API Reference

Launching the Chat Session

12/14/2025

Launching the Chat Session

There are two methods you can use to launch chat session:

- startSession — Use this entry-point method to start a new chat session, with various options for controlling the shape of that session.
- restoreSession — Use this entry-point method to launch an existing chat session. For example, in cases where a browser page with a chat window is reloaded.

startSession

Description

Entry-point method for creating new chat session.

Returned Promise

`startSession` (as well as all session commands) returns a "promise" object with two chainable methods: `done` and `fail`.

<table>
<tr><td>done</td><td>This method should be used to get access to the chat session object.

<pre>chat.startSession(options).done(function(session)
{
  // session.sendMessage,
session.onAgentConnected and all other
method are at your disposal.
});</pre></td></tr>
<tr><td>fail</td><td>Resolved with an event containing an error message describing what went wrong.

Event structure:

<table><tr><th>Parameter</th><th>Meaning</th></tr><tr><td>event.error.code</td><td>Code specifying the particular error</td></tr><tr><td>event.error.description</td><td>Description of error (English is default language).</td></tr></table>
Tip
For a list of possible error codes, see Error Codes.</td></tr>
</table>

Options

Options are expected as a single object with named properties. For example:

```
chat.startSession({
    serverUrl: '...',
    username: 'Anonymous'
});
```

Option	Type	Default Value	Mandatory	Description
serverUrl	string	undefined	Yes (except when transport option is provided)	URL of the CometD chat backend if built-in CometD transport is to be used.
username	string	'User'	No	Optional parameter. Name (nickname) of the visitor who initiated the chat session. If absent, and chat user is anonymous, default value is used.
subject	string	undefined	No	Subject of the chat session. If option is absent, parameter is left empty.
userData	Object	undefined	No	Represents a set of parameters that can\should be specified when creating the chat session. Typical use-case: providing data obtained from a registration form. `chat.startChat({ // Passed data will be used to identify and/or create user contact. userData: { FirstName: 'John', LastName: 'Doe', EmailAddress: 'johndoe@example.com' } });`

Option	Type	Default Value	Mandatory	Description
				Important Data you pass to `userData` is always merged with the default values of `userData`. `userData` contains the following keys: • `source` with value `web` • `pageTitle` with value set to current page's title, `document.title` • `url` with value set to the current page's URL, `document.location` • `referrer` with value set to the previous page's URL, `document.referrer` If you need to override and of these values, pass the values along with other `userData`. For example, the following sets `source` to `null` and leaves `startPage` as is: `chat.startChat({ userData: { FirstName: 'John', LastName: 'Doe', EmailAddress: 'johndoe@example.com', source: null } });`
`transport`	Object	[built-in CometD transport]	Only if `serverURL` is absent	Pass custom transport instance here, if needed (for example, REST-based). By default, built-in CometD chat transport is used.
`stateStorage`	{{read: function(): Object, write:	[internal built-in stateStorage]	No	Provides a method for dealing with

Option	Type	Default Value	Mandatory	Description
	function(Object|null)}}			session state persistence across page reloads. By default, built-in cookie-based state storage is used. Note that in certain cases, the `stateStorage.write` method is called with a special argument, `null`, which means that all existing state has to be cleared. For details, see Using a third-party mechanism for chat session state persistence.
`maxOfflineDuration`	number	5	No	Time (in seconds) during which session state cookies are stored after page reload/ navigation. Default is 5 seconds. If cookies expire, the chat session will not be restored via `restoreSession()`. This option only takes effect on default (built-in_ state storage. If custom state storage is passed, this option does not take effect.
`disableWebSockets`	boolean	`false`	No	Disable WebSockets for connections to the server. Only effective if the default transport is used. If you disable WebSockets, you should also pass this option to restoreSession().
`logger`	function	[internal console.log-based logger]	No	A function responsible for logging. If you pass a custom logging function to this option, the

Option	Type	Default Value	Mandatory	Description
				function should accept an arbitrary number of different argument types (similar to console.log).

restoreSession

Description

Entry-point method for launching an already existing chat session (for example, when a browser page with a chat window was reloaded).

Returned Promise

`restoreSession` returns a "promise" object with two chainable methods: `done` and `fail`.

done	This method should be used to get access to chat session object. `chat.restoreSession(options).done(function(session) { // session.sendMessage, session.onAgentConnected and all other methods are at your disposal. });`
fail	If the restore chat operation fails because of an error (and not because the chat doesn't exist), the fail callback receives an event argument with an error property similar to `startSession().fail` `callback`. `chat.restoreSession(options) .fail(function(event) { // If there was a chat session, but restoration fails, signal failure. if (event.error) { alert('chat restoration failed'); return; } // If there was no chat session, bind start chat to "start chat" button jQuery('#myChatButton').on('click', function() { chat.startChat(startChatOptions); } }) .done(function(session) { // session.sendMessage, session.onAgentConnected and all other method are at your disposal.`

	}); Tip For a list of possible error codes, see Error Codes.

Options

Options are expected as a single object with named properties. For example:

```
chat.restoreSession({
    maxOfflineDuration: 60
});
```

Option	Type	Default Value	Mandatory	Description
`transport`	Object	[built-in CometD transport]	Yes, if custom transport was passed in `startSession`	Pass custom transport here if needed (for example, REST-based). By default built-in CometD chat transport will be used.
`stateStorage`	{{read: function(): Object, write: function(Object|null)}}	[internal built-in stateStorage]	No	Provides a method for dealing with session state persistence across page reloads. By default, built-in cookie-based state storage is used. Note that in certain cases, the `stateStorage.write` method is called with a special argument, `null`, which means that all existing state has to be cleared. For details, see Using a third-party mechanism for chat session state persistence.
`maxOfflineDuration`	number	5	No	Time (in seconds) during which session state cookies are stored after page reload/ navigation. Default is 5 seconds. If cookies expire, the chat session will not be restored via `restoreSession()`. This option has no

Option	Type	Default Value	Mandatory	Description
				effect on whether custom `stateStorage` is passed.
`disableWebSockets`	boolean	`false`	No	Disable WebSockets for connections to the server. Only effective if the default transport is used. If you disable WebSockets, you should also pass this option to startSession().
`logger`	function	[internal console.log-based logger]	No	A function responsible for logging. You can pass an arbitrary number of different argument types (similar to console.log). For example: console.log.bind(console).

> **Tip**
>
> To find out how to control the chat session once it is started, see Controlling the chat session.