



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 Developer Resources

[WebChatService](#)

---

## Contents

- 1 Overview
  - 1.1 Usage
  - 1.2 Namespace
  - 1.3 Customization
- 2 Configuration
  - 2.1 Example
- 3 Localization
- 4 API commands
  - 4.1 configure
  - 4.2 startChat
  - 4.3 endChat
  - 4.4 sendMessage
  - 4.5 sendCustomNotice
  - 4.6 sendTyping
  - 4.7 sendFilteredMessage
  - 4.8 addPrefilter
  - 4.9 updateUserData
  - 4.10 poll
  - 4.11 startPoll
  - 4.12 stopPoll
  - 4.13 resetPollExceptions
  - 4.14 restore
  - 4.15 getTranscript
  - 4.16 getAgents
  - 4.17 getStats
  - 4.18 sendFile
  - 4.19 downloadFile
  - 4.20 getSessionData
  - 4.21 fetchHistory
  - 4.22 registerTypingPreviewInput
  - 4.23 registerPreProcessor

- 
- 4.24 verifySession
  - 5 API events

---

Learn how to use Genesys chat services in Genesys Cloud CX.

### Related documentation:

- 

## Feature coming soon: Web messaging

If you are a Genesys Cloud CX customer, we encourage you to use the new web messaging feature to replace web chat. To use web messaging, you configure tracking through the Messenger JavaScript SDK instead of deploying a tracking snippet.

## Overview

WebChatService exposes high-level API access to Genesys chat services, so you can monitor and modify a chat session on the front end, or develop your own custom WebChat Widget. Compared to developing a custom chat UI and using the chat REST API, WebChatService dramatically simplifies integration—improving the reliability, feature set, and compatibility of every widget on the bus.

## Usage

WebChatService and the matching WebChat Widget work together right out of the box and they share the same configuration object. Using WebChat uses WebChatService.

You can also use WebChatService as a high-level API using bus commands and events to build your own WebChat Widget or other UI features based on WebChatService events.

## Namespace

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

Type	Namespace
Configuration	webchat
CXBus— API commands & API events	WebChatService

## Customization

WebChatService has many configuration options but no customization options. It is a plug-and-play plugin and works as is.

---

## Configuration

WebChat and WebChatService share the **\_genesys.widgets.webchat** configuration namespace. WebChat contains the UI options and WebChatService contains the connection options.

### Important

Starting with version 9.0.008.04, WebChatService allows you to choose between the types of chat services available in Genesys via the transport section configuration options.

For Genesys Cloud CX, the **transport.type** property should always be set to `purecloud-v2-sockets`.

### Example

- Applicable to Genesys Cloud CX - Guest Chat APIs

```
window._genesys.widgets.webchat = { transport: {
  type: 'purecloud-v2-sockets',
  dataURL: 'https://api.mypurecloud.com', // replace with API URL matching your region
  deploymentKey : 'YOUR_DEPLOYMENTKEY_HERE', // replace with your Deployment ID
  orgGuid : 'YOUR_ORGGUID_HERE', // replace with your Organization ID
  interactionData: {
    routing: {
      targetType: 'QUEUE',
      targetAddress: 'YOUR_QUEUE_NAME_HERE',
      priority: 2
    }
  }
},
  userData: {
    addressStreet: '64472 Brown Street',
    addressCity: 'Lindgrenmouth',
    addressPostalCode: '50163-2735',
    addressState: 'FL',
    phoneNumber: '1-916-892-2045 x293',
    phoneType: 'Cell',
    customerId: '596006'
  }
}
```

### Options

Name	Type	Description	Default	Required	Introduced/ updated
transport	object	Object containing the transport service configuration	N/A	Yes	9.0.008.04

Name	Type	Description	Default	Required	Introduced/ updated
		options.			
transport.type	string	Always set to purecloud-v2-sockets for use with Genesys Cloud CX.  For more details see Widget - Version 2 in Genesys Cloud CX Developer Center.	N/A	Yes	9.0.008.04
transport.dataURLstring (URL)		The Genesys Cloud CX WebChatService URL for your region. A list of API URLs per region is available in the Platform API section.  For more details see Widget - Version 2 in Genesys Cloud CX Developer Center.	N/A	Yes	9.0.008.04
transport.deploymentKey	string	Genesys Cloud CX widget deployment key. Identifies the widget on your web page as the one you created in the previous task (Create a widget configuration object).  For more details see Widget - Version 2 in Genesys Cloud CX Developer Center.	N/A	Yes	9.0.008.04
transport.orgGuid	string	Genesys Cloud CX organization ID; a unique GUID.	N/A	Yes	9.0.008.04

Name	Type	Description	Default	Required	Introduced/ updated
		For more details see Widget - Version 2 in Genesys Cloud CX Developer Center.			
transport.pagination	boolean	Enable/disable pagination capability to restore the chat messages based on transport.maxMessagePageSize option. If set to false, chat messages will be restored all at once.	true	No	9.0.008.04
transport.maxMessagePageSize	integer	Number of messages to be received per page during chat restore.	100	No	9.0.008.04
transport.interactionData.routing.targetType	string	Always set to 'QUEUE' to route to a queue.	N/A	Yes	9.0.008.04
transport.interactionData.routing.targetAddress	string	The queue name that receives chat messages. Example: <i>Support</i> For more details see Widget - Version 2 in Genesys Cloud CX Developer Center.	N/A	Yes	9.0.008.04
transport.interactionData.routing.priority	integer	Priority level from 0 (lowest) to 10 (highest). For more details see Widget - Version 2 in Genesys Cloud CX Developer Center.	N/A	No	9.0.008.04
transport.interactionData.routing.skills	array	List of skills. Example: <i>[Computers, Printers]</i> .	N/A	No	9.0.008.04

Name	Type	Description	Default	Required	Introduced/ updated
		For more details see Widget - Version 2 in Genesys Cloud CX Developer Center.			
transport.interactionData	object	Requested agent language skill. Example: <i>English - Written</i> . For more details see Widget - Version 2 in Genesys Cloud CX Developer Center.	N/A	No	9.0.008.04
userData	object	An object of key/value pairs of arbitrary custom data. For more details see Widget - Version 2 in Genesys Cloud CX Developer Center.	N/A	No	9.0.008.04

## Localization

WebChatService 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('WebChatService.getAgents');
```

### Important

Starting with version 9.0.008.04, WebChatService allows you to choose between the types of chat API services available in Genesys via the transport section configuration options. For more information, see the Options table in configuration.

## configure

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.

## startChat

Initiates a new chat session with the chat server via GES or with the service configured under the transport section.

## Example

```
oMyPlugin.command('WebChatService.startChat', {  
    nickname: 'Jonny',  
    firstname: 'Johnathan',  
    lastname: 'Smith',  
    email: 'jon.smith@mail.com',  
    subject: 'product questions',  
    userData: {}  
}).done(function(e){  
    // WebChatService started a chat successfully  
}).fail(function(e){  
    // WebChatService failed to start chat  
});
```

## Options

Option	Type	Description
nickname	string	Chat Entry Form Data: 'nickname'.
firstname	string	Chat Entry Form Data: 'firstname'.
lastname	string	Chat Entry Form Data: 'lastname'.
email	string	Chat Entry Form Data: 'email'.

---

Option	Type	Description
subject	string	Chat Entry Form Data: 'subject'.
userData	object	Arbitrary data to attach to the chat session (AKA attachedData). Properties defined here will be merged with default userData set in the configuration object.

## Resolutions

Status	When	Returns
resolved	Server confirms session started.	(AJAX Response Object)
rejected	A chat session is already active.	There is already an active chat session.
rejected	AJAX exception occurs.	(AJAX Response Object)
rejected	Server exception occurs.	(AJAX Response Object)
rejected	userData is invalid.	malformed data object provided in userData property.

## endChat

Ends the chat session with the chat server via GES or with the service configured under transport section.

## Example

```
oMyPlugin.command('WebChatService.endChat').done(function(e){
    // WebChatService ended a chat successfully
}).fail(function(e){
    // WebChatService failed to end chat
});
```

## Resolutions

Status	When	Returns
resolved	Active session is ended successfully.	(AJAX Response Object)
rejected	No chat session is currently active.	There is no active chat session.

## sendMessage

Sends a message from the client to the chat session.

---

## Example

```
oMyPlugin.command('WebChatService.sendMessage', {message: 'hi'}).done(function(e){
    // WebChatService sent a message successfully
}).fail(function(e){
    // WebChatService failed to send a message
});
```

## Options

Option	Type	Description
message	string	The message you want to send.

## Resolutions

Status	When	Returns
resolved	Message is successfully sent.	(AJAX Response Object)
rejected	No message text provided.	No message text provided.
rejected	No chat session is currently active.	There is no active chat session.
rejected	AJAX exception occurs.	(AJAX Response Object)

## sendCustomNotice

Sends a custom notice from the client to the chat server. This request is used to deliver any custom notification between a custom client application and a custom agent desktop. Neither Genesys Widgets, nor Workspace, uses this out of the box.

## Example

```
oMyPlugin.command('WebChatService.sendCustomNotice', {message: 'bye'}).done(function(e){
    // WebChatService sent a custom message successfully
}).fail(function(e){
    // WebChatService failed to send a custom message
});
```

## Options

Option	Type	Description
message	string	A message you want to send along with the custom notice.

---

## Resolutions

Status	When	Returns	Introduced/updated
resolved	Message is successfully sent.	(AJAX Response Object)	
rejected	AJAX exception occurs.	(AJAX Response Object)	
rejected	The server doesn't support receiving custom notices.	This transport doesn't support sendCustomNotice command.	9.0.008.04

## sendTyping

Sends a "*Customer typing*" notification to the chat session. A visual indication will be shown to the agent.

### Example

```
oMyPlugin.command('WebChatService.sendTyping').done(function(e){
    // WebChatService sent typing successfully
}).fail(function(e){
    // WebChatService failed to send typing
});
```

### Options

Option	Type	Description
Message	String	The message you want to send along with the typing notification.

## Resolutions

Status	When	Returns
resolved	AJAX request is successful.	(AJAX Response Object)
rejected	AJAX exception occurs.	(AJAX Response Object)
rejected	No chat session is currently active.	There is no active chat session.

## sendFilteredMessage

Sends a message along with a regular expression to match the message and hide it from the client. Useful for sending codes and tokens through the WebChat interface to the Agent Workspace.

---

## Important

Filters are now automatically stored and recalled on chat restore for the duration of the session.

### Example

```
oMyPlugin.command('WebChatService.sendFilteredMessage', {
    message: 'filtered message',
    regex: /[a-zA-Z]/
}).done(function(e){
    // WebChatService sent filtered message successfully
}).fail(function(e){
    // WebChatService failed to send filtered message
});
```

### Options

Option	Type	Description
message	string	Message you want to send but don't want to appear in the transcript.
regex	RegExp	Regular expression to match the message.

### Resolutions

Status	When	Returns
resolved	There is an active session.	n/a
rejected	No chat session is currently active.	No active chat session.

### addPrefilter

Adds a new pre-filter regular expression to the pre-filter list. Any messages matched using the pre-filters will not be shown in the transcript

## Important

Filters are now automatically stored and recalled on chat restore for the duration of the session.

---

## Example

```
oMyPlugin.command('WebChatService.addPrefilter', {filters: /[a-zA-Z]/}).done(function(e){
    // WebChatService added filter successfully
    // e == Object of registered prefilters
}).fail(function(e){
    // WebChatService failed to add filter
});
```

## Options

Option	Type	Description
filters	RegExp or Array of RegExp	Regular Expression(s) to add to the prefilter list.

## Resolutions

Status	When	Returns
resolved	Valid filters are provided.	Array of all registered prefilters.
rejected	Invalid or missing filters provided.	Missing or invalid filters provided. Please provide a regular expression or an array of regular expressions.

## updateUserData

Updates the `userData` properties associated with the chat session. If this command is called before a chat session starts, it will update the internal `userData` object and will be sent when a chat session starts. If this command is called after a chat session starts, a request to the server will be made to update the `userData` on the server associated with the chat session.

## Example

```
oMyPlugin.command('WebChatService.updateUserData', {firstname: 'Joe'}).done(function(e){
    // WebChatService updated user data successfully
}).fail(function(e){
    // WebChatService failed to update user data
});
```

## Options

Option	Type	Description
n/a	object	<code>userData</code> object you want to send to the server for this active session.

---

## Resolutions

Status	When	Returns	Introduced/updated
resolved	Session is active and userData is successfully sent.	(AJAX Response Object)	
rejected	Session is active and AJAX exception occurs.	(AJAX Response Object)	
resolved	Session is not active and internal userData object is merged with new userData properties provided.	The internal userData object that will be sent to the server.	
rejected	Session is active and the server doesn't support updating userData.	This transport doesn't support updating userData during an active chat session.	9.0.008.04

## poll

Internal use only. Starts polling for new messages.

### Example

```
oMyPlugin.command('WebChatService.poll').done(function(e){  
    // WebChatService started polling successfully  
}).fail(function(e){  
    // WebChatService failed to start polling  
});
```

## Resolutions

Status	When	Returns	Introduced/updated
resolved	There is an active session.	n/a	
rejected	WebChatService isn't calling this command.	Access Denied to private command. Only WebChatService is allowed to invoke this command.	
rejected	No chat session is currently active.	previous poll has not finished.	
rejected	The server doesn't support polling.	This transport doesn't support polling.	9.0.008.04

---

## startPoll

Starts automatic polling for new messages.

### Example

```
oMyPlugin.command('WebChatService.startPoll').done(function(e){
    // WebChatService started polling successfully
}).fail(function(e){
    // WebChatService failed to start polling
});
```

### Resolutions

Status	When	Returns	Introduced/updated
resolved	There is an active session.	n/a	
rejected	No chat session is currently active.	No active chat session.	
rejected	The server doesn't support polling.	This transport doesn't support polling.	9.0.008.04

## stopPoll

Stops automatic polling for new messages.

### Example

```
oMyPlugin.command('WebChatService.stopPoll').done(function(e){
    // WebChatService stopped polling successfully
}).fail(function(e){
    // WebChatService failed to stop polling
});
```

### Resolutions

Status	When	Returns	Introduced/updated
resolved	There is an active session.	n/a	
rejected	No chat session is currently active.	No active chat session.	
rejected	The server doesn't support polling.	This transport doesn't support polling.	9.0.008.04



---

## resetPollExceptions

Resets the poll exception count to 0. pollExceptionLimit is set in the configuration.

### Example

```
oMyPlugin.command('WebChatService.resetPollExceptions').done(function(e){  
    // WebChatService reset polling successfully  
}).fail(function(e){  
    // WebChatService failed to reset polling  
});
```

### Resolutions

Status	When	Returns	Introduced/updated
resolved	Always.	n/a	
rejected	The server doesn't support polling.	This transport doesn't support resetPollExceptions command.	9.0.008.04

## restore

Internal use only. You should not invoke this manually unless you are using Async mode.

### Example

```
oMyPlugin.command('WebChatService.restore').done(function(e){  
    // WebChatService restored successfully  
}).fail(function(e){  
    // WebChatService failed to restore  
});
```

### Options

Option	Type	Description	Accepted values	Introduced/updated
sessionData	string	The session data that is needed to restore the WebChat in Async mode. It is a JWT token string value. Applicable only when using WebChat with Genesys Multicloud CX v3	(JWT string token)	9.0.008.04

Option	Type	Description	Accepted values	Introduced/updated
		API. For more information, see the <b>Genesys Multicloud CX v3</b> tab in the <b>Options</b> table in configuration.		

## Resolutions

Status	When	Returns	Introduced/updated
resolved	Session has been found.	n/a	
rejected	Session cannot be found.	n/a	
rejected	Restoring chat session is in progress.	Already restoring. Ignoring request.	9.0.002.06
rejected	Chat session is already active.	Chat session is already active, ignoring restore command.	9.0.002.06
rejected	Trying restore chat session manually.	Access Denied to private command. Only WebChatService is allowed to invoke this command in Non-Async mode.	9.0.002.06

## getTranscript

Fetches an array of all messages in the chat session.

### Important

For more information on the fields included in JSON response, see Digital Channels Chat V2 Response Format.

## Example

```
oMyPlugin.command('WebChatService.getTranscript').done(function(e){
    // WebChatService got transcript successfully
    // e == Object with an array of messages
}).fail(function(e){
    // WebChatService failed to get transcript
});
```

---

## Resolutions

Status	When	Returns
resolved	Always	Object with an array of messages.

## getAgents

Return a list of agents that have participated in the chat. Includes agent metadata.

### Example

```
oMyPlugin.command('WebChatService.getAgents').done(function(e){  
    // WebChatService got agents successfully  
    // e == Object with agents information in chat  
}).fail(function(e){  
    // WebChatService failed to get agents  
});
```

## Resolutions

Status	When	Returns
resolved	Always	(Object List) {name: (String), connected: (Boolean), supervisor: (Boolean), connectedTime: (int time), disconnectedTime: (int time)}

## getStats

Returns stats on chat session including start time, end time, duration, and list of agents.

### Example

```
oMyPlugin.command('WebChatService.getStats').done(function(e){  
    // WebChatService got stats successfully  
    // e == Object with chat session stats  
}).fail(function(e){  
    // WebChatService failed to get stats  
});
```

## Resolutions

Status	When	Returns
resolved	Always	{agents: (Object), startTime: (int time), endTime: (int time),

---

---

Status	When	Returns
		duration: (int time)}

## sendFile

### [Introduced: 9.0.008.04]

Sends the file from the client machine to the agent.

### Example

```
oMyPlugin.command('WebChatService.sendFile', {files: $('').attr('type', 'file') /* Only works on UI, can not dynamically change */ }).done(function(e){
    // WebChatService sent file successfully
}).fail(function(e){
    // WebChatService failed to send file
});
```

### Options

Option	Type	Description
files	File	A reference to a file input element (for example )

### Resolutions

Status	When	Returns
resolved	The file sent is a valid type and size.	(AJAX Response Object)
rejected	The file sent is an invalid type.	(AJAX Response Object)
rejected	The number of uploads is exceeded.	(AJAX Response Object)
rejected	The file size exceeds the limit.	(AJAX Response Object)
rejected	The file size is too large or an unknown error occurs.	(AJAX Response Object)
rejected	The server doesn't support file uploads.	This transport doesn't support file uploads.

## downloadFile

Downloads the file to the client machine. Example

```
oMyPlugin.command('WebChatService.downloadFile', {fileId: '1', fileName: 'myfile.txt'}).done(function(e){
    // WebChatService sent file successfully
```

```

}).fail(function(e){
    // WebChatService failed to send file
});

```

## Options

Option	Type	Description
fileId	string	This is the ID of the file to be downloaded from the session.

## Resolutions

Status	When	Returns
resolved	The file is downloaded successfully.	n/a

## getSessionData

**[Introduced: 9.0.002.06]**

Retrieves the active session data at any time.

### Example

```
oMyPlugin.command('WebChatService.getSessionData')
```

## Resolutions

Status	When	Returns	Introduced/updated
resolved	Always, when using Chat via GMS API. For more information, see the <b>GMS</b> tab in the <b>Options</b> table in configuration.	{secureKey: (string), sessionId: (number/string), alias: (number/string), userId: (number/string)}	
resolved	Always, when using Chat via Genesys Multicloud CX v3 API. For more information, see the <b>Genesys Multicloud CX v3</b> tab in the <b>Options</b> table in configuration.	{participantId: (string), sessionId: {string}, token: (string), transportId: (string)}	9.0.008.04
rejected	Never	undefined	

## fetchHistory

**[Introduced: 9.0.008.04]**

This applies only in Asynchronous mode to fetch older chat messages. It does not fetch all of the

---

messages at once; rather a certain number of messages are fetched every time this command is called. Response data will be available in the `messageReceived` event.

### Example

```
oMyPlugin.command('WebChatService.fetchHistory')
```

### Resolutions

Status	When	Returns
resolved	Old messages are retrieved.	(AJAX Response Object)
rejected	Request fails.	(AJAX Response Object)
rejected	Asynchronous mode is not enabled.	Fetching history messages applies only to Asynchronous chat.
rejected	All messages are received.	No more messages to fetch.

### registerTypingPreviewInput

Selects an HTML input to watch for key events. Used to trigger `startTyping` and `stopTyping` automatically.

### Example

```
oMyPlugin.command('WebChatService.registerTypingPreviewInput', {input: $('input')
}).done(function(e){
    // WebChatService registered input area successfully
}).fail(function(e){
    // WebChatService failed to register typing preview
});
```

### Options

Option	Type	Description
input	HTML Reference	An HTML reference to a text or textarea input.

### Resolutions

Status	When	Returns
resolved	Valid HTML input reference is provided.	n/a
rejected	Invalid or missing HTML input reference.	Invalid value provided for the <b>input</b> property. An HTML element reference to a textarea or text input is required.

---

## registerPreProcessor

Registers a function that receives the message object, allowing you to manipulate the values before it is rendered in the transcript.

### Example

```
oMyPlugin.command('WebChatService.registerPreProcessor', {preprocessor: function(message){
    message.text = message.text + ' some preprocessing text';
    return message;
} }).done(function(e){
    // WebChatService registered preprocessor function
    // e == function that was registered
}).fail(function(e){
    // WebChatService failed to register function
});
```

### Options

Option	Type	Description
preprocessor	function	The preprocessor function you want to register.

### Resolutions

Status	When	Returns
resolved	A valid preprocessor function is provided and is registered.	The registered preprocessor function.
rejected	An invalid preprocessor function is provided.	No preprocessor function provided. Type provided was "".

## verifySession

Checks for existing WebChat session before triggering a proactive invite.

### Example

```
oMyPlugin.command('WebChatService.verifySession').done(function(e){
    if(e.sessionActive) {
        // dont show chat invite
    } else if(!e.sessionActive) {
        if(oMyPlugin.data('WebChat.open') == false){
            // show chat invite
        } else {
            // dont trigger chat invite
        }
    }
});
```

```

    }
  }
});

```

## Resolutions

Status	When	Returns
resolved	A session exists or not.	A boolean <code>sessionActive</code> which holds the session state.

## 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('WebChatService.ready', function(e){});

```

Name	Description	Data	Introduced/updated
Started	Chat session has successfully started.	(AJAX Response containing session data)	9.0.008.04
restoreTimeout	Chat session restoration attempted was denied after user navigated away from originating website for longer than the time limit: default 60 seconds.	N/A	9.0.008.04
restoreFailed	Could not restore chat session after page navigation or refresh.	N/A	9.0.008.04
restored	Chat session has been restored after page navigation or refresh.	N/A	9.0.008.04
reconnected	Connection restored. This event is only	N/A	9.0.008.04



Name	Description	Data	Introduced/updated
	published after <i>disconnected</i> .		
ready	WebChatService is initialized and ready to accept commands.	N/A	9.0.008.04
messageReceived	A new message has been received from the server. Includes text messages, status messages, notices, and other message types.	{originalMessages: (object), messages: (array of objects), restoring: (boolean), sessionData: (object)}	9.0.008.04
error	An error occurred between the client and the server.	(AJAX Response)	9.0.008.04
ended	Chat session has successfully ended.	N/A	9.0.008.04
disconnected	Cannot reach servers. No connection. Either the user is offline or the server is offline.	N/A	9.0.008.04
clientTypingStopped	After a user stops typing, a countdown begins. When the countdown completes, the typing notification will clear for the agent.	N/A	9.0.008.04
clientTypingStarted	The user has started typing. Sends an event to the agent.	N/A	9.0.008.04
clientDisconnected	Indicates the user has been disconnected from the chat session.	{message: (object), agents: (object), numAgentsConnected: (number)}	9.0.008.04
clientConnected	Indicates the user has been connected to the chat session.	{message: (object), agents: (object), numAgentsConnected: (number)}	9.0.008.04
agentTypingTimeout	Agent typing event has been timed out.	(AJAX Response)	9.0.008.04
agentTypingStopped	Agent has stopped typing.	(AJAX Response)	9.0.008.04
agentTypingStarted	Agents has started typing a new message.	(AJAX Response)	9.0.008.04
agentDisconnected	Indicates an agent has disconnected from the chat.	{message: (object), agents: (object), numAgentsConnected: (number)}	9.0.008.04
agentConnected	Indicates an agent has connected to the chat.	{message: (object), agents: (object),	9.0.008.04

---

Name	Description	Data	Introduced/updated
		numAgentsConnected: (number)}	