

Oliver DeckForge — System Administration Guide

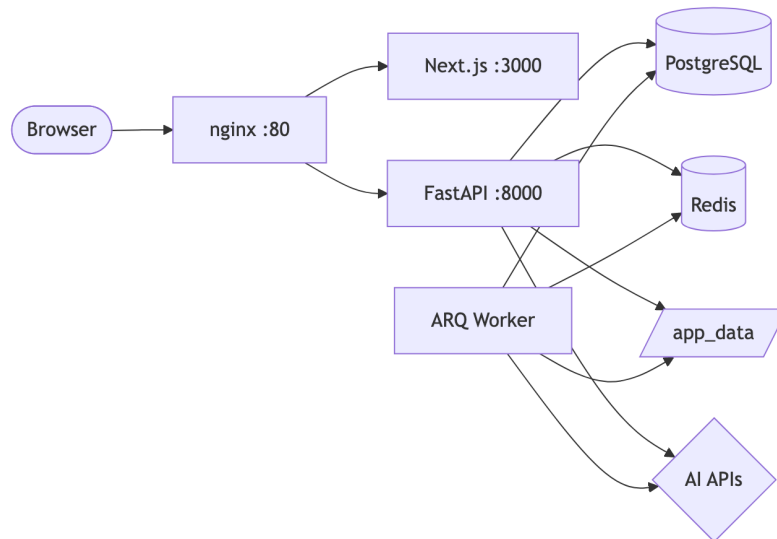
Version 1.0 | Complete Setup, Configuration & Operations Reference

Table of Contents

1. [Architecture Overview](#)
 2. [Installation & Deployment](#)
 3. [Environment Variables Reference](#)
 4. [Authentication Configuration](#)
 5. [Role-Based Access Control](#)
 6. [Admin Panel Operations](#)
 7. [Template Pipeline](#)
 8. [AI Provider Configuration](#)
 9. [Database Administration](#)
 10. [Background Jobs & Workers](#)
 11. [Nginx & Networking](#)
 12. [Storage & File Management](#)
 13. [Monitoring & Logging](#)
 14. [Backup & Recovery](#)
 15. [Security Hardening](#)
 16. [Scaling & Performance](#)
 17. [Troubleshooting](#)
 18. [API Reference](#)
-

1. Architecture Overview

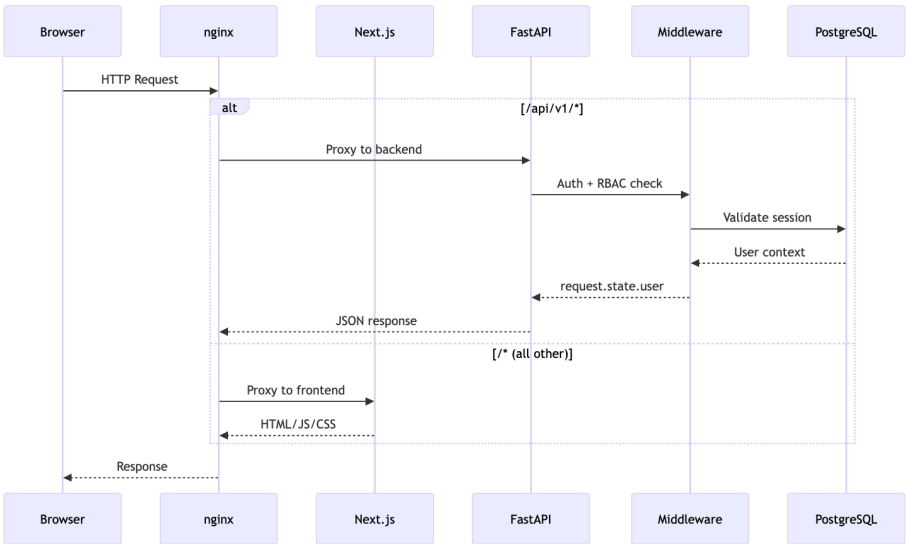
1.1 System Architecture



1.2 Service Overview

Service	Technology	Port	Purpose
nginx	nginx:alpine	80	Reverse proxy, static file serving, SSL termination
web	Next.js 14	3000	Frontend SPA, Puppeteer-based export
api	FastAPI + SQLAlchemy	8000	REST API, authentication, RBAC
worker	ARQ (Python)	—	Background AI generation, parsing, retention
postgres	PostgreSQL 16	5432	Primary relational database
redis	Redis 7	6379	Job queue, caching

1.3 Request Flow



1.4 Data Flow: Presentation Generation



2. Installation & Deployment

2.1 Prerequisites

Requirement	Minimum	Recommended
Docker	20.10+	24.0+
Docker Compose	v2.0+	v2.20+
RAM	4 GB	8 GB+
Disk	10 GB	50 GB+ (for generated assets)
CPU	2 cores	4+ cores

2.2 Quick Start

```
# 1. Clone repository
git clone <repository-url>
cd ppt-tool
```

```
# 2. Configure environment
cp .env.example .env
# Edit .env – set ANTHROPIC_API_KEY at minimum

# 3. Build and start
make dev

# 4. Run migrations
make migrate

# 5. Seed default data
make seed
```

The application is available at:

- <http://localhost> — Full application (via nginx)
- <http://localhost:3000> — Frontend directly
- <http://localhost:8000> — API directly
- <http://localhost/docs> — Swagger API documentation

2.3 Makefile Commands

Command	Description
<code>make dev</code>	Build and start all services with logs
<code>make build</code>	Build Docker images only
<code>make up</code>	Start services in background (detached)
<code>make down</code>	Stop and remove all containers
<code>make migrate</code>	Run Alembic database migrations
<code>make seed</code>	Seed default admin user and team
<code>make test</code>	Run backend pytest suite
<code>make test-e2e</code>	Run Cypress E2E tests
<code>make test-all</code>	Run all tests
<code>make logs</code>	Follow all container logs

<code>make shell-api</code>	Open bash shell in API container
<code>make shell-db</code>	Open psql shell in PostgreSQL

2.4 Local Development (Without Docker)

Backend

```
cd backend
python -m venv venv && source venv/bin/activate
pip install -r requirements.txt

export DATABASE_URL="postgresql+asyncpg://deckforge:deckforge@localhost:5432/deckforge"
export REDIS_URL="redis://localhost:6379/0"
export APP_DATA_DIRECTORY="./data"
export ANTHROPIC_API_KEY="sk-ant-..."

# Start API
uvicorn api.main:app --reload --port 8000

# Start worker (separate terminal)
python -m arq workers.main.WorkerSettings
```

Frontend

```
cd frontend
npm install
npm run dev
```

The Next.js dev server proxies `/api/v1/` requests to `http://localhost:8000` via rewrites in `next.config.mjs`.

2.5 Docker Image Details

Backend Dockerfile (multi-stage):

- **Builder stage:** `python:3.11-slim-bookworm` + `uv` package manager

- **Runtime stage:** Includes LibreOffice (PDF conversion), Chromium (browser automation), fontconfig, ONNX models
- Exposes port 8000

Frontend Dockerfile:

- **Base:** `node:20-alpine`
- Includes Chromium for server-side Puppeteer PDF/PPTX export
- Environment: `PUPPETEER_EXECUTABLE_PATH=/usr/bin/chromium-browser`
- Exposes port 3000

3. Environment Variables Reference

3.1 Database & Infrastructure

Variable	Required	Default	Description
<code>POSTGRES_PASSWORD</code>	No	<code>deckforge</code>	PostgreSQL password
<code>DATABASE_URL</code>	Auto	Set by docker-compose	Full async connection string
<code>REDIS_URL</code>	No	<code>redis://redis:6379/0</code>	Redis connection string
<code>APP_DATA_DIRECTORY</code>	No	<code>/app_data</code>	Path for images, exports, uploads
<code>TEMP_DIRECTORY</code>	No	<code>/tmp/deckforge</code>	Temporary file storage

3.2 Authentication

Variable	Required	Default
<code>JWT_SECRET_KEY</code>	Yes	<code>change-me-...</code>

<code>AZURE_AD_TENANT_ID</code>	No	—
<code>AZURE_AD_CLIENT_ID</code>	No	—
<code>AZURE_AD_CLIENT_SECRET</code>	No	—
<code>AZURE_AD_REDIRECT_URI</code>	No	<code>http://localhost/api/v1/auth/callback</code>
<code>DEV_AUTH_PASSWORD</code>	No	<code>devpass123</code>

3.3 AI Providers

Variable	Required	Default	Description
<code>LLM</code>	No	<code>anthropic</code>	Primary LLM provider
<code>ANTHROPIC_API_KEY</code>	Yes*	—	Claude API key
<code>ANTHROPIC_MODEL</code>	No	<code>claude-sonnet-4-6</code>	Claude model ID
<code>OPENAI_API_KEY</code>	No	—	OpenAI API key
<code>OPENAI_MODEL</code>	No	—	OpenAI model ID
<code>GOOGLE_API_KEY</code>	No	—	Google Gemini API key
<code>GOOGLE_MODEL</code>	No	—	Google model ID

<code>OLLAMA_URL</code>	No	—	Ollama server URL
<code>OLLAMA_MODEL</code>	No	—	Ollama model name
<code>IMAGE_PROVIDER</code>	No	<code>nanobanana_pro</code>	Image generation provider
<code>DISABLE_IMAGE_GENERATION</code>	No	—	Set to disable image gen

*Required if using Anthropic (default provider)

3.4 Application Settings

Variable	Required	Default	Description
<code>CAN_CHANGE_KEYS</code>	No	<code>false</code>	Allow runtime API key changes
<code>DISABLE_ANONYMOUS_TRACKING</code>	No	<code>true</code>	Disable analytics tracking
<code>SETTINGS_ENCRYPTION_KEY</code>	No	—	Fernet key for encrypting stored API keys
<code>EXTENDED_REASONING</code>	No	—	Enable LLM extended thinking
<code>TOOL_CALLS</code>	No	—	Enable LLM tool use
<code>WEB_GROUNDING</code>	No	—	Enable web search in generation

<code>NEXT_INTERNAL_URL</code>	No	<code>http://web:3000</code>	Backend → frontend URL (Docker)
--------------------------------	----	------------------------------	---------------------------------------

3.5 Supported AI Providers

LLM Providers:

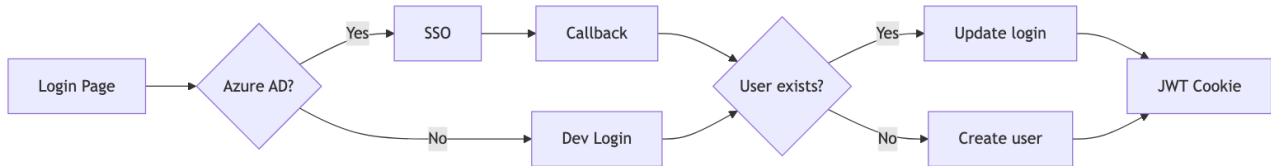
Provider	Value	Models
Anthropic	<code>anthropic</code>	claude-opus-4-6, claude-sonnet-4-6, claude-sonnet-4-5, claude-haiku-4-5
OpenAI	<code>openai</code>	gpt-4.1, gpt-4.1-mini, gpt-4o, o3, o4-mini
Google	<code>google</code>	gemini-2.5-flash, gemini-2.5-pro, gemini-2.0-flash
Ollama	<code>ollama</code>	Any locally installed model
Custom	<code>custom</code>	Any OpenAI-compatible endpoint

Image Providers:

Provider	Value	Requirements
NanoBanana Pro	<code>nanobanana_pro</code>	Google API key
Gemini Flash	<code>gemini_flash</code>	Google API key
DALL-E 3	<code>dall-e-3</code>	OpenAI API key
GPT Image 1.5	<code>gpt-image-1.5</code>	OpenAI API key
Pexels	<code>pexels</code>	Pexels API key
Pixabay	<code>pixabay</code>	Pixabay API key
ComfyUI	<code>comfyui</code>	Local ComfyUI instance

4. Authentication Configuration

4.1 Authentication Flow



4.2 Azure AD Setup (Production)

1. **Register an application** in Azure Portal > Azure AD > App Registrations
2. Set the **Redirect URI** to: `https://your-domain.com/api/v1/auth/callback`
3. Create a **client secret** under Certificates & Secrets
4. Configure environment variables:

```

AZURE_AD_TENANT_ID=your-tenant-id
AZURE_AD_CLIENT_ID=your-client-id
AZURE_AD_CLIENT_SECRET=your-client-secret
AZURE_AD_REDIRECT_URI=https://your-domain.com/api/v1/auth/callback
  
```

5. Grant API permissions: `User.Read` (delegated)

4.3 Development Mode

When `AZURE_AD_TENANT_ID` is empty or not set, the system enables development authentication:

- Login form with email + password fields
- Password validated against `DEV_AUTH_PASSWORD` environment variable
- Users are auto-created on first login
- Default role: `user`

Security Warning: Development mode should never be used in production. Always configure Azure AD or another SSO provider for production deployments.

4.4 JWT Configuration

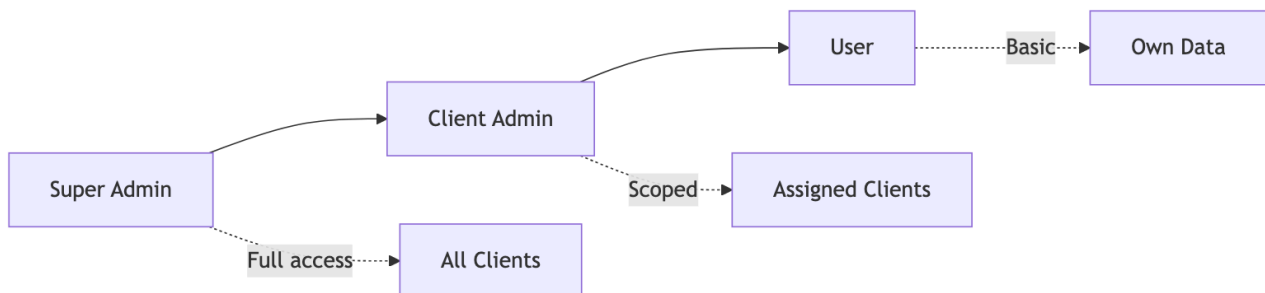
Parameter	Value
Algorithm	HS256

Expiry	24 hours
Storage	<code>session_token</code> HTTP cookie
Payload	<code>sub</code> (user UUID), <code>email</code> , <code>role</code> , <code>exp</code> , <code>iat</code>

Critical: Change `JWT_SECRET_KEY` from the default value in production. Use a cryptographically random 256-bit key. All active sessions are invalidated when this key changes.

5. Role-Based Access Control

5.1 Role Hierarchy

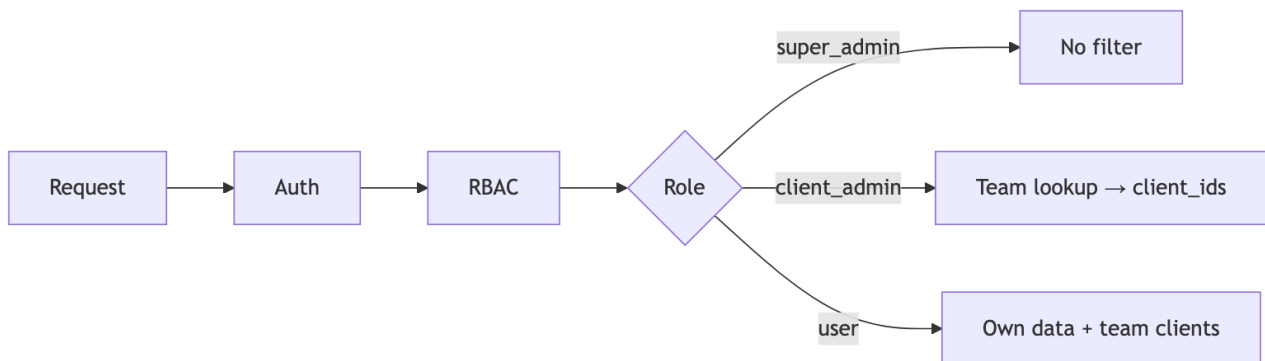


5.2 Permission Matrix

Feature	Super Admin	Client Admin	User
Presentations	All	Client-scoped	Own only
Admin Panel	Full	Limited	None
User Management	CRUD all	View team members	—
Client Management	CRUD all	View/edit assigned	—
Team Management	All teams	Assigned client teams	—
Master Decks	All	Client-scoped	—
Storage	All clients	Client-scoped	—

Analytics	Global + per-client	Client-scoped	—
Audit Logs	All	Client-scoped	—
System Settings	Full access	—	—
Brand Config	All clients	Client-scoped	—

5.3 Multi-Tenant Data Isolation



The `_resolve_client_filter()` pattern is used throughout:

- **Super Admin with no client_id param** → Returns `None` (no filter, see all)
- **Super Admin with client_id param** → Filters to specific client
- **Client Admin** → Auto-scoped to accessible clients via `TeamMembershipModel`
- **User** → Scoped to own data within accessible clients

5.4 Admin Panel Navigation

The sidebar dynamically shows menu items based on role:

Menu Item	Super Admin	Client Admin
Users	Yes	—
Clients	Yes	Yes
Storage	Yes	Yes
Audit Log	Yes	Yes
Analytics	Yes	Yes
Settings	Yes	—

6. Admin Panel Operations

6.1 User Management

Path: Admin > Users (Super Admin only)

Listing Users

- Table with columns: Name, Email, Role, Status, Last Login
- Filterable by active status and role

Changing User Roles

1. Find the user in the list
2. Click the role dropdown
3. Select new role: `super_admin`, `client_admin`, or `user`

You cannot change your own role. This prevents accidental logout.

Deactivating Users

1. Click **Deactivate** on the user row
2. Confirm the action
3. User's `is_active` is set to false — they can no longer log in
4. Their presentations remain in the system

Transferring Ownership

Before deactivating a user (e.g., for GDPR compliance):

1. Use the transfer ownership endpoint to move all presentations to another user
2. Then deactivate the original user

6.2 Client Management

Path: Admin > Clients

Creating a Client

1. Click "+ New Client"
2. Enter the client name
3. A URL-safe slug is auto-generated
4. A default team is auto-created for the client

Client Settings

Each client has configurable:

Setting	Description
Name	Display name
Slug	URL-safe identifier (unique)
Review Policy	<code>self_approve</code> or <code>require_reviewer</code>
Retention Days	Auto-delete presentations after N days (optional)

6.3 Team Management

Path: Admin > Clients > [Client] > Teams

Teams group users within a client:

- Each client has a **default team** (cannot be deleted)
- Users can belong to multiple teams
- Team membership determines client access for non-admin users

Adding Team Members

1. Navigate to the client's team page
2. Click "+ Add Member"
3. Search and select a user from the dropdown
4. The user now has access to this client's data

6.4 Brand Configuration

Path: Admin > Clients > [Client] > Brand Config

Configure branding per client:

Setting	Description
Primary Colors	Color picker, add/remove multiple colors
Secondary Colors	Color picker, add/remove multiple colors
Fonts	Heading, Body, and Accent font names
Logos	Upload multiple logo images
Voice Rules	Text guidelines for AI tone and style
Voice Examples	Good/bad example pairs for AI training
Brand Guideline	Upload a PDF/DOCX brand guide

Brand configuration is injected into AI prompts during presentation generation to ensure brand consistency.

6.5 Storage Management

Path: Admin > Storage

Summary Dashboard

Four cards show:

- **Presentations** count
- **Export Files** count
- **Master Decks** count
- **Total Size** (formatted)

Client Selector (Super Admin)

Dropdown to filter by specific client or view all clients combined.

Presentation Table

- Columns: Title, Status, Created, Files, Size
- Checkboxes for bulk selection
- Per-row actions: Download PPTX, Delete

Bulk Operations

- Select multiple presentations via checkboxes
- Click "**Delete Selected**" for bulk soft-delete

Purge Files (Super Admin Only)

- Amber banner shows count of soft-deleted presentations
- "**Purge Files**" permanently removes files from disk
- Returns statistics: files purged, bytes freed

6.6 Analytics Dashboard

Path: Admin > Analytics

Overview Metrics

- Total Presentations (all-time)
- This Month / This Week (30/7-day counts)
- Active Users (distinct users, last 30 days)
- Approval Rate (% approved or in_review)

Usage Metrics

- **Presentations per Day** — 14-day bar chart
- **Top 10 Users** — ranked by presentation count

Quality Metrics

- **Status Distribution** — draft/in_review/approved breakdown
- **Presentations with Comments** — count

Performance Metrics

- **Average Generation Time** — job completion duration
- **Total Jobs** — all-time count
- **Error Rate** — % failed jobs

AI Usage (if tracking enabled)

- Total AI Calls, Input/Output Tokens

- Usage by Provider (bar chart)
- Usage by Model (top 10)
- Daily Usage Trend

6.7 Audit Logs

Path: Admin > Audit Log

All mutating API requests are logged automatically.

Query Filters

- **Action** — text search (e.g., "admin_delete")
- **User ID** — filter by specific user
- **Resource Type** — e.g., "presentation", "storage"
- **Client ID** — scope to a client
- **Date Range** — from/to date pickers

Export

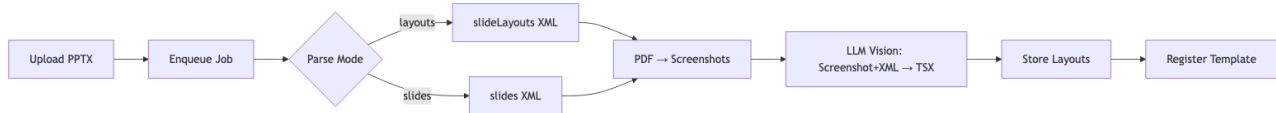
- Click "**Export Audit Log**"
- Choose format: CSV or JSON
- Downloads up to 10,000 entries

Logged Actions

Action	Trigger
<code>admin_delete</code>	Single presentation soft-delete
<code>admin_bulk_delete</code>	Bulk presentation delete
<code>admin_purge</code>	Hard-delete purged files
Role changes, team membership updates, etc.	Various admin operations

7. Template Pipeline

7.1 Master Deck → Template Flow



7.2 Upload & Parse

1. Navigate to **Admin > Clients > [Client] > Master Decks**
2. Click **"Upload PPTX"** and select a `.pptx` file
3. The deck enters **"pending"** status, then **"processing"**
4. Auto-polling every 5 seconds shows current status
5. On completion, status changes to **"completed"** and layouts appear

7.3 Parse Modes

Mode	Source	Best For
slides (default)	Actual slides (<code>ppt/slides/</code>)	Decks with unique slide designs; 1:1 screenshot match
layouts	Slide layouts (<code>ppt/slideLayouts/</code>)	Decks with reusable layout templates; may produce more layouts

7.4 Layout Management

After parsing, manage layouts in the expanded deck view:

Filtering & Search:

- Text search by layout name
- Type filter dropdown (auto-detected from layout types)
- Code filter: All / Has Code / Missing Code

Individual Actions:

- **Edit** — modify name, type, or React TSX code
- **Delete** — remove with confirmation dialog

Bulk Actions:

- Toggle **Select Mode** to show checkboxes

- **Select All / Deselect All**
- **Delete Selected** — bulk remove

After deleting layouts, the system automatically re-registers the template by recreating `PresentationLayoutCodeModel` records for the remaining layouts.

7.5 Reparsing

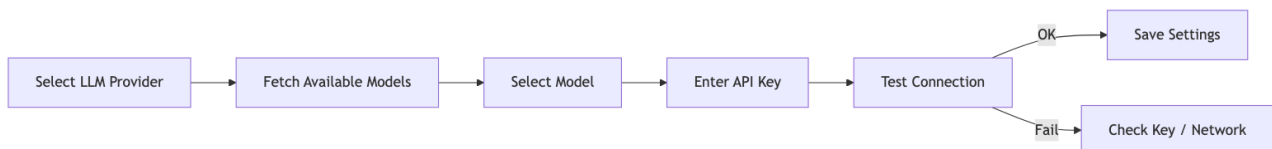
If layouts need to be regenerated (e.g., after an LLM model upgrade):

1. Click the reparse dropdown on the deck card
2. Choose "**Reparse (slides)**" or "**Reparse (layouts)**"
3. All existing layouts are replaced with freshly parsed versions

8. AI Provider Configuration

8.1 Settings Page

Path: Admin > Settings (Super Admin only)



8.2 Configuring LLM Provider

1. Open **Admin > Settings**
2. Select the **LLM Provider** from the dropdown
3. The **Model** dropdown auto-populates with available models
4. Enter the **API Key** if not already set (shown as "Set" badge if configured)
5. Click "**Test**" to verify connectivity
6. Click "**Save Changes**"

8.3 Configuring Image Provider

1. Select the **Image Provider** from the dropdown

2. Ensure the required API key is set (e.g., Google API key for NanoBanana Pro)
3. Save changes

8.4 Connection Testing

The **"Test"** button performs a lightweight API call to validate the key:

Result	Display
Success	Green check + latency in ms
Failure	Red X + error message

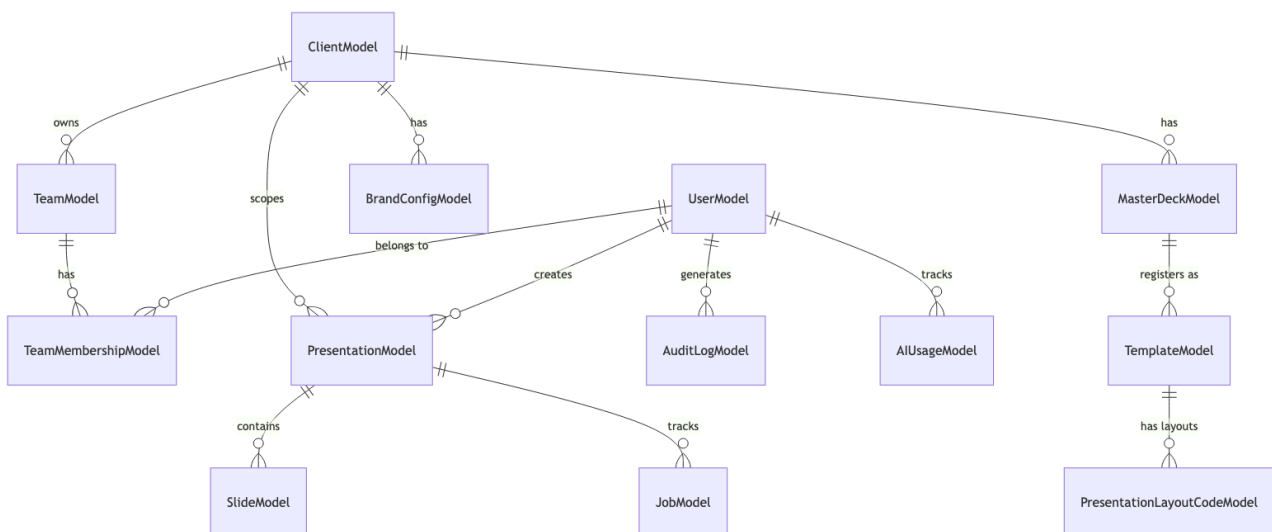
8.5 Settings Persistence

Settings are persisted to the database via `KeyValueSqlModel` :

- Survive container restarts
- API keys optionally encrypted at rest (if `SETTINGS_ENCRYPTION_KEY` is set)
- Environment variables serve as defaults — database values override them

9. Database Administration

9.1 Schema Overview



9.2 Core Tables

Table	Purpose	Key Fields
<code>usermodel</code>	User accounts	id, email, role, azure_oid, is_active
<code>clientmodel</code>	Tenant organizations	id, name, slug, retention_days, review_policy
<code>teammodel</code>	Team groupings	id, name, client_id, is_default
<code>teammembershipmodel</code>	User↔Team links	user_id, team_id, assigned_by
<code>presentationmodel</code>	Presentations	id, title, owner_id, client_id, status, content
<code>slidemodel</code>	Individual slides	id, presentation, index, content, layout
<code>jobmodel</code>	Background jobs	id, job_type, status, progress, error_message
<code>masterdeck</code>	Master PPTX decks	id, client_id, layouts (JSON), parse_status
<code>templatemodel</code>	Registered templates	id, name, description
<code>presentationlayoutcodemodel</code>	Layout TSX code	presentation, layout_name, layout_code
<code>brandconfigmodel</code>	Brand settings	client_id, colors, fonts, logos, voice_rules
<code>auditlogmodel</code>	Audit trail	user_id, action, resource_type, ip_address
<code>aiusagemodel</code>	AI usage metrics	provider, model, tokens, duration_ms
<code>keyvaluesqlmodel</code>	KV settings store	key, value (JSON)

imageasset	Generated images	id, path, is_uploaded
------------	------------------	-----------------------

9.3 Alembic Migrations

```
# View migration history
docker compose exec api alembic history

# Apply all pending migrations
make migrate
# or: docker compose exec api alembic upgrade head

# Generate new migration after model changes
docker compose exec api alembic revision --autogenerate -m "description"

# Rollback last migration
docker compose exec api alembic downgrade -1

# View current revision
docker compose exec api alembic current
```

Always review auto-generated migrations before applying. SQLAlchemy may miss rename operations (interpreting them as drop + create) or produce incorrect defaults.

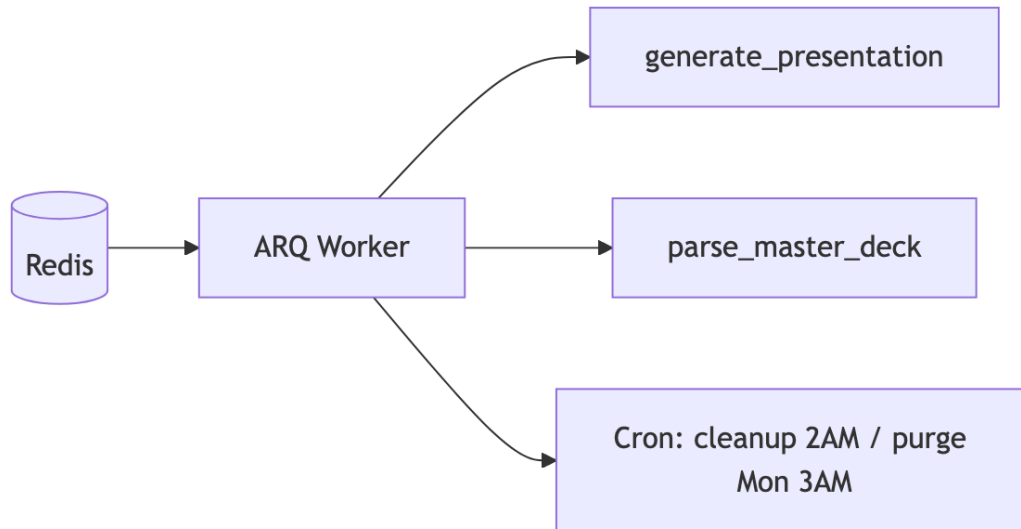
9.4 Direct Database Access

```
# Interactive psql
make shell-db

# Common queries
SELECT COUNT(*) FROM usermodel;
SELECT COUNT(*) FROM presentationmodel WHERE deleted_at IS NULL;
SELECT status, COUNT(*) FROM jobmodel GROUP BY status;
SELECT provider, SUM(total_tokens) FROM aiusagemodel GROUP BY provider;
```

10. Background Jobs & Workers

10.1 ARQ Worker Configuration



Setting	Value	Description
<code>max_jobs</code>	5	Maximum concurrent background jobs
<code>job_timeout</code>	1800s (30 min)	Per-job timeout
<code>max_tries</code>	3	Retry attempts on failure
<code>health_check_interval</code>	30s	Health check frequency

10.2 Job Types

Presentation Generation (`generate_presentation_task`)

1. Load request from PresentationModel
2. Fetch brand context (colors, fonts, voice rules)
3. Generate outlines via LLM
4. Generate per-slide structure and content
5. Run image generation for each slide
6. Save results to database
7. Update JobModel progress (0–100%)

Master Deck Parsing (`parse_master_deck_task`)

1. Extract XML layouts/slides from PPTX
2. Convert PPTX to PDF via LibreOffice
3. Split PDF into per-page screenshots
4. Send each screenshot + XML to LLM vision
5. Store generated React TSX code
6. Register as template

Retention Cleanup (Cron — Daily)

- Soft-deletes presentations exceeding client's `retention_days`
- Runs at 2:00 AM UTC

Retention Purge (Cron — Weekly)

- Permanently deletes files for presentations soft-deleted 30+ days ago
- Runs Monday 3:00 AM UTC

10.3 Monitoring Jobs

```
# View worker logs
docker compose logs -f worker

# Check job status in database
make shell-db
# Then: SELECT id, job_type, status, progress, error_message FROM jobmode`
```

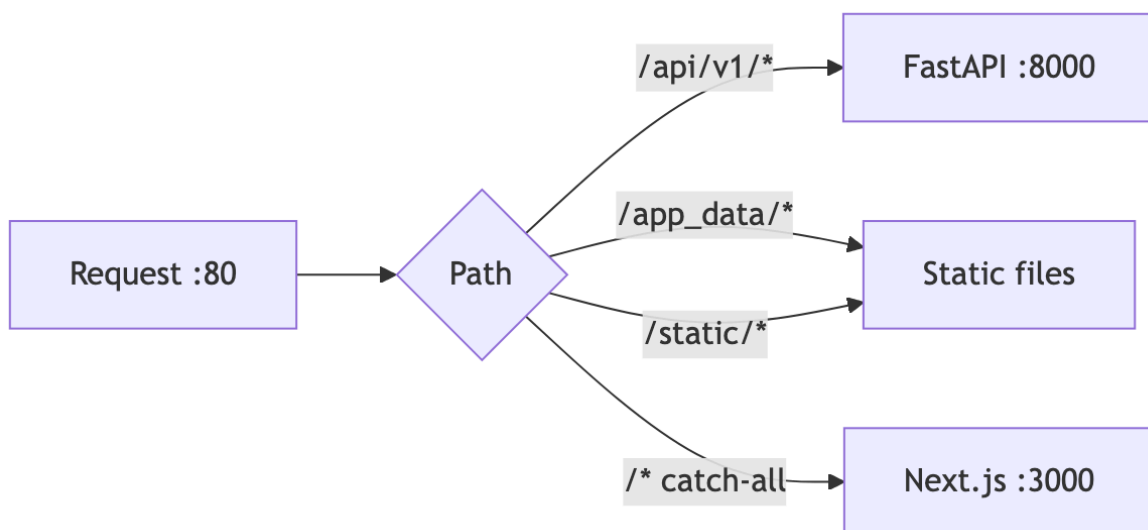
10.4 Common Job Issues

Issue	Cause	Solution
Job stuck at 0%	Worker crashed or no workers running	Restart worker: <code>docker compose restart worker</code>
Job times out	LLM response too slow	Increase <code>job_timeout</code> in WorkerSettings

Job fails repeatedly	Invalid API key or model	Check Settings page, test connection
Queue backed up	Too many concurrent requests	Scale workers horizontally

11. Nginx & Networking

11.1 Routing Rules



11.2 Key Configuration

Setting	Value	Purpose
<code>client_max_body_size</code>	100M	Allow large PPTX uploads
<code>proxy_read_timeout</code>	30m	Long-running LLM operations
<code>proxy_connect_timeout</code>	30m	Connection establishment
<code>proxy_buffering</code>	off	SSE streaming support
<code>chunked_transfer_encoding</code>	off	SSE streaming support

11.3 SSL/TLS (Production)

The default `nginx.conf` serves HTTP only. For production, add SSL:

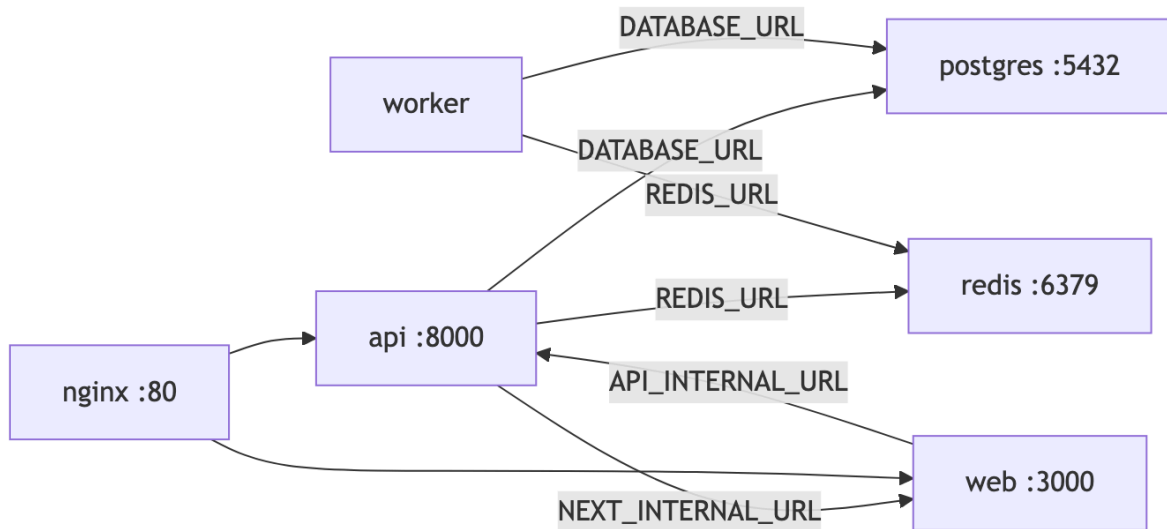
```

server {
    listen 443 ssl http2;
    ssl_certificate /etc/ssl/certs/your-cert.pem;
    ssl_certificate_key /etc/ssl/private/your-key.pem;
    # ... existing location blocks ...
}

server {
    listen 80;
    return 301 https://$host$request_uri;
}

```

11.4 Inter-Service Communication



All services communicate via Docker Compose's internal network. No ports need to be exposed to the host except:

- **80** (nginx) — user access
- **5432** (postgres) — optional, for direct DB access
- **6379** (redis) — optional, for debugging

12. Storage & File Management

12.1 Directory Structure

```

/app_data/
├── images/           # AI-generated images (UUID-named PNG files)
├── exports/          # Generated PPTX/PDF export files
├── uploads/          # User-uploaded documents (DOCX, PDF, TXT)
├── fonts/            # Custom font files
├── master_decks/     # Master deck PPTX files and screenshots
│   └── {deck_id}/
│       ├── original.pptx
│       ├── screenshots/
│       │   ├── page_1.png
│       │   ├── page_2.png
│       │   └── ...
│       └── pdf/
│           └── deck.pdf

```

12.2 File Serving

Context	Serving Method
Docker (production)	nginx serves <code>/app_data/</code> directly from volume
Local development	FastAPI <code>StaticFiles</code> mount on <code>/app_data</code>
Frontend access	Next.js rewrites <code>/app_data/*</code> to backend

12.3 Retention Policy



- **Retention days** configured per client in `ClientModel.retention_days`
- **Soft delete** runs daily at 2:00 AM UTC (sets `deleted_at` timestamp)
- **Hard purge** runs weekly Monday at 3:00 AM UTC (removes files for items soft-deleted 30+ days ago)
- **Manual purge** available via Admin > Storage > "Purge Files" button

12.4 Disk Space Monitoring

Monitor the `app_data` volume:

```
# Check volume usage
docker compose exec api du -sh /app_data/*

# Check available disk space
docker compose exec api df -h /app_data
```

13. Monitoring & Logging

13.1 Log Access

```
# All services
make logs

# Specific service
docker compose logs -f api
docker compose logs -f worker
docker compose logs -f postgres
docker compose logs -f web

# Last N lines
docker compose logs --tail=100 api
```

13.2 Audit Trail

The `AuditMiddleware` automatically logs all mutating API requests:

Field	Content
<code>user_id</code>	Authenticated user's UUID
<code>action</code>	Operation name (e.g., "admin_delete")
<code>resource_type</code>	Entity type (e.g., "presentation")
<code>resource_id</code>	Entity UUID
<code>client_id</code>	Tenant context

<code>details</code>	JSON with request/response metadata
<code>ip_address</code>	Client IP address
<code>created_at</code>	Timestamp (indexed for fast queries)

Audit logging is fire-and-forget (non-blocking) via `asyncio.create_task()`.

13.3 AI Usage Tracking

The `AIUsageModel` tracks all LLM API calls:

Metric	Description
Provider	anthropic, openai, google, ollama
Model	Specific model ID
Call Type	outline, content, vision, etc.
Input Tokens	Tokens sent to the model
Output Tokens	Tokens received
Duration (ms)	Call latency
Error Details	Error message if failed

View aggregated metrics at **Admin > Analytics > AI Usage**.

13.4 Health Checks

Service	Method	Interval	Timeout
PostgreSQL	<code>pg_isready</code>	5s	5s, 5 retries
Redis	<code>redis-cli ping</code>	5s	5s, 5 retries
API	Lifespan startup	At boot	—
Worker	ARQ health check	30s	—

14. Backup & Recovery

14.1 Backup Components

Component	Location	Strategy
Database	<code>postgres_data</code> volume	<code>pg_dump</code> to file
Redis	<code>redis_data</code> volume	Optional (transient queue data)
Files	<code>app_data</code> volume	File-level backup or snapshot
Settings	Database (KeyValueSqlModel)	Included in <code>pg_dump</code>
Migrations	<code>backend/migrations/</code>	In git repository

14.2 Database Backup

```
# Full database dump
docker compose exec postgres pg_dump -U deckforge deckforge > backup_$(date +%Y%m%d).sql

# Compressed backup
docker compose exec postgres pg_dump -U deckforge deckforge | gzip > backup_$(date +%Y%m%d).sql.gz

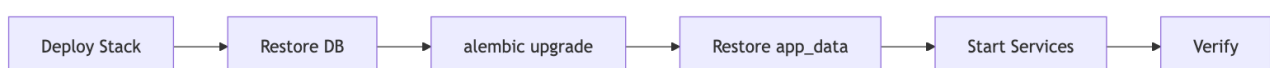
# Restore from backup
docker compose exec -T postgres psql -U deckforge deckforge < backup.sql
```

14.3 File Backup

```
# Backup app_data volume
docker run --rm -v ppt-tool_app_data:/data -v $(pwd):/backup alpine \
  tar czf /backup/app_data_$(date +%Y%m%d).tar.gz -C /data .

# Restore app_data
docker run --rm -v ppt-tool_app_data:/data -v $(pwd):/backup alpine \
  tar xzf /backup/app_data_YYYYMMDD.tar.gz -C /data
```

14.4 Disaster Recovery



1. Deploy fresh Docker Compose stack
2. Start PostgreSQL and Redis first
3. Restore database from latest `pg_dump`
4. Run `alembic upgrade head` to ensure schema is current
5. Restore `app_data` volume from backup
6. Start remaining services
7. Verify data integrity via Admin Panel

15. Security Hardening

15.1 Production Checklist

Before deploying to production, complete every item:

Item	Action	Priority
JWT Secret	Change <code>JWT_SECRET_KEY</code> to a random 256-bit key	Critical
Dev Auth	Set <code>AZURE_AD_TENANT_ID</code> to disable dev bypass	Critical
CORS	Restrict <code>allow_origins</code> from <code>*</code> to your domain	High
SSL/TLS	Configure nginx with SSL certificates	High
Database Password	Use a strong <code>POSTGRES_PASSWORD</code>	High
API Keys	Set <code>SETTINGS_ENCRYPTION_KEY</code> for at-rest encryption	High
Port Exposure	Remove host port mappings for postgres/redis	Medium
Secrets	Move <code>.env</code> to Docker secrets or vault	Medium
Rate Limiting	Add nginx rate limiting rules	Medium
Monitoring	Set up external monitoring and alerting	Medium

15.2 CORS Configuration

The default CORS configuration allows all origins. For production, modify `backend/api/main.py`:

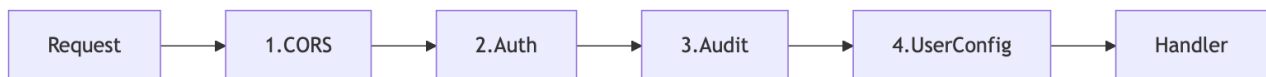
```
origins = [  
    "https://your-domain.com",  
    "https://app.your-domain.com",  
]
```

15.3 API Key Encryption

Enable at-rest encryption for stored API keys:

```
# Generate a Fernet key  
python -c "from cryptography.fernet import Fernet; print(Fernet.generate_key())"  
  
# Add to .env  
SETTINGS_ENCRYPTION_KEY=your-generated-fernet-key
```

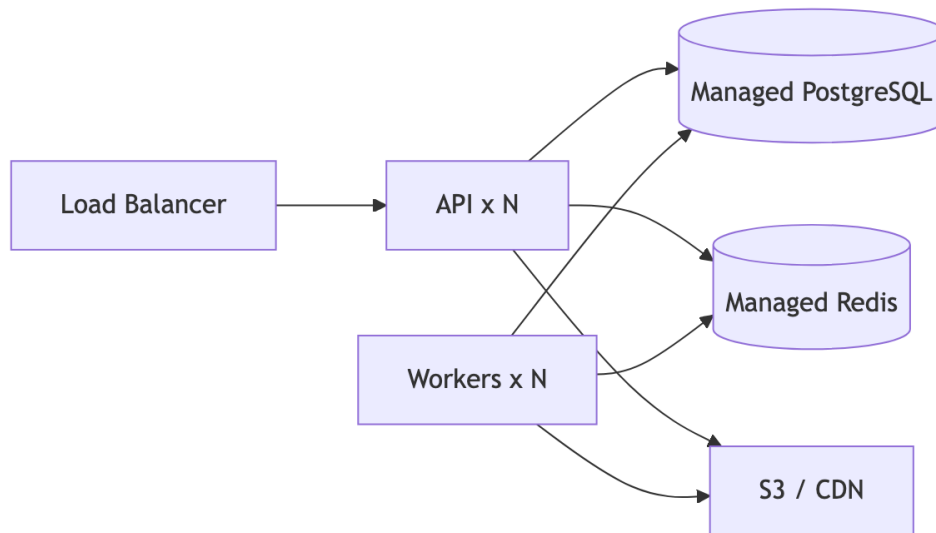
15.4 Middleware Execution Order



Middleware are added in reverse order in FastAPI (last added = first executed).

16. Scaling & Performance

16.1 Horizontal Scaling



Component	Scaling Strategy
API	Run multiple replicas behind load balancer
Worker	Run multiple instances (ARQ handles job locking)
PostgreSQL	Use managed service (RDS, Cloud SQL) with read replicas
Redis	Use managed service (ElastiCache) with clustering
Files	Replace local volume with S3 + CDN
nginx	Replace with cloud load balancer (ALB, Cloud Load Balancing)

16.2 Vertical Scaling

Setting	Location	Effect
<code>max_jobs</code>	<code>WorkerSettings</code>	More concurrent background jobs
<code>job_timeout</code>	<code>WorkerSettings</code>	Allow longer-running LLM operations
<code>worker_connections</code>	<code>nginx.conf</code>	More concurrent connections
<code>client_max_body_size</code>	<code>nginx.conf</code>	Larger file uploads

16.3 Performance Optimization

- Database indexes already exist on:

- `auditlogmodel.created_at`
- `teammembershipmodel(user_id, team_id)` (unique)
- `usermodel.email` (unique)
- `clientmodel.slug` (unique)
- **Async operations:** All database queries use `asyncpg` (async PostgreSQL driver)
- **Sync LLM calls:** Wrapped in `asyncio.to_thread()` to avoid blocking the event loop
- **SSE streaming:** Server-Sent Events for real-time progress (no polling overhead)
- **Fire-and-forget audit:** Audit logs don't block request processing

17. Troubleshooting

17.1 Common Issues

Issue	Cause	Solution
API won't start	PostgreSQL not ready	Wait for health check (15–25s)
Worker jobs not processing	Redis down or worker crashed	<code>docker compose restart worker</code>
Auth failures after restart	JWT_SECRET_KEY changed	All users must re-login
File uploads fail	Disk full or wrong permissions	Check <code>du -sh /app_data/*</code> and permissions
PPTX export 500	Puppeteer / Chromium issue	Restart web service, check memory
Slide edit timeout	LLM response too slow	Check provider status, increase timeouts
Master deck stuck "processing"	Worker died during parse	Restart worker, reparse the deck
Images not showing	Static files not served	Check FastAPI mounts and nginx config

SSE not working	Proxy buffering enabled	Ensure nginx <code>proxy_buffering off</code>
-----------------	-------------------------	--

17.2 Diagnostic Commands

```
# Service health
docker compose ps

# Container resource usage
docker stats

# API application logs
docker compose logs --tail=50 api

# Worker job processing logs
docker compose logs --tail=50 worker

# Database connection check
docker compose exec postgres pg_isready -U deckforge

# Redis connectivity
docker compose exec redis redis-cli ping

# Database query – check failed jobs
docker compose exec postgres psql -U deckforge -c \
    "SELECT id, job_type, status, error_message FROM jobmodel WHERE status="

# Disk usage
docker compose exec api du -sh /app_data/*
```

17.3 Resetting the System

```
# Full reset (WARNING: destroys all data)
docker compose down -v
docker compose up --build -d
make migrate
make seed
```

18. API Reference

18.1 Authentication Endpoints

Method	Path	Description
GET	<code>/api/v1/auth/login</code>	Redirect to Azure AD login
GET	<code>/api/v1/auth/callback</code>	OAuth callback handler
POST	<code>/api/v1/auth/dev-login</code>	Dev mode authentication
GET	<code>/api/v1/auth/dev-status</code>	Check if dev mode is enabled
POST	<code>/api/v1/auth/logout</code>	Clear session
GET	<code>/api/v1/auth/me</code>	Current user info

18.2 Admin Endpoints

Method	Path	Access
GET/PUT/DELETE	<code>/api/v1/admin/users/*</code>	Super Admin
POST/GET/PUT/DELETE	<code>/api/v1/admin/clients/*</code>	Admin
POST/GET/DELETE	<code>/api/v1/admin/teams/*</code>	Admin
GET/PUT	<code>/api/v1/admin/settings</code>	Super Admin
GET/POST	<code>/api/v1/admin/settings/models</code>	Super Admin
POST	<code>/api/v1/admin/settings/test-connection</code>	Super Admin
GET/DELETE/POST	<code>/api/v1/admin/storage/*</code>	Admin
GET	<code>/api/v1/admin/analytics/*</code>	Admin
GET	<code>/api/v1/admin/audit-log</code>	Admin
GET/PUT/POST/DELETE	<code>/api/v1/admin/master-decks/*</code>	Admin

GET/PUT/POST/DELETE	<code>/api/v1/admin/brand-config/*</code>	Admin
---------------------	---	-------

18.3 Presentation Endpoints

Method	Path	Description
POST	<code>/api/v1/ppt/presentation/create</code>	Create new presentation
GET	<code>/api/v1/ppt/presentation/all</code>	List presentations
GET	<code>/api/v1/ppt/presentation/{id}</code>	Get presentation detail
PUT	<code>/api/v1/ppt/presentation/{id}</code>	Update presentation
DELETE	<code>/api/v1/ppt/presentation/{id}</code>	Delete presentation
POST	<code>/api/v1/ppt/presentation/decompose</code>	Decompose content
POST	<code>/api/v1/ppt/presentation/prepare</code>	Prepare for generation
GET	<code>/api/v1/ppt/presentation/{id}/review</code>	Get review status
PUT	<code>/api/v1/ppt/presentation/{id}/status</code>	Change review status
POST	<code>/api/v1/ppt/presentation/{id}/comment</code>	Add review comment
POST	<code>/api/v1/ppt/jobs/generate</code>	Start generation job
GET	<code>/api/v1/ppt/jobs/{id}/status</code>	Job status (SSE)
POST	<code>/api/v1/ppt/export/pptx</code>	Export as PPTX
POST	<code>/api/v1/ppt/export/pdf</code>	Export as PDF