

WO Data API Guide

WO TRAFFIC

WO DATA API VERSION 4.2.1

© 2018 WideOrbit Inc. All rights reserved.

Under copyright law, this material may not be copied, in whole or in part, without the written consent of WideOrbit Inc. Every effort has been made to ensure that the information in this publication is accurate. WideOrbit is not responsible for printing or clerical errors. The information in this publication may change without notice.

WO Traffic, WO Network, WO International, WO Traffic Satellite, WO Sales, WO Automation for Radio, WO Program, and Morning Edge are trademarks of WideOrbit Inc.

Other company and product names mentioned herein are trademarks of their respective companies. Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. WideOrbit assumes no responsibility regarding the performance or use of these products.

Table of Contents

Introduction	4
Basics	4
Authentication	4
Detecting and Handling Request Errors	4
Choosing the Export Format Type	5
Data Limits, Performance and Chunk Size Recommendations	5
GZip Support	6
API Methods	7
Request Stations List	7
Request Data	8
<i>Data Response</i>	<i>12</i>
<i>Data Receiving on Data API Side</i>	<i>12</i>
<i>Data Response Format</i>	<i>12</i>
<i>Merging Results</i>	<i>13</i>
<i>Sample Code</i>	<i>13</i>
Request Data for FTP	16
<i>Data Receiving on FTP Side</i>	<i>21</i>
Get Request Status	22
Canceling Requests	26
<i>Cancel Single Request</i>	<i>26</i>
<i>Cancel all Requests</i>	<i>27</i>
Request Data for Local Storage	28
Request Data for E-mail	32
Request Data for AWS S3	35
Import Data Request	39
Callback Request	42

<i>Callback Response</i>	43
Diagnostic Broadcasters Status Check Request	45
Serialization	46
<i>Original Sample Request</i>	46
<i>Comparing SQL Output</i>	46
<i>SQL Server FOR XML PATH (Sample)</i>	46
<i>With Data API Serialization</i>	46
<i>TSV (CSV tab separated)</i>	46
<i>XML</i>	46
<i>JSON</i>	46
<i>RAW</i>	47
<i>RAW with SQL Serializer</i>	47
Appendix	48
SqlDbType for Query Parameters	48

Introduction

This guide walks you through various steps to access and manage Data API requests. Upon completion, you will have a basic understanding of how to accomplish the following tasks:

- Authenticate connection
- Error handling
- Creating DAPI requests
- Monitoring Data API requests

Basics

WO Data API is a web service that provides access to *WO Traffic* data in a unified version agnostic manner.

It is REST based and allows for information in CSV/TSV, JSON, RAW, or XML formats.

DAPI uses the closed authentication model, where client authentication and usage patterns are managed by WideOrbit.



NOTE: If you have questions on how to use the API, are in need of new request types, or are considering delivery format types, please contact WideOrbit Support.

Authentication

WO Data API uses HTTP request header information to authorize requests and authenticate users. Users are authenticated using the following header keys: "partner-id", "api-key", and "agreement-key".

Each combination uniquely identifies the particular area of *WO Traffic* data to be accessed and the API client requesting access.

If no valid combination is found, the system returns the unauthorized response code:

```
Status Code: 401 (Unauthorized)
Response Content:
{ "Error": "Unauthorized access" }
```

Detecting and Handling Request Errors

When an unhandled error is detected, the server returns the following response:

```
Status Code: 500 (Internal Server Error)
```

```
Response Content:  
{ "Message": "Error Message"
```



NOTE: It is recommended that the client application has code to log and preserve error messages for further assistance.

Choosing the Export Format Type

When selecting the data export format type between CSV, JSON, RAW, and XML, consider the following:

- Capability of client to process data
- Amount of data returned for processing
- Delivery Type (FTP, HTTP, email)

Depending on the software solution used to process data, CSV may be most commonly supported, but would not allow for data validation, while JSON could be easily processed from JavaScript based applications.

Delivered data could consist of just one row/cell or multiple rows and columns.

JSON, and XML serialization is not customizable and would simply reflect the structure of the original data set.

RAW format doesn't make any data transformation and just returns query result as pure text. (this format is intended to be used with the FOR XML AUTO statement).

Data Limits, Performance and Chunk Size Recommendations

The system allows data to be delivered as a continuous stream or in "chunks" allowing reduction of time for each data response and to avoid request time outs as well as optimizing system performance.

The ChunkRowCount parameter specifies how many records will be sent to the destination service (Web/FTP) in one "packet".

The recommended value depends on the following:

- Row size read from Database – If the row's data size is significant, it is recommended to reduce the number of rows returned in one chunk. Also, you are required to implement logic to "stitch" data back into one continuous dataset.
- Export format: XML, JSON, CSV, RAW - Export formats have their own size overhead due to markup "verbosity" in an increasing order: CSV<JSON<XML.



NOTE: Servers which support live gzipping of workloads would be less affected by redundant information in XML due to internal algorithms addressing the issue.

- Target system: Web, FTP, Email – The Web system can have more requests handled comparably to FTP. FTP can be set to larger values in the ChunkRowCount parameter.



NOTE: With large chunk sizes, the server max request size limit can be reached and the request will be aborted by the target server.

GZip Support

Only HTTP delivery supports GZip compression for the target server.

In order to trigger GZip support in the response, add the following header tag in the original request sent to Data API:

```
Accept-Encoding: gzip
```

Header, and data will be transferred to the final endpoint in the corresponding format.



ALERT: The HTTP receiver must support GZip processing to decompress the incoming data packet.

API Methods

The system accepts GET and POST requests depending on the request type. Please refer to the appropriate HTTP Method below.



NOTE: Please contact your support or account contact to obtain Authentication tokens and API endpoint URI {root} information.

Request Stations List

Item	Description																											
URI	{root}/api/Stations/GetAllowedStations																											
HTTP Method	GET																											
Description	Submits a Station List request, returns stations array that is available for current Agreement or all Broadcaster stations when "Include all stations" is allowed by contract.																											
Request Header	"partner-id":"PrtnID123" "api-key":"ApiKey123" "agreement-key": "Ydg4eFbl"																											
JSON Response	[{ "TrafficStationInt":1, "TrafficStationId":"8456a7cf-8a48-4d8e-b484-ce23205e29d1", "BroadcasterInt":6, "StationId":"4597c0f0-0ba7-4116-ec23-54da8c208314", "StationInt":2, "StationCallLetters":"SUNNETWORKS", "StationName":"SUNNETWORKS", "StationMainPhone":"9841899425" }]																											
Response Schema	<table border="1"> <thead> <tr> <th>Field</th> <th>Data Type</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>TrafficStationInt</td> <td>int</td> <td></td> </tr> <tr> <td>TrafficStationId</td> <td>GUID</td> <td></td> </tr> <tr> <td>BroadcasterInt</td> <td>int</td> <td></td> </tr> <tr> <td>StationId</td> <td>GUID</td> <td></td> </tr> <tr> <td>StationInt</td> <td>int</td> <td></td> </tr> <tr> <td>StationCallLetters</td> <td>string</td> <td></td> </tr> <tr> <td>StationName</td> <td>string</td> <td></td> </tr> <tr> <td>StationMainPhone</td> <td>string</td> <td></td> </tr> </tbody> </table>	Field	Data Type	Notes	TrafficStationInt	int		TrafficStationId	GUID		BroadcasterInt	int		StationId	GUID		StationInt	int		StationCallLetters	string		StationName	string		StationMainPhone	string	
Field	Data Type	Notes																										
TrafficStationInt	int																											
TrafficStationId	GUID																											
BroadcasterInt	int																											
StationId	GUID																											
StationInt	int																											
StationCallLetters	string																											
StationName	string																											
StationMainPhone	string																											

Request Data

Item	Description			
URI	{root}/api/RequestData/GetData			
HTTP Method	POST			
Description	<p>Submits a Data Export request for delivery to a HTTP endpoint. If the target request has input parameters, they need to be included.</p> <p>Data is returned asynchronously to target URL.</p> <p>Request returns Request Id or error info.</p> <p>NOTE: We do recommend, always use HTTPS secure endpoint.</p>			
Request Header	<pre>"partner-id":"PrtnID123" "api-key":"ApiKey123" "agreement-key": "Ydg4eFbl"</pre>			
JSON Request	<pre>{ "RequestType":"General.Orders", "DataExportFormat":"JSON", "TargetUrl":"https://yoursite.com/ProcessData", "ChunkRowsCount":"50", "GenerateInvalidFormat":"true", "Parameters":[{ "Name":"@station_int", "DbType":"Int", "Value":"2" },{ "Name":"@start_date", "DbType":"VarChar", "Value":"2000-1-1" },{ "Name":"@end_date", "DbType":"VarChar", "Value":"2020-1-1" },{ "Name":"@options", "DbType":"Int", "Value":"1" }] }</pre>			
Response Schema	<table border="1"> <thead> <tr> <th>Field</th> <th>Data Type</th> <th>Notes</th> </tr> </thead> </table>	Field	Data Type	Notes
Field	Data Type	Notes		

RequestType	String	The value should be taken from the Vendor System>SQL Script page in WO Data API and is provided by WO. After implementation of the namespaces concept, the value becomes a concatenation of Request Group with Request Type, separated by point, or just the request type, when the namespace is global. Example: "General.Orders" or "Orders"
DataExportFormat	String	Value that specifies desired data format to be exported to. Valid options are: JSON, XML, CSV or RAW.
TargetUrl	String	The URL in Data API that will receive exported data from <i>WO Traffic</i> .
UseProxy	bool	Optional parameter. False by default. If set to true, data is sent to client from the Data API server. Otherwise the data is sent directly from the BAS.
ChunkRowsCount	Int	Number of rows read from database

		to be sent in one chunk. Range 10 - 9999.
GenerateInvalidFormat	bool	If set to true, then each file (new, or final appended result) will have a valid format and corresponding extension. If set to false, then all the files will have the .txt extension.
SuppressCsvHeaders	bool	Optional parameter. If set to true, then the headers in resulting files will be suppressed.
CustomCsvDelimiter	string	Optional parameter. Gives the ability to create custom delimiters in CSV files. For example, it could be a semicolon instead of whitespace.
QuoteText	bool	Optional parameter. Specifies whether to wrap text cells in double quotes or not.
Parameters	JSON Array	SQL Query Parameters Array.
Name	String	SQL Query Parameter Name.
DbType	String	Possible values: "Int", "VarChar" etc. NOTE: For the full list of possible values refer to the SqlDbTypes section

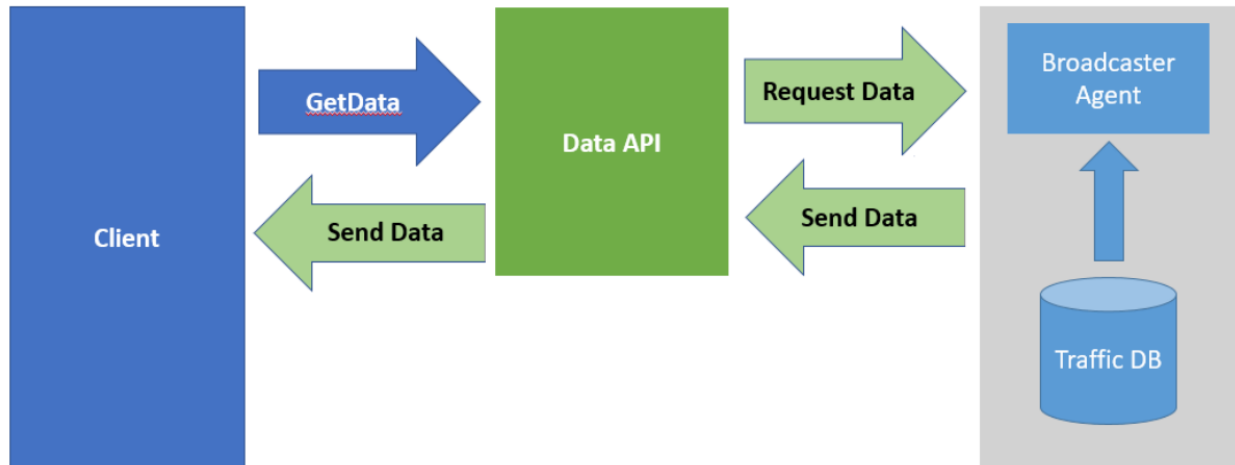
	Value	String	Value of Query Parameter.
JSON Response	"dbb0c7d0-7a7a-467b-979a-7a5478c0d93d"		
	RequestID GUID value is returned and can later be used to monitor status of the requests through the GetRequestStatus method.		

Data Response

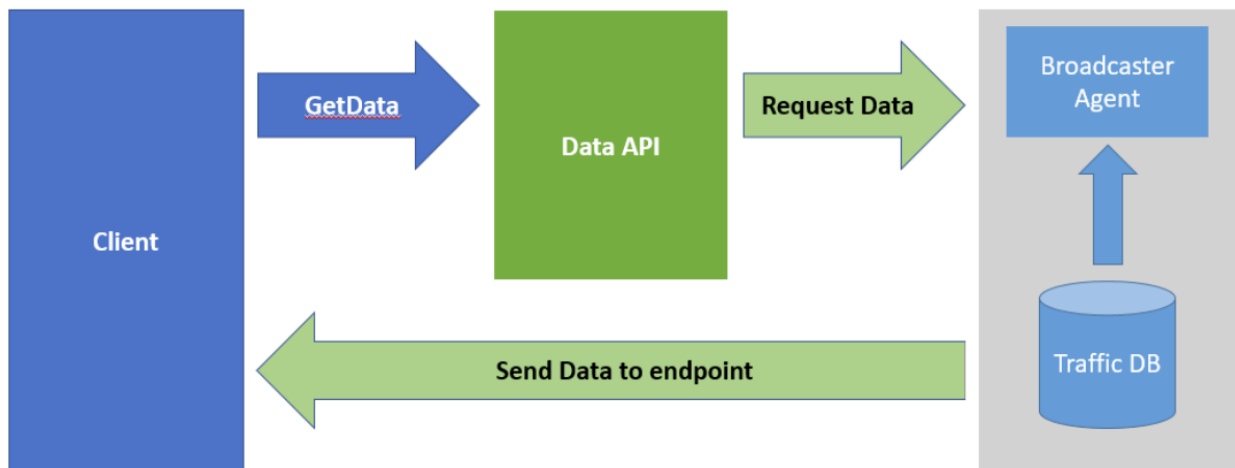
Data Receiving on Data API Side

After submitting Data API requests for data export to the HTTP endpoint, the following workflow applies:

UseProxy = true



UseProxy = false



NOTE: If the Broadcaster has firewall rules in place to prevent outbound communication, an HTTP endpoint will need to be white listed.

Data Response Format

The receiving side will need to implement the handler to unpack data from the following JSON-RPC packet.

Name	Data Type	Notes
requestId	String	The identifier to track requests that are currently uploading to the Data API system. This value is the result of the GetData API method.
chunkId	Int	Chunk Id that identifies the current block of information that was sent from <i>WO Traffic</i> . The last Chunk Id is defined as the negative total chunks count value.
resultSet	Int	Shows from which result set we are getting our data. This parameter varies only for multiple results datasets, for single result queries, it always equals 1.
data	String	Data formatted in requested format: JSON, XML, RAW or CSV Empty data response: JSON: '[]' XML: '<data></data>' RAW: '' CSV: '' If the resulting set contains the column of SQL binary array type, then it will be encoded as Base64 for security purposes.
errorMessage	String	Error message if something goes wrong during the data reading process.

Merging Results

When data is exported in chunks during the *WO Traffic* database reading process the following applies:

- Data is read from the SQL server sequentially row by row and placed in the buffer.
- When the read buffer is full, the chunk is sent to the client.
- When all the data has been processed, the last chunkId is set to the total chunks count as a negative value to indicate that there will be no more chunks to received and the client can mark the process as completed.

Sample Code

Included below is the ASP.NET MVC sample to save data as files and merge all result files when all data is received.



NOTE: The sample below takes in to consideration if gzip is enabled (GZipStream type requires inclusion of System.IO.Compression assembly).

```
[HttpPost]
[System.Web.Mvc.ValidateInput(false)]
public ActionResult ProcessData(string requestId, string data, int?
resultSet = null, int? chunkId = null, string errorMessage = "") {
    if (!_allowTestAction) {
        Response.Clear();
        Response.Write("AllowTestActionExecution is not enabled in
configuration");
        Response.TrySkipIisCustomErrors = true;
        return new HttpStatusCodeResult(HttpStatusCode.Forbidden);
    }
    if (HttpContext.Request.Headers["Content-Encoding"] != null &&
HttpContext.Request.Headers["Content-Encoding"] == "gzip") {
        var inputStream = HttpContext.Request.InputStream;
        var inputStreamLength = inputStream.Length;
        var decompressedBytes = new byte[inputStreamLength];
        using (var decompressionStream = new GZipStream(inputStream,
CompressionMode.Decompress)) {
            decompressionStream.Read(decompressedBytes, 0,
(int)inputStreamLength);
        }
        var decompressedString =
Encoding.UTF8.GetString(decompressedBytes);
        var keys = HttpUtility.ParseQueryString(decompressedString);
        requestId = keys["requestId"];
        data = keys["data"];
        resultSet = Int32.Parse(keys["resultSet"]);
        chunkId = Int32.Parse(keys["chunkId"]);
        errorMessage = Int32.Parse(keys["errorMessage "]);
    }

    var path = Server.MapPath(@"~/App_Data");
    var dir = Path.Combine(path, requestId);
    if (!Directory.Exists(dir)) {
        Directory.CreateDirectory(dir);
    }

    string fileChunkName;
    if (chunkId < 0) {
        fileChunkName = Path.Combine(dir,
String.Format("{1:D2}.Last.total({0}).txt", -chunkId, resultSet));
    } else {
```

```
        fileChunkName = Path.Combine(dir,
String.Format("{1:D2}.{0:D4}.txt", chunkId, resultSet));
    }
    System.IO.File.WriteAllText(fileChunkName, data);
    if (chunkId < 0) {
        var fileName = MergeAllFiles(dir);
    }
    return new HttpStatusCodeResult(HttpStatusCode.OK);
}
private static string MergeAllFiles(string dir) {
    var files = Directory.GetFiles(dir);
    var resFilePath = Path.Combine(dir, "resFile.txt");
    var fileCounter = 0;
    foreach (var file in files) {
        var txt = System.IO.File.ReadAllText(file);
        System.IO.File.AppendAllText(resFilePath, txt);
        fileCounter++;
    }
    return resFilePath;
}
```

Request Data for FTP

Item	Description
URI	{root}/api/RequestData/GetDataFtp
HTTP Method	POST
Description	Submits a Data Export request. Returns Request Id.
Request Header	"partner-id":"PrtnID123" "api-key":"ApiKey123" "agreement-key": "Ydg4eFbl"
JSON Response	{ "RequestType":"Orders", "DataExportFormat":"JSON", "EnableCompression":"false", "GenerateInvalidFormat":"true", "ChunkRowCount":"500", "FtpUrl":"ftp://ftp.domain.com", "FtpLogin":"ftp_login" "FtpPassword":"ftp_password" "UsePassiveMode":"true" "FtpProtocol":"FTP", "FtpProcessingOption":"Append", "Port":"567" "MaxOpenConnections":"50", "Parameters":[{" "Name":"@station_int", "DbType":"Int", "Value":"2" }], {" "Name":"@start_date", "DbType":"VarChar", "Value":"2000-1-1" }, {" "Name":"@end_date", "DbType":"VarChar", "Value":"2020-1-1" }, {" "Name":"@options", "DbType":"Int", "Value":"1" } }

Request Schema	Field	Data Type	Notes
	RequestType	String	The value should be taken from the Vendor System>SQL Script page in WO Data API and is provided by WO. After implementation of the namespaces concept, the value becomes a concatenation of the Request Group with the Request Type, separated by point, or just the request type, when the namespace is global. Example: "General.Orders" or "Orders"
	DataExportFormat	String	Value that specifies the desired data format to be exported to. Valid options are: JSON, XML, CSV, or RAW.
	UseProxy	bool	Optional parameter. True by default. If set to true, then data is sent to client from the Data API server. Otherwise the data is sent directly from the BAS.
	ChunkRowCount	Int	Number of rows read from the database to be sent

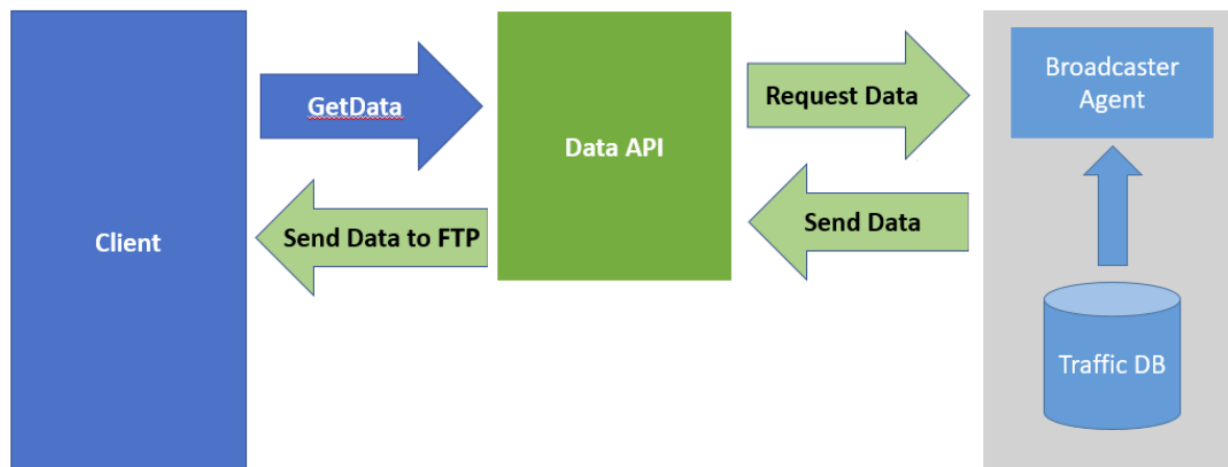
		in one chunk. Range is 10-10000.
FtpUrl	String	Target FTP location for exported data.
Port	Int?	Optional parameter. Port number.
FtpLogin	String	FTP User.
FtpPassword	String	FTP Password.
UsePassiveMode	String	FTP mode.
FtpProtocol	String	FTP Protocol – can be any of: FTP, FTP_SSL, or FTP_SSH.
FtpProcessingOption	String	Determines how the data will be processed on the FTP server. There are two options: "NewFile" or "Append".
MaxOpenConnections	Int	Number of FTP connections that can be opened at one time. This parameter is optional, and has the default value of 30.
EnableCompression	bool	If set to true, then each file will be saved as a Zip archive to optimize disc storage. Compression is allowed only for the "NewFile" mode. When both "Append" and "Enable compression" modes are selected,

		the error message " <i>Compression allowed only for NewFile processing option</i> " is displayed.
GenerateInvalidFormat	bool	If set to true, then each file (new, or final appended result) will have a valid format and corresponding extension. If set to false, then all files will have the .txt extension.
SuppressCsvHeaders	bool?	Optional parameter. If set to true, then headers in the resulting files will be suppressed.
CustomCsvDelimiter	string	Optional parameter. It gives the ability to create custom delimiters in CSV files. For example, using a semicolon instead of whitespace.
QuoteText	bool	Optional parameter. Specifies whether to wrap text cells in double quotes or not.
Parameters	JSON Array	SQL Query Parameters Array.
Name	String	SQL Query Parameter Name.
DbType	String	Possible values: "Int", "VarChar" etc. NOTE: For the full list of possible

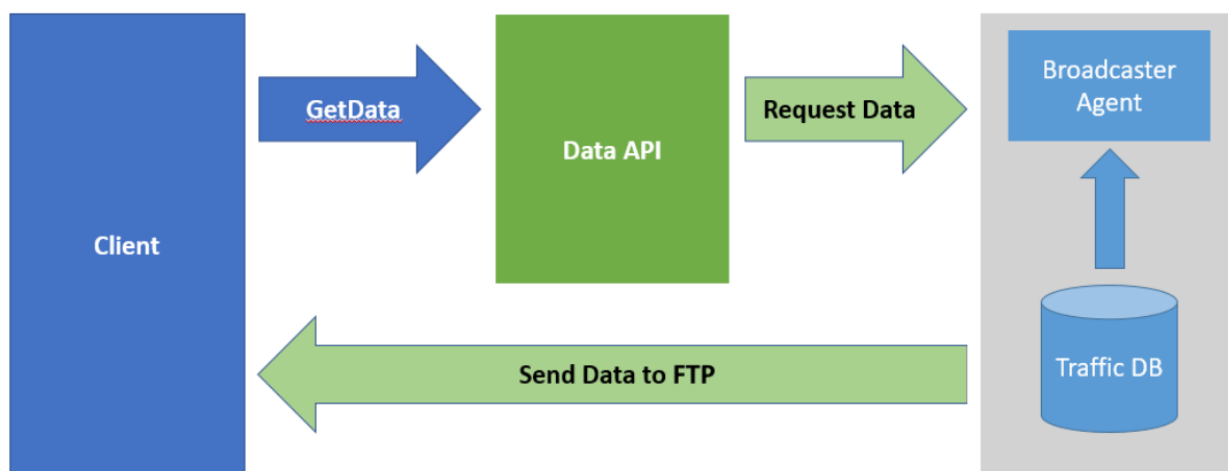
	Value	String	values refer to the SqlDbType section. Value of Query Parameter.
JSON Response	"dbb0c7d0-7a7a-467b-979a-7a5478c0d93d"		
	RequestID Guid value		

Data Receiving on FTP Side

UseProxy = true



UseProxy = false



NOTE: If the Broadcaster has firewall rules in place to prevent outbound communication, WideOrbit DAPI Server endpoint will need to be white listed.

To successfully receive exported data, the FTP server should have write permission enabled.

The data is exported in chunks and when the Append method is selected, APPE command should be allowed by the FTP Server. The file name is a result set and the request ID is divided by point, plus the '.txt' extension.

Get Request Status

Item	Description						
URI	{root}/api/RequestStats/GetRequestStatus						
HTTP Method	POST						
Description	Submits a request for Status info of currently executing or completed Request. Returns JSON-RPC data.						
Request Header	"partner-id":"PrtnID123" "api-key":"ApiKey123" "agreement-key": "Ydg4eFbl"						
JSON Request	{ "RequestId":"d8222969-5e63-4117-bad5-bc1f652f3add" }						
Response Schema	<table border="1"> <thead> <tr> <th>Field</th> <th>Data Type</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>RequestId</td> <td>String</td> <td>Request ID that was returned by GetData or GetDataFTP API methods.</td> </tr> </tbody> </table>	Field	Data Type	Notes	RequestId	String	Request ID that was returned by GetData or GetDataFTP API methods.
Field	Data Type	Notes					
RequestId	String	Request ID that was returned by GetData or GetDataFTP API methods.					
JSON Response	{ "BroadcasterExternalSystemName":"Broadcaster1", "RequestId":"d8222969-5e63-4117-bad5-bc1f652f3add", "CreateDate":"2015-05-18T09:03:15.157", "CompleteDate":"2015-05-18T09:03:49.573", "RequestStatus":"Complete", "ErrorMessage":null, "RequestDuration":"00:00:34.4160000", "BroadcasterRequestsQueueDuration":"00:00:05.35", "SqlQueryExecutionDuration":"00:00:25.65", "FtpUploadDuration":"00:00:04.416", "Destination":"https://cs50.salesforce.com" }						

Status	Description
Pending	Assigned to the request when it's waiting in the Broadcaster Agent Requests Queue on the WO Data API side.
Inactive	Returned when BAS is in Inactive state and could not accept or process the requests.
PostedToBa	When the request is posted to the Broadcaster Agent and moved to the <i>WO Central</i> database.
In_Progress	When the request is retrieved from the <i>WO Central</i> database by the Broadcaster Agent and execution is started.
Complete	Request completed without any errors.
Failed	Some errors occurred while executing the request.
Deleted	Service status, can be used manually when you don't want to show requests on the WO Data Web UI, but you need them to exist in the database.
PartiallyCompleted	Added to support integration with Salesforce, used to identify when not all the data is successfully transferred.
Canceling	Assigned when user sent the WO Data API request for canceling, or cancelled using the WO Data Web UI.
Canceled	Request is actually canceled, and it's execution is terminated.
Abandoned	Assigned to all requests in In_Progress status when the Broadcaster Agent is inactive, connection is stale (no requests are processed in a reasonable time) or on Broadcaster Agent restart. The Abandoned state will

also be assigned to the requests that expired while waiting in the Data API's or Central's queues.

Response Schema	Field	Data Type	Notes
	BroadcasterExternalSystemName	String	Name of the target Broadcaster.
	RequestId	String	Request ID being checked.
	CreateDate	DateTime	Request Start Date (UTC).
	CompleteDate	DateTime	Request End Date (UTC).
	RequestStatus	String	Status of Request. Possible values - see below.
	ErrorMessage	String	Error message, if any.
	RequestDuration	String	Request execution duration in format of HH:MM:SS.MS.
	BroadcasterRequestsQueueDuration	String	Time spent on Data API Broadcaster Requests queue in the HH:MM:SS.MS format.
	SqlQueryExecutionDuration	String	SQL query execution duration in the HH:MM:SS.MS format.
	FtpUploadDuration	String	Duration of data upload to the FTP server in

		HH:MM:SS.MS format.
Destination	String	Depending on the destination type (URL, FTP, SF, or Local Storage) returns the destination point, where the result data is uploaded.

Canceling Requests

Cancel Single Request

After submitting a Data API request, you have the ability to cancel pending requests.

When you send a CancelRequest to Data API, the request status will be set to "Canceling", and after the request is actually canceled, the request status will be set to "Canceled".

Item	Description									
URI	{root}/api/RequestStats/CancelRequest/{request_id}									
HTTP Method	GET									
Description	Submits request to cancel Data API request in progress where {request_id} is pending request id. Returns JSON data.									
Request Header	"partner-id":"PrtnID123" "api-key":"ApiKey123" "agreement-key": "Ydg4eFbl"									
JSON Response	{ "CanceledRequestsCount":"1", "Message":"Request succesfully canceled" }									
Response Schema	<table border="1"> <thead> <tr> <th>Field</th> <th>Data Type</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>CanceledRequestsCount</td> <td>Int</td> <td>Informs you about the number of requests canceled, for this type of request could be "1" or "0".</td> </tr> <tr> <td>Message</td> <td>String</td> <td>Error message if any, success message, or in case the Data API request is already processed: "<i>There are no requests with request id {your id} in progress</i>" message displays.</td> </tr> </tbody> </table>	Field	Data Type	Notes	CanceledRequestsCount	Int	Informs you about the number of requests canceled, for this type of request could be "1" or "0".	Message	String	Error message if any, success message, or in case the Data API request is already processed: " <i>There are no requests with request id {your id} in progress</i> " message displays.
Field	Data Type	Notes								
CanceledRequestsCount	Int	Informs you about the number of requests canceled, for this type of request could be "1" or "0".								
Message	String	Error message if any, success message, or in case the Data API request is already processed: " <i>There are no requests with request id {your id} in progress</i> " message displays.								

Cancel all Requests

Item	Description									
URI	{root}/api/RequestStats/CancelRequestsPerAgreement									
HTTP Method	GET									
Description	Submits request for canceling all pending Data API requests per connection.									
Request Header	"partner-id":"PrtnID123" "api-key":"ApiKey123" "agreement-key": "Ydg4eFbl"									
JSON Response	{ "CanceledRequestsCount":"10", "Message":"Requests successfully canceled" }									
Response Schema	<table border="1"> <thead> <tr> <th>Field</th> <th>Data Type</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>CanceledRequestsCount</td> <td>Int</td> <td>Informs you about the number of requests canceled.</td> </tr> <tr> <td>Message</td> <td>String</td> <td>Error message if any, success message, or in the case that Data API request is already processed: "<i>There are no requests in progress for current agreement</i>" message displays.</td> </tr> </tbody> </table>	Field	Data Type	Notes	CanceledRequestsCount	Int	Informs you about the number of requests canceled.	Message	String	Error message if any, success message, or in the case that Data API request is already processed: " <i>There are no requests in progress for current agreement</i> " message displays.
	Field	Data Type	Notes							
	CanceledRequestsCount	Int	Informs you about the number of requests canceled.							
Message	String	Error message if any, success message, or in the case that Data API request is already processed: " <i>There are no requests in progress for current agreement</i> " message displays.								

Request Data for Local Storage

Item	Description						
URI	{root}/api/RequestData/GetDataLocalStorage						
HTTP Method	POST						
Description	Submits a Local Storage Data Export request, returns Request Id Internal to WideOrbit network.						
Request Header	"partner-id":"PrtnID123" "api-key":"ApiKey123" "agreement-key": "Ydg4eFbl"						
JSON Response	{ "RequestType":"General.Stations", "DataExportFormat":"JSON", "ChunkRowsCount":"100", "EnableCompression":"true", "GenerateInvalidFormat":"true", "ProcessingOption":"Append", "Parameters":[{" "Name":"@station_int", "DbType":"Int", "Value":"2" }], {" "Name":"@start_date", "DbType":"VarChar", "Value":"2000-1-1" }, {" "Name":"@end_date", "DbType":"VarChar", "Value":"2020-1-1" }, {" "Name":"@options", "DbType":"Int", "Value":"1" } }						
Request Schema	<table border="1"> <thead> <tr> <th>Field</th> <th>Data Type</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>RequestType</td> <td>String</td> <td>The value should be taken from the Vendor System>SQL Script</td> </tr> </tbody> </table>	Field	Data Type	Notes	RequestType	String	The value should be taken from the Vendor System>SQL Script
Field	Data Type	Notes					
RequestType	String	The value should be taken from the Vendor System>SQL Script					

		<p>page in WO Data API and is provided by WO. After implementation of the namespaces concept, the value becomes a concatenation of Request Group with Request Type, separated by point, or just the request type, when the namespace is global.</p> <p>Example: "General.Orders" or "Orders"</p>
DataExportFormat	String	Value that specifies the desired data format to be exported to. Valid options are: JSON, XML, CSV, or RAW.
ChunkRowsCount	Int	Number of rows read from the database to be sent in one chunk. Range is 10-10000.
EnableCompression	bool	If set to true, then each file will be saved as a Zip archive to optimize disc storage. Compression is allowed only for "NewFile" mode. When both "Append" and "Enable compression" modes are selected,

		the error message " <i>Compression allowed only for NewFile processing option</i> " is displayed.
GenerateInvalidFormat	bool	If set to true, then each file (new, or final appended result) will have a valid format and the corresponding extension. If set to false, then all the files will have the .txt extension.
ProcessingOption	String	Determines how the data will be processed. At the moment there are two options: "NewFile" or "Append".
SuppressCsvHeaders	bool?	Optional parameter. If set to true, then headers in the resulting files will be suppressed.
CustomCsvDelimiter	String	Optional parameter. It gives the ability to create custom delimiters in CSV files. For example, using a semicolon instead of whitespace.
QuoteText	bool	Optional parameter. Specifies whether to wrap text cells in double quotes or not.
Parameters	JSON Array	SQL Query Parameters Array

	Name	String	SQL Query Parameter Name.
	DbType	String	Possible values: "Int", "VarChar" etc. For the full list of possible values refer to the SqlDbTypes section.
	Value	String	Value of Query Parameter.
JSON Response	"dbb0c7d0-7a7a-467b-979a-7a5478c0d93d" RequestID GUID value		

Request Data for E-mail

Item	Description						
URI	{root}/api/RequestData/GetDataEmail						
HTTP Method	POST						
Description	Submits the E-mail Data Export request, and returns the Request Id.						
Request Header	"partner-id":"PrtnID123" "api-key":"ApiKey123" "agreement-key": "Ydg4eFbl"						
JSON Request	{ "RequestType":"General.Stations", "DataExportFormat":"JSON", "EmailTitle":"Your e-mail title", "EmailBodyText":"Your e-mail body text", "EnableCompression":"false", "ChunkRowsCount":"50", "EmbeddedResult":"true", "SuppressEmptyResultSets":"true", "ReceiversEmails":["receiveremail1@mail.com", "receiveremail2@mail.com", "receivermail3@mail.com"], "Parameters":[{ "Name":"@station_int", "DbType":"Int", "Value":"2" },{ "Name":"@start_date", "DbType":"VarChar", "Value":"2000-1-1" }] }						
Request Schema	<table border="1"> <thead> <tr> <th>Field</th> <th>Data Type</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>RequestType</td> <td>String</td> <td>The value should be taken form the Vendor System>SQL Script page in WO Data</td> </tr> </tbody> </table>	Field	Data Type	Notes	RequestType	String	The value should be taken form the Vendor System>SQL Script page in WO Data
Field	Data Type	Notes					
RequestType	String	The value should be taken form the Vendor System>SQL Script page in WO Data					

		API and provided is by WO. After implementation of the namespaces concept, the value becomes a concatenation of the Request Group with the Request Type, separated by point, or just the request type, when the namespace is global. Example: "General.Orders" or "Orders"
DataExportFormat	String	Value that specifies the desired data format to be exported to. Valid options are: JSON, XML, CSV, or RAW.
EmailTitle	String	Specifies the title of the mail message.
EmailBodyText	String	Text that will be shown in the mail message body.
EnableCompression	bool	If set to true, then each file will be saved as a Zip archive.
EmbeddedResult	Bool?	Optional parameter. If set to true, then the returned result will be inserted directly into the e-mail body serialized as CSV, JSON, or XML. (depends on DataExportFormat)

	SuppressEmptyResultSets	bool?	Optional parameter. If set to true, then empty result sets will be suppressed and not sent to the receiver.
	ChunkRowCount	int	Number of rows read from the database to be sent in one e-mail attachment. Range is 10-10000.
	QuoteText	bool	Optional parameter. Specifies whether to wrap text cells in double quotes or not.
	ReceiversEmails	String Array	E-mail addresses that data is delivered to.
	Parameters	JSON Array	SQL Query Parameters Array.
	Name	String	SQL Query Parameter Name.
	DbType	String	Possible values: "Int", "VarChar" etc. For the full list of possible values refer to the SqlDbTypes section.
	Value	String	Value of Query Parameter.
JSON Response	"dbb0c7d0-7a7a-467b-979a-7a5478c0d93d" RequestID GUID value		

Request Data for AWS S3

Item	Description
URI	{root}/api/RequestData/GetDataAWSS3
HTTP Method	POST
Description	Submits the E-mail Data Export request, and returns the Request Id.
Request Header	"partner-id":"PrtnID123" "api-key":"ApiKey123" "agreement-key": "Ydg4eFbl"
JSON Request	{ "RequestType":"Default.Orders", "DataExportFormat":"JSON", "EnableCompression":"false", "GenerateInvalidFormat":"true", "ChunkRowCount":"500", "AccessKeyId": "KJHKJGHGJGJHGJHGJHG", "SecretAccessKey": "xldskfjsvncsnvlksrioneorrfoijflsldfj", "BucketName": "dataapiextract", "BucketRegionSystemName": "us-west-2", "RootPath": "data", "MaxOpenConnections":"50", "SuppressCsvHeaders": false, "QuoteText": true, "CustomCsvDelimiter": "", "UseProxy": false, "Retry": "0", "Parameters":[{ "Name":"@station_int", "DbType":"Int", "Value":"2" },{ "Name":"@start_date", "DbType":"VarChar", "Value":"2000-1-1" },{ "Name":"@end_date", "DbType":"VarChar", "Value":"2020-1-1" },{ "Name":"@options", "DbType":"Int",

```

        "Value": "1"
    }
  ]
}
    
```

Request Schema	Field	Data Type	Notes
	RequestType	String	The value should be taken from the Vendor System > SQL Script page in WO Data API and provided is by WO. For example: "General.Orders" or "Orders".
	DataExportFormat	String	Value that specifies the desired data format to be exported to. Valid options are: JSON, XML, CSV, or RAW.
	ChunkRowCount	Int	Number of rows read from DB to be sent in one chunk. The range is 10-10000.
	AccessKeyId	String	Amazon access key ID.
	SecretAccessKey	String	Amazon secret access key.
	BucketName	String	The Bucket name for uploaded data.
	BucketRegionSystemName	String	Region system name.
	RootPath	String	Optional parameter. Path to data location inside of the bucket. For example: /data/orders.

Retry	Int?	Optional parameter. Retry number for a failed connection. If absent, the Amazon default value is used.
MaxOpenConnections	Int	Optional parameter. Number of FTP connections that can be opened at once. This parameter has a default value of 30.
EnableCompression	bool	If set to true, then each file will be saved as a Zip archive to optimize disc storage. Compression is allowed only for "NewFile" mode. When both "Append" and "Enable compression" modes are selected, the error message " <i>Compression allowed only for NewFile processing option</i> " displays.
GenerateInvalidFormat	bool	If set to true, then each file (new, or final appended result) will have a valid format and corresponding extension. If set to false, then all the files will have a .txt extension.

	SuppressCsvHeaders	Bool?	Optional parameter. If set to true, then headers in the resulting files are suppressed.
	CustomCsvDelimiter	String	Optional parameter. Gives you the ability to create custom delimiters in CSV files. For example, it could be semicolon instead of whitespace.
	QuoteText	bool	Optional parameter. Specifies whether to wrap text cells into double quotes or not.
	Parameters	JSON Array	SQL Query parameters array.
	Name	String	SQL Query parameter name.
	DbType	String	Possible values: "Int", "VarChar", etc. For the full list of possible values refer to the SqlDbTypes section.
	Value	String	Value of query parameter.
JSON Response	"dbb0c7d0-7a7a-467b-979a-7a5478c0d93d" RequestID GUID value		

Import Data Request

Item	Description						
URI	{root}/api/ImportData						
HTTP Method	POST						
Description	Submits Import Data request, returns the Request Id.						
Request Header	"partner-id" : "PrtnID123" "api-key" : "ApiKey123" "agreement-key" : "Ydg4eFbl" "Content-Encoding" : "gzip" "Content-Type" : "application/json"						
JSON Request	{ "RequestType" : "RPE.Impressions", // for Staging Table mode "Data" : "1,573,598,1,2017-01- 12,39600000,40499000,W,153304,1732,1660,1186,841,672,168,1190,1 345,1377,1308" // for Stored Procedure mode "Parameters":[{ "Name": "@station_int", "DbType": "Int", "Value": "2" }},{ "Name": "@start_date", "DbType": "VarChar", "Value": "2000-1-1" }},{ "Name": "@end_date", "DbType": "VarChar", "Value": "2020-1-1" }},{ "Name": "@options", "DbType": "Int", "Value": "1" }] }						
Request Schema	<table border="1"> <thead> <tr> <th>Field</th> <th>Data Type</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>RequestType</td> <td>String</td> <td>The value should be taken from the Vendor System>SQL Script</td> </tr> </tbody> </table>	Field	Data Type	Notes	RequestType	String	The value should be taken from the Vendor System>SQL Script
Field	Data Type	Notes					
RequestType	String	The value should be taken from the Vendor System>SQL Script					

		page in WO Data API and is provided by WO. After implementation of the namespaces concept, the value becomes a concatenation of the Request Group with the Request Type, separated by point, or just the request type, when the namespace is global. Example: "General.Orders" or "Orders"
Data	String	TSV formatted file (column headers are optional).
Parameters	JSON Array	SQL Query Parameters Array (for Stored Procedure mode only).
Name	String	SQL Query Parameter Name (for Stored Procedure mode only).
DbType	String	Possible values: "Int", "VarChar" etc. (for Stored Procedure mode only). For the full list of possible values refer to the SqlDbTypes section.
Value	String	Value of Query Parameter (for

	Stored Procedure mode only).
JSON Response	"dbb0c7d0-7a7a-467b-979a-7a5478c0d93d" NOTE: Returned RequestID Guid can be later used to track requests to the Data API server data handler.

Callback Request

Item	Description																					
URI	Receiver URI																					
HTTP Method	POST																					
Description	Submits scheduled request finalization.																					
Request Header	"partner-id" : "236C8668-23D2-4D38-904F-04B257DFCFCB" "api-key" : "ABD3C411-D835-42E8-BB35-85C8C1C1F196" "authorization": "AAAAAAA-D835-42E8-BB35-85C8C1C1F196" "Content-Type" : "application/json"																					
JSON Request	{ "RequestType":"DubList", "Requestid":"fdbef825-d32e-401d-9c58-063e4cdb1634", "Path":"c:\\Data", "BroadcasterId":6, "AgreementId":1, "BroadcasterExternalSystemName":"SUN Networks 5.2", "CreateDate":"2017-10-05T08:20:13.173", "CompleteDate":"2017-10-05T08:20:15.147", "RequestStatus":"Complete", "ErrorMessage":null, "RequestDuration":19740000, "BroadcasterRequestsQueueDuration":18340000, "SqlQueryExecutionDuration":1400000, "FtpUploadDuration":0, "LoadStatus":null, "LoadFlag":0, "RowsExtracted":14, "ScheduledTaskInt":1, "TrackingField":"CREATE_DATE", "TrackingEnabled":true }																					
Request Schema	<table border="1"> <thead> <tr> <th>Field</th> <th>Data Type</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>RequestType</td> <td>String</td> <td></td> </tr> <tr> <td>Requestid</td> <td>Guid</td> <td></td> </tr> <tr> <td>Path</td> <td>String</td> <td></td> </tr> <tr> <td>BroadcasterId</td> <td>Integer</td> <td></td> </tr> <tr> <td>AgreementId</td> <td>Integer</td> <td></td> </tr> <tr> <td>BroadcasterExternalSystem Name</td> <td>String</td> <td></td> </tr> </tbody> </table>	Field	Data Type	Notes	RequestType	String		Requestid	Guid		Path	String		BroadcasterId	Integer		AgreementId	Integer		BroadcasterExternalSystem Name	String	
Field	Data Type	Notes																				
RequestType	String																					
Requestid	Guid																					
Path	String																					
BroadcasterId	Integer																					
AgreementId	Integer																					
BroadcasterExternalSystem Name	String																					

CreateDate	Date
CompleteDate	Date
RequestStatus	String
ErrorMessage	String
RequestDuration	Integer
BroadcasterRequestsQueueDuration	Integer
SqlQueryExecutionDuration	Integer
FtpUploadDuration	Integer
LoadStatus	String
LoadFlag	Integer
RowsExtracted	Integer
ScheduledTaskInt	Integer
TrackingField	String
TrackingEnabled	Bool
Response	Empty response with status code OK.

Callback Response

Item	Description									
URI	{root}/api/Requests/Update/{requestId}									
HTTP Method	POST									
Description	Submits processed callback request.									
Request Header	"partner-id" : "236C8668-23D2-4D38-904F-04B257DFCFCB" "api-key" : "ABD3C411-D835-42E8-BB35-85C8C1C1F196" "Content-Type" : "application/json"									
JSON Request	{ "IsValidated":true, "TrackingFieldValue":"2014-02-05 00:00:00.000" }									
Request Schema	<table border="1"> <thead> <tr> <th>Field</th> <th>Data Type</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>IsValidated</td> <td>Bool</td> <td>Request successfully handled.</td> </tr> <tr> <td></td> <td>Var</td> <td>Tracking field value. Empty value if tacking field is not used.</td> </tr> </tbody> </table>	Field	Data Type	Notes	IsValidated	Bool	Request successfully handled.		Var	Tracking field value. Empty value if tacking field is not used.
	Field	Data Type	Notes							
IsValidated	Bool	Request successfully handled.								
	Var	Tracking field value. Empty value if tacking field is not used.								

Response	Empty response with status code OK.
----------	-------------------------------------

Diagnostic Broadcasters Status Check Request

Item	Description
URI	{root}/api/Diagnostic/status_check[?full=true]
HTTP Method	GET
Description	Get current broadcasters status.
Request Header	"DiagnosticSecurityKey": "38A539B7-EAF9-4871-A7A1-E7A113BFFF95" "Content-Type" : "application/json"
GET Parameters	"full=true" - Returns full list of failing connections. Otherwise only returns changed since the last call.
JSON Response Failed	{ "alert_needed": "yes", "details": [{ "connection_name": "DataApiTest3", "last_status": "inactive", "last_status_change": "2018-08-06T15:02:09.3", "last_check": "2018-08-06T15:03:59.34" }] }
JSON Response OK	{ "alert_needed": "no", "details": null }

Serialization

WO DATA API supports various serialization methods while producing universally understandable data representations.

Original Sample Request

```
SELECT
    0 AS FLD_INT
  , 0.00 AS FLD_DOUBLE
  , 'ABC, ASDS, &, <TD>
SADSADAS' AS FLD_STR,
  GETDATE() AS FLD_DATE
```

Comparing SQL Output

SQL Server FOR XML PATH (Sample)

```
<Sample>
  <FLD_INT>0</FLD_INT>
  <FLD_DOUBLE>0.00</FLD_DOUBLE>
  <FLD_STR>ABC, ASDS, &;, &lt;TD&gt;&#x0D;
SADSADAS</FLD_STR>
  <FLD_DATE>2018-04-14T14:05:26.963</FLD_DATE>
</Sample>
```

With Data API Serialization

TSV (CSV tab separated)

```
FLD_INT FLD_DOUBLE FLD_STR FLD_DATE
0 0.00 "ABC, ASDS, &, <TD> SADSADAS" 2018-04-14T17:47:00.47
```

XML

```
<data>
  <record>
    <FLD_INT>0</FLD_INT>
    <FLD_DOUBLE>0.00</FLD_DOUBLE>
    <FLD_STR>ABC, ASDS, &;, &lt;TD&gt;
SADSADAS</FLD_STR>
    <FLD_DATE>2018-04-14T17:49:47.45</FLD_DATE>
  </record>
</data>
```

JSON

```
[{"FLD_INT": "0", "FLD_DOUBLE": "0.00", "FLD_STR": "ABC, ASDS, &, <TD>
SADSADAS", "FLD_DATE": "2018-04-14T17:51:23.08"}]
```

RAW

```
00.00ABC, ASDS, &, <TD>  
SADSADAS4/14/2018 5:53:53 PM
```

RAW with SQL Serializer

```
<Sample>  
  <FLD_INT>0</FLD_INT>  
  <FLD_DOUBLE>0.00</FLD_DOUBLE>  
  <FLD_STR>ABC,  
  ASDS, &, &lt;TD&gt;  
SADSADAS</FLD_STR>  
  <FLD_DATE>2018-04-14T17:56:39.560</FLD_DATE>  
</Sample>
```

Appendix

SqlDbType for Query Parameters

- BigInt
- Binary
- Bit
- Char
- DateTime
- Decimal
- Float
- Image
- Int
- Money
- NChar
- NText
- NVarChar
- Real
- UniqueIdentifier
- SmallDateTime
- SmallInt
- SmallMoney
- Text
- Timestamp
- TinyInt
- VarBinary
- VarChar
- Variant
- Xml
- Udt
- Structured
- Date

- Time
- DateTime2
- DateTimeOffset

Additional Resources

General information can be found in Help (from within *WO Traffic* click **Help** from the top menu). Additional questions can be directed to your WideOrbit Account Manager or the Support Department at wotrafficsupport@wideorbit.com.

WIDEORBIT

North American Headquarters:

1160 Battery Street

Suite 300

San Francisco, CA 94111

+1.415.675.6700

www.wideorbit.com