



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.

# Workspace Web Edition Developer's Guide and API Reference

Web Services and Applications 8.6.0

# Table of Contents

<b>Workspace Web Edition Developer's Guide and API Reference</b>	<b>3</b>
<b>Service Client API</b>	<b>4</b>
Agent Namespace	13
Email Namespace	16
Interaction Namespace	17
Media Namespace	29
System Namespace	33
Voice Namespace	38

# Workspace Web Edition Developer's Guide and API Reference

Welcome to the *Workspace Web Edition Developer's Guide and API Reference*. This document provides information about customizing Workspace Web Edition and working with its JavaScript-based APIs.

## JavaScript APIs

---

**Service Client API** — Use this API to integrate your own web application with Workspace Web Edition.

# Service Client API

## API Overview

You can use the Service Client API to customize how your web application or website integrates with Workspace Web Edition. Genesys provides this API, which is based on `window.postMessage`, so that your application can access the Workspace Web Edition object model and bypass the cross-domain security limitations.

You can use the Service Client API to perform the following actions:

- [Controlling call recording from a third-party application](#)
- [Embedding multiple third-party applications in Workspace](#)
- [Updating attached data from a third-party application](#)
- [Enabling click-to-dial from a third-party application](#)
- [Enabling Service Client API to invoke toast in Agent Desktop](#)
- [Controlling Case Selection from a Third Party Application](#)

## Controlling Call Recording from a Third-Party Application

Review the following methods for details about call recording control:

- [pauseCallRecording](#)
- [resumeCallRecording](#)
- [startCallRecording](#)
- [stopCallRecording](#)

The call recording state is stored in the `recordingState` attribute on the [interaction.Interaction](#) object.

## Embedding Multiple Third-Party Applications in Workspace

You can now set the [interaction.web-content](#) option to a list of option section names that correspond to web extension views. This means that you can configure Workspace to include more than one third-party web application, displayed as either a tab, a popup window, in the background at the interaction level, or hidden.

You should also make sure that the [service-client-api.accepted-web-content-origins](#) option references all the websites that should use the Service Client API.

See [Enabling integration of web applications in the agent interface](#) for details about setting up multiple web applications in Workspace.

## Updating Attached Data from a Third-Party Application

Review the following methods for details about updating attached data:

- `deleteUserData`
- `getByInteractionId`
- `getInteractions`
- `setUserData`

The user data is stored in the `userData` attribute on the `interaction.Interaction` object.

You should also be sure to configure the `service-client-api.user-data.read-allowed` and `service-client-api.user-data.write-allowed` options to enable read and write access to user data.

## Enabling Click-to-Dial from a Third-Party Application

If you configure Workspace Web Edition to display your web application in a new tab in the Workspace user interface (as described in [Enabling integration of web applications in the agent interface](#)), then the service API only gives access to the `dial` operation.

## Enabling Service Client API to invoke toast in Agent Desktop

Review the following methods for details about enabling and updating toast:

- `system.popupToast`
- `system.updateToast`
- `system.closeToast`

## Controlling Case Selection from a Third Party Application

Review the following method for details about case selecting control:

- `selectCaseByCaseId`

The case selection state is stored in the `isCaseSelected` attribute and the `isCaseExpanded` attribute on the `interaction.Interaction` object.

## Getting Started

Here's an overview of the steps you should to follow to access the API:

1. You have a web application that you've integrated in Workspace Web Edition. See [Enabling integration of web applications in the agent interface](#) for details.
2. Download the sample application: [service-client-api.zip](#).

3. Copy the **wws-service-client-api.js** file in the sample application to a location your web application can access.
4. Set the options described below in [Security Configuration](#).
5. Review [Working with the API](#) for more information about how to use the API.
6. Review the methods and types available in each namespace:
  - [Agent Namespace](#)
  - [Email Namespace](#)
  - [Interaction Namespace](#)
  - [Media Namespace](#)
  - [System Namespace](#)
  - [Voice Namespace](#)

## Security Configuration

The Service Client API involves two parties inside the agent's web browser: the service (the main web page) and the client (in an iframe on the same web page as the service). In order for the client web page to access the API, you need to set a few configuration options to work around [web browser security restrictions](#) for cross-origin requests and to enable request limits. You set these options on the **WWEWS Cluster** application only at the Application level; you can't set these options at the Agent or Agent Group level. Check out the [Service Client API](#) topic in the Workspace Web Edition Configuration Guide for a full list of the options available to configure the API.

### Origin

First, to work around web browser security restrictions set the [service-client-api.accepted-web-content-origins](#) option to the domain you want to be able to access to the API. For example, if you want to give access to a web page located at `http://my-web-server/path/page.html`, then you would set **service-client-api.accepted-web-content-origins** to `http://my-web-server`.

If you have several pages that need access to the API and they're located at different domains, you can also provide **service-client-api.accepted-web-content-origins** with a list. For example: `http://my-web-server, http://my-second-web-server, http://my-third-web-server`.

Finally, if you want to allow *any* page to access the API, just set **service-client-api.accepted-web-content-origins** to `*`.

You can also set the **service-client-api.accepted-web-content-origins** option to values that filter by API request, using any of the following keywords:

- `agent.get`
- `agent.getState`
- `agent.getStateList`
- `agent.setState`

- `email.create`
- `interaction.deleteUserData`
- `interaction.getByInteractionId`
- `interaction.getInteractions`
- `interaction.selectCaseByCaseId`
- `interaction.setUserData`
- `media.getMediaList`
- `media.setState`
- `voice.dial`
- `voice.pauseCallRecording`
- `voice.resumeCallRecording`
- `voice.startCallRecording`
- `voice.stopCallRecording`

For example, you could set **`service-client-api.accepted-web-content-origins`** to `http://my-web-server0`, `http://my-web-server1 (*)`, `http://my-web-server2 (agent.*, voice.dial)`, `http://my-web-server3 (agent.*, interaction.*)`. In this example, everything is allowed for the `http://my-web-server0` and `http://my-web-server1`. For the `http://my-web-server2` domain, only the `agent.get`, `agent.getStateList`, `agent.setState`, `agent.getState` and `voice.dial` requests are allowed.

As seen in the example above, you can also filter by wildcards, using the asterisk in parenthesis. For example, `http://my-web-server1 (*)` or `http://my-web-server3 (agent.*, interaction.*)`.

## Rate Limit

You can limit the maximum number of requests per minute on any Service Client API request by setting the **`service-client-api.rate-limit`** option. For example, setting the value to 50 would restrict the number of requests to 50 per minute. Set the value to 0 for unlimited requests.

If you want to limit the maximum number of requests per minute on a particular Service Client API request, use **`service-client-api.rate-limit.<service-name>`**.

Consider the following sample configuration:

```
service-client-api.rate-limit=0
service-client-api.rate-limit.voice.dial=4
service-client-api.rate-limit.email.create=2
```

In this example, there are no limits globally, but `voice.dial` requests are limited to 4 requests per minute and `email.create` requests are limited to 2 requests per minute.

Workspace calculates the limitation as a fixed interval of time, each minute (this is not calculated on a costly sliding window).

When the number of requests reaches the limit, Workspace ignores all further requests of the same type for a configurable period of time, known as the quarantine delay. In response, Workspace Web

Edition sends a result with an explicit error message to the first request it receives after the limit is reached:

```
{
  "errorMessage": "The rate limit for the request 'voice.dial' has been reached.\nFurther requests of the same type will be ignored for 30 seconds.",
  "request": "agent.getState"
}
```

To specify the global quarantine delay, set the [service-client-api.rate-limit-quarantine-delay](#) option. For example, setting the option to 60 means that Workspace Web Edition ignores requests for 60 seconds after the limit is reached. A value of 0 means that Workspace Web Edition ignores further requests forever, so use this value carefully.

## Attached Data Access

Workspace offers two configuration options to limit the read or write access to the key/value pairs in user data:

- [service-client-api.user-data.write-allowed](#) specifies the list of keys in user data that can be written with the [interaction.setUserData\(\)](#) or [interaction.deleteUserData\(\)](#) functions.
- [service-client-api.user-data.read-allowed](#) specifies the list of keys in user data that can be read. This applies in the `userData` property of the [Interaction](#) object returned by a function or an event.

For example, consider the following configuration:

```
service-client-api.user-data.write-allowed=Key1,Key3
service-client-api.user-data.read-allowed=Key1,Key2,Key3
```

This configuration lets you read the attached data with they keys Key1, Key2, and Key3, but only allows writes on keys Key1, and Key3.

## Working with the API

After you've completed the setup and security steps, you're ready to start working with the Service Client API. The first thing you need to do is add a `<script>` tag to your web application that points to the **wws-service-client-api.js** file (remember, you stored it somewhere accessible in Step 3 above).

Now you can access the API through the **genesys.wws.service** namespace. For example:

```
<html>
  <head>
    <script src="wws-service-client-api.js"></script>
    <script>
      function test() {
        genesys.wws.service.sendMessage({
          request: "agent.get"
        }, function(result) {
          console.debug("SUCCEEDED, result: " + JSON.stringify(result, null, '\t'));
        }, function(result) {
          console.debug("FAILED, result: " + JSON.stringify(result, null, '\t'));
        });
      }
    </script>
  </head>
</html>
```



```
    }

    function eventHandler(message)
    {
        console.debug("Event: " + JSON.stringify(message, null, '\t'));
    }

    genesys.wwe.service.subscribe([ "agent", "interaction" ], eventHandler, this);

</script>
</head>
<body>
    Hello world
</body>
</html>
```

Here's an example of how you could modify attached data:

```
genesys.wwe.service.interaction.setUserData("1",
{
    MyKEY1: "MyValue1",
    MyKEY2: "MyValue2"
})
```

In the above example, the request is `interaction.setUserData` and the parameters are the `interactionId` of 1 and the keyValues of `MyKEY1` and `MyKEY2`.

All methods provided in the Service Client API are asynchronous, so to get the successful or failed result, just add the matching callback:

```
genesys.wwe.service.interaction.setUserData("1",
{
    MyKEY1: "MyValue1",
    MyKEY2: "MyValue2"
}, function(result){
    console.debug("SUCCEEDED, result: " + JSON.stringify(result, null, '\t'));
}, function(result){
    console.debug("FAILED, result: " + JSON.stringify(result, null, '\t'));
})
```

The global template for a service call is:

```
genesys.wwe.service.<Service name>.<Service function>( <... function parameters ...>,
[<optional done() callback>, [<optional fail() callback>]]);
```

The `done()` callback is called when a request is successfully sent without an error.

The `fail()` callback is called when a request generates an error or an exception.

The result of these functions is provided in a JSON object as a unique parameter.

## Notifications

You can use the following code to subscribe to **agent** and **interaction** notifications:

```
function eventHandler(message)
{
    console.debug("Event: " + JSON.stringify(message, null, '\t'));
}
```

```
genesys.wwe.service.subscribe([ "agent", "interaction" ], eventHandler, context);
```

In the above example, `eventHandler` is the event handler function and `context` is an optional contextual object.

Here's an example with an agent `STATE_CHANGED` to Ready:

```
{
  "event": "agent",
  "data": {
    "eventType": "STATE_CHANGED",
    "mediaState": "READY"
  }
}
```

Here's an example with an agent `STATE_CHANGED` to Not Ready with a reason:

```
{
  "event": "agent",
  "data": {
    "eventType": "STATE_CHANGED",
    "mediaState": "NOT_READY_ACTION_CODE",
    "reason": "Break",
    "reasonCode": "1511"
  }
}
```

Finally, here's an example with an `ATTACHED_DATA_CHANGED` event on a voice interaction:

```
{
  "event": "interaction",
  "data": {
    "eventType": "ATTACHED_DATA_CHANGED",
    "media": "voice",
    "interaction": {
      "interactionId": "1",
      "caseId": "4ddalab6-aeab-4a33-f5d0-0153c9fdb43b",
      "userData": {
        "IWAttachedDataInformation": {
          "DispositionCode.Label": "DispositionCode",
          "Option.interaction.case-data.header-foreground-
color": "#FFFFFF",
          "CaseDataBusinessAttribute": "CaseData",
          "DispositionCode.Key": "ChooseDisposition",
          "Option.interaction.case-data.frame-color": "#17849D"
        },
        "IW_CaseUid": "4ddalab6-aeab-4a33-f5d0-0153c9fdb43b",
        "IW_BundleUid": "dfaca66c-4149-42a1-7244-337e949a12b5"
      },
      "parties": [
        {
          "name": "5001"
        }
      ],
      "callUuid": "4L6JGNEE9H7DT671FRPTKE6CQ000000G",
      "state": "DIALING",
      "previousState": "UNKNOWN",
      "isConsultation": false,
      "direction": "OUT",
      "callType": "Internal",
      "dnis": "5001",

```

```

    "isMainCaseInteraction": true
  }
}

```

## Event Type References

The system `eventType` field can be one of the following:

eventType	Description
CUSTOM_TOAST_BUTTON_CLICK	<p>Uses the following parameters:</p> <ul style="list-style-type: none"> <li><b>customToastId</b>: The identifier of the toast where the button has been clicked. The identifier is returned by the <code>popupToast</code> method.</li> <li><b>buttonIndex</b>: The index of the clicked button. The index starts by 0.</li> </ul>

The interaction `eventType` field can be one of the following:

eventType	Description
Common events to all interaction types	
UNKNOWN	An unknown event occurs.
ADDED	The interaction has been added in the list of interactions.
REMOVED	The interaction has been removed from the list of interactions.
ATTACHED_DATA_CHANGED	The attached data have changed in the interaction.
CASE_OR_BUNDLE_ID_CHANGED	The case or the bundle identifier of this interaction has changed.
NEW_MESSAGE	This event represents a new message.
ERROR	An error occurs in the interaction.
Voice events	
CALL_RECORDING_STATE_CHANGED	The call recording state changed.
DIALING	The outbound call starts ringing.
ESTABLISHED	The call has been established.
HELD	The call has been held.
PARTY_CHANGED	The list of party has been changed in the interaction.
RELEASED	The call has been released.
RINGING	The inbound call starts ringing.
OpenMedia events	

---

<b>eventType</b>	<b>Description</b>
ACCEPTED	The open media interaction is accepted.
COMPLETED	The open media interaction has been completed (Mark as done).
COMPOSING	The open media interaction is in composing mode.
CREATED	The open media interaction has been created.
INSERT_STANDARD_RESPONSE	A standard response has been inserted in the interaction.
INVITED	The open media interaction is an invitation.
INVITED_CONFERENCE	The open media interaction receive a conference invitation.
IN_QUEUE_FAILED	The place in queue has failed.
IN_WORKBIN	The interaction has been placed in the work-bin.
IN_WORKBIN_FAILED	The place in work-bin has failed.
LEFT_CONFERENCE	The open media interaction has left the conference.
PULLED	The open media interaction has been pulled from a work-bin.
PULL_FAILED	The pull from the queue has failed.
PULL_WORKBIN_FAILED	The pull from the work-bin has failed.
REVOKED	The open media interaction has been revoked.
TRANSFER_COMPLETED	The open media interaction has been transferred and the transfer has been completed.
Chat events (inherit from OpenMedia events)	
ENDED	The chat has been ended.
JOIN_FAILED	The connection with the chat server failed.
JOIN_PENDING	The interaction is trying to join the chat session.
Outbound email events (inherit from OpenMedia events)	
CANCELLED	The outbound email has been cancelled.
SENT	The outbound email has been sent.

---

# Agent Namespace

## Methods

The Agent namespace includes the following methods:

- [get](#)
- [getState](#)
- [getStateList](#)
- [setState](#)

### get

<b>Signature</b>	<static> get() → { <a href="#">agent.Agent</a> }
<b>Description</b>	Gets the agent's attributes.
<b>Returns</b>	<a href="#">agent.Agent</a>

### getState

<b>Signature</b>	<static> getState() → { <a href="#">media.State</a> }
<b>Description</b>	Gets the agent's state.
<b>Returns</b>	<a href="#">media.State</a>

### getStateList

<b>Signature</b>	<static> getStateList() → {Array.< <a href="#">media.State</a> >}
<b>Description</b>	Gets the list of possible agent states.
<b>Returns</b>	Array.< <a href="#">media.State</a> >

### setState

<b>Signature</b>	<static> setState( <i>stateOperationName</i> )								
<b>Description</b>	Sets the agent's state.								
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>stateOperationName</i></td> <td>string</td> <td>An operationName from the</td> </tr> </tbody> </table>	Name	Type	Description	<i>stateOperationName</i>	string	An operationName from the		
Name	Type	Description							
<i>stateOperationName</i>	string	An operationName from the							

Signature	<static> setState( <i>stateOperationName</i> )		
	Name	Type	Description
			agent states list. See <a href="#">State</a> .

## Type Definitions

The agent namespace includes the following object types:

- [Agent](#)

### Agent

Description	Represents the JSON structure of the agent.		
Type	Object		
	Name	Type	Description
<b>Properties</b>	employeeId	string	The agent's unique identifier used for routing purposes.
	firstname	string	The agent's first name.
	lastname	string	The agent's last name.
	username	string	The agent's username. This is a global unique ID.
	roles	Array.<string>	An array of the agent's roles. Possible roles are: <ul style="list-style-type: none"> <li>• <code>ROLE_AGENT</code> — mandatory for users of Workspace Web Edition</li> </ul>

Description	Represents the JSON structure of the agent.								
	<table border="1"> <thead> <tr> <th data-bbox="820 294 1031 336">Name</th> <th data-bbox="1034 294 1242 336">Type</th> <th data-bbox="1245 294 1453 336">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="820 340 1031 903"></td> <td data-bbox="1034 340 1242 903"></td> <td data-bbox="1245 340 1453 903"> <ul style="list-style-type: none"> <li>• <b>ROLE_SUPERVISOR</b> — enables users to perform supervisor operations like monitoring.</li> <li>• <b>ROLE_ADMIN</b> — provides administrator access to the <b>Genesys Web Services API</b>.</li> </ul> </td> </tr> </tbody> </table>			Name	Type	Description			<ul style="list-style-type: none"> <li>• <b>ROLE_SUPERVISOR</b> — enables users to perform supervisor operations like monitoring.</li> <li>• <b>ROLE_ADMIN</b> — provides administrator access to the <b>Genesys Web Services API</b>.</li> </ul>
Name	Type	Description							
		<ul style="list-style-type: none"> <li>• <b>ROLE_SUPERVISOR</b> — enables users to perform supervisor operations like monitoring.</li> <li>• <b>ROLE_ADMIN</b> — provides administrator access to the <b>Genesys Web Services API</b>.</li> </ul>							

# Email Namespace

## Methods

The Email namespace includes the following methods:

- [create](#)

### create

<b>Signature</b>	<static> create( <i>destination</i> , <i>userData</i> )			
<b>Description</b>	Creates a new empty email.			
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Argument</b>	<b>Description</b>
	destination	string		The destination address for the email.
	userData	object	<optional>	The attached user data key/value object that is updated with each interaction event.



# Interaction Namespace

## Methods

The Interaction namespace includes the following methods:

- [deleteUserData](#)
- [getByInteractionId](#)
- [getInteractions](#)
- [selectCaseByCaseId](#)
- [setUserData](#)
- [markdone](#)
- [blockMarkdone](#)
- [unblockMarkdone](#)

### deleteUserData

<b>Signature</b>	<code>&lt;static&gt; deleteUserData(interactionId, key)</code>											
<b>Description</b>	Deletes the user data attached to the interaction. The <a href="#">service-client-api.user-data.write-allowed</a> option might restrict the allowed key/value pairs.											
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>interactionId</td> <td>string</td> <td>The unique identifier for the interaction.</td> </tr> <tr> <td>key</td> <td>string</td> <td>The key to delete from the attached data.</td> </tr> </tbody> </table>	Name	Type	Description	interactionId	string	The unique identifier for the interaction.	key	string	The key to delete from the attached data.		
Name	Type	Description										
interactionId	string	The unique identifier for the interaction.										
key	string	The key to delete from the attached data.										

### getByInteractionId

<b>Signature</b>	<code>&lt;static&gt; getByInteractionId(interactionId) → {interaction.Interaction}</code>
<b>Description</b>	Gets an interaction by its unique identifier.

<b>Signature</b>	<code>&lt;static&gt; getByInteractionId(interactionId) → {interaction.Interaction}</code>								
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>interactionId</td> <td>string</td> <td>The unique identifier for the interaction.</td> </tr> </tbody> </table>	Name	Type	Description	interactionId	string	The unique identifier for the interaction.		
Name	Type	Description							
interactionId	string	The unique identifier for the interaction.							
<b>Returns</b>	<code>interaction.Interaction</code> or null if the interaction doesn't exist.								

### getInteractions

<b>Signature</b>	<code>&lt;static&gt; getInteractions() → {Array.&lt;interaction.Interaction&gt;}</code>
<b>Description</b>	Gets all the interactions.
<b>Returns</b>	<code>Array.&lt;interaction.Interaction&gt;</code>

### selectCaseByCaseId

<b>Signature</b>	<code>&lt;static&gt; genesys.wwe.service.interaction.selectCaseByCaseId(caseId, succeeded, failed)</code>
<b>Description</b>	<p>Select the case in the UI by case identifier. If you subscribe to the "interaction" events (<code>genesys.wwe.service.subscribe([ "interaction" ], eventHandler, this);</code>), you will receive the following event:</p> <pre>Received interaction event: {   "event": "interaction",   "data": {     "eventType": "CASE_COLLAPSED",     "selectedCaseId": "4401820b- c4e6-4994-69c2-6ae7fdb4905"   },   "userAgent": "WWE Server",   "protocolVersion": 2 } Received interaction event: {   "event": "interaction",   "data": {     "eventType": "CASE_EXPANDED",     "selectedCaseId": "4401820b- c4e6-4994-69c2-6ae7fdb4905"   },   "userAgent": "WWE Server",   "protocolVersion": 2 } Received interaction event: {</pre>

<b>Signature</b>	<static> genesys.wwe.service.interaction.selectCaseByCaselId(caselId, succeeded, failed)								
	<pre>       "event": "interaction",       "data": {         "eventType": "CASE_SELECTED",         "selectedCaseId": "d4187b87-9fe1-4db8-0515-6a91e666e22d"       },       "userAgent": "WWE Server",       "protocolVersion": 2     } </pre>								
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>caselId</td> <td>string</td> <td>The unique identifier for the case.</td> </tr> </tbody> </table>	Name	Type	Description	caselId	string	The unique identifier for the case.		
Name	Type	Description							
caselId	string	The unique identifier for the case.							

## setUserData

<b>Signature</b>	<static> setUserData( <i>interactionId</i> , <i>keyValues</i> )											
<b>Description</b>	Sets the user data on the live interaction (for voice, this means the <b>interaction is not in the IDLE state</b> ). This request overwrites any existing keys on the user data. The <b>service-client-api.user-data.write-allowed</b> option might restrict the allowed key/value pairs.											
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>interactionId</td> <td>string</td> <td>The unique identifier for the interaction.</td> </tr> <tr> <td>keyValues</td> <td>object</td> <td>The key value pairs to set on the user data.</td> </tr> </tbody> </table>	Name	Type	Description	interactionId	string	The unique identifier for the interaction.	keyValues	object	The key value pairs to set on the user data.		
Name	Type	Description										
interactionId	string	The unique identifier for the interaction.										
keyValues	object	The key value pairs to set on the user data.										

## markdone

<b>Signature</b>	<static> markdone( <i>interactionId</i> )								
<b>Description</b>	Mark done the selected interaction.								
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>interactionId</td> <td>string</td> <td>The unique</td> </tr> </tbody> </table>	Name	Type	Description	interactionId	string	The unique		
Name	Type	Description							
interactionId	string	The unique							

<b>Signature</b>	<static> markdone( <i>interactionId</i> )		
	<b>Name</b>	<b>Type</b>	<b>Description</b>
			identifier for the interaction.

## blockMarkdone

<b>Signature</b>	<static> blockMarkdone( <i>interactionId</i> , <i>warningMessage</i> )		
<b>Description</b>	Block the mark done operation on the selected interaction. The "markdone" event must be subscribed to receive the event which informs that there is a delay in blocking the markdone operation with this method.		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
	interactionId	string	The unique interaction identifier of the interaction to prevent the mark done operation.
	warningMessage	string	The warning message.

## unlockMarkdone

<b>Signature</b>	<static> unlockMarkdone( <i>interactionId</i> )		
<b>Description</b>	Unlock the mark done operation on the selected interaction that was previously blocked.		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
	interactionId	string	The unique interaction identifier of the interaction to prevent the mark done operation.

## Type Definitions

The Interaction namespace includes the following object types:

- [Interaction](#)
- [Party](#)

### Interaction

<b>Description</b>	Represents the JSON structure of an interaction. Attributes specific to voice interactions are: callUuid, direction, callType, ani, dnis and recordingState.		
<b>Type</b>	Object		
<b>Properties</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
	interactionId	string	The unique identifier for the interaction. <b>Note:</b> This is a client-side ID that is lost on the next session or refresh.
	parentInteractionId	string	The unique identifier for the parent interaction. <b>Note:</b> This is a client-side ID that is lost on the next session or refresh.
	caseId	string	This identifier targets the case that this interaction is part of.
	userData	object	The attached user data key/value object that is updated with each interaction event.
	state	string	The current

<p><b>Description</b></p>	<p>Represents the JSON structure of an interaction. Attributes specific to voice interactions are: callUuid, direction, callType, ani, dnis and recordingState.</p>								
	<table border="1"> <thead> <tr> <th data-bbox="824 384 1032 426">Name</th> <th data-bbox="1032 384 1239 426">Type</th> <th data-bbox="1239 384 1445 426">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td data-bbox="1239 426 1445 1831"> <p>state of the interaction. Possible values are:</p> <ul style="list-style-type: none"> <li>• UNKNOWN — An unknown state.</li> <li>• IDLE — Specifies a non-active interaction which could be closed.</li> <li>• RINGING — The inbound call is ringing.</li> <li>• DIALING — The outbound call is ringing.</li> <li>• TALKING — The call is established.</li> <li>• HELD — The call is on hold.</li> <li>• PREVIEW — The interaction is a call preview.</li> <li>• INVITED — The open media interaction is inviting.</li> <li>• ACCEPTED — The</li> </ul> </td> </tr> </tbody> </table>			Name	Type	Description			<p>state of the interaction. Possible values are:</p> <ul style="list-style-type: none"> <li>• UNKNOWN — An unknown state.</li> <li>• IDLE — Specifies a non-active interaction which could be closed.</li> <li>• RINGING — The inbound call is ringing.</li> <li>• DIALING — The outbound call is ringing.</li> <li>• TALKING — The call is established.</li> <li>• HELD — The call is on hold.</li> <li>• PREVIEW — The interaction is a call preview.</li> <li>• INVITED — The open media interaction is inviting.</li> <li>• ACCEPTED — The</li> </ul>
Name	Type	Description							
		<p>state of the interaction. Possible values are:</p> <ul style="list-style-type: none"> <li>• UNKNOWN — An unknown state.</li> <li>• IDLE — Specifies a non-active interaction which could be closed.</li> <li>• RINGING — The inbound call is ringing.</li> <li>• DIALING — The outbound call is ringing.</li> <li>• TALKING — The call is established.</li> <li>• HELD — The call is on hold.</li> <li>• PREVIEW — The interaction is a call preview.</li> <li>• INVITED — The open media interaction is inviting.</li> <li>• ACCEPTED — The</li> </ul>							

<p><b>Description</b></p>	<p>Represents the JSON structure of an interaction. Attributes specific to voice interactions are: callUuid, direction, callType, ani, dnis and recordingState.</p>								
	<table border="1"> <thead> <tr> <th data-bbox="824 384 1032 426">Name</th> <th data-bbox="1032 384 1240 426">Type</th> <th data-bbox="1240 384 1446 426">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="824 426 1032 1822"></td> <td data-bbox="1032 426 1240 1822"></td> <td data-bbox="1240 426 1446 1822"> <p>open media interaction is accepted.</p> <ul style="list-style-type: none"> <li>• <b>CREATED</b> — The open media interaction has been created.</li> <li>• <b>PULLED</b> — The open media interaction has been pulled from a workbin.</li> <li>• <b>REVOKED</b> — The open media interaction has been revoked.</li> <li>• <b>COMPLETED</b> — The open media interaction has been completed (Mark as done).</li> <li>• <b>ERROR</b> — The open media interaction has an error.</li> <li>• <b>SAVED</b> — The open</li> </ul> </td> </tr> </tbody> </table>			Name	Type	Description			<p>open media interaction is accepted.</p> <ul style="list-style-type: none"> <li>• <b>CREATED</b> — The open media interaction has been created.</li> <li>• <b>PULLED</b> — The open media interaction has been pulled from a workbin.</li> <li>• <b>REVOKED</b> — The open media interaction has been revoked.</li> <li>• <b>COMPLETED</b> — The open media interaction has been completed (Mark as done).</li> <li>• <b>ERROR</b> — The open media interaction has an error.</li> <li>• <b>SAVED</b> — The open</li> </ul>
Name	Type	Description							
		<p>open media interaction is accepted.</p> <ul style="list-style-type: none"> <li>• <b>CREATED</b> — The open media interaction has been created.</li> <li>• <b>PULLED</b> — The open media interaction has been pulled from a workbin.</li> <li>• <b>REVOKED</b> — The open media interaction has been revoked.</li> <li>• <b>COMPLETED</b> — The open media interaction has been completed (Mark as done).</li> <li>• <b>ERROR</b> — The open media interaction has an error.</li> <li>• <b>SAVED</b> — The open</li> </ul>							

<p><b>Description</b></p>	<p>Represents the JSON structure of an interaction. Attributes specific to voice interactions are: callUuid, direction, callType, ani, dnis and recordingState.</p>																							
	<table border="1"> <thead> <tr> <th data-bbox="824 384 1036 426">Name</th> <th data-bbox="1036 384 1239 426">Type</th> <th data-bbox="1239 384 1445 426">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td data-bbox="1239 426 1445 573"> <p>media interaction has been saved.</p> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1239 573 1445 783"> <ul style="list-style-type: none"> <li>• TRANSFERRING — The open media interaction is being transferred.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1239 783 1445 1119"> <ul style="list-style-type: none"> <li>• TRANSFER_COMPLETED — The open media interaction has been transferred and the transfer has been completed.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1239 1119 1445 1350"> <ul style="list-style-type: none"> <li>• INVITED_CONFERENCE — The open media interaction receives a conference invitation.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1239 1350 1445 1581"> <ul style="list-style-type: none"> <li>• LEFT_CONFERENCE — The open media interaction has left the conference.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1239 1581 1445 1791"> <ul style="list-style-type: none"> <li>• USER_DATA_ATTACHED — Data has been attached to the interaction.</li> </ul> </td> </tr> </tbody> </table>			Name	Type	Description			<p>media interaction has been saved.</p>			<ul style="list-style-type: none"> <li>• TRANSFERRING — The open media interaction is being transferred.</li> </ul>			<ul style="list-style-type: none"> <li>• TRANSFER_COMPLETED — The open media interaction has been transferred and the transfer has been completed.</li> </ul>			<ul style="list-style-type: none"> <li>• INVITED_CONFERENCE — The open media interaction receives a conference invitation.</li> </ul>			<ul style="list-style-type: none"> <li>• LEFT_CONFERENCE — The open media interaction has left the conference.</li> </ul>			<ul style="list-style-type: none"> <li>• USER_DATA_ATTACHED — Data has been attached to the interaction.</li> </ul>
Name	Type	Description																						
		<p>media interaction has been saved.</p>																						
		<ul style="list-style-type: none"> <li>• TRANSFERRING — The open media interaction is being transferred.</li> </ul>																						
		<ul style="list-style-type: none"> <li>• TRANSFER_COMPLETED — The open media interaction has been transferred and the transfer has been completed.</li> </ul>																						
		<ul style="list-style-type: none"> <li>• INVITED_CONFERENCE — The open media interaction receives a conference invitation.</li> </ul>																						
		<ul style="list-style-type: none"> <li>• LEFT_CONFERENCE — The open media interaction has left the conference.</li> </ul>																						
		<ul style="list-style-type: none"> <li>• USER_DATA_ATTACHED — Data has been attached to the interaction.</li> </ul>																						



Description	Represents the JSON structure of an interaction. Attributes specific to voice interactions are: callUuid, direction, callType, ani, dnis and recordingState.																													
	<table border="1"> <thead> <tr> <th data-bbox="824 384 1036 426">Name</th> <th data-bbox="1036 384 1239 426">Type</th> <th data-bbox="1239 384 1450 426">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td data-bbox="1263 457 1563 657"> <ul style="list-style-type: none"> <li>• USER_DATA_UPDATED — The attached data has changed in the interaction.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1263 678 1466 814"> <ul style="list-style-type: none"> <li>• JOIN_PENDING — Trying to join the chat session.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1263 835 1433 1024"> <ul style="list-style-type: none"> <li>• JOIN_FAILED — The connection with the chat server failed.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1263 1045 1580 1213"> <ul style="list-style-type: none"> <li>• HISTORY_IN_PROGRESS — Loading the content of the chat interaction.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1263 1234 1482 1423"> <ul style="list-style-type: none"> <li>• HISTORY_DONE — The content of the chat interaction has been loaded.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1263 1444 1433 1581"> <ul style="list-style-type: none"> <li>• CANCELLED — The outbound email is cancelled.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1263 1602 1417 1738"> <ul style="list-style-type: none"> <li>• SENT — The outbound email is sent.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1263 1759 1401 1791"> <ul style="list-style-type: none"> <li>• READY —</li> </ul> </td> </tr> </tbody> </table>			Name	Type	Description			<ul style="list-style-type: none"> <li>• USER_DATA_UPDATED — The attached data has changed in the interaction.</li> </ul>			<ul style="list-style-type: none"> <li>• JOIN_PENDING — Trying to join the chat session.</li> </ul>			<ul style="list-style-type: none"> <li>• JOIN_FAILED — The connection with the chat server failed.</li> </ul>			<ul style="list-style-type: none"> <li>• HISTORY_IN_PROGRESS — Loading the content of the chat interaction.</li> </ul>			<ul style="list-style-type: none"> <li>• HISTORY_DONE — The content of the chat interaction has been loaded.</li> </ul>			<ul style="list-style-type: none"> <li>• CANCELLED — The outbound email is cancelled.</li> </ul>			<ul style="list-style-type: none"> <li>• SENT — The outbound email is sent.</li> </ul>			<ul style="list-style-type: none"> <li>• READY —</li> </ul>
Name	Type	Description																												
		<ul style="list-style-type: none"> <li>• USER_DATA_UPDATED — The attached data has changed in the interaction.</li> </ul>																												
		<ul style="list-style-type: none"> <li>• JOIN_PENDING — Trying to join the chat session.</li> </ul>																												
		<ul style="list-style-type: none"> <li>• JOIN_FAILED — The connection with the chat server failed.</li> </ul>																												
		<ul style="list-style-type: none"> <li>• HISTORY_IN_PROGRESS — Loading the content of the chat interaction.</li> </ul>																												
		<ul style="list-style-type: none"> <li>• HISTORY_DONE — The content of the chat interaction has been loaded.</li> </ul>																												
		<ul style="list-style-type: none"> <li>• CANCELLED — The outbound email is cancelled.</li> </ul>																												
		<ul style="list-style-type: none"> <li>• SENT — The outbound email is sent.</li> </ul>																												
		<ul style="list-style-type: none"> <li>• READY —</li> </ul>																												

Description	Represents the JSON structure of an interaction. Attributes specific to voice interactions are: callUuid, direction, callType, ani, dnis and recordingState.																				
	<table border="1"> <thead> <tr> <th data-bbox="833 384 1032 426">Name</th> <th data-bbox="1032 384 1239 426">Type</th> <th data-bbox="1239 384 1437 426">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="833 426 1032 835"></td> <td data-bbox="1032 426 1239 835"></td> <td data-bbox="1239 426 1437 835">                             The call preview is ready.                             <ul style="list-style-type: none"> <li>• CANCELED — The call preview is cancelled.</li> <li>• REJECTED — The call preview is rejected.</li> </ul> </td> </tr> <tr> <td data-bbox="833 835 1032 930">previousState</td> <td data-bbox="1032 835 1239 930">string</td> <td data-bbox="1239 835 1437 930">The previous state of the interaction.</td> </tr> <tr> <td data-bbox="833 930 1032 1087">parties</td> <td data-bbox="1032 930 1239 1087">Array.&lt;interactionParty&gt;</td> <td data-bbox="1239 930 1437 1087">A collection of all the parties involved in the interaction.</td> </tr> <tr> <td data-bbox="833 1087 1032 1287">isConsultation</td> <td data-bbox="1032 1087 1239 1287">boolean</td> <td data-bbox="1239 1087 1437 1287">This property is true if the interaction is a consultation; otherwise, it's false.</td> </tr> <tr> <td data-bbox="833 1287 1032 1827">isMainCaseInteraction</td> <td data-bbox="1032 1287 1239 1827">boolean</td> <td data-bbox="1239 1287 1437 1827">This property is true if the interaction is the main interaction in the customer case; otherwise, it's false. In Workspace Web Edition, the main interaction is related to Case Information, Disposition, Note, Contact Profile, and so</td> </tr> </tbody> </table>			Name	Type	Description			The call preview is ready. <ul style="list-style-type: none"> <li>• CANCELED — The call preview is cancelled.</li> <li>• REJECTED — The call preview is rejected.</li> </ul>	previousState	string	The previous state of the interaction.	parties	Array.<interactionParty>	A collection of all the parties involved in the interaction.	isConsultation	boolean	This property is true if the interaction is a consultation; otherwise, it's false.	isMainCaseInteraction	boolean	This property is true if the interaction is the main interaction in the customer case; otherwise, it's false. In Workspace Web Edition, the main interaction is related to Case Information, Disposition, Note, Contact Profile, and so
Name	Type	Description																			
		The call preview is ready. <ul style="list-style-type: none"> <li>• CANCELED — The call preview is cancelled.</li> <li>• REJECTED — The call preview is rejected.</li> </ul>																			
previousState	string	The previous state of the interaction.																			
parties	Array.<interactionParty>	A collection of all the parties involved in the interaction.																			
isConsultation	boolean	This property is true if the interaction is a consultation; otherwise, it's false.																			
isMainCaseInteraction	boolean	This property is true if the interaction is the main interaction in the customer case; otherwise, it's false. In Workspace Web Edition, the main interaction is related to Case Information, Disposition, Note, Contact Profile, and so																			

Description	Represents the JSON structure of an interaction. Attributes specific to voice interactions are: callUuid, direction, callType, ani, dnis and recordingState.																										
	<table border="1"> <thead> <tr> <th data-bbox="833 384 1032 426">Name</th> <th data-bbox="1032 384 1239 426">Type</th> <th data-bbox="1239 384 1438 426">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="833 426 1032 468"></td> <td data-bbox="1032 426 1239 468"></td> <td data-bbox="1239 426 1438 468">on.</td> </tr> <tr> <td data-bbox="833 468 1032 621">callUuid</td> <td data-bbox="1032 468 1239 621">string</td> <td data-bbox="1239 468 1438 621">The UUID of the call. This attribute is only on voice interactions.</td> </tr> <tr> <td data-bbox="833 621 1032 915">direction</td> <td data-bbox="1032 621 1239 915">string</td> <td data-bbox="1239 621 1438 915">The call direction. Possible values are: IN, OUT or UNKNOWN. This attribute is only on voice interactions.</td> </tr> <tr> <td data-bbox="833 915 1032 1257">callType</td> <td data-bbox="1032 915 1239 1257">string</td> <td data-bbox="1239 915 1438 1257">The call type. Possible values are: INTERNAL, INBOUND, OUTBOUND, CONSULT or UNKNOWN. This attribute is only on voice interactions.</td> </tr> <tr> <td data-bbox="833 1257 1032 1470">ani</td> <td data-bbox="1032 1257 1239 1470">string</td> <td data-bbox="1239 1257 1438 1470">The Automatic Number Identification service. This attribute is only on voice interactions.</td> </tr> <tr> <td data-bbox="833 1470 1032 1675">dnis</td> <td data-bbox="1032 1470 1239 1675">string</td> <td data-bbox="1239 1470 1438 1675">The Dialed Number Identification Service. This attribute is only on voice interactions.</td> </tr> <tr> <td data-bbox="833 1675 1032 1822">recordingState</td> <td data-bbox="1032 1675 1239 1822">string</td> <td data-bbox="1239 1675 1438 1822">The call recording state. Possible values are: STOPPED,</td> </tr> </tbody> </table>			Name	Type	Description			on.	callUuid	string	The UUID of the call. This attribute is only on voice interactions.	direction	string	The call direction. Possible values are: IN, OUT or UNKNOWN. This attribute is only on voice interactions.	callType	string	The call type. Possible values are: INTERNAL, INBOUND, OUTBOUND, CONSULT or UNKNOWN. This attribute is only on voice interactions.	ani	string	The Automatic Number Identification service. This attribute is only on voice interactions.	dnis	string	The Dialed Number Identification Service. This attribute is only on voice interactions.	recordingState	string	The call recording state. Possible values are: STOPPED,
Name	Type	Description																									
		on.																									
callUuid	string	The UUID of the call. This attribute is only on voice interactions.																									
direction	string	The call direction. Possible values are: IN, OUT or UNKNOWN. This attribute is only on voice interactions.																									
callType	string	The call type. Possible values are: INTERNAL, INBOUND, OUTBOUND, CONSULT or UNKNOWN. This attribute is only on voice interactions.																									
ani	string	The Automatic Number Identification service. This attribute is only on voice interactions.																									
dnis	string	The Dialed Number Identification Service. This attribute is only on voice interactions.																									
recordingState	string	The call recording state. Possible values are: STOPPED,																									

<b>Description</b>	Represents the JSON structure of an interaction. Attributes specific to voice interactions are: callUuid, direction, callType, ani, dnis and recordingState.		
	<b>Name</b>	<b>Type</b>	<b>Description</b>
			RECORDING or PAUSED. This attribute is only on voice interactions.
	isCaseSelected	boolean	Is true if the case containing this interaction is selected, otherwise is false.
	isCaseExpanded	boolean	Is true if the case containing this interaction is expanded, otherwise is false.
	interactionUUID	string	The attr_itx_id for a multimedia interaction or the callUuid for a voice interaction.

### Party

<b>Description</b>	Represents the JSON structure of a party.		
<b>Type</b>	Object		
<b>Properties</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
	name	string	The name of the party.

# Media Namespace

## Methods

The Media namespace includes the following methods:

- [getMediaList](#)
- [getMediaByName](#)
- [setState](#)

### getMediaList

<b>Signature</b>	<static> getMediaList() → {Array.< <a href="#">media.Media</a> >}
<b>Description</b>	Get the list of media with attributes.
<b>Returns</b>	Array.< <a href="#">media.Media</a> >

### getMediaByName

<b>Signature</b>	<static> getMediaByName( <i>name</i> )								
<b>Description</b>	Get the media attributes.								
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>string</td> <td>The media name.</td> </tr> </tbody> </table>	Name	Type	Description	name	string	The media name.		
Name	Type	Description							
name	string	The media name.							

### setState

<b>Signature</b>	<static> setState( <i>name</i> , <i>stateOperationName</i> )											
<b>Description</b>	Sets the media state.											
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>string</td> <td>The media name.</td> </tr> <tr> <td>stateOperationName</td> <td>string</td> <td>An operationName from the agent states</td> </tr> </tbody> </table>	Name	Type	Description	name	string	The media name.	stateOperationName	string	An operationName from the agent states		
Name	Type	Description										
name	string	The media name.										
stateOperationName	string	An operationName from the agent states										

Signature	<static> setState( <i>name</i> , <i>stateOperationName</i> )		
	<b>Name</b>	<b>Type</b>	<b>Description</b>
			list. See <a href="#">State</a> .

## Type Definitions

The Media namespace includes the following object types:

- [Media](#)
- [State](#)
- [Device](#)

### Media

Description	Represents the JSON structure of a media.		
Type	Object		
<b>Properties</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
	name	string	The media name.
	state	<a href="#">media.State</a>	The media state object.

### State

Description	Represents the JSON structure of a media state.		
Type	Object		
<b>Properties</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
	type	string	The type of operation. Possible values are: <ul style="list-style-type: none"> <li>• LOGOUT</li> <li>• READY</li> <li>• PARTIAL_READY *</li> <li>• NOT_READY</li> </ul>

Description	Represents the JSON structure of a media state.														
	<table border="1"> <thead> <tr> <th data-bbox="824 302 1032 344">Name</th> <th data-bbox="1032 302 1240 344">Type</th> <th data-bbox="1240 302 1451 344">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="824 344 1032 800"></td> <td data-bbox="1032 344 1240 800"></td> <td data-bbox="1240 344 1451 800"> <ul style="list-style-type: none"> <li>• NOT_READY_ACTION_CODE</li> <li>• NOT_READY_AFTER_CALLW</li> <li>• NOT_READY_AFTER_CALLW</li> <li>• DND_ON</li> <li>• OUT_OF_SERVICE *</li> <li>• LOGOUT_DND_ON *</li> <li>• UNKNOWN *</li> </ul> </td> </tr> <tr> <td data-bbox="824 800 1032 894">displayName</td> <td data-bbox="1032 800 1240 894">string</td> <td data-bbox="1240 800 1451 894">The display name of the state.</td> </tr> <tr> <td data-bbox="824 894 1032 1071">operationName</td> <td data-bbox="1032 894 1240 1071">string</td> <td data-bbox="1240 894 1451 1071">The operation name to use with <code>agent.setState</code> and <code>media.setState</code>.</td> </tr> </tbody> </table>			Name	Type	Description			<ul style="list-style-type: none"> <li>• NOT_READY_ACTION_CODE</li> <li>• NOT_READY_AFTER_CALLW</li> <li>• NOT_READY_AFTER_CALLW</li> <li>• DND_ON</li> <li>• OUT_OF_SERVICE *</li> <li>• LOGOUT_DND_ON *</li> <li>• UNKNOWN *</li> </ul>	displayName	string	The display name of the state.	operationName	string	The operation name to use with <code>agent.setState</code> and <code>media.setState</code> .
Name	Type	Description													
		<ul style="list-style-type: none"> <li>• NOT_READY_ACTION_CODE</li> <li>• NOT_READY_AFTER_CALLW</li> <li>• NOT_READY_AFTER_CALLW</li> <li>• DND_ON</li> <li>• OUT_OF_SERVICE *</li> <li>• LOGOUT_DND_ON *</li> <li>• UNKNOWN *</li> </ul>													
displayName	string	The display name of the state.													
operationName	string	The operation name to use with <code>agent.setState</code> and <code>media.setState</code> .													

\* States that are limited to an event and can't be applied by code

## Device

Description	Represents the JSON structure of a media.											
Type	Object											
<p><b>Properties</b></p>	<table border="1"> <thead> <tr> <th data-bbox="824 1373 1032 1415">Name</th> <th data-bbox="1032 1373 1240 1415">Type</th> <th data-bbox="1240 1373 1451 1415">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="824 1415 1032 1696">number</td> <td data-bbox="1032 1415 1240 1696">string</td> <td data-bbox="1240 1415 1451 1696">The phone number configured for an agent - the physical DN. <b>Note:</b> This property is applicable only for voice data.</td> </tr> <tr> <td data-bbox="824 1696 1032 1816">dynamicPhoneNumber</td> <td data-bbox="1032 1696 1240 1816">string</td> <td data-bbox="1240 1696 1451 1816">The dynamic phone number configured for the agent for</td> </tr> </tbody> </table>			Name	Type	Description	number	string	The phone number configured for an agent - the physical DN. <b>Note:</b> This property is applicable only for voice data.	dynamicPhoneNumber	string	The dynamic phone number configured for the agent for
Name	Type	Description										
number	string	The phone number configured for an agent - the physical DN. <b>Note:</b> This property is applicable only for voice data.										
dynamicPhoneNumber	string	The dynamic phone number configured for the agent for										

Description	Represents the JSON structure of a media.								
	<table border="1"> <thead> <tr> <th data-bbox="820 296 1031 338">Name</th> <th data-bbox="1034 296 1237 338">Type</th> <th data-bbox="1240 296 1445 338">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="820 342 1031 693"></td> <td data-bbox="1034 342 1237 693"></td> <td data-bbox="1240 342 1445 693">                     the session.   <b>Note:</b> This property is applicable only for voice data. This property is applicable only when there is an alternate phone number and applicable for the current session only.                 </td> </tr> </tbody> </table>			Name	Type	Description			the session.  <b>Note:</b> This property is applicable only for voice data. This property is applicable only when there is an alternate phone number and applicable for the current session only.
Name	Type	Description							
		the session.  <b>Note:</b> This property is applicable only for voice data. This property is applicable only when there is an alternate phone number and applicable for the current session only.							



# System Namespace

## Methods

The System namespace includes the following methods:

- [getAllowedServices](#)
- [triggerActivity](#)
- [closeToast](#)
- [popupToast](#)
- [updateToast](#)

### getAllowedServices

<b>Signature</b>	<static> getAllowedServices() → {Array.<string>}
<b>Description</b>	Gets the list of allowed services, as determined by the <a href="#">Security Configuration</a> . If the domain of the web application that calls this method isn't listed in the <a href="#">service-client-api.accepted-web-content-origins</a> option, then this method fails.
<b>Returns</b>	Array.<string>

### triggerActivity

<b>Signature</b>	<static> triggerActivity()
<b>Description</b>	Triggers a fake activity to prevent the <a href="#">inactivity timer</a> from closing the agent session.

### popupToast

<b>Signature</b>	<static> popupToast(parameters) → {string}											
<b>Description</b>	Pops up a new custom toast.											
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>title</td> <td>string</td> <td>The title</td> </tr> <tr> <td>iconUrl</td> <td>string</td> <td>The URL of the icon you want to display in the title bar of the custom</td> </tr> </tbody> </table>	Name	Type	Description	title	string	The title	iconUrl	string	The URL of the icon you want to display in the title bar of the custom		
Name	Type	Description										
title	string	The title										
iconUrl	string	The URL of the icon you want to display in the title bar of the custom										

Signature	<static> popupToast(parameters) → {string}		
	Name	Type	Description
			toast popup.
	subject	string	Optional. The subject
	message	string	Optional. The message
	keyValues	string	Optional. JSON object used to fill the key value pair list. For example: {"key1" ; "value one", "key2" ; "value two", "key3" ; "value three"}.
	buttons	Array.<string>	Optional. Each character string in this array becomes a button.
	buttonShowDismiss	boolean	Optional. If set to true, displays the <b>Show</b> and <b>Dismiss</b> buttons and pops up the current iframe if the <b>Show</b> button is pushed. If set to false, displays "OK" or custom buttons based on the parameter's buttons.
	autoCloseTimeout	object	Optional. If set to greater than 0, the popup is automatically closed after the specified milliseconds.

<b>Signature</b>	<static> popupToast(parameters) → {string}								
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #f2f2f2;">Name</th> <th style="background-color: #f2f2f2;">Type</th> <th style="background-color: #f2f2f2;">Description</th> </tr> </thead> <tbody> <tr> <td>sendToMyMessageobject</td> <td></td> <td>Optional. If set to true, sends the <b>subject</b>, <b>iconUrl</b>, <b>title</b>, <b>keyValues</b>, and <b>message</b> parameters to the <b>MyMessage</b> panel.</td> </tr> </tbody> </table>			Name	Type	Description	sendToMyMessageobject		Optional. If set to true, sends the <b>subject</b> , <b>iconUrl</b> , <b>title</b> , <b>keyValues</b> , and <b>message</b> parameters to the <b>MyMessage</b> panel.
Name	Type	Description							
sendToMyMessageobject		Optional. If set to true, sends the <b>subject</b> , <b>iconUrl</b> , <b>title</b> , <b>keyValues</b> , and <b>message</b> parameters to the <b>MyMessage</b> panel.							
<b>Returns</b>	A unique identifier								

### updateToast

<b>Signature</b>	<static> updateToast(id, parameters) → {boolean}																								
<b>Description</b>	Updates the specified toast.																								
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #f2f2f2;">Name</th> <th style="background-color: #f2f2f2;">Type</th> <th style="background-color: #f2f2f2;">Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>string</td> <td>The identifier of the toast to update. The identifier is returned by the <b>popupToast</b> method.</td> </tr> <tr> <td></td> <td></td> <td style="text-align: center;"> <table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #f2f2f2;">Name</th> <th style="background-color: #f2f2f2;">Type</th> <th style="background-color: #f2f2f2;">Description</th> </tr> </thead> <tbody> <tr> <td>title</td> <td>string</td> <td>The title</td> </tr> <tr> <td>iconUrl</td> <td>string</td> <td>The URL of the icon you want to display in the title bar of the custom toast</td> </tr> </tbody> </table> </td> </tr> <tr> <td style="background-color: #f2f2f2;"><b>Parameters</b></td> <td>parameters</td> <td>object</td> <td></td> </tr> </tbody> </table>			Name	Type	Description	id	string	The identifier of the toast to update. The identifier is returned by the <b>popupToast</b> method.			<table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #f2f2f2;">Name</th> <th style="background-color: #f2f2f2;">Type</th> <th style="background-color: #f2f2f2;">Description</th> </tr> </thead> <tbody> <tr> <td>title</td> <td>string</td> <td>The title</td> </tr> <tr> <td>iconUrl</td> <td>string</td> <td>The URL of the icon you want to display in the title bar of the custom toast</td> </tr> </tbody> </table>	Name	Type	Description	title	string	The title	iconUrl	string	The URL of the icon you want to display in the title bar of the custom toast	<b>Parameters</b>	parameters	object	
Name	Type	Description																							
id	string	The identifier of the toast to update. The identifier is returned by the <b>popupToast</b> method.																							
		<table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #f2f2f2;">Name</th> <th style="background-color: #f2f2f2;">Type</th> <th style="background-color: #f2f2f2;">Description</th> </tr> </thead> <tbody> <tr> <td>title</td> <td>string</td> <td>The title</td> </tr> <tr> <td>iconUrl</td> <td>string</td> <td>The URL of the icon you want to display in the title bar of the custom toast</td> </tr> </tbody> </table>	Name	Type	Description	title	string	The title	iconUrl	string	The URL of the icon you want to display in the title bar of the custom toast														
Name	Type	Description																							
title	string	The title																							
iconUrl	string	The URL of the icon you want to display in the title bar of the custom toast																							
<b>Parameters</b>	parameters	object																							

Signature	<static> updateToast(id, parameters) → {boolean}		
	Name	Type	Description
			popup.
	subject	string	Optional. The subject.
	message	string	Optional. The subject.
	keyValues	JSON object	Optional. JSON object used to fill the key value pair list. For example: {"key1" : "value one", "key2" : "value two", "key3" : "value three"}.
	buttonArray	Array<string>	Each character string in this array becomes a button.
	buttonStyle	boolean	If set to true, displays <b>Show</b> and <b>Dismiss</b> buttons and pops

<b>Signature</b>	<static> updateToast(id, parameters) → {boolean}														
	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td> <table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td>up the current iframe if the <b>Show</b> button is pushed. If set to false, displays "OK" or custom buttons based on the parameter's buttons.</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>			Name	Type	Description			<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td>up the current iframe if the <b>Show</b> button is pushed. If set to false, displays "OK" or custom buttons based on the parameter's buttons.</td> </tr> </tbody> </table>	Name	Type	Description			up the current iframe if the <b>Show</b> button is pushed. If set to false, displays "OK" or custom buttons based on the parameter's buttons.
Name	Type	Description													
		<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td>up the current iframe if the <b>Show</b> button is pushed. If set to false, displays "OK" or custom buttons based on the parameter's buttons.</td> </tr> </tbody> </table>	Name	Type	Description			up the current iframe if the <b>Show</b> button is pushed. If set to false, displays "OK" or custom buttons based on the parameter's buttons.							
Name	Type	Description													
		up the current iframe if the <b>Show</b> button is pushed. If set to false, displays "OK" or custom buttons based on the parameter's buttons.													
<b>Returns</b>	true if the toast has been updated; false if the toast identifier has not been found.														

### closeToast

<b>Signature</b>	<static> closeToast(id) → {boolean}								
<b>Description</b>	Closes the specified toast.								
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>string</td> <td>The identifier of the toast to close. The identifier is returned by the <b>popupToast</b> method.</td> </tr> </tbody> </table>			Name	Type	Description	id	string	The identifier of the toast to close. The identifier is returned by the <b>popupToast</b> method.
Name	Type	Description							
id	string	The identifier of the toast to close. The identifier is returned by the <b>popupToast</b> method.							
<b>Returns</b>	true if the toast has been updated; false if the toast identifier has not been found.								

# Voice Namespace

## Methods

The Voice namespace includes the following methods:

- [answer](#)
- [dial](#)
- [hangUp](#)
- [hold](#)
- [resume](#)
- [pauseCallRecording](#)
- [resumeCallRecording](#)
- [startCallRecording](#)
- [stopCallRecording](#)
- [isMicrophoneMute](#)
- [muteMicrophone](#)
- [unmuteMicrophone](#)
- [isSpeakerMute](#)
- [muteSpeaker](#)
- [unmuteSpeaker](#)

### answer

<b>Signature</b>	answer('interactionId')			
<b>Description</b>	Answers the incoming call.			
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Argument</b>	<b>Description</b>
	interaction	string		The interaction identifier

### dial

Name	Type	Argument	Description
destination	string		The call destination number.
userData	object		The attached user data key/value object that is updated with each interaction event.

## hangUp

Signature	hangUp('interactionId')								
Description	Releases the incoming call.								
Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Argument</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>interaction</td> <td>string</td> <td></td> <td>The interaction identifier</td> </tr> </tbody> </table>	Name	Type	Argument	Description	interaction	string		The interaction identifier
Name	Type	Argument	Description						
interaction	string		The interaction identifier						

## hold

Signature	hold('interactionId')								
Description	Holds the incoming call.								
Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Argument</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>interaction</td> <td>string</td> <td></td> <td>The interaction identifier</td> </tr> </tbody> </table>	Name	Type	Argument	Description	interaction	string		The interaction identifier
Name	Type	Argument	Description						
interaction	string		The interaction identifier						

## resume

Signature	resume('interactionId')								
Description	Resumes the held call.								
Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Argument</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>interaction</td> <td>string</td> <td></td> <td>The interaction identifier</td> </tr> </tbody> </table>	Name	Type	Argument	Description	interaction	string		The interaction identifier
Name	Type	Argument	Description						
interaction	string		The interaction identifier						

## pauseCallRecording

Name	Type	Description
interactionId	string	The unique identifier for the interaction.

## resumeCallRecording

Name	Type	Description
interactionId	string	The unique identifier for the interaction.

## startCallRecording

Name	Type	Description
interactionId	string	The unique identifier for the interaction.

## stopCallRecording

Name	Type	Description
interactionId	string	The unique identifier for the interaction.

## isMicrophoneMute

<b>Signature</b>	isMicrophoneMute()
<b>Description</b>	Get the mute state of the microphone of the SIP Endpoint.
<b>Parameters</b>	None.



## muteMicrophone

<b>Signature</b>	muteMicrophone()
<b>Description</b>	Mute the microphone of the SIP Endpoint.
<b>Parameters</b>	None.

## unmuteMicrophone

<b>Signature</b>	unmuteMicrophone()
<b>Description</b>	Unmute the microphone of the SIP Endpoint.
<b>Parameters</b>	None.

## isSpeakerMute

<b>Signature</b>	isSpeakerMute()
<b>Description</b>	Get the mute state of the speaker of the SIP Endpoint.
<b>Parameters</b>	None.

## muteSpeaker

<b>Signature</b>	muteSpeaker()
<b>Description</b>	Mute the speaker of the SIP Endpoint.
<b>Parameters</b>	None.

## unmuteSpeaker

<b>Signature</b>	unmuteSpeaker()
<b>Description</b>	Unmute the speaker of the SIP Endpoint.
<b>Parameters</b>	None.