SGCWEBSOCKETS · TECHNICAL DOCUMENT

Binance API

Binance spot and futures WebSocket and REST client — public streams, user data and authenticated trading.

eSeGeCe

WWW.ESEGECE.COM

Overview

Binance is an international multi-language cryptocurrency exchange. It offers some APIs to access Binance data. The following APIs are supported:

At a glance

COMPONENT CLASS `TsgcWSAPI_Binance`	STANDARDS / SPEC **Binance Spot API documentation**
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	Binance Spot API documentation · Binance USDT-M Futures API
Component class	`TsgcWSAPI_Binance` (unit `sgcWebSocket_API_Binance`)
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.

`Client`	Published or public property used to configure or query the component.
`OnBinanceHTTPException`	Published or public property used to configure or query the component.
`Binance`	Published or public property used to configure or query the component.
`RawMessages`	Published or public property used to configure or query the component.
`REST_API`	Published or public property used to configure or query the component.
`ListenKey`	Published or public property used to configure or query the component.
`Version`	Published or public property used to configure or query the component.

Main methods

The principal public methods exposed by the component.

`SubscribeAggregateTrades()`	Public function exposed by the component.
`UnSubscribeAggregateTrades()`	Public function exposed by the component.
`SubscribeTrades()`	Public function exposed by the component.
`UnSubscribeTrades()`	Public function exposed by the component.
`SubscribeKLine()`	Public function exposed by the component.

`UnSubscribeKLine()`	Public function exposed by the component.
`SubscribeMiniTicker()`	Public function exposed by the component.
`UnSubscribeMiniTicker()`	Public function exposed by the component.
`SubscribeAllMiniTickers()`	Public function exposed by the component.
`UnSubscribeAllMiniTickers()`	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 **Binance | Connect WebSocket API** configuration sourced from the online help.

> **About this scenario.** In order to connect to Binance WebSocket API, just create a new Binance API client and attach to TsgcWebSocketClient.

Delphi (VCL / FireMonkey)

```
oClient := TsgcWebSocketClient.Create(nil);
oBinance := TsgcWSAPI_Binance.Create(nil);
oBinance.Client := oClient;
oClient.Active := True;
```

C++ Builder

```
TsgcWebSocketClient *oClient = new TsgcWebSocketClient(NULL);
TsgcWSAPI_Binance *oBinance = new TsgcWSAPI_Binance(NULL);
oBinance->Client = oClient;
oClient->Active = true;
```

.NET (C#)

```
TsgcWebSocketClient oClient = new TsgcWebSocketClient();
TsgcWSAPI_Binance oBinance = new TsgcWSAPI_Binance();
oBinance.Client = oClient;
oClient.Active = true;
```

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 · Binance | Subscribe WebSocket Channel

Binance offers a variety of channels where you can subscribe to get real-time updates of market data, orders... Find below a sample of how to subscribe to a Ticker:

Delphi (VCL / FireMonkey)

```
oClient := TsgcWebSocketClient.Create(nil);
oBinance := TsgcWSAPI_Binance.Create(nil);
oBinance.Client := oClient;
oBinance.SubscribeTicker('bnbbtc');

procedure OnMessage(Connection: TsgcWSConnection; const aText: string);
begin
// here you will receive the ticker updates
end;
```

C++ Builder

```
TsgcWebSocketClient *oClient = new TsgcWebSocketClient(NULL);
TsgcWSAPI_Binance *oBinance = new TsgcWSAPI_Binance(NULL);
oBinance->Client = oClient;
oBinance->SubscribeTicker("bnbbtc");

void OnMessage(TsgcWSConnection *Connection, const string aText)
{
// here you will receive the ticker updates
}
```

.NET (C#)

```
TsgcWebSocketClient oClient = new TsgcWebSocketClient();
TsgcWSAPI_Binance oBinance = new TsgcWSAPI_Binance();
oBinance.Client = oClient;
oBinance.SubscribeTicker("bnbbtc");

void OnMessage(TsgcWSConnection Connection, const string aText)
{
// here you will receive the ticker updates
}
```

2 · Place an Order

To place a new order, just call the method REST_API.NewOrder of the Binance Client Component.

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

```
oBinance := TsgcWSAPI_Binance.Create(nil);
oBinance.Binance.ApiKey := '<api key>';
oBinance.Binance.ApiSecret := '<api secret>';
ShowMessage(oBinance.REST_API.NewOrder('BNBBTC', 'BUY', 'MARKET', '', 1));
```

```
C++ Builder
```

```
TsgcWSAPI_Binance *oBinance = new TsgcWSAPI_Binance(this);
oBinance->Binance->ApiKey = "<api key>";
oBinance->Binance->ApiSecret = "<api secret>";
ShowMessage(oBinance->REST_API->NewOrder("BNBBTC", "BUY", "MARKET", "", 1));
```

```
.NET (C#)
```

```
TsgcWSAPI_Binance oBinance = new TsgcWSAPI_Binance();
oBinance.Binance.ApiKey = "<api key>";
oBinance.Binance.ApiSecret = "<api secret>";
MessageBox.Show(oBinance.REST_API.NewOrder("BNBBTC", "BUY", "MARKET", "", 1));
```

3 · Binance | Get Market Data

Binance offers public Market Data through REST Endpoints, when you call one of these endpoints, you will get a snapshot of the market data requested.

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

```
oBinance := TsgcWSAPI_Binance.Create(nil);
ShowMessage(oBinance.REST_API.GetPriceTicker('BNBBTC'));
```

C++ Builder

```
TsgcWSAPI_Binance *oBinance = new TsgcWSAPI_Binance(this);
ShowMessage(oBinance->REST_API->GetPriceTicker("BNBBTC"));
```

.NET (C#)

```
TsgcWSAPI_Binance oBinance = new TsgcWSAPI_Binance();
MessageBox.Show(oBinance.REST_API.GetPriceTicker("BNBBTC"));
```

4 · Binance | Private REST API

The Binance REST API offers public and private endpoints. The Private endpoints require that messages are signed to increase the security of transactions.

Delphi (VCL / FireMonkey)

```
oBinance := TsgcWSAPI_Binance.Create(nil);
oBinance.Binance.ApiKey := '<your api key>';
oBinance.Binance.ApiSecret := '<your api secret>';
ShowMessage(oBinance.REST_API.GetAccountInformation);
```

C++ Builder

```
TsgcWSAPI_Binance *oBinance = new TsgcWSAPI_Binance(this);
oBinance->Binance->ApiKey = "<your api key>";
oBinance->Binance->ApiSecret = "<your api secret>";
ShowMessage(oBinance->REST_API->GetAccountInformation());
```

.NET (C#)

```
TsgcWSAPI_Binance oBinance = new TsgcWSAPI_Binance();
oBinance.Binance.ApiKey = "<your api key>";
oBinance.Binance.ApiSecret = "<your api secret>";
MessageBox.Show(oBinance.REST_API.GetAccountInformation());
```

5 · Binance | Withdraw

Binance allows you to use the Wallet API to submit a Withdraw request, only the following parameters are mandatory:

Delphi (VCL / FireMonkey)

```
oBinance := TsgcWSAPI_Binance.Create(nil);
oBinance.Binance.ApiKey := '<your api key>';
oBinance.Binance.ApiSecret := '<your api secret>';
ShowMessage(oBinance.REST_API.WalletWithdraw('BTC', '7213fea8e94b4a5593d507237e5a555b', 0.25));
```

C++ Builder

```
TsgcWSAPI_Binance *oBinance = new TsgcWSAPI_Binance(this);
oBinance->Binance->ApiKey = "<your api key>";
oBinance->Binance->ApiSecret = "<your api secret>";
ShowMessage(oBinance->REST_API->WalletWithdraw("BTC", "7213fea8e94b4a5593d507237e5a555b", 0.25))
```

.NET (C#)

```
TsgcWSAPI_Binance oBinance = new TsgcWSAPI_Binance();
oBinance.Binance.ApiKey = "<your api key>";
oBinance.Binance.ApiSecret = "<your api secret>";
MessageBox.Show(oBinance.REST_API.WalletWithdraw("BTC", "7213fea8e94b4a5593d507237e5a555b", 0.25
```

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 — Binance Spot API documentation	developers.binance.com/docs/binance-spot-api-docs
Primary standard / spec — Binance USDT-M Futures API	developers.binance.com/docs/derivatives/usds-margined-futures/general-info
Online help — component page	www.esegece.com/help/sgcWebSockets/Components/APIs/API/API_Binance.htm
Delphi demo project (in the sgcWebSockets package)	`Demos\05.Crypto\01.Binance`
.NET demo project (in the sgcWebSockets package)	`.net\demos\05.Crypto\01.Binance`
Component page	www.esegece.com/products/websockets/apis/binance/
Product page	www.esegece.com/products/websockets/

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