SGCWEBSOCKETS · TECHNICAL DOCUMENT

Google Calendar API

Google Calendar v3 API client — events, calendars, ACLs and notifications with OAuth 2.0 from Delphi.

eSeGeCe

WWW.ESEGECE.COM

Overview

The Google Calendar API lets you integrate your app with Google Calendar, creating new ways for you to engage your users. The Calendar API lets you display, create and modify calendar events as well as work with many other calendar-related objects, such as calendars or access controls.

At a glance

COMPONENT CLASS

`TsgcHTTPGoogleCloud_Calendar_Client`

STANDARDS / SPEC

Google Calendar API v3 reference

TRANSPORTS

TCP, TLS

PLATFORMS

Windows, macOS, Linux, iOS, Android

FRAMEWORKS

VCL, FireMonkey, Lazarus / FPC

EDITION

Standard / Professional / Enterprise

Features

- Native Delphi implementation with full ANSI/Unicode support.

Technical specification

Standards & specs	Google Calendar API v3 reference
Component class	`TsgcHTTPGoogleCloud_Calendar_Client` (unit `sgcHTTP_GoogleCloud_Calendar_Client`)
Frameworks	VCL, FireMonkey, Lazarus / FPC
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`	Published or public property used to configure or query the component.
`OnAuthToken`	Published or public property used to configure or query the component.
`OnAuthTokenError`	Published or public property used to configure or query the component.
`GoogleCloudOptions`	Published or public property used to configure or query the component.
`LogFile`	Published or public property used to configure or query the component.
`Scopes`	Published or public property used to configure or query the component.
`OnGetCalendar`	Published or public property used to configure or query the component.
`OnGetCalendarEvent`	Published or public property used to configure or query the component.
`OnError`	Published or public property used to configure or query the component.
`Calendars`	Published or public property used to configure or query the component.

Main methods

The principal public methods exposed by the component.

`ACL_Get()`	Public function exposed by the component.

`CalendarList_Delete()`	Public function exposed by the component.
`CalendarList_Get()`	Public function exposed by the component.
`CalendarList_Patch()`	Public function exposed by the component.
`Calendar_Get()`	Public function exposed by the component.
`Color_Get()`	Public function exposed by the component.
`Event_Get()`	Public function exposed by the component.
`DeleteCalendar()`	Public function exposed by the component.
`DeleteEvent()`	Public function exposed by the component.
`ACL_Delete()`	Public function exposed by the component.

Quick Start

Drop the component on a form, configure the properties below and activate it. The snippet that follows shows the typical **Google Calendar | Load Calendars** configuration sourced from the online help.

> **About this scenario.** The process to get all calendars from your account is very easy. Just follow the next steps:

Delphi (VCL / FireMonkey)

```
oGoogleCalendar := TsgcHTTPGoogleCloud_Calendar_Client.Create(nil);
// ... configure OAuth2 options
oGoogleCalendar.GoogleCloudOptions.OAuth2.ClientId := 'google ClientId';
oGoogleCalendar.GoogleCloudOptions.OAuth2.ClientSecret := 'google ClientSecret';
// ... request calendars
if oGoogleCalendar.LoadCalendars then
begin
// ... get calendars data
for i := 0 to oGoogleCalendar.Calendars.Count - 1 do
vCalendarTitle := oGoogleCalendar.Calendars.Calendar[i].Summary;
end
else
raise Exception.Create('Error Calendar Sync');
```

C++ Builder

```
TsgcHTTPGoogleCloud_Calendar_Client oGoogleCalendar = new TsgcHTTPGoogleCloud_Calendar_Client(NU
// ... configure OAuth2 options
oGoogleCalendar->GoogleCloudOptions->OAuth2->ClientId = "google ClientId";
oGoogleCalendar->GoogleCloudOptions->OAuth2->ClientSecret = "google ClientSecret";
// ... request calendars
if (oGoogleCalendar->LoadCalendars)
{
// ... get calendars data
for (int i = 0; i < oGoogleCalendar->Calendars->Count; i++)
{
vCalendarTitle = oGoogleCalendar->Calendars->Calendar[i]->Summary;
}
}
else
{
throw Exception("Error Calendar Sync");
}
```

.NET (C#)

```
TsgcHTTPGoogleCloud_Calendar_Client oGoogleCalendar = new TsgcHTTPGoogleCloud_Calendar_Client();
// ... configure OAuth2 options
oGoogleCalendar.GoogleCloudOptions.OAuth2.ClientId = "google ClientId";
oGoogleCalendar.GoogleCloudOptions.OAuth2.ClientSecret = "google ClientSecret";
// ... request calendars
if (oGoogleCalendar.LoadCalendars())
{
  // ... get calendars data
  for (int i = 0; i < oGoogleCalendar.Calendars.Count; i++)
  {
    string vCalendarTitle = oGoogleCalendar.Calendars.Calendar[i].Summary;
  }
}
else
{
  throw new Exception("Error Calendar Sync");
}
```

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 · Google Calendar | Sync Events

The process to get all events from a calendar is very easy. Just follow the next steps:

Delphi (VCL / FireMonkey)

```
oGoogleCalendar := TsgcHTTPGoogleCloud_Calendar_Client.Create(nil);
// ... configure OAuth2 options
oGoogleCalendar.GoogleCloudOptions.OAuth2.ClientId := 'google ClientId';
oGoogleCalendar.GoogleCloudOptions.OAuth2.ClientSecret := 'google ClientSecret';
// ... request calendars first
oGoogleCalendar.LoadCalendars;
// ... request events from first calendar
oCalendar := TsgcGoogleCalendarItem(oGoogleCalendar.Calendars.Calendar[0]);
if oGoogleCalendar.LoadEvents(oCalendar.ID) then
begin
// ... get events data
for i := 0 to oCalendar.Events.Count - 1 do
vEventTitle := oCalendar.Events[i].Summary;
end
else
raise Exception.Create('Error Event Sync');
```

C++ Builder

```
TsgcHTTPGoogleCloud_Calendar_Client oGoogleCalendar = new TsgcHTTPGoogleCloud_Calendar_Client(NU
// ... configure OAuth2 options
oGoogleCalendar→GoogleCloudOptions→OAuth2→ClientId = "google ClientId";
oGoogleCalendar→GoogleCloudOptions→OAuth2→ClientSecret = "google ClientSecret";
// ... request calendars first;
oGoogleCalendar→LoadCalendars;
// ... request events from first calendar
oCalendar = TsgcGoogleCalendarItem(oGoogleCalendar→Calendars→Calendar[0]);
if (oGoogleCalendar→LoadEvents(oCalendar→ID))
{
// ... get events data
for (int i = 0; i < oCalendar→Events→Count; i++)
{
vEventTitle = oCalendar→Events[i]→Summary;
}
}
else
{
throw Exception("Error Event Sync");
}
```

.NET (C#)

```
TsgcHTTPGoogleCloud_Calendar_Client oGoogleCalendar = new TsgcHTTPGoogleCloud_Calendar_Client();
// ... configure OAuth2 options
oGoogleCalendar.GoogleCloudOptions.OAuth2.ClientId = "google ClientId";
oGoogleCalendar.GoogleCloudOptions.OAuth2.ClientSecret = "google ClientSecret";
// ... request calendars first
oGoogleCalendar.LoadCalendars();
// ... request events from first calendar
TsgcGoogleCalendarItem oCalendar = (TsgcGoogleCalendarItem)oGoogleCalendar.Calendars.Calendar[0]
if (oGoogleCalendar.LoadEvents(oCalendar.ID))
{
  // ... get events data
  for (int i = 0; i < oCalendar.Events.Count; i++)
  {
    string vEventTitle = oCalendar.Events[i].Summary;
  }
}
else
{
  throw new Exception("Error Event Sync");
}
```

2 · Google Calendar — Configuration

Google Calendar component client has the following properties:

Delphi (VCL / FireMonkey)

```
oCalendar := TsgcHTTPGoogleCloud_Calendar_Client.Create(nil);
oCalendar.TLSOptions.IOHandler := iohOpenSSL;
oCalendar.TLSOptions.Version := tls1_3;
oCalendar.TLSOptions.VerifyCertificate := True;
oCalendar.TLSOptions.OpenSSL_Options.LibPath := oslpDefaultFolder;
```

C++ Builder

```
TsgcHTTPGoogleCloud_Calendar_Client *oCalendar = new TsgcHTTPGoogleCloud_Calendar_Client(NULL);
oCalendar->TLSOptions->IOHandler = iohOpenSSL;
oCalendar->TLSOptions->Version = tls1_3;
oCalendar->TLSOptions->VerifyCertificate = true;
oCalendar->TLSOptions->OpenSSL_Options->LibPath = oslpDefaultFolder;
```

.NET (C#)

```
oCalendar = new TsgcHTTPGoogleCloud_Calendar_Client();
oCalendar.TLSOptions.IOHandler = TwsTLSIOHandler.iohOpenSSL;
oCalendar.TLSOptions.Version = TwsTLSVersions.tls1_3;
oCalendar.TLSOptions.VerifyCertificate = true;
oCalendar.TLSOptions.OpenSSL_Options.LibPath = oslpDefaultFolder;
```

3 · Using RefreshToken

The first time you authenticate, use the OnAuthToken event to save the RefreshToken if it exists. You can save it in an INI file, for example:

Delphi (VCL / FireMonkey)

```
procedure OnGoogleCalendarAuthToken(Sender: TObject; const TokenType, Token, Data: string);
var
  oINI: TINIFile;
  oJSON: TsgcJSON;
begin
  oJSON := TsgcJSON.Create(nil);
  Try
    oJSON.Read(Data);
    if oJSON.Node['refresh_token']  nil then
    begin
      oINI := TINIFile.Create(ChangeFileExt(Application.ExeName, '.ini'));
      Try
        oINI.WriteString('OAUTH2', 'Token', oJSON.Node['refresh_token'].Value);
      Finally
        oINI.Free;
      End;
    end;
  Finally
    oJSON.Free;
  End;
end;
```

C++ Builder

```
void OnGoogleCalendarAuthToken(TObject *Sender, string TokenType, string Token, string Data)
{
  TsgcJSON *oJSON = new TsgcJSON();
  try
  {
    oJSON->Read(Data);
    if (oJSON->Node["refresh_token"] != NULL)
    {
      TINIFile *oINI = new TINIFile(ChangeFileExt(Application->ExeName, ".ini"));
      try
      {
        oINI->WriteString("OAUTH2", "Token", oJSON->Node["refresh_token"]->Value);
      }
      __finally
      {
        oINI->Free();
      }
    }
  }
  __finally
  {
    oJSON->Free();
  }
}
```

.NET (C#)

```
void OnGoogleCalendarAuthToken(object sender, string TokenType, string Token, string Data)
{
  TsgcJSON oJSON = new TsgcJSON();
  oJSON.Read(Data);
  if (oJSON.Node["refresh_token"] =/= null)
  {
    // Save refresh token to your preferred storage
    string refreshToken = oJSON.Node["refresh_token"].Value;
    System.IO.File.WriteAllText("refresh_token.txt", refreshToken);
  }
}
```

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 — Google Calendar API v3 reference	developers.google.com/calendar/api/v3/reference
Online help — component page	www.esegece.com/help/sgcWebSockets/Components/HTTP/Google/Calendar/Google_Calendar.htm
Delphi demo project (in the sgcWebSockets package)	`Demos\20.HTTP_Protocol\03.Google\02.Google_Calendar`
Component page	www.esegece.com/products/websockets/http/google-calendar/
Product page	www.esegece.com/products/websockets/

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