

DataSnap — WebBroker Bridge (WebSockets)

TsgcWSHTTPWebBrokerBridgeServer — pure-Delphi WebBroker / DataSnap host with WebSocket transport on top of TsgcWebSocketHTTPServer.

Overview

TsgcWSHTTPWebBrokerBridgeServer make use of TIdHttpWebBrokerBridge as server base and is useful if you want to use a single server for DataSnap, HTTP and WebSocket connections.

At a glance

COMPONENT CLASS

TsgcWSHTTPWebBrokerBridgeServer

STANDARDS / SPEC

WebSocket Protocol — RFC 6455

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	WebSocket Protocol — RFC 6455
Component class	<code>TsgcWSHTTPWebBrokerBridgeServer</code> (unit <code>sgcWebSocket_HTTP_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>DefaultPort</code>	Published or public property used to configure or query the component.
<code>HTTPWebBrokerBridge</code>	Published or public property used to configure or query the component.
<code>NotifyEvents</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>OnBeforeCommand</code>	Published or public property used to configure or query the component.
<code>OnCommandGet</code>	Published or public property used to configure or query the component.
<code>OnCommandOther</code>	Published or public property used to configure or query the component.
<code>OnCreateSession</code>	Published or public property used to configure or query the component.
<code>OnInvalidSession</code>	Published or public property used to configure or query the component.
<code>OnSessionEnd</code>	Published or public property used to configure or query the component.

Main methods

The principal public methods exposed by the component.

<code>PushPromiseAddPreLoadLinks()</code>	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: 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

Online help — component page www.esegece.com/help/sgcWebSockets/Components/Datasnap/Servers/TsgcWSHTTPWebBrokerBridgeServer.htm

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

Component page www.esegece.com/products/websockets/datasnap/webbrokerbridge-websockets/

Product page www.esegece.com/products/websockets/

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