SGCWEBSOCKETS · TECHNICAL DOCUMENT

DataSnap — WebBroker Bridge (WS + HTTP/2)

TsgcWSHTTP2WebBrokerBridgeServer — WebBroker / DataSnap host with both WebSocket and RFC 9113 HTTP/2 transports.

eSeGeCe

WWW.ESEGECE.COM

Overview

TsgcWSHTTP2WebBrokerBridgeServer use TsgcWebSocketHTTPServer 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

`TsgcWSHTTP2WebBrokerBridgeServer`

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	WebSocket Protocol — RFC 6455 · HTTP/2 — RFC 9113
Component class	`TsgcWSHTTP2WebBrokerBridgeServer` (unit `sgcWebSocket_HTTP2_WebBrokerBridge`)
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.

`DefaultPort`	Published or public property used to configure or query the component.
`OnCommandRequest`	Published or public property used to configure or query the component.
`OnBeforeCommand`	Published or public property used to configure or query the component.
`OnCommandGet`	Published or public property used to configure or query the component.
`OnCommandOther`	Published or public property used to configure or query the component.
`OnCreateSession`	Published or public property used to configure or query the component.
`OnInvalidSession`	Published or public property used to configure or query the component.
`OnSessionEnd`	Published or public property used to configure or query the component.
`OnSessionStart`	Published or public property used to configure or query the component.
`OnBeforeForwardHTTP`	Published or public property used to configure or query the component.

Main methods

The principal public methods exposed by the component.

`RegisterWebModuleClass()`	Public procedure exposed by the component.

`PushPromiseAddPreLoadLinks()`	Public procedure exposed by the component.
`PushPromiseRemovePreLoadLinks()`	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: T
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-Contro
  Handled := True;
 end;
 if FServerFunctionInvokerAction <> nil then
  FServerFunctionInvokerAction.Enabled := AllowServerFunctionInvoker;
end;
```

C++ Builder

```
void __fastcall TWebModule1::WebModuleBeforeDispatch(TObject *Sender, TWebRequest *Request, TWeb
{
  Response->SetCustomHeader("Access-Control-Allow-Origin", "*");
  if (Trim(Request->GetFieldByName("Access-Control-Request-Headers")) != "")
  {
    Response->SetCustomHeader("Access-Control-Allow-Headers", Request->GetFieldByName("Access-Co
    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
Primary standard / spec — HTTP/2 — RFC 9113	datatracker.ietf.org/doc/html/rfc9113
Online help — component page	www.esegece.com/help/sgcWebSockets/Components/Datasnap/Servers/TsgcWSHTTP2WebBrokerBridgeServer.htm
Delphi demo project (in the sgcWebSockets package)	`Demos\40.DataSnap`
Component page	www.esegece.com/products/websockets/datasnap/webbrokerbridge-ws-http2/
Product page	www.esegece.com/products/websockets/

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