

# 3.11 Desktop JavaScript API Specification

## Bright Pattern Documentation

Generated: 11/01/2024 12:29 am

Content is available under license unless otherwise noted.

## Table of Contents

Table of Contents	2
Purpose	3
Audience	3
General Information	3
Dial Number	3
Select Service	3
Single-Step Transfer	4
Single-Step Conference	4
Terminate Interaction	4
Set Notes	4
Set Disposition	5
Complete Interaction	5
Complete Interaction with Disposition and Notes	5
Get Call Recording Status	5
Start Call Recording	5
Stop Call Recording	6
Mute Call Recording	6
Unmute Call Recording	6
Get Screen Recording Status	6
Stop Screen Recording	7
Pause Screen Recording	7
Resume Screen Recording	7
Callback	7

# Purpose

ServicePattern Desktop JavaScript API Specification describes the Desktop JavaScript API that provides access to a number of functions of the ServicePattern Agent Desktop application from the web pages or IFrames loaded into this application from other domains.

For general information about the ServicePattern Agent Desktop application, see [ServicePattern Agent Guide](#). For information about loading web pages into the Agent Desktop from scenarios, see description of the [Web Screen Pop](#) block in the *ServicePattern Scenario Builder Reference Guide*.

# Audience

This guide is intended for the IT personnel responsible for the data infrastructure of the ServicePattern-based contact centers. Readers of this guide are expected to have expertise in web application development as well as 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 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>
```

# Dial Number

<b>Syntax</b>	<i>dialNumber(number);</i>
<b>Description</b>	Makes a call to the specified number. For more information, see sections <a href="#">How to Make an Internal Call</a> and <a href="#">How to Make an Outbound Call</a> of the <i>ServicePattern Agent Guide</i> .
<b>Parameters</b>	<i>number</i> – string containing the number to be dialed; required

# Select Service

<b>Syntax</b>	<i>selectService(name);</i>
<b>Description</b>	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 sections <a href="#">How to Make an Internal Call</a> and <a href="#">How to Make an Outbound Call</a> of the <i>ServicePattern Agent Guide</i> .
<b>Parameters</b>	<i>name</i> – string containing the name of the selected service; optional; if omitted or empty, subsequent call attempts will not be associated with any service

## Single-Step Transfer

<b>Syntax</b>	<i>singleStepTransfer(number);</i>
<b>Description</b>	Initiates a <a href="#">single-step (blind) transfer</a> of the current call to the specified number.
<b>Parameters</b>	<i>number</i> – string containing the number that the call is to be transferred to; required

## Single-Step Conference

<b>Syntax</b>	<i>singleStepConference(number);</i>
<b>Description</b>	Initiates a <a href="#">single-step conference</a> with the current party on the call and the party at the specified number.
<b>Parameters</b>	<i>number</i> – string containing the number that the call is to be conferenced with; required

## Terminate Interaction

<b>Syntax</b>	<i>terminateInteraction();</i>
<b>Description</b>	Unlike <a href="#">Complete Interaction</a> , this method only <a href="#">releases the communication channel</a> of the current interaction. I.e., if after-call work (ACW) is configured for the corresponding service, interaction processing will continue until ACW is completed.

## Set Notes

<b>Syntax</b>	<i>setNotes(notes);</i>
<b>Description</b>	Sets the <a href="#">free-form notes</a> for the current interaction to the specified string.
<b>Parameters</b>	<i>notes</i> – string containing the interaction notes

## Set Disposition

<b>Syntax</b>	<code>setDisposition(dispositionCode);</code>
<b>Description</b>	Sets the <a href="#">disposition</a> for the current interaction to the value corresponding to the <a href="#">alphanumeric code specified for this disposition</a> .
<b>Parameters</b>	<code>dispositionCode</code> – string containing the numeric code of the desired disposition

## Complete Interaction

<b>Syntax</b>	<code>completeInteraction();</code>
<b>Description</b>	<a href="#">Completes</a> the current interaction. If the interaction <a href="#">requires a disposition</a> in order to be completed, the Agent Desktop will show a warning message.

## Complete Interaction with Disposition and Notes

<b>Syntax</b>	<code>completeInteractionWithDisp(dispositionCode, notes);</code>
<b>Description</b>	<a href="#">Completes</a> the current interaction and <a href="#">sets its disposition and notes</a> to the specified values (i.e., combines the actions of the <a href="#">Set Notes</a> , <a href="#">Set Disposition</a> , and <a href="#">Complete Interaction</a> methods).
<b>Parameters</b>	<code>dispositionCode</code> – string containing the <a href="#">alphanumeric code</a> of the desired disposition <code>notes</code> – string containing the interaction notes

## Get Call Recording Status

<b>Syntax</b>	<code>getCallRecordingStatus(callback);</code>
<b>Description</b>	Requests the current status of voice recording of the current call.
<b>Parameters</b>	<a href="#">callback</a> – function that will be called upon completion of the method

## Start Call Recording

<b>Syntax</b>	<code>startCallRecording(callback);</code>
<b>Description</b>	Starts voice recording for the current call. For more information, see section <a href="#">How to Record a Call</a> of the <i>ServicePattern Agent Guide</i> .

<b>Parameters</b>	<a href="#">callback</a> – function that will be called upon completion of the method; see <i>Callback</i>
<b>Note</b>	Prior to version 3.8, this method was called <i>startCurrentCallRecording()</i> ; backward compatibility is preserved

## Stop Call Recording

<b>Syntax</b>	<i>stopCallRecording(callback);</i>
<b>Description</b>	Stops voice recording for the current call.
<b>Parameters</b>	<a href="#">callback</a> – function that will be called upon completion of the method; see <i>Callback</i>
<b>Note</b>	Prior to version 3.8, this method was called <i>stopCurrentCallRecording()</i> ; backward compatibility is preserved

## Mute Call Recording

<b>Syntax</b>	<i>muteCallRecording(callback);</i>
<b>Description</b>	Mutes voice recording for the current call. Unlike <a href="#">Stop Call Recording</a> , this method will continue voice recording for the current call, but any voice signal will be replaced with silence.
<b>Parameters</b>	<a href="#">callback</a> – function that will be called upon completion of the method

## Unmute Call Recording

<b>Syntax</b>	<i>unmuteCallRecording(callback);</i>
<b>Description</b>	Unmutes the previously <a href="#">muted</a> voice recording for the current call.
<b>Parameters</b>	<a href="#">callback</a> – function that will be called upon completion of the method

## Get Screen Recording Status

<b>Syntax</b>	<i>getScreenRecordingStatus(callback);</i>
<b>Description</b>	Requests the current status of screen recording of the user session.
<b>Parameters</b>	<a href="#">callback</a> – function that will be called upon completion of the method

## Stop Screen Recording

<b>Syntax</b>	<code>stopScreenRecording(callback);</code>
<b>Description</b>	Stops screen recording of the user session.
<b>Parameters</b>	<a href="#">callback</a> - function that will be called upon completion of the method

## Pause Screen Recording

<b>Syntax</b>	<code>pauseScreenRecording(callback);</code>
<b>Description</b>	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.
<b>Parameters</b>	<a href="#">callback</a> - function that will be called upon completion of the method

## Resume Screen Recording

<b>Syntax</b>	<code>resumeScreenRecording(callback);</code>
<b>Description</b>	Resumes the previously <a href="#">paused</a> screen recording of the user session.
<b>Parameters</b>	<a href="#">callback</a> - function that will be called upon completion of the method

## Callback

<b>Syntax</b>	<pre>callback = function(data) {     data.recording = 1/0;     data.muted = 1/0;     data.paused = 1/0; }</pre>
<b>Description</b>	Function that is called upon completion of the call/screen recording methods.

<b>Parameters</b>	<p><i>data.recording</i> – 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</p> <p><i>data.muted</i> – boolean; indicates if voice recording is currently muted; returned only for the voice recording methods</p> <p><i>data.paused</i> – boolean; indicates if screen recording is currently paused; returned only for the screen recording methods</p>
-------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------