



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.

## API Reference

`restoreChat(options)`

# restoreChat(options)

## Contents

- [1 restoreChat\(options\)](#)
  - [1.1 Description](#)
  - [1.2 Returned promise](#)
  - [1.3 Options](#)

## Description

Use this method to restore the chat widget after page reload/navigation.

### Important

Currently, this method only works for "embedded" mode.

If the previous page was unloaded during the registration phase, `restoreChat` will try to restore the chat widget with the registration form.

## Returned promise

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

### done

This method can be used to get access to chat session service API.

```
chat.restoreChat(options).done(function(session) {  
  // session.sendMessage, session.onAgentConnected and all other methods are at your disposal.  
});
```

See [Chat Service JS API](#) for documentation about using the session API. If you need to access Chat Session API, you will probably want to get the access not only in cases when session is restored, but also when it is started fresh. You can use `startChat`'s "done" callback for this. See [Getting access to Chat Session API](#) for more info.

### Important

The promise is not resolved before the chat session is created. This means that if the chat widget is restored during the registration phase (the registration form is displayed to the user), the **done** callback is not sent until the registration is complete and processed by server.

### fail

If chat restoration fails because of an error (and not because the chat session does not exist), the **fail** callback receives an event argument with an error property, similar to the `startChat().fail` callback.

```
chat.restoreChat(options)  
  .fail(function(event) {
```

## restoreChat(options)

---

```
// If there was 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

Some chat states are restored automatically after page reload/navigation. However, most options must be passed to `restoreChat` directly. Supported options are:

- **embedded** — Must be explicitly passed as `true`. Otherwise an error occurs.
- **transport** — Include if providing a custom transport.
- **registration** — If you are using a custom function for registration and you want this registration to be restored, you must pass this option to both `startChat` and `restoreChat`.
- **ui** — If you want to customize/disable the chat UI, pass this option to both `startChat` and `restoreChat`.
- **localization** — For localization, pass the custom localization URL to both `startChat` and `restoreChat`.
- **debug** — If you want logs enabled, you must pass this option to `restoreChat` explicitly.
- **logger**
- **maxOfflineDuration**
- **disableWebSockets**