

3.18 Desktop JavaScript API Specification

Bright Pattern Documentation

Generated: 11/01/2024 12:03 am

Content is available under license unless otherwise noted.

Table of Contents

Table of Contents	2
Purpose	4
Audience	4
General Information	4
Dial Number	4
Syntax	5
Parameters	5
Select Service	5
Syntax	5
Parameters	5
Single-Step Transfer	5
Syntax	5
Parameters	5
Single-Step Conference	6
Syntax	6
Parameters	6
Terminate Interaction	6
Syntax	6
Set Notes	6
Syntax	6
Parameters	6
Set Disposition	7
Syntax	7
Parameters	7
Complete Interaction	7
Syntax	7
Complete Interaction with Disposition and Notes	7
Syntax	7
Parameters	7
Get Call Recording Status	8
Syntax	8
Parameters	8
Start Call Recording	8
Syntax	8
Parameters	8
Stop Call Recording	8
Syntax	8
Parameters	9
Mute Call Recording	9
Syntax	9
Parameters	9
Unmute Call Recording	9
Syntax	9
Parameters	9
Get Screen Recording Status	9
Syntax	10
Parameters	10
Stop Screen Recording	10
Syntax	10
Parameters	10
Pause Screen Recording	10
Syntax	10

Parameters	10
Resume Screen Recording	11
Syntax	11
Parameters	11
Pause Call and Screen Recording	11
Resume Call and Screen Recording	11
Callback	12
Syntax	12
Parameters	12

Purpose

The Bright Pattern Contact Center *Desktop JavaScript API Specification* describes the Desktop JavaScript API that provides access to a number of functions of the Bright Pattern Contact Center Agent Desktop application from the web pages or IFrames loaded into this application from other domains.

For general information about the Bright Pattern Contact Center Agent Desktop application, see the [Agent Guide](#). For information about loading web pages into Agent Desktop from scenarios, see the *Scenario Builder Reference Guide*, section [Web Screen Pop](#) block.

[Next >](#)

Audience

This guide is intended for the IT personnel responsible for the data infrastructure of Bright Pattern Contact Center-based contact centers. Readers of this guide are expected to have expertise in web application development as well as a solid understanding of contact center operations.

[< Previous](#) | [Next >](#)

General Information

The API supports a number of desktop telephony functions, including call initiation, voice and screen recording, interaction completion, and the setting of interaction dispositions and notes.

These functions are available via a global object called *window.bpspat.api*.

For example:

```
window.bpspat.api.dialNumber("1234567");
```

The object is created when the script file is included in the web page loaded into the Agent Desktop application as follows:

```
<script type="text/javascript" src="[agent-desktop-web-server]/app/libs/servicepatternapi-dev.js"></script>
```

For example:

```
<script type="text/javascript" src="https://barco.brightpattern.com/app/libs/servicepatternapi-dev.js"></script>
```

[< Previous](#) | [Next >](#)

Dial Number

This function makes a call to the specified number.

For more information, see the Bright Pattern Contact Center *Agent Guide*, section [How to Make an Internal Call](#) and section [How to Make an Outbound Call](#).

Syntax

dialNumber(number);

Parameters

number – String containing the number to be dialed; required

[< Previous](#) | [Next >](#)

Select Service

This function selects the service that will be associated with subsequent call attempts (until another service is selected using this method or via Agent Desktop)

For more information, see the Bright Pattern Contact Center *Agent Guide*, section [How to Make an Internal Call](#) and section [How to Make an Outbound Call](#).

Syntax

selectService(name);

Parameters

name – String containing the name of the selected service; optional

If omitted or empty, subsequent call attempts will not be associated with any service.

[< Previous](#) | [Next >](#)

Single-Step Transfer

This function initiates a [single-step \(blind\) transfer](#) of the current call to the specified number.

Syntax

singleStepTransfer(number);

Parameters

number – string containing the number to which the call is to be transferred; required

[< Previous](#) | [Next >](#)

Single-Step Conference

This function initiates a [single-step conference](#) with the current party on the call and the party at the specified number.

Syntax

```
singleStepConference(number);
```

Parameters

number – string containing the number for which the call is to be conferenced; required

[< Previous](#) | [Next >](#)

Terminate Interaction

Unlike [Complete Interaction](#), this method only [releases the communication channel](#) of the current interaction.

For example, if after-call work (ACW) is configured for the corresponding service, interaction processing will continue until ACW is completed.

Syntax

```
terminateInteraction();
```

[< Previous](#) | [Next >](#)

Set Notes

This function sets the [free-form notes](#) for the current interaction to the specified string.

Syntax

```
setNotes(notes);
```

Parameters

notes – String containing the interaction notes

[< Previous](#) | [Next >](#)

Set Disposition

This function sets the [disposition](#) for the current interaction to the value corresponding to the [alphanumeric code specified for this disposition](#).

Syntax

```
setDisposition(dispositionCode);
```

Parameters

dispositionCode – String containing the numeric code of the desired disposition

[< Previous](#) | [Next >](#)

Complete Interaction

This function [completes](#) the current interaction.

If the interaction [requires a disposition](#) in order to be completed, Agent Desktop will show a warning message.

Syntax

```
completeInteraction();
```

[< Previous](#) | [Next >](#)

Complete Interaction with Disposition and Notes

This function [completes](#) the current interaction and [sets its disposition and notes](#) to the specified values (i.e., combines the actions of the [Set Notes](#), [Set Disposition](#), and [Complete Interaction](#) methods).

Syntax

```
completeInteractionWithDisp(dispositionCode, notes);
```

Parameters

dispositionCode – String containing the [alphanumeric code](#) of the desired disposition

notes – String containing the interaction notes

Get Call Recording Status

This function requests the current status of voice recording of the current call.

Syntax

```
getCallRecordingStatus(callback);
```

Parameters

[*callback*](#) – Function that will be called upon completion of the method

Start Call Recording

This function starts voice recording for the current call.

For more information, see the Bright Pattern Contact Center *Agent Guide*, section [How to Record a Call](#).

Syntax

```
startCallRecording(callback);
```

Parameters

[*callback*](#) – Function that will be called upon completion of the method; see *Callback*

Note that prior to Bright Pattern Contact Center version 3.8, this method was called *startCurrentCallRecording()*; and backward compatibility is preserved.

Stop Call Recording

This function stops voice recording for the current call.

Syntax

```
stopCallRecording(callback);
```


Parameters

callback – Function that will be called upon completion of the method; see [Callback](#)

Note that prior to Bright Pattern Contact Center version 3.8, this method was called *stopCurrentCallRecording()*; and backward compatibility is preserved.

[< Previous](#) | [Next >](#)

Mute Call Recording

This function mutes voice recording for the current call. Unlike [Stop Call Recording](#), this method will continue voice recording for the current call, but any voice signal will be replaced with silence.

Syntax

```
muteCallRecording(callback);
```

Parameters

[callback](#) – Function that will be called upon completion of the method

[< Previous](#) | [Next >](#)

Unmute Call Recording

This function unmutes the previously [muted](#) voice recording for the current call.

Syntax

```
unmuteCallRecording(callback);
```

Parameters

[callback](#) – Function that will be called upon completion of the method

[< Previous](#) | [Next >](#)

Get Screen Recording Status

This function requests the current status of screen recording of the user session.

Syntax

```
getScreenRecordingStatus(callback);
```

Parameters

[*callback*](#) – Function that will be called upon completion of the method

[< Previous](#) | [Next >](#)

Stop Screen Recording

This function stops screen recording of the user session.

Syntax

```
stopScreenRecording(callback);
```

Parameters

[*callback*](#) – Function that will be called upon completion of the method

[< Previous](#) | [Next >](#)

Pause Screen Recording

This function pauses screen recording of the user session.

For the period when screen recording is paused, the recording will contain a static snapshot of the desktop at the moment when pause was applied.

Syntax

```
pauseScreenRecording(callback);
```

Parameters

[*callback*](#) – Function that will be called upon completion of the method

Resume Screen Recording

This function resumes the previously [paused](#) screen recording of the user session.

Syntax

```
resumeScreenRecording(callback);
```

Parameters

[callback](#) - Function that will be called upon completion of the method

Pause Call and Screen Recording

Syntax	<code>parent.frames.postMessage({'command': "MUTE_CALL_RECORDINGS"}, '*');</code>
Description	Mutes call recording (e.g., when an agent opens a payment screen, a third-party web application loaded into Agent Desktop may issue the Javascript command). Note that it does not require loading API code on the application's page.

Syntax	<code>parent.frames.postMessage({'command': "MUTE_SCREEN_RECORDINGS"}, '*');</code>
Description	Mutes screen recording (e.g., when an agent opens a payment screen, a third-party web application loaded into Agent Desktop may issue the Javascript command). Note that it does not require loading API code on the application's page.

Resume Call and Screen Recording

Syntax	<code>parent.frames.postMessage({'command': "UNMUTE_CALL_RECORDINGS"}, '*');</code>
Description	Resumes call recording (e.g., when an agent exits a payment screen, a third-party web application loaded into Agent Desktop may issue the Javascript command). Note that it does not require loading API code on the application's page.

Syntax	<code>parent.frames.postMessage({'command': "UNMUTE_SCREEN_RECORDINGS"}, '*');</code>
Description	Resumes screen recording (e.g., when an agent exits a payment screen, a third-party web application loaded into Agent Desktop may issue the Javascript command). Note that it does not require loading API code on the application's page.

[< Previous](#) | [Next >](#)

Callback

This is the function that is called upon completion of the call/screen recording methods.

Syntax

```
callback = function(data) {  
  
  data.recording = 1/0;  
  
  data.muted = 1/0;  
  
  data.paused = 1/0;  
  
}
```

Parameters

data.recording – Boolean; indicates if voice/screen recording is currently in progress; note that voice recording is applied to interactions (calls), while screen recording is applied to user sessions

data.muted – Boolean; indicates if voice recording is currently muted; returned only for the voice recording methods

data.paused – Boolean; indicates if screen recording is currently paused; returned only for the screen recording methods

[< Previous](#)