SGCWEBSOCKETS · TECHNICAL DOCUMENT

Telegram API

Telegram TDLib (full client) and Bot API REST client — chat, message, file and bot endpoints for Delphi.

eSeGeCe

WWW.ESEGECE.COM

Overview

Telegram TDLib (full client) and Bot API REST client — chat, message, file and bot endpoints for Delphi. The component is part of the sgcWebSockets library.

At a glance

COMPONENT CLASS
`TsgcTDLib_Telegram`

STANDARDS / SPEC
Telegram Bot API

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	Telegram Bot API · TDLib documentation
Component class	`TsgcTDLib_Telegram` (unit `sgcTDLib_Telegram`)
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.

`OnAuthenticationPassword`	Published or public property used to configure or query the component.
`OnAuthorizationStatus`	Published or public property used to configure or query the component.
`OnAuthenticationCode`	Published or public property used to configure or query the component.
`Client`	Published or public property used to configure or query the component.
`Active`	Published or public property used to configure or query the component.
`OnEvent`	Published or public property used to configure or query the component.
`OnBeforeReadEvent`	Published or public property used to configure or query the component.
`OnConnectionStatus`	Published or public property used to configure or query the component.
`OnRegisterUser`	Published or public property used to configure or query the component.
`OnMessageText`	Published or public property used to configure or query the component.

Main methods

The principal public methods exposed by the component.

`getChatSponsoredMessage()`	Public procedure exposed by the component.
`CheckAuthenticationBotToken()`	Public procedure exposed by the component.

`DeleteSavedOrderInfo()`	Public procedure exposed by the component.
`EditInlineMessageText()`	Public procedure exposed by the component.
`EditMessageText()`	Public procedure exposed by the component.
`SearchCallMessages()`	Public procedure exposed by the component.
`SendChatSetTtlMessage()`	Public procedure exposed by the component.
`TestGetDifference()`	Public procedure exposed by the component.
`WriteGeneratedFilePart()`	Public procedure exposed by the component.
`TDLibSend()`	Public procedure 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 **Telegram | Send Bot Message With Buttons** configuration sourced from the online help.

> **About this scenario.** Telegram API allows you to send messages with buttons to request data from the user (this option is only available for bots).

Delphi (VCL / FireMonkey)

```
oReplyMarkup := TsgcTelegramReplyMarkupShowKeyboard.Create;
Try
oReplyMarkup.AddButtonTypeRequestPhoneNumber('Give me your phone');
sgcTelegram.SendTextMessage('123456', 'Please provide the information below', nil, oReplyMarkup)
Finally
oReplyMarkup.Free;
End;
```

C++ Builder

```
oReplyMarkup = new TsgcTelegramReplyMarkupShowKeyboard();
oReplyMarkup->AddButtonTypeRequestPhoneNumber("Give me your phone");
sgcTelegram->SendTextMessage("123456", "Please provide the information below", null, oReplyMarku
oReplyMarkup->Free();
```

.NET (C#)

```
oReplyMarkup = new TsgcTelegramReplyMarkupShowKeyboard();
oReplyMarkup.AddButtonTypeRequestPhoneNumber("Give me your phone");
sgcTelegram.SendTextMessage("123456", "Please provide the information below", null, oReplyMarkup
```

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 · Telegram | Send Telegram Invoice Message

If your bot supports inline mode, users can also send invoices to other chats via the bot, including to one-on-one chats with other users.

Delphi (VCL / FireMonkey)

```
procedure SendInvoice;
var
  oInvoice: TsgcTelegramSendInvoice;
begin
  oInvoice := TsgcTelegramSendInvoice.Create;
  Try
    oInvoice.Title := 'Invoice Title Test';
    oInvoice.Description := 'Description Invoice Test';
    oInvoice.Invoice.Currency := 'EUR';
    oInvoice.Invoice.Total := 800;
    oInvoice.Invoice.IsTest := True;
        oInvoice.Invoice.Payload := 'payload';
        oInvoice.Invoice.ProviderToken := 'provider_token';
        oInvoice.Invoice.ProviderData := 'provider_data';

    sgcTelegram.SendInvoiceMessage('3284239872', oInvoice);
  Finally
    oInvoice.Free;
  End;
end;
```

C++ Builder

```
private void SendInvoice()
{
  TsgcTelegramSendInvoice *oInvoice = new TsgcTelegramSendInvoice();
  Try
  {
    oInvoice->Title = "Invoice Title Test";
    oInvoice->Description = "Description Invoice Test";
    oInvoice->Invoice->Currency = 'EUR';
    oInvoice->Invoice->Total = 800;
    oInvoice->Invoice->IsTest = True;
        oInvoice->Invoice->Payload := "payload";
        oInvoice->Invoice->ProviderToken := "provider_token";
        oInvoice->Invoice->ProviderData := "provider_data";

    sgcTelegram->SendInvoiceMessage("3284239872", oInvoice);
  __finally
  {
    oInvoice->Free();
  }
}
```

.NET (C#)

```
private void SendInvoice()
{
  TsgcTelegramSendInvoice oInvoice = new TsgcTelegramSendInvoice();
  oInvoice.Title = 'Invoice Title Test';
  oInvoice.Description = 'Description Invoice Test';
  oInvoice.Invoice.Currency = 'EUR';
  oInvoice.Invoice.Total = 800;
  oInvoice.Invoice.IsTest = True;
  oInvoice.Invoice.Payload := "payload";
  oInvoice.Invoice.ProviderToken := "provider_token";
  oInvoice.Invoice.ProviderData := "provider_data";

  sgcTelegram.SendInvoiceMessage("3284239872", oInvoice);
}
```

2 · Telegram | Send Telegram Message With Inline Buttons

Telegram API allows you to send messages with inline buttons to select options as an answer (this option is only available for bots).

Delphi (VCL / FireMonkey)

```
oReplyMarkup := TsgcTelegramReplyMarkupInlineKeyboard.Create;
Try
  oReplyMarkup.AddButtonTypeCallback('Yes', 'I like it');
  oReplyMarkup.AddButtonTypeCallback('No', 'I hate it');
  oReplyMarkup.AddButtonTypeUrl('Poll', 'https://www.yoursite.com/telegram/poll');
  sgcTelegram.SendTextMessage('123456', 'Do you like the message?', oReplyMarkup);
Finally
  oReplyMarkup.Free;
End;

      procedure OnNewCallbackQuery(Sender: TObject; CallbackQuery: TsgcTelegramCallbackQuery);
begin
  if CallbackQuery.PayloadData.Data = 'I like it' then
    ShowMessage('yes')
  else
    ShowMessage('no');
end;
```

C++ Builder

```
TsgcTelegramReplyMarkupInlineKeyboard *oReplyMarkup = new TsgcTelegramReplyMarkupInlineKeyboard(
try
{
  oReplyMarkup->AddButtonTypeCallback("Yes", "I like it");
  oReplyMarkup->AddButtonTypeCallback("No", "I hate it");
  oReplyMarkup->AddButtonTypeUrl("Poll", "https://www.yoursite.com/telegram/poll");
  sgcTelegram->SendTextMessage("123456", "Do you like the message?", oReplyMarkup);
}
__finally
{
  oReplyMarkup->Free();
}

void OnNewCallbackQuery(TObject *Sender, TsgcTelegramCallbackQuery *CallbackQuery)
{
  if (CallbackQuery->PayloadData->Data == "I like it") then
  {
    ShowMessage("yes")
  }
  else
  {
    ShowMessage("no");
  }
}
```

.NET (C#)

```
TsgcTelegramReplyMarkupInlineKeyboard oReplyMarkup = new TsgcTelegramReplyMarkupInlineKeyboard()
oReplyMarkup.AddButtonTypeCallback("Yes", "I like it");
oReplyMarkup.AddButtonTypeCallback("No", "I hate it");
oReplyMarkup.AddButtonTypeUrl("Poll", "https://www.yoursite.com/telegram/poll");
sgcTelegram.SendTextMessage("123456", "Do you like the message?", oReplyMarkup);

void OnNewCallbackQuery(TObject Sender, TsgcTelegramCallbackQuery CallbackQuery)
{
  if (CallbackQuery.PayloadData.Data == "I like it") then
  {
    MessageBox.Show("yes")
  }
  else
  {
    MessageBox.Show("no");
  }
}
```

3 · Add Proxy

In order to configure a HTTP Proxy, first you must add the proxy to telegram configuration, to do this, just call AddProxyHTTP and if successful, a message will be returned with the new proxy added. Once the proxy has been added to the list, just call EnableProxy and pass the ID of the proxy received on the confirmation message.

Delphi (VCL / FireMonkey)

```
Telegram.AddProxyHTTP('8.8.8.8', 8080, '', '', True);
// ... read the confirmation message and save the ID of the proxy.
Telegram.EnableProxy(2);
```

C++ Builder

```
Telegram->AddProxyHTTP("8.8.8.8", 8080, "", "", true);
// ... read the confirmation message and save the ID of the proxy.
Telegram->EnableProxy(2);
```

.NET (C#)

```
Telegram.AddProxyHTTP("8.8.8.8", 8080, "", "", true);
// ... read the confirmation message and save the ID of the proxy.
Telegram.EnableProxy(2);
```

4 · Creating your Telegram Application

In order to obtain an API id and develop your own application using the Telegram API you need to do the following:

Delphi (VCL / FireMonkey)

```
oTelegram := TsgcTDLib_Telegram.Create(nil);
oTelegram.Telegram.API.ApiHash := 'your api hash';
oTelegram.Telegram.API.ApiId := 'your api id';
oTelegram.PhoneNumber := 'your phone number';
oTelegram.ApplicationVersion := '1.0';
oTelegram.DeviceModel := 'Desktop';
oTelegram.LanguageCode := 'en';
oTelegram.SystemVersion := 'Windows';
oTelegram.Active := true;
```

C++ Builder

```
TsgcTDLib_Telegram oTelegram = new TsgcTDLib_Telegram();
oTelegram->Telegram->API->ApiHash = "your api hash";
oTelegram->Telegram->API->ApiId = "your api id";
oTelegram->PhoneNumber = "your phone number";
oTelegram->ApplicationVersion = "1.0";
oTelegram->DeviceModel = "Desktop";
oTelegram->LanguageCode = "en";
oTelegram->SystemVersion = "Windows";
oTelegram->Active = true;
```

.NET (C#)

```
TsgcTDLib_Telegram oTelegram = new TsgcTDLib_Telegram();
oTelegram.Telegram.API.ApiHash = "your api hash";
oTelegram.Telegram.API.ApiId = "your api id";
oTelegram.PhoneNumber = "your phone number";
oTelegram.ApplicationVersion = "1.0";
oTelegram.DeviceModel = "Desktop";
oTelegram.LanguageCode = "en";
oTelegram.SystemVersion = "Windows";
oTelegram.Active = true;
```

5 · Get Sponsored Messages

Send a request to the channel asking if there are sponsored messages available, just call the method GetChatSponsoredMessage.

Delphi (VCL / FireMonkey)

```
oTelegram := TsgcTDLib_Telegram.Create(nil);
oTelegram.Telegram.API.ApiHash := 'ABCDEFGHIJKLMN';
oTelegram.Telegram.API.ApiId := '1234';
oTelegram.PhoneNumber := '008745744155';
oTelegram.Active := true;
oTelegram.getChatSponsoredMessage('100');
```

C++ Builder

```
TsgcTDLib_Telegram *oTelegram = new TsgcTDLib_Telegram();
oTelegram->Telegram->API->ApiHash = "ABCDEFGHIJKLMN";
oTelegram->Telegram->API->ApiId = "1234";
oTelegram->PhoneNumber = "008745744155";
oTelegram->Active = true;
oTelegram->getChatSponsoredMessage("100");
```

.NET (C#)

```
TsgcTDLib_Telegram oTelegram = new TsgcTDLib_Telegram();
oTelegram.Telegram.API.ApiHash = "ABCDEFGHIJKLMN";
oTelegram.Telegram.API.ApiId = "1234";
oTelegram.PhoneNumber = "008745744155";
oTelegram.Active = true;
oTelegram->getChatSponsoredMessage("100");
```

6 · Telegram | Get SuperGroup Members

Telegram API allows you to get information about members of a SuperGroup. Use the method GetSuperGroupMembers to get information about members or banned users in a supergroup or channel. Can be used only if SupergroupFullInfo.can_get_members is true; additionally, administrator privileges may be required for some filters.

Delphi (VCL / FireMonkey)

```
Telegram.GetSupergroupMembers(1452979380);
<br/>

procedure OnTelegramEvent(Sender: TObject; const Event, Text: string);
begin
if Event = 'chatMembers' then
ReadJSON(Text);
end;
```

C++ Builder

```
Telegram->GetSupergroupMembers(1452979380);
<br/>

private void OnTelegramEvent(TObject *Sender, const string Event, const string Text)
{
if (Event == "chatMembers")
{
ReadJSON(Text);
}
}
```

.NET (C#)

```
Telegram.GetSupergroupMembers(1452979380);
<br/>

private void OnTelegramEvent(TObject Sender, const string Event, const string Text)
{
if (Event == "chatMembers")
{
ReadJSON(Text);
}
}
```

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 — Telegram Bot API	core.telegram.org/bots/api
Primary standard / spec — TDLib documentation	core.telegram.org/tdlib
Delphi demo project (in the sgcWebSockets package)	`Demos\50.Other\01.Telegram_Client`
Component page	www.esegece.com/products/websockets/apis/telegram/
Product page	www.esegece.com/products/websockets/

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