indition™

INDITION AI CORE

Feature List

The platform AI infrastructure layer — providers, models, credentials, request types, generation, embeddings, logging, and cost.

AI Core Module · Version 1.0 · June 2026

Indition AI Core — Feature List

> **Indition AI Core** is the platform's shared AI infrastructure layer. It gives every Indition module a single, provider-agnostic way to call AI: register providers and models, store credentials securely, define structured request types, generate text/images/email, create embeddings, and log every request with full cost attribution. Product modules such as **AI Assistant** build on it rather than talking to AI providers directly.

This document catalogues what AI Core provides, grouped by area. It is an administrator- and integrator-facing reference; there is no end-user surface. The currently supported providers are those you configure (for example OpenAI and Google/Gemini).

At a glance

Area	What it delivers
Providers & models	Register AI providers and their models with pricing and capability flags.
Credentials	Encrypted-at-rest API keys, per provider, with default selection.
Request types	Reusable, schema-validated request definitions with system prompts.
Generation	Text, rewrite, image, and structured AI-email generation endpoints.
Embeddings	Single and batch embedding service for retrieval/RAG.
Document extraction	Pull text from PDF, DOCX, and TXT for use as context.
Logging & cost	Request logs with redaction, retention, attribution, and token-cost calculation.
Sharing & governance	Share providers/models/request types to customers; tunable safety/limit settings.

1. Providers, models & credentials

PROVIDERS

- Register external AI providers (code, name, base URL, status, provider config).
- Mark a **default** provider; optionally **share by default** with all customers.
- Provider-agnostic client routes calls to the right vendor (e.g. OpenAI, Google/Gemini).

MODELS

- Register models under a provider with a code, name, and context window.
- **Pricing**: per-million-token input and output cost rates used for cost calculation.
- **Capability flags**: completion support, vision (image input), and image generation.
- Status and default-share controls per model.

CREDENTIALS

- Store provider API keys **encrypted at rest** (AES), never in plain text.
- Multiple labelled credentials per provider, with a default selection.
- Provider-specific metadata kept alongside the encrypted key.

2. Request types & generation

REQUEST TYPES

- Define reusable request types with a code, name, and handler class.
- **Structured output**: JSON-schema or JSON-object response formats with a stored schema and example.
- **System prompt** per type, a default model, and an on-invalid-JSON policy.
- Retry handling for malformed responses.

GENERATION ENDPOINTS

- **Text generation** — complete and block-level generation.
- **Rewrite** — transform/improve existing text.
- **Image generation** — create and edit images (provider-permitting).
- **AI email generation** — produce structured, multi-block emails from a validated email schema (with optional image blocks, image limits, and retry on invalid JSON).
- **Vision** — send images as context (base64 or URL, configurable).
- Streaming responses supported by the provider client.

DOCUMENT EXTRACTION

- Extract text from **PDF, DOCX, and TXT** uploads for use as model context.
- Whitespace normalization and a configurable character cap.
- Allowed document and image types, max file size, and per-request document/image counts are all configurable.

3. Embeddings

- Shared **embedding service** for knowledge/RAG use cases.
- Single-text and **batch** embeddings (configurable batch size).
- Reuses the same provider, credential, retry, logging, and cost infrastructure.
- Logs store only input hashes/lengths and embedding counts/dimensions — never raw vectors or full text.
- Attribution and context tags supported so embeddings trace back to the calling feature.

4. Request logging & cost

- **Request log** for every AI call: provider, model, credential, endpoint, status, HTTP status, latency, tokens, and cost.
- **Redaction** of sensitive fields (e.g. api_key, authorization, secret) before payloads are stored.
- **Retention** window (default 365 days) for pruning old logs.
- **Attribution**: source product, module, feature, operation, correlation id, actor, and a primary business context.
- **Context rows** link a single request to multiple business objects (assistant, collection, conversation, message, resource, ingestion job, …).
- **Cost calculation** from normalized token usage and model rates.
- Admin grid to filter and inspect logs.

5. Sharing, settings & access

CUSTOMER SHARING

- Share **providers, models, and request types** with specific customers (or by default to all).
- Multi-tenant visibility controls via a dedicated sharing grid.

CONFIGURABLE SETTINGS

- **Master switch**: `AI_CORE_ENABLED` turns all AI on or off.
- **Logging**: enable/disable, retention days, redaction fields.
- **Reliability**: request timeout and max retries.
- **Documents/vision**: max file size, allowed doc/image types, max info/image docs, extracted-text cap, force-base64 for vision.
- **Email generation**: images on/off, max images, hide model selection, invalid-response retries.

ADMINISTRATION & ACCESS

- Owner/admin menus: AI Settings, Manage AI Providers, Manage AI Models, Manage AI Request Types, Manage AI Request Logs, and Miscellaneous AI Settings.
- Component-driven access control (owner-settings and admin-settings scopes).

6. Platform role

- Single integration point for AI across the platform — modules call AI Core, not vendors.
- Consumed today by **AI Assistant** (generation, embeddings, extraction, logging, cost) and available to any future AI feature.
- Centralizes provider strategy, credential security, cost visibility, and audit in one place.