

# 5.3 Embedded Agent Desktop SDK Specification

## Bright Pattern Documentation

Generated: 9/25/2021 8:33 am

Content is available under license unless otherwise noted.

## Table of Contents

Table of Contents	2
Purpose	3
Try It Now	3
Audience	3
General Information About the Embedded Agent Desktop API	3
Supported Methods	3
Supported Events	4
Overview	4
Embedding Agent Desktop	4
Sample Code and Example	4
Try It Now	4
Code	4
Overview	6

# Purpose

The *Embedded Agent Desktop SDK Specification* describes the Bright Pattern Agent Desktop SDK, which incorporates the Bright Pattern [Embedded Agent Desktop API](#).

The SDK supports requests for information about agent states and interactions, as well as control over agent and supervisor login, status, call flow, call recordings, transfers, and outbound dialing. The SDK is available in the latest release of Bright Pattern Contact Center.

## Try It Now

You can give the SDK a spin by copying the following URL into your web browser's address field and replacing the placeholder "corp" with the name of your Bright Pattern Contact Center instance:

<https://corp.brightpattern.com/agentdesktop/AgentDesktopSdkTest.jsp>

Then log in to Agent Desktop and click **Get State** to get started.

ControlRecord1b-53.png



## Audience

This specification 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 and resources that are involved in such operations.

## General Information About the Embedded Agent Desktop API

The Embedded Agent Desktop API is used for requesting for information about agent states and interactions, as well as controlling agent and supervisor login, status, call flow, call recordings, transfers, and outbound dialing.

The methods and events of the API are specified in the [Embedded Agent Desktop API Specification](#).

### Supported Methods

The Embedded Agent Desktop API supports the following types of methods: agent state, dialing and transfer, completing and terminating an interaction, and call recordings control.

**Agent State** methods are used to request an agent's state and interaction information, as well as to change an agent's state.

- [getState](#)
- [setStatus](#)

**Dialing and Transfer** methods are used for dialing phone numbers and transferring calls.

- [dialNumber](#)
- [selectService](#)
- [singleStepTransfer](#)
- [singleStepConference](#)

**Completing and Terminating an Interaction** methods are used for completing and terminating interactions.

- [setDisposition](#)
- [terminateInteraction](#)
- [completeInteraction](#)
- [completeInteractionWithDisp](#)

**Call Recordings Control** methods are used for controlling call recordings and getting the status of call recordings.

- [muteCallRecordings\(\)](#)
- [unmuteCallRecordings\(\)](#)
- [getCallRecordingStatus\(callback\)](#)
- [startCallRecording\(callback\)](#)

- [stopCallRecording\(callback\)](#)
- [muteCallRecording\(callback\)](#)
- [unmuteCallRecording\(callback\)](#)

## Supported Events

The Embedded Agent Desktop API supports the following types of events: agent login, agent status, and call flow.

**Agent Login** events are used for showing whether an agent is logged in or out of Agent Desktop.

- [onAgentLogin\(callback\)](#)

**Agent Status** events are used for retrieving an agent's state and reason.

- [onStatusChange\(callback\)](#)

**Call Flow** events are used for getting information about active interactions and completed interactions.

- [addInteractionRenderedHandler\(callback\)](#)
- [addInteractionCompletedHandler\(callback\)](#)

## Overview

There are many ways to call the Embedded Agent Desktop API methods and events. This section provides [SDK sample code](#), which embeds the Agent Desktop widget on a basic webpage and attaches all the methods and events of the Embedded Agent Desktop API to web elements.

See the [Use Cases](#) section of this guide for more ways to use the SDK.

## Embedding Agent Desktop

The Embedded Agent Desktop widget incorporates the Embedded Agent Desktop API, a browser-based JavaScript API that allows a webpage to communicate with the Bright Pattern Agent Desktop application embedded on that page in iframe. Iframe (Inline Frame) is an HTML element that is used to insert interactive content from a different source onto a website.

Some domains do not allow content from other sources to be embedded using iframe, and you will experience errors when trying to embed Agent Desktop.

If this occurs, please contact your service provider to enable a domain for your contact center.

## Sample Code and Example

This section provides a functional example of the SDK, along with sample code for a webpage that includes all the Embedded Agent Desktop API methods and events that are called from Bright Pattern's embedded Agent Desktop widget.

### Try It Now

You can give the SDK a spin by copying the following URL into your web browser's address field and replacing the placeholder "corp" with the name of your Bright Pattern Contact Center instance:

<https://corp.brightpattern.com/agentdesktop/AgentDesktopSdkTest.jsp>

### Code

This sample code is provided for reference only, as an example of what you can do with the SDK and the API it incorporates. You should copy all and save it as HTML for viewing in your web browser.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>AgentDesktop SDK Example</title><!--This is a test HTML page.-->
    <script type="text/javascript" src="libs/servicepattern-sdk-v1.js"></script>//The API in JavaScript. You paste the script on the webpage.
  </head>
  <body>
    <iframe id="repeater" style="position:absolute; right: 8px; width: 30%; height: 90%;"></iframe>
    //You can adjust the appearance of the Agent Desktop widget by tweaking iframe properties. Repeater is the Agent Desktop widget.

    <button onclick="bpspat.api.getState(log)">Get State</button>
    //Gets the agent state. In this example, simple HTML buttons are used for executing the Outpage JS API events. The buttons handle the events.
    <br>
    <br>
    <button onclick="bpspat.api.setStatus('READY')">Go To Ready</button>
    //Sets agent state to "Ready".
    <br>
    <br>
    <button onclick="bpspat.api.logout()">Logout</button>
    //Logs out agent.
    <br>
```

```

<br>
<button onclick="bpspat.api.muteCallRecordings()">Mute Call Recordings</button>
//Mutes call recordings.
<button onclick="bpspat.api.unmuteCallRecordings()">Unmute Call Recordings</button>
//Unmutes call recordings.
<button onclick="bpspat.api.muteScreenRecordings()">Mute Screen Recordings</button>
//Mutes screen recordings.
<button onclick="bpspat.api.unmuteScreenRecordings()">Unmute Screen Recordings</button>
//Unmutes screen recordings.
<br>
<br>
<input id="service" type="text" placeholder="Service name"/>
<button onclick="bpspat.api.selectService(document.getElementById('service').value)">Change Service by Name</button>
//Selects the name of the contact center service.
<br>
<br>
<input id="number" type="text" placeholder="Dial number"/>
//Parameters of bpspat.api.dialNumber() event.
<button onclick="bpspat.api.dialNumber(document.getElementById('number').value)">Dial</button>
//Dials phone number. Note that you have to declare the phone number as the "input id" parameter value above.
<br>
<br>
<input id="item" type="text" placeholder="Interaction Id, see Logs"/>
//Parameters of bpspat.api.terminateInteraction() event.
<br>
<br>
<button onclick="bpspat.api.terminateInteraction(document.getElementById('item').value)">Terminate</button>
//Ends the interaction. Note that you have to declare the interaction ID number as the "input id" parameter value above.
<button onclick="bpspat.api.completeInteraction(document.getElementById('item').value)">Complete</button>
//Marks the interaction as complete. Note that you have to declare the interaction ID number as the "input id" parameter value above.
<br>
<br>
<button onclick="bpspat.api.getCallRecordingStatus(document.getElementById('item').value)">Get Call Recording Status</button>
//Gets call recording status using the interaction ID number. Note that you have to declare the interaction ID number as the "input id" above.
<button onclick="bpspat.api.startCallRecording(document.getElementById('item').value)">Start Call Recording</button>
//Starts call recording using the interaction ID number. Note that you have to declare the interaction ID number as the "input id" above.
<button onclick="bpspat.api.stopCallRecording(document.getElementById('item').value)">Stop Call Recording</button>
//Stops call recording using the interaction ID number. Note that you have to declare the interaction ID number as the "input id" above.
<button onclick="bpspat.api.muteCallRecording(document.getElementById('item').value)">Mute Call Recording</button>
//Mutes call recording of the specified interaction ID.
<button onclick="bpspat.api.unmuteCallRecording(document.getElementById('item').value)">Unmute Call Recording</button>
//Unmutes call recording of the specified interaction ID.
<br>
<br>
<button onclick="bpspat.api.getScreenRecordingStatus(document.getElementById('item').value)">Get Screen Recording Status</button>
//Gets screen recording status of the specified interaction ID.
<button onclick="bpspat.api.muteScreenRecording(document.getElementById('item').value)">Mute Screen Recording</button>
//Mutes screen recording of the specified interaction ID.
<button onclick="bpspat.api.unmuteScreenRecording(document.getElementById('item').value)">Unmute Screen Recording</button>
//Unmutes screen recording of the specified interaction ID.
<button onclick="bpspat.api.stopScreenRecording(document.getElementById('item').value)">Stop Screen Recording</button>
//Stops screen recording of the specified interaction ID.
<br>
<br>
<input id="transfer_number" type="text" placeholder="Number"/>
//Parameters of bpspat.api.singleStepTransfer() event and bpspat.api.singleStepConference() event.
<button onclick="bpspat.api.singleStepTransfer(document.getElementById('transfer_number').value)">Single Step Transfer</button>
//Transfers the active call to a specified phone number. Note that you have to specify the phone number as the "transfer_number" parameter in the input id above.
<button onclick="bpspat.api.singleStepConference(document.getElementById('transfer_number').value)">Single Step Conference</button>
//Initiates a conference call with a specified phone number. Note that you have to specify the phone number as the "transfer_number" parameter in the input id above.
<br>
<br>
<input id="dispositionItem" type="text" placeholder="Item_id"/>
//Parameters of bpspat.api.setDispositionByName() event.
<input id="dispositionCode" type="text" placeholder="Disposition code"/>
//Parameters of bpspat.api.setDisposition() event.
<br>
<button onclick="bpspat.api.setDisposition(document.getElementById('dispositionItem').value, document.getElementById('dispositionCode').value)">Set Disposition By Code</button>
//Sets disposition of interaction by code (string).
<br>
<button onclick="bpspat.api.setDispositionByName(document.getElementById('dispositionItem').value, document.getElementById('dispositionCode').value)">Set Disposition By Name</button>
//Sets disposition of interaction by disposition name (string).
<br>
<br>
<input id="variableName" type="text" placeholder="Variable name"/>
//Parameters of bpspat.api.postVariable() event.
<input id="variableValue" type="text" placeholder="Variable value"/>
<button onclick="bpspat.api.postVariable(document.getElementById('dispositionItem').value, document.getElementById('variableName').value, document.getElementById('variableValue').value)">Se
</button>
</div>
</div>
<div id="log">Logs...</div>
<script type="text/javascript">
//initiate script with origin of AgentDesktop embedded on the page
window.bpspat.api.init(window.location.href.substring(0, window.location.href.indexOf("/agentdesktop/AgentDesktopSdkTest.jsp"));
//remove suffix, pass origin
document.getElementById("repeater").src = window.location.href.replace("AgentDesktopSdkTest", "UniversalRepeater");
function log(data) {
let div = document.createElement("div");
div.innerHTML = JSON.stringify(data);
document.getElementById("log").appendChild(div);
}
//Event handlers
window.bpspat.api.addAgentLoginHandler(log);
window.bpspat.api.addStatusChangeHandler(log);
window.bpspat.api.addInteractionRenderedHandler(log);
window.bpspat.api.addInteractionCompletedHandler(log);
window.bpspat.api.addInboundCallOfferedHandler(log);

```

```
window.bpspat.api.addInboundCallAnsweredHandler(log);
window.bpspat.api.addOutboundCallDialingHandler(log);
window.bpspat.api.addOutboundCallConnectedHandler(log);
window.bpspat.api.addCallEndedHandler(log);
window.bpspat.api.addCallHeldHandler(log);
window.bpspat.api.addCallResumedHandler(log);
window.bpspat.api.addBlindTransferHandler(log);
window.bpspat.api.addConsultTransferHandler(log);
window.bpspat.api.addCallRecordingStatusHandler(log);
window.bpspat.api.addScreenRecordingStatusHandler(log);
</script>
</body>
</html>
```

## Overview

This section describes some ways to use the Embedded Agent Desktop SDK, such as retrieving an agent's state and interaction status information, dialing numbers, transferring calls, wrapping up interactions, and controlling call recordings.