

# DataSnap — HTTP.sys WebSocket + HTTP/2

---

TsgcWSServer\_HTTPAPI\_WebBrokerBridge — kernel-mode (http.sys) WebBroker / DataSnap host with WebSocket and HTTP/2 transports.

## Overview

---

TsgcWSServer\_HTTPAPI\_WebBrokerBridge use TsgcWebSocketServer\_HTTPAPI with HTTP/2 protocol enabled as server base and is useful if you want to use a single server for DataSnap, HTTP/2 and WebSocket connections.

## At a glance

---

### COMPONENT CLASS

TsgcWSServer\_HTTPAPI\_WebBrokerBridge

### STANDARDS / SPEC

WebSocket Protocol — RFC 6455

### TRANSPORTS

TCP, TLS, HTTP, HTTPS

### 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	<a href="#">WebSocket Protocol — RFC 6455</a> · <a href="#">HTTP/2 — RFC 9113</a>
Component class	<code>TsgcWSServer_HTTPAPI_WebBrokerBridge</code> (unit <code>sgcWebSocket_HTTPAPI_WebBrokerBridge</code> )
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.

<code>APIKeyManager</code>	Published or public property used to configure or query the component.
<code>DefaultPort</code>	Published or public property used to configure or query the component.
<code>BindingOptions</code>	Published or public property used to configure or query the component.
<code>SSLOptions</code>	Published or public property used to configure or query the component.
<code>OnCommandRequest</code>	Published or public property used to configure or query the component.
<code>Timeouts</code>	Published or public property used to configure or query the component.
<code>FineTune</code>	Published or public property used to configure or query the component.
<code>Asynchronous</code>	Published or public property used to configure or query the component.
<code>MaxConnections</code>	Published or public property used to configure or query the component.
<code>MaxBandwidth</code>	Published or public property used to configure or query the component.

---

## Main methods

The principal public methods exposed by the component.

<code>Start()</code>	Public procedure exposed by the component.
----------------------	--

---

---

<code>Stop()</code>	Public procedure exposed by the component.
<code>ReStart()</code>	Public procedure exposed by the component.
<code>ShareList()</code>	Public function exposed by the component.
<code>UnShareList()</code>	Public procedure exposed by the component.
<code>LockList()</code>	Public function exposed by the component.
<code>UnLockList()</code>	Public procedure exposed by the component.
<code>RegisterWebModuleClass()</code>	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 **Load Balancer** configuration sourced from the online help.

**About this scenario.** If the server is behind the TsgcWebSocketLoadBalancerServer, you may have issues with CORS, to avoid these issues, use the following code

### Delphi (VCL / FireMonkey)

```
procedure TWebModule1.WebModuleBeforeDispatch(Sender: TObject; Request: TWebRequest; Response: TWebResponse)
begin
    Response.SetCustomHeader('Access-Control-Allow-Origin','*');
    if Trim(Request.GetFieldByName('Access-Control-Request-Headers')) <> '' then
    begin
        Response.SetCustomHeader('Access-Control-Allow-Headers', Request.GetFieldByName('Access-Control-Request-Headers'));
        Response.Handled := True;
    end;
    if FServerFunctionInvokerAction <> nil then
        FServerFunctionInvokerAction.Enabled := AllowServerFunctionInvoker;
end;
```

### C++ Builder

```
void __fastcall TWebModule1::WebModuleBeforeDispatch(TObject *Sender, TWebRequest *Request, TWebResponse *Response)
{
    Response->SetCustomHeader("Access-Control-Allow-Origin", "*");
    if (Trim(Request->GetFieldByName("Access-Control-Request-Headers")) != "")
    {
        Response->SetCustomHeader("Access-Control-Allow-Headers", Request->GetFieldByName("Access-Control-Request-Headers"));
        Response->Handled = true;
    }
    if (FServerFunctionInvokerAction != nullptr)
    {
        FServerFunctionInvokerAction->Enabled = AllowServerFunctionInvoker;
    }
}
```

## .NET (C#)

```
void WebModuleBeforeDispatch(object Sender, TWebRequest Request, TWebResponse Response, ref bool
{
    Response.SetCustomHeader("Access-Control-Allow-Origin", "*");
    if (Request.GetFieldByName("Access-Control-Request-Headers").Trim() ≠ "")
    {
        Response.SetCustomHeader("Access-Control-Allow-Headers", Request.GetFieldByName("Access-Cont
        Handled = true;
    }
    if (FServerFunctionInvokerAction ≠ null)
    {
        FServerFunctionInvokerAction.Enabled = AllowServerFunctionInvoker;
    }
}
```

## 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 — WebSocket Protocol — RFC 6455 [datatracker.ietf.org/doc/html/rfc6455](https://datatracker.ietf.org/doc/html/rfc6455)

---

Primary standard / spec — HTTP/2 — RFC 9113 [datatracker.ietf.org/doc/html/rfc9113](https://datatracker.ietf.org/doc/html/rfc9113)

---

Online help — component page [www.egegece.com/help/sgcWebSockets/Components/Datasnap/Servers/TsgcWSServer\\_HTTPAPI\\_WebBrokerBridge.htm](http://www.egegece.com/help/sgcWebSockets/Components/Datasnap/Servers/TsgcWSServer_HTTPAPI_WebBrokerBridge.htm)

---

Delphi demo project (in the sgcWebSockets package) `Demos\40.DataSnap`

---

Component page [www.egegece.com/products/websockets/datasnap/httpsys-ws-http2/](http://www.egegece.com/products/websockets/datasnap/httpsys-ws-http2/)

---

Product page [www.egegece.com/products/websockets/](http://www.egegece.com/products/websockets/)

**Document scope.** This document covers the publicly-documented surface of the DataSnap — HTTP.sys WebSocket + HTTP/2 component shipped with sgcWebSockets. For full property, method and event reference consult the online help linked above.