src 'self'` 로 잠겨 있어 스니펫을 그냥 붙이면 브라우저가 조용히 차단합니다. 추적을 켜면 서비스는 다음을 요청마다 합니다.

1. 무작위 nonce를 만들어 정책의 `script-src` 에 `'nonce-…'` 로 넣고, 스니펫의 **모든** `<script>` 태그에 같은 값을 붙입니다. `'unsafe-inline'` 은 쓰지 않습니다. 한 번 풀면 그 앱의 모든 인라인 스크립트가 함께 허용되고 추적을 끈 뒤에도 느슨한 채 남기 때문입니다.
2. provider가 정한 출처(예: Matomo 주소)와 붙여 넣은 스니펫에서 읽어 낸 `http(s)` 출처, 추가 허용 출처를 `script-src` · `connect-src` · `img-src` 에 더합니다.
3. `report-uri /api/v1/analytics/csp-report` 를 정책에 넣어 브라우저가 거부한 요청을 신고하게 합니다. 이 신고는 추적이 켜진 동안에만 옵니다.

`/api/*` · `/mcp` · `/health*` · `/momento/*` 같은 비화면 경로는 `default-src 'none'` 으로 오히려 더 좁고, 관리자 콘솔은 *관리자 콘솔에서도 추적*을 켰을 때만 스니펫을 받습니다. 추적을 끄면 정책은 7.2의 원래 문자열로 즉시 돌아갑니다.

정책이 차단한 출처. 같은 화면 아래에 브라우저가 신고한 차단이 출처·지시어·횟수로 모입니다(메모리에 최근 100건, 같은 출처는 한 줄). 화면이 비어 보이는데 여기에 출처가 있으면 그 출처가 막힌 것입니다. **허용에 추가**를 누르면 추가 허용 출처에 들어가고 다음 페이지부터 정책에 실립니다. 스니펫을 고친 뒤에는 **기록 비우기**로 지우고 다시 띄워 남는 차단이 없는지 확인합니다. Momento 프록시를 쓰면 외부 출처가 없으니 이 표가 비어 있는 것이 정상입니다.

---

5. 운영

5.1. 상태 점검 엔드포인트

라우트 등록 자리(`internal/httpapi/server.go`)에서 확인한 메서드와 경로입니다.

메서드	경로	응답	용도
`GET`	`/health/live` (별칭 `/healthz`)	`{"status":"ok","uptimeSeconds":…}`	프로세스 살아 있는지. 컨테이너 HEALTHCHECK가 이 경로를 씁니다
`GET`	`/health/ready` (별칭 `/readyz`)	200 `{"status":"ready"}` / 503 `{"status":"not_ready"}`	DB ping(2초 timeout). load balancer는 이 경로를 봅니다
`GET`	`/api/version`	version · commit · buildDate	배포된 빌드 확인
`GET`	`/openapi.json`	OpenAPI 3.1 문서	
`GET`	`/docs`	API 문서 화면	

5.2. 로그

로그는 표준 출력으로 나가는 **JSON 한 줄**입니다(`slog` JSON handler, level INFO). 파일에 쓰지 않으므로 `docker logs` 또는 조직의 로그 수집기로 봅니다.

```
docker logs --tail 200 appstore
docker compose logs --tail 200 appstore
```

실제로 찍히는 메시지:

msg	함께 나오는 필드	언제
`appstore listening`	`address`, `version`	기동 완료
`http request`	`method`, `path`, `status`, `bytes`, `duration_ms`, `request_id`	모든 요청
`panic recovered`	`error`, `request_id`, `stack`	처리 중 패닉. 응답은 500 `INTERNAL_ERROR`
`appstore shutting down`	—	SIGINT/SIGTERM 수신, 20초 유예
`appstore stopped`	`error`	기동 실패 또는 비정상 종료(종료 코드 1)

사용자가 오류 화면에서 알려 준 **요청 ID**로 `request_id`를 검색하면 해당 요청 한 건을 찾을 수 있습니다. 요청 ID는 응답의 `X-Request-ID` 헤더로도 나가고, 클라이언트가 보낸 `X-Request-ID`가 8~128자이면 그 값을 그대로 씁니다.

5.3. 백업

백업 대상은 **PostgreSQL과 `ENCRYPTION_KEY` 두 가지**이고, 서로 다른 저장소에 둡니다. 컨테이너에는 백업할 상태가 없습니다.

```
backup_stamp="$(date -u +%Y%m%dT%H%M%SZ)"
pg_dump --format=custom --no-owner --no-privileges \
  --file="appstore-$backup_stamp.dump" "$POSTGRES_DSN"
pg_restore --list "appstore-$backup_stamp.dump" >/dev/null
sha256sum "appstore-$backup_stamp.dump" >"appstore-$backup_stamp.dump.sha256"
```

dump만으로는 완전한 복구가 아닙니다. OIDC Client Secret과 AI API Key는 `ENCRYPTION_KEY`로 암호화돼 있어 원래 키가 없으면 복호화할 수 없습니다. 키 사본을 승인된 vault에 최소 두 벌 두고, DB 백업 접근자와 키 접근자를 분리하세요.

복원은 격리된 데이터베이스에 먼저 시험합니다.

```
pg_restore --clean --if-exists --no-owner --no-privileges \
  --dbname="$RESTORE_POSTGRES_DSN" appstore-20260901T080000Z.dump
```

복원 뒤에는 같은 버전의 이미지와 원래 `ENCRYPTION_KEY`로 기동해 `/health/ready`와 `/api/version`을 확인하고, 앱 수·사용자·역할·감사 로그 기간과 OIDC·AI 연결 테스트 성공을 대조합니다. 자세한 절차는 백업·복구 가이드에 있습니다.

5.4. 업그레이드

마이그레이션은 기동 시 자동 적용됩니다. 그래서 **새 이미지를 올리기 전에 복구 가능한 백업**을 확보하는 것이 가장 중요합니다.

1. 현재 `/api/version`, 이미지 태그, 백업 시각을 기록합니다.
2. 위 5.3의 dump를 뜨고 `pg_restore --list`로 읽히는지 확인합니다.
3. 새 archive의 SHA-256을 대조하고 `docker load` 후 label과 non-root 사용자를 확인합니다.
4. 교체합니다.

```
APPSTORE_VERSION=v2.6.0 docker compose up -d --no-build
docker compose ps
docker compose logs --tail 200 appstore
```

5. `/health/live`, `/health/ready`, `/api/version`이 새 버전인지 확인하고, 익명 탐색 → 로그인 → `/admin/users` 새로고침 → 앱 등록·검토 → 키 발급·회전·폐기 → AI 스트리밍 순으로 실제 화면을 확인합니다.

여러 인스턴스를 운영한다면 한 대만 먼저 올려 마이그레이션을 끝낸 뒤 나머지를 교체합니다. 같은 데이터베이스를 두 버전이 동시에 마이그레이션하지 않게 하려는 것입니다.

5.5. 롤백

자동 down 마이그레이션은 실행되지 않습니다. 새 마이그레이션이 이전 바이너리와 호환되지 않으면 이미지만 되돌려서는 안 되고 백업도 함께 복원해야 합니다.

이전 스키마와 호환되는 경우:

```
docker rm -f appstore
docker rename appstore-v2.5.8-stopped appstore
docker start appstore
curl --fail http://127.0.0.1:8080/health/ready
```

호환되지 않는 경우: 트래픽을 차단하고 → 변경 창 직전 dump를 별도 DB에 복원하고 → 원래 `ENCRYPTION_KEY`와 이전 이미지를 연결하고 → health·로그인·핵심 화면을 확인한 뒤 트래픽을 복구합니다. 롤백 원인과 유실 가능 구간을 운영 기록에 남깁니다. 자세한 절차는 업그레이드·롤백 가이드에 있습니다.

---

6. 장애 대응

증상	확인할 곳	조치
컨테이너가 뜨자마자 죽는다. 로그에 `missing required environment variables: ...`	`docker logs appstore` 첫 줄	목록에 적힌 변수를 env 파일에 채웁니다
로그에 `BOOTSTRAP_ADMIN_PASSWORD must be at least 12 characters`	같은 곳	12자 이상으로 바꾸고 다시 기동합니다
로그에 `initialize database (migrate): ...`	PostgreSQL 도달 가능성, DSN, 계정 권한	DSN의 host·port·sslmode와 스키마 변경 권한을 확인합니다
로그에 `ENCRYPTION_KEY does not match the initialized database`	env 파일의 키	이 DB를 처음 초기화한 키를 써야 합니다. 키를 잃었다면 secret 재입력 계획을 세운 뒤 새 DB로 시작합니다
`/health/live`는 200인데 `/health/ready`가 503 `{"status":"not_ready"}`	PostgreSQL, 네트워크	DB가 죽었거나 연결이 끊겼습니다. 2초 안에 ping이 되어야 합니다
사용자에게 500과 요청 ID가 보인다	`docker logs`에서 `request_id`로 검색	`panic recovered`면 `stack`을, 아니면 해당 `http request`의 `status`와 `path`를 봅니다
SSO 로그인이 안 된다	관리자 → 인증·SSO → **연결 테스트**	3.3의 실패 메시지 표를 따릅니다
로그인은 되는데 화면이 403	사용자 → 역할, 인증·SSO → Role Mapping	외부 역할 값과 Role Claim Path가 실제 token과 맞는지 확인합니다
익명 사용자가 아무 화면도 못 본다	시스템 설정 → 공개 모드	꺼져 있으면 익명 탐색이 차단됩니다
API 호출이 `자동화 API가 비활성화되어 있습니다.`	관리자 → REST API	API 사용을 켭니다
API·MCP 호출이 429 `요청이 너무 많습니다.`	REST API / MCP의 분당 요청 제한	한도를 올리거나 클라이언트가 `Retry-After` 초만큼 기다리게 합니다. reverse proxy 뒤에서는 제한이 프록시 IP 기준으로 잡힙니다
로고 업로드가 `이미지는 1MB 이하여야 합니다.`	파일 크기	1MB 이하로 줄입니다. 서버는 본문을 다 받기 전에 거절합니다
로고 주소 가져오기가 `이미지 주소가 HTTP ...를 반환했습니다.`	그 주소가 컨테이너에서 열리는지	폐쇄망에서 도달 가능한 주소인지, 인증이 필요한 주소는 아닌지 확인합니다

증상	확인할 곳	조치
검토 대기 목록이 비어 있다	관리자 → 승인 워크플로	*Workflow가 활성화된 경우에만 등록 건이 이 목록에 나타납니다.*
팀장이 `팀장 검토에는 사용자 팀 정보가 필요합니다.` 를 받는다	사용자 → 담당팀	SSO에서 팀 정보가 오지 않았습니다. 사용자 정보의 팀을 채웁니다
AI 응답이 `사용 가능한 AI Provider 가 없습니다.`	관리자 → AI 공급자	Provider와 모델을 등록하고 활성화합니다
추적을 켰는데 수집이 들어오지 않는다	관리자 → 방문 추적 → **정책이 차단한 출처**	출처가 보이면 **허용에 추가**합니다. 비어 있으면 브라우저 콘솔의 `Refused to load` 줄과 스니펫의 `〈script〉` 에 `nonce` 가 붙었는지 확인합니다. Momento는 같은 오리진 프록시를 켜는 편이 가장 확실합니다
`/momento/*` 가 404	관리자 → 방문 추적	추적 사용·provider Momento·같은 오리진 프록시가 모두 켜져 있어야 열립니다
`/momento/*` 가 502	`docker logs` 에서 `momento proxy failed`	컨테이너가 Momento 수집기 주소에 15초 안에 닿아야 합니다

---

7. 보안

7.1. 기본값 중 바꿔야 하는 것

항목	기본값	운영에서
`BOOTSTRAP_ADMIN_PASSWORD`	`.env.example` 의 placeholder	길고 무작위한 값으로. vault에 보관하고 일상 계정으로 공유하지 않습니다
OIDC	꺼짐	조직 SSO를 붙이고 SSO Super Admin을 따로 지정합니다
서비스 접속 URL	빈 값	실제 HTTPS origin으로 채웁니다. redirect URL이 여기서 만들어집니다
API·MCP 익명 조회	켬	조직 정책에 따라 끄거나 최소 조회로 둡니다
공개 모드	켬	카탈로그를 사내에만 보여야 한다면 끕니다
API 120 / MCP 60 req·min	그대로	실제 사용량에 맞춰 조정합니다
방문 추적	꺼짐	켜야 한다면 Momento + 같은 오리진 프록시로. 외부 provider는 방문 데이터가 밖으로 나갑니다

7.2. 노출 범위

컨테이너는 평문 HTTP 8080만 listen합니다. **8080을 외부에 직접 열지 말고** reverse proxy가 `127.0.0.1:8080`에 붙게 하세요(`-p 127.0.0.1:8080:8080`). PostgreSQL은 서비스 네트워크 안에만 두고 외부에 열지 않습니다.

서비스가 스스로 붙이는 응답 헤더: `X-Content-Type-Options: nosniff`, `X-Frame-Options: DENY`, `Referrer-Policy: strict-origin-when-cross-origin`, `Permissions-Policy`(camera·microphone·geolocation·payment 모두 차단), `Cross-Origin-Opener-Policy: same-origin`, 그리고 `default-src 'self'` 기반 CSP(`frame-ancestors 'none'`, `object-src 'none'`, `script-src 'self'`). 방문 추적을 켜면 페이지 응답의 `script-src`에 요청마다 다른 `'nonce-…'`와 provider 출처가 더해지고(4.9), `/api/*`·`/mcp`·`/momento/*`는 `default-src 'none'`을 받습니다. reverse proxy에서 이 헤더를 약하게 덮어쓰거나 캐시하지 마세요 — nonce는 응답마다 달라야 합니다.

7.3. 비밀값 취급

- OIDC Client Secret과 AI API Key는 `ENCRYPTION_KEY`로 인증 암호화되어 DB에 저장되고, **조회 API가 없습니다.** 변경만 가능합니다.
- 개인 API·MCP 키는 prefix와 keyed hash만 저장됩니다. 원문은 생성 순간 한 번만 표시되고 서버도 복원할 수 없습니다.
- `.env` 파일, DSN, 비밀번호, API Key를 커밋하거나 이미지 build argument로 넘기지 마세요.
- 로고·파비콘·배너 문구는 로그인 전 화면에도 보입니다. 비공개 정보를 넣지 마세요.

7.4. 컨테이너

이미지는 non-root(`appstore:appstore`, uid/gid 10001)로 실행하고, 배포는 `--read-only --cap-drop ALL --security-opt no-new-privileges:true`로 굳힙니다. 쓰기가 필요한 곳은 64MB tmpfs `/tmp` 하나뿐입니다.

7.5. 운영 전 점검

- [ ] Bootstrap 관리자와 SSO Super Admin이 서로 다른 승인된 계정이다.
- [ ] OIDC·AI secret이 재조회되지 않고 변경만 가능하다.
- [ ] 공개 API와 익명 MCP가 조직 정책대로 설정돼 있다.
- [ ] 서비스 URL, OIDC redirect와 logout URL이 운영 HTTPS origin만 허용한다.
- [ ] PostgreSQL 백업과 `ENCRYPTION_KEY` 복구 절차를 각각 시험했다.
- [ ] 관리자·일반 사용자·익명 세션으로 200 / 401 / 403 경계를 확인했다.
- [ ] 감사 로그에 로그인과 설정 변경이 남는 것을 확인했다.

더 읽을 것

- 사용자 가이드 — 화면 사용법, 자주 하는 작업, 오류 메시지
- 오프라인 설치 · 업그레이드와 롤백 · 백업과 복구 · 릴리스 Runbook
- 서비스의 `/docs`와 `/openapi.json`에 REST API 계약이 있습니다.