



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 Reference

API Commands

---

## Contents

- 1 API Commands
  - 1.1 configure
  - 1.2 startChat
  - 1.3 endChat
  - 1.4 sendMessage
  - 1.5 sendCustomNotice
  - 1.6 sendTyping
  - 1.7 sendFilteredMessage
  - 1.8 addPrefilter
  - 1.9 updateUserData
  - 1.10 poll
  - 1.11 startPoll
  - 1.12 stopPoll
  - 1.13 resetPollExceptions
  - 1.14 restore
  - 1.15 getTranscript
  - 1.16 getAgents
  - 1.17 getStats
  - 1.18 sendFile
  - 1.19 downloadFile
  - 1.20 getFileLimits
  - 1.21 getSessionData
  - 1.22 fetchHistory
  - 1.23 registerTypingPreviewInput
  - 1.24 registerPreProcessor
  - 1.25 verifySession

# API Commands

Once you've registered your own plugin on the bus, you can call commands on other registered plugins. Below we'll quickly register a new plugin on the bus using the global bus object.

## Important

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

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

## 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 GMS. Intended to be used by WebChat widgets only. Should not be invoked manually.

## 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'.
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	When server confirms session started	(AJAX Response Object)
rejected	When a chat session is already active	'There is already an active chat session'
rejected	When AJAX exception occurs	(AJAX Response Object)
rejected	When server exception occurs	(AJAX Response Object)
rejected	When userData is invalid	'malformed data object provided in userData property'

## endChat

Ends the chat session with the chat server via GMS. Intended to be used by WebChat widgets only. Should not be invoked manually.

## 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	When active session is ended successfully	(AJAX Response Object)
rejected	If no chat session is currently active	'There is no active chat session'

## sendMessage

Send a message from the client to the chat session. Intended to be used by WebChat widgets only. Should not be invoked manually.

## 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	When message is successfully sent	(AJAX Response Object)
rejected	If no message text provided	'No message text provided'
rejected	If no chat session is currently active	'There is no active chat session'
rejected	When AJAX exception occurs	(AJAX Response Object)

## sendCustomNotice

Send a custom notice from the client to the chat server.

### 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
resolved	When message is successfully sent	(AJAX Response Object)
rejected	When AJAX exception occurs	(AJAX Response Object)

---

## sendTyping

Send 'customer typing' notification to chat session. A visual indication will be shown to agent. Intended to be used by WebChat widgets only. Should not be invoked manually.

### Example

```
oMyPlugin.command('WebChatService.sendTyping', {message: 'hi'}).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	When AJAX request is successful	(AJAX Response Object)
rejected	When AJAX exception occurs	(AJAX Response Object)
rejected	If no chat session is currently active	'There is no active chat session'

## sendFilteredMessage

Send 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 Desktop.

### 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	When there is an active session	n/a
rejected	If no chat session is currently active	'No active chat session'

## addPrefilter

Add a new regular expression prefilter to the prefilter list. Any messages matched using the prefilters 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	When valid filters are provided	Array of all registered prefilters.
rejected	When 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	userData object you want to send to the server for this active session

## Resolutions

Status	When	Returns
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

## poll

Internal use only. Start polling for new messages. Intended to be used by WebChat widgets only. Should not be invoked manually.

## Example

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

```
});
```

## Resolutions

Status	When	Returns
resolved	When 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	If no chat session is currently active	'previous poll has not finished.'

## startPoll

Start automatic polling for new messages. Intended to be used by WebChat widgets only. Should not be invoked manually.

## Example

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

## Resolutions

Status	When	Returns
resolved	When there is an active session	n/a
rejected	When no chat session is currently active	No active chat session
rejected	When CometD is enabled	Polling is not supported when using CometD

## stopPoll

Stop automatic polling for new messages. Intended to be used by WebChat widgets only. Should not be invoked manually.

### Example

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

### Resolutions

Status	When	Returns
resolved	When there is an active session	n/a
rejected	If no chat session is currently active	No active chat session

## resetPollExceptions

Reset 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
resolved	Always	n/a
rejected	Never	undefined

## restore

Internal use only. Intended to be used by WebChatService only. Should not be invoked manually, except when 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	object	Applicable when using Async mode only. The session data that is needed to restore the WebChat in Async mode. It is a Key value pair object containing the values mentioned.	secureKey, userId, alias, sessionID	9.0.002.06

## Resolutions

Status	When	Returns	Introduced / Updated
resolved	Session has been found.	n/a	

Status	When	Returns	Introduced / Updated
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

Fetch 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

Return 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), duration: (int time)}

## sendFile

Sends the file from the client machine to the agent.

## Example

```
oMyPlugin.command('WebChatService.sendFile', {files: $('<input/>').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 <input type="file"/>)

## Resolutions

Status	When	Returns
resolved	When the file sent is a valid type and size	(AJAX Response Object)
rejected	When the file sent is an invalid type	(AJAX Response Object)
rejected	When the number of uploads is exceeded	(AJAX Response Object)
rejected	When the file size exceeds the limit	(AJAX Response Object)



---

Status	When	Returns
rejected	When the file size is too large or an unknown error occurs	(AJAX Response Object)
rejected	When CometD is enabled	File Uploads are not currently supported when using CometD

## 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
fileName	string	This is the name of the file to be downloaded from the session. It is an optional field.

### Resolutions

Status	When	Returns
resolved	When the file is downloaded successfully	n/a

## getFileLimits

This optional request can be used before uploading a large file. If size, type, or other constraints are

---

not met, then uploading the file will fail, avoiding network and CPU overhead.

### Example

```
oMyPlugin.command('WebChatService.getFileLimits').done(function(e){
    // WebChatService got file limits successfully
}).fail(function(e){
    // WebChatService failed to get file limits
});
```

### Resolutions

Status	When	Returns
resolved	When the file limits request succeeds	(AJAX Response Object)
rejected	When the file limits request fails	(AJAX Response Object)
rejected	When CometD is enabled	File Uploads are not currently supported when using CometD

## getSessionData

### [Introduced: 9.0.002.06]

To retrieve the active session data at any time.

### Example

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

### Resolutions

Status	When	Returns
resolved	Always	{secureKey: (string), sessionId: (number/string), alias: (number/string), userId: (number/string)}
rejected	Never	undefined

## fetchHistory

### [Introduced: 9.0.002.06]

For use with WebChat Widget only. This applies only in Asynchronous mode to fetch older chat messages. It does not fetch all at a time, rather a certain number of messages are fetched every time this command is called. Response data will be available in the **messageReceived** event. This internal command determines the last received message index and, based on this information, fetches older messages whenever it is called.

### Example

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

### Resolutions

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

## registerTypingPreviewInput

Select an HTML input to watch for key events. Used to trigger **startTyping** and **stopTyping** automatically. Intended to be used by WebChat widgets only. Should not be invoked manually.

### 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	When valid HTML input reference is provided	n/a
rejected	When invalid or missing HTML input reference	'Invalid value provided for the 'input' property. An HTML element reference to a textarea or text input is required.'

## registerPreProcessor

Allows you to register 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	When a valid preprocessor function is provided and is registered.	The registered preprocessor function.
rejected	When an invalid preprocessor function is provided.	No preprocessor function provided. Type provided was '<DATATYPE>'.

## 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 'sessionActive' which holds the session state.