

5.3 Desktop JavaScript API Specification

Bright Pattern Documentation

Generated: 6/14/2021 6:25 am

Content is available under license unless otherwise noted.

Table of Contents

Table of Contents	2
Purpose	4
Audience	4
General Information	4
Example	4
Object Creation	4
Example	4
Dial Number	5
Request	5
Syntax	5
Parameters	5
Select Service	5
Request	5
Syntax	5
Parameters	5
setRescheduleWindow	6
Request	6
Syntax	6
Parameters	6
Single-Step Transfer	6
Request	6
Syntax	6
Parameters	6
Single-Step Conference	7
Request	7
Syntax	7
Parameters	7
Complete Interaction	7
Request	7
Syntax	7
Complete Interaction with Disposition and Notes	7
Request	8
Syntax	8
Parameters	8
postVariable	8
Syntax	8
Parameters	8
Set Disposition	8
Request	9
Syntax	9
Parameters	9
Set Disposition By Name	9
Request	9
Syntax	9
Parameters	9
Set Notes	9
Request	9
Syntax	9
Parameters	9
Set Reporting Custom Field	10
Request	10
Syntax	10
Parameters	10
Terminate Interaction	10
Request	10
Syntax	10

Callback	11
Syntax	11
Parameters	11
Pause Call and Screen Recording	11
Mute Call Recordings	11
Syntax	11
Mute Screen Recordings	11
Syntax	12
Resume Call and Screen Recording	12
Unmute Call Recordings	12
Syntax	12
Unmute Screen Recordings	12
Syntax	12
Get Call Recording Status	12
Request	12
Syntax	13
Parameters	13
Mute Call Recording	13
Request	13
Syntax	13
Parameters	13
Unmute Call Recording	13
Request	13
Syntax	13
Parameters	13
Start Call Recording	14
Request	14
Syntax	14
Parameters	14
Stop Call Recording	14
Request	14
Syntax	14
Parameters	14
Get Screen Recording Status	15
Request	15
Syntax	15
Parameters	15
Pause Screen Recording	15
Request	15
Syntax	15
Parameters	15
Resume Screen Recording	16
Request	16
Syntax	16
Parameters	16
Stop Screen Recording	16
Request	16
Syntax	16
Parameters	16

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.

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.

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*.

Example

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

Object Creation

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]/libs/servicepatternapi-dev.js"></script>
```

or

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

Example

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

or

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

Dial Number

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](#).

Request

Syntax

```
dialNumber(number);
```

Parameters

Parameter	Type	Optional/Required	Description	Example
number	String	Required	The number to be dialed	"11234567"

Select Service

Selects the service that will be associated with subsequent call attempts (until another service is selected using this method or via Agent Desktop). If omitted or empty, subsequent call attempts will not be associated with any service.

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](#).

Request

Syntax

```
selectService(name);
```

Parameters

Parameter	Type	Optional/Required	Description	Example
name	String	Optional	The name of the selected service	"Voice Service"

setRescheduleWindow

Allows you to reschedule outbound dialing retry time to be within a specific timeframe with the option to specify a time zone.

The reschedule window will only affect outbound campaigns when a non-final disposition is selected. The data is retained in Agent Desktop until changed or the interaction is completed.

Request

Syntax

```
setRescheduleWindow(numberToDial, fromTime, untilTime, timezoneName);
```

Parameters

Parameter	Data Type	Required/Optional	Description	Example
numberToDial	String	Required	The phone number to dial	"11234567"
fromTime	String	Required	The start of the reschedule timeframe in "YYYY-MM-DD HH24:MM:SS" format	"2019-09-12 15:30:00"
untilTime	String	Required	The end of the reschedule timeframe in "YYYY-MM-DD HH24:MM:SS" format (must be after the starting time)	"2019-09-13 15:30:00"
timezoneName	String	Optional	The name of the timezone; if omitted, the timezone will be assumed to be the record's detected timezone	"America/Los_Angeles"

Single-Step Transfer

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

Request

Syntax

```
singleStepTransfer(number);
```

Parameters

Parameter	Type	Optional/Required	Description	Example
-----------	------	-------------------	-------------	---------

number	String	Required	The number to which the call is to be transferred	14151234567
--------	--------	----------	---	-------------

Single-Step Conference

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

Request

Syntax

```
singleStepConference(number);
```

Parameters

Parameter	Type	Optional/Required	Description	Example
number	String	Required	The number for which the call is to be conferenced	"16501234567"

Complete Interaction

[Completes](#) the current interaction.

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

Request

Syntax

```
completeInteraction();
```

Complete Interaction with Disposition and Notes

[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).

Request

Syntax

completeInteractionWithDisp(dispositionCode, notes);

Parameters

Parameter	Type	Optional/Required	Description	Example
dispositionCode	String	Required	The alphanumeric code of the desired disposition	"12345"
notes	String	Optional	The interaction notes	"Could not resolve. Not enough information."

postVariable

This function enables a variable to be pushed to a scenario as if the [Set Variable](#) block is included; the variable is then available in scenarios and [workflows](#).

When invoking the postVariable method for a variable that starts with *ActivityHistory*, Agent Desktop memorizes the value of the variable to post it in activity history (see the *Form Builder Reference Guide*, section [How to Configure Activity History Forms](#) for more information on mapping activity history values). If an activity history field was marked for export in campaign results, it will be possible to export it with campaign results. The data is retained in Agent Desktop until changed or the interaction is completed. If a form is displaying that data, the form is also updated to reflect the new value. If the data changes in the form, it changes the value to be submitted.

Syntax

postVariable(name, value);

Parameters

Parameter	Data Type	Description
name	String	The name of the desired variable
value	String	The resulting value of the variable

Set Disposition

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

Request

Syntax

```
setDisposition(dispositionCode);
```

Parameters

Parameter	Type	Optional/Required	Description	Example
dispositionCode	String	Required	The numeric code of the desired disposition	"12345"

Set Disposition By Name

Sets the [disposition](#) for the current interaction by the [name specified of the disposition](#).

Request

Syntax

```
setDispositionByName(name);
```

Parameters

Parameter	Type	Optional/Required	Description	Example
name	String	Required	The name of the desired disposition	"Product sold"

Set Notes

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

Request

Syntax

```
setNotes(notes);
```

Parameters

Parameter	Type	Optional/Required	Description	Example
notes	String	Required	The interaction notes	"Third time calling today."

Set Reporting Custom Field

Allows [custom reporting fields](#) to be entered during an agent's interaction with a customer; the method works like the [Set Custom Reporting Field](#) scenario block.

Request

Syntax

```
setReportingCustomField(name, value);
```

Parameters

Parameter	Type	Optional/Required	Description	Example
name	String	Required	The name of the custom reporting field	"custom1"
value	String	Required	The resulting value of the custom reporting field	"\$(variable_in_field)"

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.

Request

Syntax

```
terminateInteraction();
```

Callback

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

Parameter	Type	Description	Example
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	"1" (for true; recording is on)
data.muted	Boolean	Indicates if voice recording is currently muted; returned only for the voice recording methods	"0" (for false; recording is not muted)
data.paused	Boolean	Indicates if screen recording is currently paused; returned only for the screen recording methods	"1" (for false, recording is not paused)

Pause Call and Screen Recording

Mutes call recordings and screen recordings using the following commands.

Mute Call Recordings

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

```
parent.frames.postMessage({'command': "MUTE_CALL_RECORDINGS"}, '*');
```

Mute Screen Recordings

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.

Syntax

```
parent.frames.postMessage({"command": "MUTE_SCREEN_RECORDINGS"}, '*');
```

Resume Call and Screen Recording

Uses the following commands to resume a call recording or screen recording that is currently muted.

Unmute Call Recordings

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

```
parent.frames.postMessage({"command": "UNMUTE_CALL_RECORDINGS"}, '*');
```

Unmute Screen Recordings

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.

Syntax

```
parent.frames.postMessage({"command": "UNMUTE_SCREEN_RECORDINGS"}, '*');
```

Get Call Recording Status

Requests the current status of voice recording of the current call.

Request

Syntax

```
getCallRecordingStatus(callback);
```

Parameters

Parameter	Type	Optional/Required	Description
callback	Function	Required	Function that will be called upon completion of the method

Mute Call Recording

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.

Request

Syntax

```
muteCallRecording(callback);
```

Parameters

Parameter	Type	Optional/Required	Description
callback	Function	Required	Function that will be called upon completion of the method

Unmute Call Recording

Unmutes the previously [muted](#) voice recording for the current call.

Request

Syntax

```
unmuteCallRecording(callback);
```

Parameters

Parameter	Type	Optional/Required	Description
callback	Function	Required	Function that will be called upon completion of the method

Start Call Recording

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

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

Request

Syntax

```
startCallRecording(callback);
```

Parameters

Parameter	Type	Optional/Required	Description
callback	Function	Required	Function that will be called upon completion of the method

Stop Call Recording

Stops voice recording for the current call.

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

Request

Syntax

```
stopCallRecording(callback);
```

Parameters

Parameter	Type	Optional/Required	Description
callback	Function	Required	Function that will be called upon completion of the method

Get Screen Recording Status

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

Request

Syntax

```
getScreenRecordingStatus(callback);
```

Parameters

Parameter	Type	Optional/Required	Description
callback	Function	Required	Function that will be called upon completion of the method

Pause Screen Recording

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.

Request

Syntax

```
pauseScreenRecording(callback);
```

Parameters

Parameter	Type	Optional/Required	Description
callback	Function	Required	Function that will be called upon completion of the method

Resume Screen Recording

Resumes the previously [paused](#) screen recording of the user session.

Request

Syntax

```
resumeScreenRecording(callback);
```

Parameters

Parameter	Type	Optional/Required	Description
callback	Function	Required	Function that will be called upon completion of the method

Stop Screen Recording

Stops screen recording of the user session.

Request

Syntax

```
stopScreenRecording(callback);
```

Parameters

Parameter	Type	Optional/Required	Description
callback	Function	Required	Function that will be called upon completion of the method