

AzureResponsesProvider

September 4, 2026

AzureResponsesProvider

Kind: Class

Source: [atloria-monorepo/packages/doc-automation/src/v2/providers/azure-responses.provider.ts](#)

Implements: [AIProvider](#)

Methods

Method	Signature	Returns
<code>generate</code>	<code>generate(prompt: string, options: AIGenerateOptions, stage: JobStage)</code>	<code>Promise<AIResponse></code>
<code>generateWithStreaming</code>	<code>generateWithStreaming(prompt: string, options: AIGenerateOptions, stage: JobStage)</code>	<code>Promise<AIResponse></code>
<code>generateText</code>	<code>generateText(prompt: string, options: Omit<AIGenerateOptions, 'messages'>, stage: JobStage)</code>	<code>Promise<string></code>
<code>generateJSON</code>	<code>generateJSON(prompt: string, options: AIGenerateOptions, stage: JobStage)</code>	<code>Promise<{ data: T; response: AIResponse }></code>
<code>generateWithStructuredOutput</code>	<code>`generateWithStructuredOutput(options: {`</code>	

```
prompt: string;
schema: object;
model?: string;
```

`) | Promise<{ content: T; usage: { inputTokens: number; outputTokens: number } }> |`
`| isAvailable | isAvailable() | Promise |`

```
| getThrottlingStatus | getThrottlingStatus() | { consecutiveFailures: number;  
adaptiveDelayMs: number; lastFailureTime: number }` |
```

Properties

Property	Type
name	string

Where it refuses work

- `AzureResponsesProvider` stops the work with `Error` when `!config.endpoint` — “Azure Responses provider requires endpoint”.
- `AzureResponsesProvider` stops the work with `Error` when `!config.apiKey` — “Azure Responses provider requires apiKey”.
- `AzureResponsesProvider` stops the work with `AIProviderError` when `content.length === 0 && inputTokens === 0` — “Azure returned empty stream — throttled (429)”.
- `AzureResponsesProvider` stops the work with an early return when `error instanceof AIProviderError`.
- `AzureResponsesProvider` stops the work with an early return when `!response.ok`.

When something fails

- `AzureResponsesProvider` handles failure in 5 places: it lets it reach the caller in 2, logs it and continues in 1, turns it into a return value in 1, and discards it silently in 1. A failure discarded silently leaves no trace for whoever debugs this later.