



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Widgets API Reference

StatsService

---

## Contents

- [1 Overview](#)
  - [1.1 Usage](#)
  - [1.2 Namespace](#)
  - [1.3 Customization](#)
- [2 Configuration](#)
  - [2.1 Description](#)
  - [2.2 Example](#)
  - [2.3 Options](#)
- [3 Localization](#)
- [4 API commands](#)
  - [4.1 configure](#)
  - [4.2 getStats](#)
- [5 API events](#)
- [6 Estimated wait time](#)
  - [6.1 API versions](#)
  - [6.2 Where to look for EWT data](#)

- 
- Developer

Learn how to fetch estimated wait time (EWT) details for a channel.

### Related documentation:

- 

## Overview

StatsService exposes high-level API access to Genesys statistics services, allowing you to fetch estimated wait time (EWT) details for each channel, such as WebChat or Callback, and display these details across the channels.

## Usage

StatsService and the Channel Selector widget work together right out of the box to display EWT details across all channels. Using the Channel Selector widget uses StatsService.

You can also use StatsService as a high-level API with bus commands and events and integrate in your own widget.

## Namespace

The StatsService plugin has the following namespaces, tied to each of the following types:

Type	Namespace
Configuration	stats
CXBus—API commands & API events	StatsService

## Customization

StatsService doesn't have any customization options. It is a plug-and-play plugin and works as is.

## Configuration

## Description

StatsService shares the **\_genesys.widgets.stats** configuration namespace and has connection settings to fetch EWT details from each channel.

---

## Example

```
window._genesys.widgets.stats =
ajaxTimeout: 3000,
ewt: {
  dataURL: 'http://10.0.0.121:7777/genesys/1/service/ewt-for-vq',
  apikey: 'n3exxxxxXREBMjGxxxx8VA',
  apiVersion: 'v1',
  mode: 'urs2'}
};
```

## Options

Name	Type	Description	Default	Required	Accepted Values
ajaxTimeout	number	Number of milliseconds to wait before AJAX timeout	3000	N/A	N/A
ewt.apikey	string	Apigee Proxy secure token.  If apiVersion is v3, this holds the x-api-key value.	N/A	Yes, if using Apigee Proxy or v3 API.	N/A
ewt.dataURL	URL String	URL to the API endpoint for estimated wait time (EWT)	N/A	Always	N/A
ewt.apiVersion	string	Version of EWT API.  <b>Note:</b> This value determines the version of EWT API in GMS/v3. That is: 'v1' - GMS EWT v1 'v2' - GMS EWT v2 'v3' - EWT v3  Only GET request type with virtual queue name as query parameters	v1'	Yes, if using GMS EWT v2 or EWT v3 dataURL	'v1', 'v2', 'v3'

Name	Type	Description	Default	Required	Accepted Values
		are supported.			
ewt.mode	string	EWT mode parameter for GMS/v3 API.  This value will vary based on the above apiVersion.	Will vary based on the above apiVersion as shown below.  'urs2' for 'v1' 'ewt2' for 'v2' 'mode2' for 'v3'	N/A	'urs', 'urs2' or 'stat' for 'v1'  'ewt1', 'ewt2' or 'ewt3' for 'v2'  'mode1', 'mode2' or 'mode3' for 'v3'

## Localization

StatsService doesn't have any localization options.

## API commands

Once you've registered your plugin on the bus, you can call commands on other registered plugins. Here's how to use the global bus object to register a new plugin on the bus.

### Important

The global bus object is a debugging tool. When implementing Widgets on your own site, do not use the global bus object to register your custom plugins. Instead, see Genesys Widgets Extensions for more information about extending Genesys Widgets.

```
var oMyPlugin = window._genesys.widgets.bus.registerPlugin('MyPlugin');
oMyPlugin.command('StatsService.getStats');
```

## configure

This is for internal use only. The main App plugin shares configuration settings to widgets using each widget's configure command. The configure command can only be called once at startup. Calling configure again after startup may result in unpredictable behavior.

---

## Example

```
oMyPlugin.command('StatsService.configure', {
    ewt:{
        apikey: '12345',
        dataURL: 'http://localhost:8080/foo/bar'
    },
    ajaxTimeout: 10000
}).done(function(e){
    // StatsService configured successfully
}).fail(function(e){
    // StatsService failed to configure
});
```

## Options

Option	Type	Description
ewt.apikey	string	API access token. Please contact your Genesys representative to obtain your API access token.
ewt.dataURL	URL String	URL of GES server.
ajaxTimeout	number	Number of milliseconds to wait before AJAX timeout.

## Resolutions

Status	When	Returns
resolved	Configuration options are provided and set.	n/a
rejected	No configuration options are provided.	'Invalid configuration'

## getStats

Ask the Genesys Stat server to fetch EWT details.

## Example

```
oMyPlugin.command('StatsService.getStats', {
    group: 'EWT',
    vqName: 'chat_ewt_test_eservices',
    mode: 'urs2'
```

```

}).done(function(e){
    // StatsService got stats successfully
}).fail(function(e){
    // StatsService failed to get stats
});

```

## Options

Option	Type	Description
group	string	Mention specific group name you would like to request, such as EWT, etc.
vqname	string/array	Specify a single virtual queue name as a string or a list of virtual queue names as an array. EWT will be fetched only for these virtual queues specified here. If nothing is specified, EWT will be fetched for all the available virtual queues.
mode	string	Specify EWT mode. This will vary based on apiVersion. Refer to mode configuration option for possible values.

## Resolutions

Status	When	Returns
resolved	Server returns EWT data.	(AJAX Response Object)
rejected	Server fail request fails.	'EWT request failed due to unknown reason'
rejected	No EWT dataURL provided.	'Invalid EWT configuration'

## API events

Once you've registered your plugin on the bus, you can subscribe to and listen for published events. Here's how to use the global bus object to register a new plugin on the bus.

### Important

The global bus object is a debugging tool. When implementing Widgets on your own site, do not use the global bus object to register your custom plugins. Instead, see

---

Genesys Widgets Extensions for more information about extending Genesys Widgets.

```
var oMyPlugin = window._genesys.widgets.bus.registerPlugin('MyPlugin');  
oMyPlugin.subscribe('StatsService.ready', function(e){});
```

Name	Description	Data
ready	StatsService is initialized and ready to accept commands.	n/a
updated	Latest Stats data is available.	EWT AJAX Response data
error.ewt	An error occurred between the client and the server for EWT.	{{(AJAX data Response)}}

## Estimated wait time

Estimated wait time (EWT) is displayed in the ChannelSelector and Callback widgets. These widgets use the `getStats` command to fetch EWT data from the GMS or GES server. These servers support multiple API versions and this document will explain how to configure the StatsService plugin to use the version that you need.

Use the `ewt.apiVersion` configuration option to specify the API version. Each version value corresponds to a particular API of GMS/GES. For all possible version values and their mapping, refer to the Description section of the `ewt.apiVersion` configuration option.

Sample configuration:

```
_genesys.widgets.stats.ewt.apiVersion =
```

## API versions

### v1

If `ewt.apiVersion` is configured to **v1** (this is also the default value), the `ewt.dataURL` configured must be a valid GMS 8.5.1 EWT API url. If not, an incorrect EWT value might be displayed.

Depending on this API version, the `ewt.mode` configuration option can hold a set of predefined possible values for this version. They are 'urs', 'urs2' and 'stat', where 'urs2' is the default value if not specified.

### Default example

```
_genesys.widgets.stats = {  
  ewt: {  
    apiVersion: "v1"  
    dataURL: http://somedomain/genesys/1/service/ewt-for-vq  
    mode: "urs2"  
  }  
}
```

---

For the above configuration, the StatsService plugin will construct the relevant dataURL as shown below.

`http://somedomain/genesys/1/service/ewt-for-vq?name=vq1&aqt=urs2`

'vq1' is added to the URL via the **vqName** option passed into the getStats command.

v2

If `ewt.apiVersion` is configured to **v2**, the `ewt.dataURL` configured must be a valid GMS 8.5.2 EWT API url. If not, incorrect EWT may be displayed. For this `apiVersion`, the possible values for `ewt.mode` are 'ewt1', 'ewt2' and 'ewt3'. 'ewt2' is the default value.

### Example

```
_genesys.widgets.stats = {
  ewt: {
    apiVersion: "v2"
    dataURL: http://somedomain/genesys/2/ewt
    mode: "ewt2"
  }
}
```

For the above configuration, the StatsService plugin will construct the relevant dataURL as shown below.

`http://somedomain/genesys/2/ewt/ewt2?vq=vq1,vq2`

'vq1' and 'vq2' are added to the URL via the **vqName** option passed into the getStats command.

v3

If `ewt.apiVersion` is set to **v3**, the `ewt.dataURL` configured must be a valid GES EWT API url. If not, incorrect EWT may be displayed. For this `apiVersion`, the possible values for `ewt.mode` are 'mode1', 'mode2' and 'mode3', where 'mode2' will be the default value if not specified.

### Example

```
_genesys.widgets.stats = {
  ewt: {
    apiVersion: "v3"
    dataURL: http://somedomain/engagement/v3/estimated-wait-time
    mode: "mode2"
  }
}
```

For the above configuration, the StatsService plugin will construct the relevant dataURL as shown below.

`http://somedomain/engagement/v3/estimated-wait-time?virtual-queues=vq1,vq2&mode=mode2`

'vq1' and 'vq2' are added to the URL via the **vqName** option passed into the getStats command.

## Where to look for EWT data

When the getStats command is called, it fetches the EWT data from either GMS/GES server based on

---

the configuration. This response data is included in the updated event in a standard format as shown below. In this data format, the **ewt** section will contain the virtual queue name and the estimated wait time as a key value pair. The **response** section contains the original raw data from the server and may vary between each server API.

```
{
  ewt: {
    "VQ_GMS_Callback_Out": 9.999 // consolidated standardized EWT data for each
virtual queue.
    "VQ_GMS_Callback": 5.12
    ...
  },
  response: { // Original raw data from GMS.
    "VQ_GMS_Callback_Out": {
      "time": 1506021728,
      "wt": 0,
      "calls": 0,
      "wcalls": 0,
      "pos": 1,
      "wpos": 1,
      "aqt": 9.999,
      "ewt": 9.999,
      "hit": 0
    },
    "VQ_GMS_Callback": {
      ...
    }
  }
}
```