

OpenAI Component

High-level OpenAI HTTP API client — chat completions, responses, embeddings, images, audio, files, fine-tuning, batches and assistants.

Overview

OpenAI REST API client covering Chat Completions, Images, Audio (TTS/STT), Embeddings, Files, Fine-tuning, Moderation, Assistants, Threads, Runs, Messages and Vector Stores.

At a glance

COMPONENT CLASS

TsgcHTTP_API_OpenAI

STANDARDS / SPEC

OpenAI API reference

TRANSPORTS

TCP, TLS

PLATFORMS

Windows, macOS, Linux, iOS, Android

FRAMEWORKS

VCL, FireMonkey, Lazarus / FPC, .NET

EDITION

Standard / Professional / Enterprise

Features

- Native Delphi implementation with full ANSI/Unicode support.

Technical specification

Standards & specs	OpenAI API reference · OpenAI Chat Completions
Component class	<code>TsgcHTTP_API_OpenAI</code> (unit <code>sgcHTTP_API_OpenAI</code>)
Frameworks	VCL, FireMonkey, Lazarus / FPC, .NET
Platforms	Windows, macOS, Linux, iOS, Android

Main properties

The principal published / public properties used to configure and drive the component. Consult the online help for the full list.

<code>TLSoptions</code>	TLS/SSL configuration used for HTTPS connections to the OpenAI API
<code>OpenAIOptions</code>	Container for all OpenAI REST client settings including credentials, organization, HTTP, logging and retry options
<code>ReadTimeout</code>	HTTP read timeout in milliseconds applied to every OpenAI REST request
<code>CircuitBreaker</code>	Retry and rate-limit handling policy used when OpenAI endpoints fail transiently
<code>Version</code>	Read-only <code>sgcWebSockets</code> library version string

Main methods

The principal public methods exposed by the component.

<code>ListInputItems()</code>	Lists the input items that were provided when a response was created
<code>CreateThread()</code>	Creates a new Thread optionally seeded with initial messages and tool resources
<code>ModifyThread()</code>	Updates the tool resources and metadata of an existing Thread
<code>SubmitToolOutputsToRun()</code>	Returns tool function outputs to a Run that is waiting in <code>requires_action</code> state

<code>DeleteResponse()</code>	Deletes a stored response from the OpenAI Responses API
<code>DeleteFile()</code>	Permanently deletes a file previously uploaded to OpenAI
<code>DeleteFineTuneModel()</code>	Deletes a fine-tuned model so it is no longer available for inference
<code>DeleteAssistant()</code>	Permanently deletes an Assistant by its identifier
<code>RetrieveThread()</code>	Retrieves a Thread definition by its identifier
<code>DeleteThread()</code>	Permanently deletes a Thread and all of its messages

Public events

The component exposes the following published events; consult the online help for full event-handler signatures.

<code>OnHTTPAPIException</code>	Fires when an uncaught exception is raised during an OpenAI REST call
<code>OnHTTPAPISSE</code>	Fires for each Server-Sent Events chunk received during a streaming OpenAI call

Quick Start

Drop the component on a form, configure the properties below and activate it. The snippet that follows shows the typical **OpenAI Chat — Simple Example** configuration sourced from the online help.

About this scenario. Interact with ChatGPT by sending a Hello message.

Delphi (VCL / FireMonkey)

```
OpenAI := TsgcHTTP_API_OpenAI.Create(nil);
OpenAI.OpenAIOptions.ApiKey := 'API_KEY';
WriteLn(OpenAI._CreateChatCompletion('gpt-3.5-turbo', 'Hello!'));
```

C++ Builder

```
TsgcHTTP_API_OpenAI *OpenAI = new TsgcHTTP_API_OpenAI(NULL);
OpenAI->OpenAIOptions->ApiKey = "API_KEY";
ShowMessage(OpenAI->_CreateChatCompletion("gpt-3.5-turbo", "Hello!"));
```

.NET (C#)

```
TsgcHTTP_API_OpenAI OpenAI = new TsgcHTTP_API_OpenAI();
OpenAI.OpenAIOptions.ApiKey = "API_KEY";
MessageBox.Show(OpenAI._CreateChatCompletion("gpt-3.5-turbo", "Hello!"));
```

Common scenarios

The following scenarios are lifted verbatim from the online help. Each shows the configuration and method calls needed to drive the component through a specific real-world flow.

1 · OpenAI Batch — Simple Example

Create a batch job and retrieve its status.

Delphi (VCL / FireMonkey)

```
OpenAI := TsgcHTTP_API_OpenAI.Create(nil);
OpenAI.OpenAIOptions.ApiKey := 'API_KEY';

// Create a batch
WriteLn(OpenAI._CreateBatch('file-abc123', '/v1/chat/completions'));

// List batches
WriteLn(OpenAI._ListBatches());
```

C++ Builder

```
TsgcHTTP_API_OpenAI *OpenAI = new TsgcHTTP_API_OpenAI(NULL);
OpenAI->OpenAIOptions->ApiKey = "API_KEY";

// Create a batch
ShowMessage(OpenAI->_CreateBatch("file-abc123", "/v1/chat/completions"));

// List batches
ShowMessage(OpenAI->_ListBatches());
```

.NET (C#)

```
TsgcHTTP_API_OpenAI OpenAI = new TsgcHTTP_API_OpenAI();
OpenAI.OpenAIOptions.ApiKey = "API_KEY";

// Create a batch
MessageBox.Show(OpenAI._CreateBatch("file-abc123", "/v1/chat/completions"));

// List batches
MessageBox.Show(OpenAI._ListBatches());
```

2 · OpenAI Completion — Simple Example

Use the text-davinci-003 model to get a predicted completion.

Delphi (VCL / FireMonkey)

```
OpenAI := TsgcHTTP_API_OpenAI.Create(nil);
OpenAI.OpenAIOptions.ApiKey := 'API_KEY';
WriteLn(OpenAI._CreateCompletion('text-davinci-003', 'Say this is a test'));
```

C++ Builder

```
TsgcHTTP_API_OpenAI *OpenAI = new TsgcHTTP_API_OpenAI(NULL);
OpenAI->OpenAIOptions->ApiKey = "API_KEY";
ShowMessage(OpenAI->_CreateCompletion("text-davinci-003", "Say this is a test"));
```

.NET (C#)

```
TsgcHTTP_API_OpenAI OpenAI = new TsgcHTTP_API_OpenAI();
OpenAI.OpenAIOptions.ApiKey = "API_KEY";
MessageBox.Show(OpenAI._CreateCompletion("text-davinci-003", "Say this is a test"));
```

3 · OpenAI Edit — Simple Example

Tell OpenAI to fix the spelling mistakes of a prompt.

Delphi (VCL / FireMonkey)

```
OpenAI := TsgcHTTP_API_OpenAI.Create(nil);
OpenAI.OpenAIOptions.ApiKey := 'API_KEY';
WriteLn(OpenAI._CreateEdit('text-davinci-edit-001', 'Fix the spelling mistakes', 'What day of th
```

C++ Builder

```
TsgcHTTP_API_OpenAI *OpenAI = new TsgcHTTP_API_OpenAI(NULL);
OpenAI->OpenAIOptions->ApiKey = "API_KEY";
ShowMessage(OpenAI->_CreateEdit("text-davinci-edit-001", "Fix the spelling mistakes", "What day
```

.NET (C#)

```
TsgcHTTP_API_OpenAI OpenAI = new TsgcHTTP_API_OpenAI();
OpenAI.OpenAIOptions.ApiKey = "API_KEY";
MessageBox.Show(OpenAI._CreateEdit("text-davinci-edit-001", "Fix the spelling mistakes", "What d
```

4 · OpenAI Fine-Tuning — Simple Example

Create a fine-tuning job and list existing jobs.

Delphi (VCL / FireMonkey)

```
OpenAI := TsgcHTTP_API_OpenAI.Create(nil);
OpenAI.OpenAIOptions.ApiKey := 'API_KEY';

// Create a fine-tuning job
WriteLn(OpenAI._CreateFineTuningJob('gpt-4o-mini-2024-07-18', 'file-abc123'));

// List fine-tuning jobs
WriteLn(OpenAI._ListFineTuningJobs());
```

C++ Builder

```
TsgcHTTP_API_OpenAI *OpenAI = new TsgcHTTP_API_OpenAI(NULL);
OpenAI->OpenAIOptions->ApiKey = "API_KEY";

// Create a fine-tuning job
ShowMessage(OpenAI->_CreateFineTuningJob("gpt-4o-mini-2024-07-18", "file-abc123"));

// List fine-tuning jobs
ShowMessage(OpenAI->_ListFineTuningJobs());
```

.NET (C#)

```
TsgcHTTP_API_OpenAI OpenAI = new TsgcHTTP_API_OpenAI();
OpenAI.OpenAIOptions.ApiKey = "API_KEY";

// Create a fine-tuning job
MessageBox.Show(OpenAI._CreateFineTuningJob("gpt-4o-mini-2024-07-18", "file-abc123"));

// List fine-tuning jobs
MessageBox.Show(OpenAI._ListFineTuningJobs());
```

5 · OpenAI Moderation — Simple Example

Moderate the following text

Delphi (VCL / FireMonkey)

```
OpenAI := TsgcHTTP_API_OpenAI.Create(nil);
OpenAI.OpenAIOptions.ApiKey := 'API_KEY';

WriteLn(OpenAI._CreateModeration('I want to kill them.'));
```

C++ Builder

```
TsgcHTTP_API_OpenAI *OpenAI = new TsgcHTTP_API_OpenAI(NULL);
OpenAI->OpenAIOptions->ApiKey = "API_KEY";

ShowMessage(OpenAI->_CreateModeration("I want to kill them."));
```

.NET (C#)

```
TsgcHTTP_API_OpenAI OpenAI = new TsgcHTTP_API_OpenAI();
OpenAI.OpenAIOptions.ApiKey = "API_KEY";

MessageBox.Show(OpenAI._CreateModeration("I want to kill them."));
```

6 · OpenAI Responses — Simple Example

Create a response using the Responses API.

Delphi (VCL / FireMonkey)

```
OpenAI := TsgcHTTP_API_OpenAI.Create(nil);
OpenAI.OpenAIOptions.ApiKey := 'API_KEY';
WriteLn(OpenAI._CreateResponse('gpt-4o', 'What is the capital of France?'));
```

C++ Builder

```
TsgcHTTP_API_OpenAI *OpenAI = new TsgcHTTP_API_OpenAI(NULL);
OpenAI->OpenAIOptions->ApiKey = "API_KEY";
ShowMessage(OpenAI->_CreateResponse("gpt-4o", "What is the capital of France?"));
```

.NET (C#)

```
TsgcHTTP_API_OpenAI OpenAI = new TsgcHTTP_API_OpenAI();  
OpenAI.OpenAIOptions.ApiKey = "API_KEY";  
MessageBox.Show(OpenAI._CreateResponse("gpt-4o", "What is the capital of France?"));
```

Sources used to build this document

Every external claim links back to a primary source. The online-help references decode the canonical deep-link the company maintains for this component.

Primary standard / spec — OpenAI API reference platform.openai.com/docs/api-reference

Primary standard / spec — OpenAI Chat Completions platform.openai.com/docs/api-reference/chat

Online help — component page www.egegece.com/help/sgcWebSockets/Components/AI/OpenAI/TsgcHTTP_API_OpenAI.htm

Delphi demo project (in the sgcWebSockets package) `Demos\15.AI\01.QuickStart\01.OpenAI`

.NET demo project (in the sgcWebSockets package) `.net\demos\15.AI\01.QuickStart\01.OpenAI`

Component page www.egegece.com/products/websockets/ai/openai/

Product page www.egegece.com/products/websockets/

Document scope. This document covers the publicly-documented surface of the OpenAI Component component shipped with sgcWebSockets. For full property, method and event reference consult the online help linked above.