

AzureClaudeProvider

September 4, 2026

AzureClaudeProvider

Kind: Class

Source: [atloria-monorepo/apps/api/src/ai/providers/azure-claude.provider.ts](https://github.com/atloria-monorepo/apps/api/src/ai/providers/azure-claude.provider.ts)

Azure AI Foundry Claude Provider

Uses native Anthropic Messages API endpoint through Azure AI Foundry

Endpoint: /anthropic/v1/messages

Authentication priority:

1. API Key (ANTHROPIC_FOUNDRY_API_KEY) - simplest, no az login needed
2. Azure AD (DefaultAzureCredential/AzureCliCredential) - fallback

`AzureClaudeProvider` integrates Claude models hosted in Azure AI Foundry with the application's AI provider layer. It calls Azure's native Anthropic Messages API endpoint (`/anthropic/v1/messages`) for standard, usage-aware, and streaming text generation, then exposes analysis results through the shared provider interface. Authentication prefers `ANTHROPIC_FOUNDRY_API_KEY` and falls back to Azure AD credentials when no API key is configured.

Implements: `AIProvider`

Methods

Method	Signature	Returns
<code>getAzureAIConfig</code>	<code>getAzureAIConfig()</code>	<code>`Promise<{ endpoint: string; accessToken?: string; apiKey?: string; modelName: string }`</code>
<code>isAvailable</code>	<code>isAvailable()</code>	<code>Promise<boolean></code>
<code>generateText</code>	<code>generateText(prompt: string, options: GenerateOptions)</code>	<code>Promise<string></code>

Method	Signature	Returns
<code>generateTextWithUsage</code>	<code>generateTextWithUsage(prompt: string, options: GenerateOptions)</code>	<code>Promise<AIResponse></code>
<code>generateStream</code>	<code>generateStream(prompt: string, options: GenerateOptions)</code>	<code>AsyncIterableIterator<string></code>
<code>analyze</code>	<code>analyze(content: string, type: AnalysisType)</code>	<code>Promise<AnalysisResult></code>

Properties

Property	Type
<code>name</code>	<code>any</code>

Where it refuses work

- `AzureClaudeProvider` stops the work with `Error` when `!projectEndpoint` — “Azure AI Project configuration missing. Please set `AZURE_EXISTING_AIPROJECT_ENDPOINT`”.
- `AzureClaudeProvider` stops the work with an early return when `this.tokenCache && this.tokenCache.expiresAt > Date.now()` .
- `AzureClaudeProvider` stops the work with an early return when `this.apiKey` .
- `AzureClaudeProvider` stops the work with an early return when `data === '[DONE]'` .

When something fails

- `AzureClaudeProvider` handles failure in 5 places: it turns it into a return value in 2, lets it reach the caller in 2, and logs it and continues in 1.

Diagram

```
graph LR
  App[Application AI Request] --> Provider[AzureClaudeProvider]
  Provider --> Config[getAzureAIConfig]

  Config --> APIKey{ANTHROPIC_FOUNDRY_API_KEY available?}
  APIKey -->|Yes| KeyAuth[API Key Authentication]
  APIKey -->|No| AzureAD[DefaultAzureCredential / AzureCliCredential]
```

```

KeyAuth --> Endpoint[Azure AI Foundry<br/>/anthropic/v1/messages]
AzureAD --> Endpoint

Endpoint --> Text[generateText]
Endpoint --> Usage[generateTextWithUsage]
Endpoint --> Stream[generateStream]
Text --> Analysis[analyze]
Usage --> Response[AIResponse]
Stream --> Chunks[AsyncIterableIterator<string>]

```

Usage

```

import { AzureClaudeProvider } from './ai/providers/azure-claude.provider';

const provider = new AzureClaudeProvider();

if (!(await provider.isAvailable())) {
  throw new Error(
    'Azure Claude is unavailable. Configure ANTHROPIC_FOUNDRY_API_KEY or Azure AD credentials.'
  );
}

const response = await provider.generateTextWithUsage(
  'Summarize the following incident report in three bullet points.',
  {
    systemPrompt: 'You are a concise engineering incident assistant.',
  },
);

console.log(response.content);
console.log('Token usage:', response.usage);

// Stream responses when incremental output is needed.
for await (const chunk of provider.generateStream('Explain this error log.')) {
  process.stdout.write(chunk);
}

```

AI Coding Instructions

- Use `getAzureAIConfig()` as the single source of truth for endpoint, model, and authentication configuration; do not duplicate credential resolution in request methods.
- Prefer `ANTHROPIC_FOUNDRY_API_KEY` for local development and service deployments because it does not require an active Azure CLI session.

- Preserve Azure AD fallback behavior with `DefaultAzureCredential` or `AzureCliCredential` when the API key is absent.
- Send requests using the native Anthropic Messages API contract at `/anthropic/v1/messages` ; do not convert requests to Azure OpenAI chat-completions format.
- Ensure streaming methods yield text chunks incrementally and that usage-aware methods return the shared `AIResponse` structure.

Referenced By

- `AIModule` (MODULE_PROVIDES)
- `AIModule` (MODULE_EXPORTS)
- `AIService` (DEPENDS_ON)
- `BusinessContextParserService` (DEPENDS_ON)
- `DocumentationService` (DEPENDS_ON)
- `CompetitorDiscoveryService` (DEPENDS_ON)
- `DocAutomationService` (DEPENDS_ON)
- `FeatureDocsGeneratorService` (DEPENDS_ON)
- `Neo4jDocsOrchestratorService` (DEPENDS_ON)
- `TutorialGeneratorService` (DEPENDS_ON)
- `UserDocsGeneratorService` (DEPENDS_ON)
- `WorkflowDiscoveryService` (DEPENDS_ON)