SGCWEBSOCKETS · TECHNICAL DOCUMENT

Anthropic Component

High-level Anthropic Claude API client — Messages, streaming, tool use, vision and structured-output JSON schema.

eSeGeCe

WWW.ESEGECE.COM

Overview

Anthropic REST API client for Claude Messages, Files, Batches and token counting.

At a glance

COMPONENT CLASS

`TsgcHTTP_API_Anthropic`

STANDARDS / SPEC

Anthropic API — Getting started

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	Anthropic API — Getting started · Anthropic API — Messages
Component class	`TsgcHTTP_API_Anthropic` (unit `sgcHTTP_API_Anthropic`)
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.

`TLSOptions`	Configures the TLS/SSL layer used for HTTPS connections to the Anthropic API
`AnthropicOptions`	Holds connection settings for the Anthropic API, including API key, HTTP options, logging, retries and the anthropic-version header
`ReadTimeout`	Maximum time in milliseconds to wait for a response from the Anthropic API before aborting the request
`CircuitBreaker`	Protects the client from cascading failures by short-circuiting requests when the Anthropic API becomes unhealthy
`Version`	Read-only string containing the current version of the sgc Anthropic API client component

Main methods

The principal public methods exposed by the component.

`DeleteFile()`	Deletes a previously uploaded file from the Anthropic Files API
`CreateMessage()`	Sends a chat request to Claude and returns the generated message
`CountTokens()`	Counts the number of input tokens a request would consume before sending it to Claude

`ListFiles()`	Lists files previously uploaded to the Anthropic Files API, with optional pagination
`ListBatches()`	Lists all Message Batches in the Anthropic workspace, with optional pagination
`CancelBatch()`	Cancels an in-progress Message Batch on the Anthropic API
`UploadFile()`	Uploads a local file to the Anthropic Files API for later use in messages

Public events

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

`OnHTTPAPIException`	Fires when a call to the Anthropic API raises an unhandled exception
`OnHTTPAPISSE`	Fires for every Server-Sent Event received from a streaming Anthropic response

Quick Start

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

> **About this scenario.** Send a PDF document with a question using the convenience method.

Delphi (VCL / FireMonkey)

```
Anthropic := TsgcHTTP_API_Anthropic.Create(nil);
Anthropic.AnthropicOptions.ApiKey := 'API_KEY';

// Load PDF file and encode to base64
vBase64 := sgcBase64Encode(LoadFileToBytes('document.pdf'));
WriteLn(Anthropic._CreateDocumentMessage('claude-sonnet-4-20250514',
  'Summarize this document.', vBase64, 'application/pdf'));
```

C++ Builder

```
TsgcHTTP_API_Anthropic *Anthropic = new TsgcHTTP_API_Anthropic(NULL);
Anthropic->AnthropicOptions->ApiKey = "API_KEY";

// Load PDF file and encode to base64
AnsiString vBase64 = sgcBase64Encode(LoadFileToBytes("document.pdf"));
ShowMessage(Anthropic->_CreateDocumentMessage("claude-sonnet-4-20250514",
  "Summarize this document.", vBase64, "application/pdf"));
```

.NET (C#)

```
TsgcHTTP_API_Anthropic Anthropic = new TsgcHTTP_API_Anthropic();
Anthropic.AnthropicOptions.ApiKey = "API_KEY";

// Load PDF file and encode to base64
string vBase64 = sgcBase64Encode(LoadFileToBytes("document.pdf"));
MessageBox.Show(Anthropic._CreateDocumentMessage("claude-sonnet-4-20250514",
  "Summarize this document.", vBase64, "application/pdf"));
```

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 · Anthropic Extended Thinking — Simple Example

Send a message with extended thinking enabled using the convenience method. The temperature is automatically set to 1.0 (required by the API when thinking is enabled).

Delphi (VCL / FireMonkey)

```
Anthropic := TsgcHTTP_API_Anthropic.Create(nil);
Anthropic.AnthropicOptions.ApiKey := 'API_KEY';
WriteLn(Anthropic._CreateMessageWithThinking('claude-sonnet-4-20250514',
  'How many r''s are in the word strawberry?', 10000));
```

C++ Builder

```
TsgcHTTP_API_Anthropic *Anthropic = new TsgcHTTP_API_Anthropic(NULL);
Anthropic->AnthropicOptions->ApiKey = "API_KEY";
ShowMessage(Anthropic->_CreateMessageWithThinking("claude-sonnet-4-20250514",
  "How many r's are in the word strawberry?", 10000));
```

.NET (C#)

```
TsgcHTTP_API_Anthropic Anthropic = new TsgcHTTP_API_Anthropic();
Anthropic.AnthropicOptions.ApiKey = "API_KEY";
MessageBox.Show(Anthropic._CreateMessageWithThinking("claude-sonnet-4-20250514",
  "How many r's are in the word strawberry?", 10000));
```

2 · Anthropic MCP Connector — Simple Example

Use the convenience method to create a message with an MCP server.

Delphi (VCL / FireMonkey)

```
Anthropic := TsgcHTTP_API_Anthropic.Create(nil);
Anthropic.AnthropicOptions.ApiKey := 'API_KEY';
Anthropic.AnthropicOptions.BetaHeaders := 'mcp-client-2025-11-20';

WriteLn(Anthropic._CreateMessageWithMCP('claude-sonnet-4-20250514',
  'What tools are available?',
  'https://my-mcp-server.example.com/sse',
  'my-mcp-server'));
```

C++ Builder

```
TsgcHTTP_API_Anthropic *Anthropic = new TsgcHTTP_API_Anthropic(NULL);
Anthropic->AnthropicOptions->ApiKey = "API_KEY";
Anthropic->AnthropicOptions->BetaHeaders = "mcp-client-2025-11-20";

ShowMessage(Anthropic->_CreateMessageWithMCP("claude-sonnet-4-20250514",
  "What tools are available?",
  "https://my-mcp-server.example.com/sse",
  "my-mcp-server"));
```

.NET (C#)

```
TsgcHTTP_API_Anthropic Anthropic = new TsgcHTTP_API_Anthropic();
Anthropic.AnthropicOptions.ApiKey = "API_KEY";
Anthropic.AnthropicOptions.BetaHeaders = "mcp-client-2025-11-20";

MessageBox.Show(Anthropic._CreateMessageWithMCP("claude-sonnet-4-20250514",
  "What tools are available?",
  "https://my-mcp-server.example.com/sse",
  "my-mcp-server"));
```

3 · Anthropic Messages — Simple Example

Send a Hello message to Claude.

Delphi (VCL / FireMonkey)

```
Anthropic := TsgcHTTP_API_Anthropic.Create(nil);
Anthropic.AnthropicOptions.ApiKey := 'API_KEY';
WriteLn(Anthropic._CreateMessage('claude-sonnet-4-20250514', 'Hello!'));
```

C++ Builder

```
TsgcHTTP_API_Anthropic *Anthropic = new TsgcHTTP_API_Anthropic(NULL);
Anthropic->AnthropicOptions->ApiKey = "API_KEY";
ShowMessage(Anthropic->_CreateMessage("claude-sonnet-4-20250514", "Hello!"));
```

.NET (C#)

```
TsgcHTTP_API_Anthropic Anthropic = new TsgcHTTP_API_Anthropic();
Anthropic.AnthropicOptions.ApiKey = "API_KEY";
MessageBox.Show(Anthropic._CreateMessage("claude-sonnet-4-20250514", "Hello!"));
```

4 · Anthropic Structured Outputs — Simple Example

Use the convenience method to create a message with JSON schema output.

Delphi (VCL / FireMonkey)

```
Anthropic := TsgcHTTP_API_Anthropic.Create(nil);
Anthropic.AnthropicOptions.ApiKey := 'API_KEY';

vSchema := '{"type":"object","properties":{"name":{"type":"string"},' +
  '"age":{"type":"integer"}},"required":["name","age"],' +
  '"additionalProperties":false}';

WriteLn(Anthropic._CreateMessageJSON('claude-sonnet-4-20250514',
  'Extract the name and age: John is 30 years old.', vSchema));
```

C++ Builder

```
TsgcHTTP_API_Anthropic *Anthropic = new TsgcHTTP_API_Anthropic(NULL);
Anthropic->AnthropicOptions->ApiKey = "API_KEY";

String vSchema = "{\"type\":\"object\",\"properties\":{\"name\":{\"type\":\"string\"},"
  "\"age\":{\"type\":\"integer\"}},\"required\":[\"name\",\"age\"],"
  "\"additionalProperties\":false}";

ShowMessage(Anthropic->_CreateMessageJSON("claude-sonnet-4-20250514",
  "Extract the name and age: John is 30 years old.", vSchema));
```

.NET (C#)

```
TsgcHTTP_API_Anthropic Anthropic = new TsgcHTTP_API_Anthropic();
Anthropic.AnthropicOptions.ApiKey = "API_KEY";

string vSchema = "{\"type\":\"object\",\"properties\":{\"name\":{\"type\":\"string\"}," +
  "\"age\":{\"type\":\"integer\"}},\"required\":[\"name\",\"age\"]," +
  "\"additionalProperties\":false}";

MessageBox.Show(Anthropic._CreateMessageJSON("claude-sonnet-4-20250514",
  "Extract the name and age: John is 30 years old.", vSchema));
```

5 · Anthropic Vision — Simple Example

Send an image with a prompt asking Claude to describe it.

```
Delphi (VCL / FireMonkey)
```

```
Anthropic := TsgcHTTP_API_Anthropic.Create(nil);
Anthropic.AnthropicOptions.ApiKey := 'API_KEY';

// Load image and encode to base64
oStream := TFileStream.Create('photo.png', fmOpenRead);
Try
  oBytes := TBytesStream.Create;
  Try
    oBytes.CopyFrom(oStream, 0);
    vBase64 := EncodeBase64(oBytes.Memory, oBytes.Size);
  Finally
    oBytes.Free;
  End;
Finally
  oStream.Free;
End;

WriteLn(Anthropic._CreateVisionMessage('claude-sonnet-4-20250514',
  'What is in this image?', vBase64, 'image/png'));
```

```
C++ Builder
```

```
TsgcHTTP_API_Anthropic *Anthropic = new TsgcHTTP_API_Anthropic(NULL);
Anthropic->AnthropicOptions->ApiKey = "API_KEY";

// Load image and encode to base64
TFileStream *oStream = new TFileStream("photo.png", fmOpenRead);
try {
  TBytesStream *oBytes = new TBytesStream();
  try {
    oBytes->CopyFrom(oStream, 0);
    vBase64 = EncodeBase64(oBytes->Memory, oBytes->Size);
  } __finally {
    oBytes->Free();
  }
} __finally {
  oStream->Free();
}

ShowMessage(Anthropic->_CreateVisionMessage("claude-sonnet-4-20250514",
  "What is in this image?", vBase64, "image/png"));
```

.NET (C#)

```
TsgcHTTP_API_Anthropic Anthropic = new TsgcHTTP_API_Anthropic();
Anthropic.AnthropicOptions.ApiKey = "API_KEY";

// Load image and encode to base64
byte[] fileBytes = System.IO.File.ReadAllBytes("photo.png");
string vBase64 = Convert.ToBase64String(fileBytes);

MessageBox.Show(Anthropic._CreateVisionMessage("claude-sonnet-4-20250514",
  "What is in this image?", vBase64, "image/png"));
```

6 · Anthropic Web Search — Simple Example

Use the convenience method to create a message with web search enabled.

Delphi (VCL / FireMonkey)

```
Anthropic := TsgcHTTP_API_Anthropic.Create(nil);
Anthropic.AnthropicOptions.ApiKey := 'API_KEY';
WriteLn(Anthropic._CreateMessageWithWebSearch('claude-sonnet-4-20250514',
  'What are the latest news about Delphi programming?'));
```

C++ Builder

```
TsgcHTTP_API_Anthropic *Anthropic = new TsgcHTTP_API_Anthropic(NULL);
Anthropic->AnthropicOptions->ApiKey = "API_KEY";
ShowMessage(Anthropic->_CreateMessageWithWebSearch("claude-sonnet-4-20250514",
  "What are the latest news about Delphi programming?"));
```

.NET (C#)

```
TsgcHTTP_API_Anthropic Anthropic = new TsgcHTTP_API_Anthropic();
Anthropic.AnthropicOptions.ApiKey = "API_KEY";
MessageBox.Show(Anthropic._CreateMessageWithWebSearch("claude-sonnet-4-20250514",
  "What are the latest news about Delphi programming?"));
```

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 — Anthropic API — Getting started	docs.anthropic.com/en/api/getting-started
Primary standard / spec — Anthropic API — Messages	docs.anthropic.com/en/api/messages
Online help — component page	www.esegece.com/help/sgcWebSockets/Components/AI/Anthropic/TsgcHTTP_API_Anthropic.htm
Delphi demo project (in the sgcWebSockets package)	`Demos\15.AI\01.QuickStart\07.Anthropic`
.NET demo project (in the sgcWebSockets package)	`.net\demos\15.AI\01.QuickStart\07.Anthropic`
Component page	www.esegece.com/products/websockets/ai/anthropic/
Product page	www.esegece.com/products/websockets/

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