

# WEBCON

## AI Proxy Self-hosted

**2026.2.135.35.**

history of changes for version 2026.2.135.35.

## 1. New features

- Added a web-based administration interface for configuring AI providers, connections, and models. The interface is always available in self-hosted mode.
- Added a configurable timeout for connections to AI providers (`TimeoutSeconds`). When the timeout is exceeded, the call is aborted and the request is forwarded to the next configured model.

## 2. Improvements

- Updated the Gemini integration by replacing the previous library with the Google Gen AI SDK.
- Updated the default model configuration by adding Gemini 2.5 Pro and defining expiration dates for Gemini models scheduled for retirement.
- Improved how the AI process design assistant (Designer Desk) reproduces workflows and data models described in attached files. The assistant now faithfully reproduces the attached definition instead of adding unspecified fields, and each file is processed only once, reducing token usage.
- Improved the prompts used by the AI process design assistant and switched the assistant to the Gemini 2.5 Flash model.
- Added scripts and documentation for building and running AI Proxy locally in self-hosted mode using Docker. The resources are available in the `WEBCON-BPS/WebCon-AiProxy-GettingStarted` repository.

### 3. Bug fixes

- Fixed fallback to the next provider when a model repeatedly returned rate-limit errors (429). After all retries are exhausted, the request is now forwarded to the next model instead of failing.
- Fixed an issue where the AI process design assistant generated unwanted workflow elements or form fields when the task concerned only one of these areas. Design tools are now selected based on the content of the task.
- Fixed inconsistent references between phases, steps, paths, and data sources in AI process design commands after identifier reassignment, as well as a unique identifier error that occurred when adding elements.
- Fixed an issue where the AI process design assistant duplicated the start step and phases instead of updating the existing elements. Generated path names were also corrected.
- Fixed an error when parsing the `PickersToSet` collection in AI process design commands.
- Fixed incorrect collection creation in AI translations when using the gpt-4.1-mini model.
- Fixed an error when retrieving an AI Proxy token caused by a missing limit value in the database.
- Fixed model configuration binding and loading when communicating with AI models through a proxy running in self-hosted mode.
- Fixed a "Parameter cannot be null" error in the end-of-day session cleanup job.

# WEBCON

## AI Proxy Self-hosted

### 2026.2.64.17.

history of changes for version 2026.2.64.17.

## 1. New features

- Added handling for models marked as expired, allowing AI Proxy to correctly recognize models that should no longer be used.
- Added model selection support in v4 endpoints for AI Concierge, translations, transcriptions, and AI Agent sessions.
- Added information about the model used to the AI operation result.
- Added an endpoint for retrieving available models together with their identifiers.
- Added `Chat with Tools` support for OpenAI and Gemini models.
- Added an OpenAI API-compatible connector, enabling integration with LiteLLM, LM Studio, OpenRouter and similar providers.
- Added inline file upload support for OpenAI, Azure, Gemini, and the OpenAI API-compatible connector.
- Added support for Azure authentication with `DefaultAzureCredential` as an alternative to API key authentication.
- Added OIDC support in services using the OpenAI client.

## 2. Improvements

- Improved AI Proxy configuration loading and validation, including support for model identifiers and provider configuration.
- Renamed `MethodType` to grouping `AiTaskType` in `aiconfiguration.json` to better reflect the purpose of the configured AI operation.
- Renamed the `Authorization` node to `OIDCAuthorization` to make the OIDC authorization configuration more explicit.
- Aligned the behavior of the OpenAI API-compatible connector with OpenAI handling, including cleanup mechanisms and file handling.
- Improved API key passing and handling in services using OpenAI.
- Added logging of the model used to process an AI request.
- Improved cancellation handling in `Chat with Tools`.
- Improved AI Concierge responses during navigation and enhanced handling of content returned by tools.

### 3. Bug fixes

- Fixed an issue with uploading files to Azure by switching to the OpenAI client.
- Fixed .txt attachment handling in Gemini chat conversations.
- Fixed file handling in the OpenAI connector.
- Fixed AI Concierge command parsing in the OpenAI API-compatible connector.
- Fixed an exception caused by a missing API key in `OpenAiService`.
- Fixed v4 endpoints handling when the `method` parameter is missing.
- Fixed prompt handling in selected AI Concierge scenarios.
- Fixed the transcription name in the enum.
- Improved error logging when loading the AI configuration.

# WEBCON

## AI Proxy Self-hosted

### 2026.1.27.108.

history of changes for version 2026.1.27.108.

## 1. New features

- Added the Concierge v3 endpoint with support for navigation tools, making **AI Concierge** available across the entire Portal, where it can assist users with navigation and suggest moving to the appropriate locations. The actual navigation is performed after the user clicks the navigation button. Support was also extended to include the `//goto` command, which forces the use of the navigation mechanism. This change requires WEBCON version 2026.1.6 or later to work.

# WEBCON

## AI Proxy Self-hosted

### 2026.1.24.106

history of changes for version 2026.1.24.106

## 1. New features

- Added support for the Azure AI Foundry connector in AI Proxy, together with AI Connector logic. Clarified the requirements for Azure-side resource configuration and network access needed to run and test the integration locally.
- Changed the version numbering scheme for AI Proxy Self-hosted. The version number is now calculated automatically and indicates compatibility with a specific WEBCON version.

# WEBCON

## AI Proxy Self-hosted

### 1.0.0.235

history of changes for version 1.0.0.235

# 1. New features

- Released **AI Proxy Self-hosted** — a new component that enables customers to use AI functionality in WEBCON based on their own infrastructure, completely bypassing WEBCON infrastructure. The solution is available under the Enterprise subscription license and the perpetual WEBCON Agentic AI Framework license.

AI Proxy Self-hosted can be installed in the customer's environment (on-premise or private cloud), with the option to connect and configure the customer's own AI models — hosted in Google Vertex, OpenAI, or Azure AI Foundry. This gives customers full control over the data and infrastructure used for AI processing, making it easier to meet security requirements, privacy policies, and industry regulations.

A dedicated AI Proxy Self-hosted mode has been introduced, along with a new configuration node for this mode, added to the `appsettings.json` configuration files used to run AI Proxy, including:

- the option to enable Azure Key Vault for storing sensitive data,
- a variable storing the path to a .pem or .pfx certificate along with its password.

An alternative class implementation has also been prepared which, in this mode, allows Azure Key Vault to be replaced with environment variables.

A new, unified AI Proxy Self-hosted provider-configuration model based on priorities has been designed and released. It allows defining multiple providers, automatically switching to lower-priority providers, and assigning different AI models to specific methods (text, image, and audio models). The configuration is defined in the `aiconfiguration.json` file, which is loaded when the AI Proxy application starts.

The ability to select the application's operating mode in the configuration file has been added, along with support for local certificates (.pem/.pfx) used to sign and verify access tokens — without the need to communicate with external authorization services (including LAS) or databases. In AI Proxy Self-hosted mode, dependencies on external infrastructure components have been reduced: the application can run without connections to licensing services, databases, or cloud data stores, and the required mechanisms are automatically selected based on configuration.