SGCWEBSOCKETS · TECHNICAL DOCUMENT

Coinbase Advanced Trade API

Coinbase Advanced Trade WebSocket and REST client — public market data plus authenticated trading on Coinbase Exchange.

eSeGeCe

WWW.ESEGECE.COM

Overview

The WebSocket feed is publicly available and provides real-time market data updates for orders and trades. Two endpoints are supported in production:

At a glance

COMPONENT CLASS `TsgcWSAPI_Coinbase`	STANDARDS / SPEC Coinbase Advanced Trade API
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	Coinbase Advanced Trade API
Component class	`TsgcWSAPI_Coinbase` (unit `sgcWebSocket_API_Coinbase`)
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.
`OnCoinbaseSubscriptions`	Published or public property used to configure or query the component.
`OnCoinbaseHeartbeats`	Published or public property used to configure or query the component.
`OnConnect`	Published or public property used to configure or query the component.
`OnCoinbaseMessage`	Published or public property used to configure or query the component.
`OnCoinbaseError`	Published or public property used to configure or query the component.
`OnCoinbaseHTTPException`	Published or public property used to configure or query the component.
`OnDisconnect`	Published or public property used to configure or query the component.
`Coinbase`	Published or public property used to configure or query the component.
`RawMessages`	Published or public property used to configure or query the component.

Main methods

The principal public methods exposed by the component.

`SubscribeMarketTrades()`	Public procedure exposed by the component.
`UnSubscribeMarketTrades()`	Public procedure exposed by the component.

`SubscribeTickerBatch()`	Public procedure exposed by the component.
`UnSubscribeTickerBatch()`	Public procedure exposed by the component.
`SubscribeHeartBeat()`	Public procedure exposed by the component.
`UnSubscribeHeartBeat()`	Public procedure exposed by the component.
`SubscribeStatus()`	Public procedure exposed by the component.
`UnSubscribeStatus()`	Public procedure exposed by the component.
`SubscribeCandles()`	Public procedure exposed by the component.
`UnSubscribeCandles()`	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 **Coinbase | Connect WebSocket API** configuration sourced from the online help.

> **About this scenario.** In order to connect to Coinbase WebSocket API, just create a new Coinbase API client and attach to TsgcWebSocketClient. See below an example:

Delphi (VCL / FireMonkey)

```
oClient := TsgcWebSocketClient.Create(nil);
oCoinbase := TsgcWSAPI_Coinbase.Create(nil);
oCoinbase.Client := oClient;
oClient.Active := True;
```

C++ Builder

```
TsgcWebSocketClient oClient = new TsgcWebSocketClient();
TsgcWSAPI_Coinbase oCoinbase = new TsgcWSAPI_Coinbase();
oCoinbase->Client = oClient;
oClient->Active = true;
```

.NET (C#)

```
TsgcWebSocketClient oClient = new TsgcWebSocketClient();
TsgcWSAPI_Coinbase oCoinbase = new TsgcWSAPI_Coinbase();
oCoinbase.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 · Coinbase | Subscribe WebSocket Channel

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

Delphi (VCL / FireMonkey)

```
oClient := TsgcWebSocketClient.Create(nil);
oCoinbase := TsgcWSAPI_Coinbase.Create(nil);
oCoinbase.Client := oClient;
oCoinbase.SubscribeTicker('ETH-USD');

procedure OnCoinbaseMessage(Sender: TObject; aType, aRawMessage: string);
begin
// here you will receive the ticker updates
end;
```

C++ Builder

```
TsgcWebSocketClient oClient = new TsgcWebSocketClient();
TsgcWSAPI_Coinbase oCoinbase = new TsgcWSAPI_Coinbase();
oCoinbase->Client = oClient;
oCoinbase->SubscribeTicker("ETH-USD");

void OnCoinbaseMessage(TObject *Sender, string aType, string aRawMessage)
{
// here you will receive the ticker updates
}
```

.NET (C#)

```
TsgcWebSocketClient oClient = new TsgcWebSocketClient();
TsgcWSAPI_Coinbase oCoinbase = new TsgcWSAPI_Coinbase();
oCoinbase.Client = oClient;
oCoinbase.SubscribeTicker("ETH-USD");

void OnCoinbaseMessage(TObject Sender, string aType, string aRawMessage)
{
// here you will receive the ticker updates
}
```

2 · Coinbase Pro SandBox Account

Coinbase allows you to use a SandBox account where you can trade without real funds. This account requires creating API keys different from the production account.

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

```
oCoinbase := TsgcWSAPI_Coinbase.Create(nil);
oCoinbase.Coinbase.ApiKey := 'your api key';
oCoinbase.Coinbase.ApiSecret := 'your api secret';
oCoinbase.Coinbase.SandBox := True;
ShowMessage(oCoinbase.REST_API.ListAccounts);
```

```
C++ Builder
```

```
TsgcWSAPI_Coinbase oCoinbase = new TsgcWSAPI_Coinbase(this);
oCoinbase→Coinbase→ApiKey = "your api key";
oCoinbase→Coinbase→ApiSecret = "your api secret";
oCoinbase→Coinbase→SandBox = true;
ShowMessage(oCoinbase→REST_API>ListAccounts);
```

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

```
TsgcWSAPI_Coinbase oCoinbase = new TsgcWSAPI_Coinbase();
oCoinbase.Coinbase.ApiKey = "your api key";
oCoinbase.Coinbase.ApiSecret = "your api secret";
oCoinbase.Coinbase.SandBox = true;
MessageBox.Show(oCoinbase.REST_API.ListAccounts);
```

3 · Coinbase Pro | Get Market Data

Coinbase 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)

```
oCoinbase := TsgcWSAPI_Coinbase.Create(nil);
ShowMessage(oCoinbase.REST_API.GetPublicProductBook('BTC-USD'));
```

C++ Builder

```
TsgcWSAPI_Coinbase oCoinbase = new TsgcWSAPI_Coinbase(this);
ShowMessage(oCoinbase→REST_API→GetPublicProductBook("BTC-USD"));
```

.NET (C#)

```
TsgcWSAPI_Coinbase oCoinbase = new TsgcWSAPI_Coinbase();
MessageBox.Show(oCoinbase.REST_API.GetPublicProductBook("BTC-USD"));
```

4 · Coinbase Pro | Place Orders

In order to place new orders in Coinbase, you first need your API keys to access your private data. Check the following article: How to Use Private REST API.

Delphi (VCL / FireMonkey)

```
oCoinbase := TsgcWSAPI_Coinbase.Create(nil);
oCoinbase.Coinbase.ApiKey := 'your api key';
oCoinbase.Coinbase.ApiSecret := 'your api secret';
oCoinbase.Coinbase.ApiPassphrase := 'your passphrase';
ShowMessage(oCoinbase.REST_API.PlaceMarketOrder(coisBuy, 'BTC-USD', 0.002, 0));
```

C++ Builder

```
TsgcWSAPI_Coinbase oCoinbase = new TsgcWSAPI_Coinbase(this);
oCoinbase→Coinbase→ApiKey = "your api key";
oCoinbase→Coinbase→ApiSecret = "your api secret";
oCoinbase→Coinbase→ApiPassphrase = "your passphrase";
ShowMessage(oCoinbase→REST_API>PlaceMarketOrder(coisBuy, "BTC-USD", 0.002</code><code class="de
```

.NET (C#)

```
TsgcWSAPI_Coinbase oCoinbase = new TsgcWSAPI_Coinbase();
oCoinbase.Coinbase.ApiKey = "your api key";
oCoinbase.Coinbase.ApiSecret = "your api secret";
oCoinbase.Coinbase.ApiPassphrase = "your passphrase";
MessageBox.Show(oCoinbase.REST_API.PlaceMarketOrder(coisBuy, "BTC-USD", 0.002</code><code class=
```

5 · Coinbase Pro | Private REST API

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

Delphi (VCL / FireMonkey)

```
oCoinbase := TsgcWSAPI_Coinbase.Create(nil);
oCoinbase.Coinbase.ApiKey := '<your api key>';
oCoinbase.Coinbase.ApiSecret := '<your api secret>';
ShowMessage(oCoinbase.REST_API.ListAccounts);
```

C++ Builder

```
TsgcWSAPI_Coinbase oCoinbase = new TsgcWSAPI_Coinbase(this);
oCoinbase→Coinbase→ApiKey = "<your api key>";
oCoinbase→Coinbase→ApiSecret = "<your api secret>";
ShowMessage(oCoinbase→REST_API→ListAccounts);
```

.NET (C#)

```
TsgcWSAPI_Coinbase oCoinbase = new TsgcWSAPI_Coinbase();
oCoinbase.Coinbase.ApiKey = "<your api key>";
oCoinbase.Coinbase.ApiSecret = "<your api secret>";
MessageBox.Show(oCoinbase.REST_API.ListAccounts);
```

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 — Coinbase Advanced Trade API	docs.cdp.coinbase.com/advanced-trade/docs/welcome
Online help — component page	www.esegece.com/help/sgcWebSockets/Components/APIs/API/API_Coinbase.htm
Delphi demo project (in the sgcWebSockets package)	`Demos\05.Crypto\09.Coinbase`
.NET demo project (in the sgcWebSockets package)	`.net\demos\05.Crypto\09.Coinbase`
Component page	www.esegece.com/products/websockets/apis/coinbase-pro/
Product page	www.esegece.com/products/websockets/

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