

# AzureOpenAIProvider

September 4, 2026

## AzureOpenAIProvider

**Kind:** Class

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

`AzureOpenAIProvider` integrates Azure OpenAI models into the API's AI provider layer. It checks provider availability, generates complete or streamed text responses, returns usage metadata when needed, and performs structured analysis through the shared AI abstractions.

**Implements:** `AIProvider`

## Methods

| Method                             | Signature                                                                    | Returns                                          |
|------------------------------------|------------------------------------------------------------------------------|--------------------------------------------------|
| <code>isAvailable</code>           | <code>isAvailable()</code>                                                   | <code>Promise&lt;boolean&gt;</code>              |
| <code>generateText</code>          | <code>generateText(prompt: string, options: GenerateOptions)</code>          | <code>Promise&lt;string&gt;</code>               |
| <code>generateTextWithUsage</code> | <code>generateTextWithUsage(prompt: string, options: GenerateOptions)</code> | <code>Promise&lt;AIResponse&gt;</code>           |
| <code>generateStream</code>        | <code>generateStream(prompt: string, options: GenerateOptions)</code>        | <code>AsyncIterableIterator&lt;string&gt;</code> |
| <code>analyze</code>               | <code>analyze(content: string, type: AnalysisType)</code>                    | <code>Promise&lt;AnalysisResult&gt;</code>       |

## Properties

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

## Where it refuses work

- `AzureOpenAIProvider` stops the work with an early return when `data === '[DONE]'`.

## When something fails

- `AzureOpenAIProvider` handles failure in 4 places: it lets it reach the caller in 2, logs it and continues in 1, and turns it into a return value in 1.

## Diagram

```
graph LR
  Client[API service or feature] --> Provider[AzureOpenAIProvider]
  Provider --> Availability[isAvailable()]
  Provider --> Text[generateText()]
  Provider --> Usage[generateTextWithUsage()]
  Provider --> Stream[generateStream()]
  Provider --> Analysis[analyze()]
  Provider --> Azure[Azure OpenAI endpoint]
  Azure --> Provider
```

## Usage

```
import { AzureOpenAIProvider } from './ai/providers/azure-openai.provider';

const provider = new AzureOpenAIProvider(
  /* inject the Azure OpenAI configuration/dependencies used by the application */
);

if (!(await provider.isAvailable())) {
  throw new Error('Azure OpenAI is not currently available');
}

const text = await provider.generateText({
  prompt: 'Summarize the following support ticket in two sentences.',
  input: ticketContent,
});

const response = await provider.generateTextWithUsage({
  prompt: 'Create a concise customer-facing reply.',
  input: ticketContent,
});

console.log(response);

for await (const chunk of provider.generateStream({
  prompt: 'Draft a response one section at a time.',
```

```
input: ticketContent,
})) {
  process.stdout.write(chunk);
}
```

## AI Coding Instructions

- Use the shared AI provider request and response types when calling `generateText` , `generateTextWithUsage` , `generateStream` , and `analyze` .
- Check `isAvailable()` before routing work to Azure OpenAI, especially when the system supports provider fallback behavior.
- Prefer `generateTextWithUsage()` when token or cost tracking is required; use `generateText()` for simple text-only workflows.
- Consume `generateStream()` with `for await...of` and forward chunks incrementally rather than buffering the entire response.
- Keep Azure endpoint, deployment, API version, and credentials in application configuration; do not hardcode Azure credentials in callers.

## Referenced By

- `AIModule` (`MODULE_PROVIDES`)
- `AIModule` (`MODULE_EXPORTS`)
- `AIService` (`DEPENDS_ON`)
- `FeatureDocsGeneratorService` (`DEPENDS_ON`)
- `TutorialGeneratorService` (`DEPENDS_ON`)
- `UserDocsGeneratorService` (`DEPENDS_ON`)