



## Service Client API Reference

4/14/2021

# Table of Contents

<b>Get started</b>	
Service Client API	4
<b>Namespaces</b>	
Agent namespace	13
Configuration namespace	16
Email namespace	21
Interaction namespace	23
Media namespace	37
System namespace	42
Voice namespace	49

---

Search the table of all articles in this guide, listed in alphabetical order, to find the article you need.

Article	Description
Article	Description

"> Agent namespace Learn about the Agent namespace methods and type definitions in the Service Client API."> Auth Namespace Learn about the Auth namespace methods and type definitions in the Service Client API."> Configuration namespace Learn about the Configuration namespace methods and type definitions in the Service Client API."> Email namespace Learn about the Email namespace methods in the Service Client API."> Interaction namespace Learn about the Interaction namespace methods and type definitions in the Service Client API."> Media namespace Learn about the Media namespace methods and type definitions in the Service Client API."> Outbound namespace Learn about the Outbound namespace methods in the Service Client API."> Service Client API Learn how to use the Service Client API to customize the way your web application integrates with Agent Desktop."> System namespace Learn about the System namespace methods in the Service Client API."> Voice namespace Learn about the Voice namespace methods in the Service Client API.

# Service Client API

## Contents

- [1 API overview](#)
  - [1.1 Controlling call recording from a third-party application](#)
  - [1.2 Embedding multiple third-party applications in Agent Desktop](#)
  - [1.3 Updating attached data from a third-party application](#)
  - [1.4 Enabling click-to-dial from a third-party application](#)
  - [1.5 Enabling Service Client API to invoke toast in Agent Desktop](#)
  - [1.6 Controlling case selection from a third-party application](#)
- [2 Getting started](#)
- [3 Working with the API](#)
  - [3.1 Notifications](#)
  - [3.2 Event Type references](#)

Learn how to use the Service Client API to customize the way your web application integrates with Agent Desktop.

### Important

You must contact your Genesys representative to configure Agent Desktop to use the Service Client API.

## API overview

You can use the Service Client API to customize how your web application or website integrates with Agent Desktop. This JavaScript API is based on `window.postMessage` and provides methods your application can use to communicate cross domain with Agent Desktop while maintaining secured isolation.

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 Agent Desktop
- 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 Agent Desktop

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 Agent Desktop 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.

Contact your Genesys representative to enable embedding multiple third-party applications in Agent Desktop.

## 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 options related to user data in the Service Client section of Agent Setup to enable read and write access to user data.

## Enabling click-to-dial from a third-party application

If you configure Agent Desktop to display your web application in a new tab in the Agent Desktop user 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 Agent Desktop—contact your Genesys representative to enable integration of web applications in Agent Desktop.
2. Download the sample application from GitHub.
3. Copy the **wwe-service-client-api.js** file in the sample application to a location your web application can access.
4. Set configuration options related to security—contact your Genesys representative to provision this 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
  - Configuration Namespace **Note:** You must work with your Genesys representative to enable and use this part of the Service Client API.
  - Email Namespace
  - Interaction Namespace
  - Media Namespace
  - System Namespace
  - Voice Namespace
  - Outbound Namespace
  - Auth Namespace

## 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 tag to your web application that points to the **wwe-service-client-api.js** file (remember, you stored it somewhere accessible in Step 3 above).

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

```
Hello world
```

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.<(<... function parameters ...>, [, []]);
```

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	Uses the following parameters: <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 popupToast method.</li></ul>

eventType	Description
	<ul style="list-style-type: none"> <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	
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.

eventType	Description
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.

### Events for outbound preview

The following table lists the SCAPI event details for Pull Preview, Push Preview and Direct Push Preview records.

Mode	UI Event	Event Type	State	Call Type	Capabilities
Pull Preview	Preview record received	ADDED	PREVIEWING	OUTBOUND_PREVIEW	CALL, REJECT_RECORD, CANCEL_RECORD
		PREVIEWING	PREVIEWING	OUTBOUND_PREVIEW	CALL, REJECT_RECORD, CANCEL_RECORD
	Make call from preview	ADDED	DIALING	OUTBOUND	HANGUP
		DIALING	DIALING	OUTBOUND	HANGUP
		REMOVED	IDLE	OUTBOUND_PREVIEW	
	Release and mark done	RELEASED	IDLE	OUTBOUND	MARK_DONE
		MARKDONE_APPLIED	IDLE	OUTBOUND	MARK_DONE
		REMOVED	IDLE	OUTBOUND	-
	Reject record	STATE_CHANGE	REJECTED	OUTBOUND_PREVIEW	MARK_DONE
	Cancel record	STATE_CHANGE	CANCELED	OUTBOUND_PREVIEW	MARK_DONE
Regular Push Preview	Record received	ADDED	INVITED	OUTBOUND_PUSH_PREVIEW	ACCEPT, REJECT
		INVITED	INVITED	OUTBOUND_PUSH_PREVIEW	ACCEPT, REJECT
	Accepted	PREVIEWING	PREVIEWING	OUTBOUND_PUSH_PREVIEW	CALL, REJECT_RECORD, CANCEL_RECORD

Mode	UI Event	Event Type	State	Call Type	Capabilities
	Rejected	REMOVED	REJECTED	OUTBOUND_PUSH_PREVIEW	
	Make call	ADDED	DIALING	OUTBOUND	HANGUP
		DIALING	DIALING	OUTBOUND	HANGUP
		ESTABLISHED	TALKING	OUTBOUND	HANGUP, HOLD
	Release and mark done	RELEASED	IDLE	OUTBOUND	MARK_DONE
		MARKDONE_APPLIED	IDLE	OUTBOUND	MARK_DONE
		REMOVED	IDLE	OUTBOUND_PUSH_PREVIEW	MARK_DONE
		REMOVED	IDLE	OUTBOUND	-
	Reject record	STATE_CHANGE	REJECTED	OUTBOUND_PUSH_PREVIEW	MARK_DONE
Cancel record	STATE_CHANGE	CANCELED	OUTBOUND_PUSH_PREVIEW	MARK_DONE	
Direct Push Preview	Record received	ADDED	INVITED	OUTBOUND_PREVIEW	ACCEPT, REJECT
		INVITED	INVITED	OUTBOUND_PREVIEW	ACCEPT, REJECT
	Accepted	PREVIEWING	PREVIEWING	OUTBOUND_PREVIEW	CALL, REJECT_RECORD, CANCEL_RECORD
	Rejected	REMOVED	REJECTED	OUTBOUND_PREVIEW	
	Make call	ADDED	DIALING	OUTBOUND	HANGUP
		DIALING	DIALING	OUTBOUND	HANGUP
		ESTABLISHED	TALKING	OUTBOUND	HANGUP
		REMOVED	IDLE	OUTBOUND_PREVIEW	
	Release and mark done	RELEASED	IDLE	OUTBOUND	MARK_DONE
		MARKDONE_APPLIED	IDLE	OUTBOUND	MARK_DONE
		REMOVED	IDLE	OUTBOUND	-
	Reject record	STATE_CHANGE	REJECTED	OUTBOUND_PREVIEW	MARK_DONE
	Cancel record	STATE_CHANGE	CANCELED	OUTBOUND_PREVIEW	MARK_DONE

# Agent namespace

## Contents

- [1 Methods](#)
  - [1.1 get](#)
  - [1.2 getState](#)
  - [1.3 getStateList](#)
  - [1.4 setState](#)
- [2 Type definitions](#)
  - [2.1 Agent](#)

Learn about the Agent namespace methods and type definitions in the Service Client API.

## Methods

The Agent namespace includes the following methods:

- `get`
- `getState`
- `getStateList`
- `setState`

### get

<b>Signature</b>	<code>get()</code> → <code>topic=AgentNamespace</code>
<b>Description</b>	Gets the agent's attributes.
<b>Returns</b>	<code>agent.Agent</code>

### getState

<b>Signature</b>	<code>getState()</code> → <code>topic=MediaNamespace</code>
<b>Description</b>	Gets the agent's state.
<b>Returns</b>	<code>media.State</code>

### getStateList

<b>Signature</b>	<code>getStateList()</code> → <code>{Array.}</code>
<b>Description</b>	Gets the list of possible agent states.
<b>Returns</b>	<code>Array.</code>

### setState

<b>Signature</b>	<code>setState(<i>stateOperationName</i>)</code>
<b>Description</b>	Sets the agent's state.

<b>Signature</b>	setState( <i>stateOperationName</i> )								
<b>Parameters</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>stateOperationName</td> <td>string</td> <td>An operationName from the agent states list. See State.</td> </tr> </tbody> </table>			Name	Type	Description	stateOperationName	string	An operationName from the agent states list. See State.
Name	Type	Description							
stateOperationName	string	An operationName from the agent states list. See State.							

## Type definitions

The agent namespace includes the following object types:

- Agent

### Agent

<b>Description</b>	Represents the JSON structure of the agent.																	
<b>Type</b>	Object																	
<b>Properties</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>employeeid</td> <td>string</td> <td>The agent's unique identifier used for routing purposes.</td> </tr> <tr> <td>firstname</td> <td>string</td> <td>The agent's first name.</td> </tr> <tr> <td>lastname</td> <td>string</td> <td>The agent's last name.</td> </tr> <tr> <td>username</td> <td>string</td> <td>The agent's username. This is a global unique ID.</td> </tr> </tbody> </table>			Name	Type	Description	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.
Name	Type	Description																
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.																

# Configuration namespace

## Contents

- [1 Methods](#)
  - [1.1 `getOption`](#)
- [2 Type definitions](#)
  - [2.1 Section](#)



Learn about the Configuration namespace methods and type definitions in the Service Client API.

### Important

You must work with your Genesys representative to enable and use this part of the Service Client API.

## Methods

The Configuration namespace includes the following methods:

- `getOption`

### getOption

<b>Signature</b>	<code>getOption(<i>options</i>) → {Array. }</code>			
<b>Description</b>	Get configuration options and values for a specific option name or a subset of options from the <b>[interaction-workspace]</b> section or a custom section.			
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Argument</b>	<b>Description</b>
	options	string		An array of configuration options or sections to return. Unless otherwise specified, the API will return options from the <b>[interaction-workspace]</b> section by default. You can specify any

Signature	getOption( <i>options</i> ) → {Array. }		
	Name	Type	<p data-bbox="1286 382 1390 436">of the following:</p> <ul data-bbox="1299 457 1620 1579" style="list-style-type: none"> <li data-bbox="1299 457 1620 667">• A single option: genesys.wwe.service. answer', succeeded, failed)</li> <li data-bbox="1299 688 1620 982">• A single option in a specific section: genesys.wwe.service. option.custom.custom succeeded, failed)</li> <li data-bbox="1299 1003 1620 1264">• Multiple options: genesys.wwe.service. answer', 'privilege.email.can- mark- done'], succeeded, failed)</li> <li data-bbox="1299 1285 1620 1579">• Multiple options in different sections: genesys.wwe.service. 'CustomSection/ option.custom.custom succeeded, failed)</li> </ul> <p data-bbox="1286 1612 1409 1768">You can use an asterisk '*' as a wildcard, but only at the end of each word. For example:</p>

<b>Signature</b>	getOption( <i>options</i> ) → {Array. }		
	<b>Name</b>	<b>Type</b>	<b>Argument</b>
			<ul style="list-style-type: none"> <li>• voice.*</li> <li>• voice.auto*</li> <li>• sipendpoint.*</li> <li>• CustomAPI/test.*</li> </ul> <p>You cannot use an asterisk at the start of an option or section. For example, the following values are not allowed:</p> <ul style="list-style-type: none"> <li>• *.markdone</li> <li>• *.auto</li> </ul>
<b>Returns</b>	Array.		

## Type definitions

The Configuration namespace includes the following object types:

- Section

### Section

<b>Description</b>	Represents the JSON structure of a configuration section. Each section includes a list of key/value pairs for the matching option(s).		
<b>Type</b>	Object		
<b>Properties</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
	name	string	The name of the configuration option.

Description	Represents the JSON structure of a configuration section. Each section includes a list of key/value pairs for the matching option(s).								
	<table border="1"><thead><tr><th data-bbox="818 344 1019 401">Name</th><th data-bbox="1019 344 1219 401">Type</th><th data-bbox="1219 344 1422 401">Description</th></tr></thead><tbody><tr><td data-bbox="818 401 1019 527">value</td><td data-bbox="1019 401 1219 527">string or array of strings</td><td data-bbox="1219 401 1422 527">The value of the configuration option.</td></tr></tbody></table>			Name	Type	Description	value	string or array of strings	The value of the configuration option.
Name	Type	Description							
value	string or array of strings	The value of the configuration option.							

# Email namespace

## Contents

- [1 Methods](#)
  - [1.1 create](#)

Learn about the Email namespace methods in the Service Client API.

## Methods

The Email namespace includes the following methods:

- create

### create

<b>Signature</b>	<code>create(destination, userData)</code>												
<b>Description</b>	Creates a new empty email.												
<b>Parameters</b>	<table border="1"><thead><tr><th>Name</th><th>Type</th><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td>destination</td><td>string</td><td></td><td>The destination address for the email.</td></tr><tr><td>userData</td><td>object</td><td></td><td>The attached user data key/value object that is updated with each interaction event.</td></tr></tbody></table>	Name	Type	Argument	Description	destination	string		The destination address for the email.	userData	object		The attached user data key/value object that is updated with each interaction event.
Name	Type	Argument	Description										
destination	string		The destination address for the email.										
userData	object		The attached user data key/value object that is updated with each interaction event.										

# Interaction namespace

## Contents

- 1 Methods
  - 1.1 deleteUserData
  - 1.2 getByInteractionId
  - 1.3 getInteractions
  - 1.4 selectCaseByCaseId
  - 1.5 setUserData
  - 1.6 markdone
  - 1.7 blockMarkdone
  - 1.8 unblockMarkdone
  - 1.9 accept
  - 1.10 reject
- 2 Type definitions
  - 2.1 Interaction
  - 2.2 Party

Learn about the Interaction namespace methods and type definitions in the Service Client API.

## Methods

The Interaction namespace includes the following methods:

- deleteUserData
- getByInteractionId
- getInteractions
- selectCaseByCaseId
- setUserData
- markdone
- blockMarkdone
- unblockMarkdone
- accept
- reject

### deleteUserData

<b>Signature</b>	deleteUserData( <i>interactionId</i> , <i>key</i> )											
<b>Description</b>	Deletes the user data attached to the interaction. The List of User Data Write Allowed option in Agent Setup 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>	getByInteractionId( <i>interactionId</i> ) → topic=InteractionNamespace						
<b>Description</b>	Gets an interaction by its unique identifier.						
<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>	interaction.Interaction or null if the interaction doesn't exist.						

## getInteractions

<b>Signature</b>	getInteractions() → {Array.}
<b>Description</b>	Gets all the interactions.
<b>Returns</b>	Array.

## selectCaseByCaseId

<b>Signature</b>	genesys.wwe.service.interaction.selectCaseByCaseId(caseId, succeeded, failed)
<b>Description</b>	<p>Select the case in the UI by case identifier. If you subscribe to the "interaction" events (genesys.wwe.service.subscribe(["interaction"], eventHandler, this);), 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: {   "event": "interaction",   "data": {     "eventType": "CASE_SELECTED",</pre>

<b>Signature</b>	genesys.wwe.service.interaction.selectCaseByCaseId(caseId, succeeded, failed)								
	<pre>"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>caseId</td> <td>string</td> <td>The unique identifier for the case.</td> </tr> </tbody> </table>			Name	Type	Description	caseId	string	The unique identifier for the case.
Name	Type	Description							
caseId	string	The unique identifier for the case.							

### setUserData

<b>Signature</b>	setUserData( <i>interactionId</i> , <i>keyValues</i> )											
<b>Description</b>	Sets the user data on the live interaction (for voice, this means the interaction is not in the IDLE state). This request overwrites any existing keys on the user data. The List of User Data Write Allowed option in Agent Setup 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>	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 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.							

## blockMarkdone

<b>Signature</b>	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>	<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 interaction identifier of the interaction to prevent the mark done operation.</td> </tr> <tr> <td>warningMessage</td> <td>string</td> <td>The warning message.</td> </tr> </tbody> </table>	Name	Type	Description	interactionId	string	The unique interaction identifier of the interaction to prevent the mark done operation.	warningMessage	string	The warning message.		
Name	Type	Description										
interactionId	string	The unique interaction identifier of the interaction to prevent the mark done operation.										
warningMessage	string	The warning message.										

## unblockMarkdone

<b>Signature</b>	unblockMarkdone( <i>interactionId</i> )								
<b>Description</b>	Unblock the mark done operation on the selected interaction that was previously blocked.								
<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 interaction identifier of the interaction to prevent the mark done operation.</td> </tr> </tbody> </table>	Name	Type	Description	interactionId	string	The unique interaction identifier of the interaction to prevent the mark done operation.		
Name	Type	Description							
interactionId	string	The unique interaction identifier of the interaction to prevent the mark done operation.							

## accept

<b>Signature</b>	accept( <i>interactionId</i> , <i>succeeded</i> , <i>failed</i> )								
<b>Description</b>	Accept an interaction when it is ringing in Agent Desktop.								
<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 interaction identifier of the interaction to be accepted.</td> </tr> </tbody> </table>	Name	Type	Description	interactionId	string	The unique interaction identifier of the interaction to be accepted.		
Name	Type	Description							
interactionId	string	The unique interaction identifier of the interaction to be accepted.							

Signature	accept( <i>interactionId</i> , <i>succeeded</i> , <i>failed</i> )		
	<b>Name</b>	<b>Type</b>	<b>Description</b>
	succeeded	string	The callback function to use if the operation succeeded.
	failed	string	The callback function to use if the operation failed.

## reject

Signature	reject( <i>interactionId</i> , <i>succeeded</i> , <i>failed</i> )		
Description	Reject an interaction when it is Ringing in Agent Desktop.		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
	interactionId	string	The unique interaction identifier of the interaction to be rejected.
	succeeded	string	The callback function to use if the operation succeeded.
	failed	string	The callback function to use if the operation failed.

## 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: <code>callUuid</code> , <code>direction</code> , <code>callType</code> , <code>ani</code> , <code>dnis</code> and <code>recordingState</code> .																				
<b>Type</b>	Object																				
<b>Properties</b>	<table border="1"> <thead> <tr> <th style="text-align: center;">Name</th> <th style="text-align: center;">Type</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">interactionId</td> <td style="text-align: center;">string</td> <td>The unique identifier for the interaction. <b>Note:</b> This is a client-side ID that is lost on the next session or refresh.</td> </tr> <tr> <td style="text-align: center;">parentInteractionId</td> <td style="text-align: center;">string</td> <td>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.</td> </tr> <tr> <td style="text-align: center;">caseId</td> <td style="text-align: center;">string</td> <td>This identifier targets the case that this interaction is part of.</td> </tr> <tr> <td style="text-align: center;">userData</td> <td style="text-align: center;">object</td> <td>The attached user data key/value object that is updated with each interaction event.</td> </tr> <tr> <td style="text-align: center;">state</td> <td style="text-align: center;">string</td> <td>The current state of the interaction. Possible values are:                             <ul style="list-style-type: none"> <li>• UNKNOWN — An unknown state.</li> </ul> </td> </tr> </tbody> </table>			Name	Type	Description	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 state of the interaction. Possible values are: <ul style="list-style-type: none"> <li>• UNKNOWN — An unknown state.</li> </ul>
Name	Type	Description																			
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 state of the interaction. Possible values are: <ul style="list-style-type: none"> <li>• UNKNOWN — An unknown state.</li> </ul>																			

Description	Represents the JSON structure of an interaction. Attributes specific to voice interactions are: <code>callUuid</code> , <code>direction</code> , <code>callType</code> , <code>ani</code> , <code>dnis</code> and <code>recordingState</code> .																																
	<table border="1"> <thead> <tr> <th data-bbox="818 394 1019 436">Name</th> <th data-bbox="1019 394 1221 436">Type</th> <th data-bbox="1221 394 1422 436">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td data-bbox="1247 464 1409 632"> <ul style="list-style-type: none"> <li>• <b>IDLE</b> — Specifies a non-active interaction which could be closed.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1247 653 1409 768"> <ul style="list-style-type: none"> <li>• <b>RINGING</b> — The inbound call is ringing.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1247 789 1409 926"> <ul style="list-style-type: none"> <li>• <b>DIALING</b> — The outbound call is ringing.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1247 947 1409 1062"> <ul style="list-style-type: none"> <li>• <b>TALKING</b> — The call is established.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1247 1083 1409 1167"> <ul style="list-style-type: none"> <li>• <b>HELD</b> — The call is on hold.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1247 1188 1409 1325"> <ul style="list-style-type: none"> <li>• <b>PREVIEW</b> — The interaction is a call preview.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1247 1346 1409 1482"> <ul style="list-style-type: none"> <li>• <b>INVITED</b> — The open media interaction is inviting.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1247 1503 1409 1640"> <ul style="list-style-type: none"> <li>• <b>ACCEPTED</b> — The open media interaction is accepted.</li> </ul> </td> </tr> <tr> <td></td> <td></td> <td data-bbox="1247 1661 1409 1797"> <ul style="list-style-type: none"> <li>• <b>CREATED</b> — The open media interaction</li> </ul> </td> </tr> </tbody> </table>			Name	Type	Description			<ul style="list-style-type: none"> <li>• <b>IDLE</b> — Specifies a non-active interaction which could be closed.</li> </ul>			<ul style="list-style-type: none"> <li>• <b>RINGING</b> — The inbound call is ringing.</li> </ul>			<ul style="list-style-type: none"> <li>• <b>DIALING</b> — The outbound call is ringing.</li> </ul>			<ul style="list-style-type: none"> <li>• <b>TALKING</b> — The call is established.</li> </ul>			<ul style="list-style-type: none"> <li>• <b>HELD</b> — The call is on hold.</li> </ul>			<ul style="list-style-type: none"> <li>• <b>PREVIEW</b> — The interaction is a call preview.</li> </ul>			<ul style="list-style-type: none"> <li>• <b>INVITED</b> — The open media interaction is inviting.</li> </ul>			<ul style="list-style-type: none"> <li>• <b>ACCEPTED</b> — The open media interaction is accepted.</li> </ul>			<ul style="list-style-type: none"> <li>• <b>CREATED</b> — The open media interaction</li> </ul>
Name	Type	Description																															
		<ul style="list-style-type: none"> <li>• <b>IDLE</b> — Specifies a non-active interaction which could be closed.</li> </ul>																															
		<ul style="list-style-type: none"> <li>• <b>RINGING</b> — The inbound call is ringing.</li> </ul>																															
		<ul style="list-style-type: none"> <li>• <b>DIALING</b> — The outbound call is ringing.</li> </ul>																															
		<ul style="list-style-type: none"> <li>• <b>TALKING</b> — The call is established.</li> </ul>																															
		<ul style="list-style-type: none"> <li>• <b>HELD</b> — The call is on hold.</li> </ul>																															
		<ul style="list-style-type: none"> <li>• <b>PREVIEW</b> — The interaction is a call preview.</li> </ul>																															
		<ul style="list-style-type: none"> <li>• <b>INVITED</b> — The open media interaction is inviting.</li> </ul>																															
		<ul style="list-style-type: none"> <li>• <b>ACCEPTED</b> — The open media interaction is accepted.</li> </ul>																															
		<ul style="list-style-type: none"> <li>• <b>CREATED</b> — The open media interaction</li> </ul>																															

Description	Represents the JSON structure of an interaction. Attributes specific to voice interactions are: <code>callUuid</code> , <code>direction</code> , <code>callType</code> , <code>ani</code> , <code>dnis</code> and <code>recordingState</code> .								
	<table border="1"> <thead> <tr> <th data-bbox="818 390 1021 436">Name</th> <th data-bbox="1021 390 1222 436">Type</th> <th data-bbox="1222 390 1425 436">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td data-bbox="1222 436 1425 1780">                     has been created.                     <ul style="list-style-type: none"> <li>• PULLED — The open media interaction has been pulled from a workbin.</li> <li>• REVOKED — The open media interaction has been revoked.</li> <li>• COMPLETED — The open media interaction has been completed (Mark as done).</li> <li>• ERROR — The open media interaction has an error.</li> <li>• SAVED — The open media interaction has been saved.</li> <li>• TRANSFERRING — The open media interaction is being transferred.</li> </ul> </td> </tr> </tbody> </table>			Name	Type	Description			has been created. <ul style="list-style-type: none"> <li>• PULLED — The open media interaction has been pulled from a workbin.</li> <li>• REVOKED — The open media interaction has been revoked.</li> <li>• COMPLETED — The open media interaction has been completed (Mark as done).</li> <li>• ERROR — The open media interaction has an error.</li> <li>• SAVED — The open media interaction has been saved.</li> <li>• TRANSFERRING — The open media interaction is being transferred.</li> </ul>
Name	Type	Description							
		has been created. <ul style="list-style-type: none"> <li>• PULLED — The open media interaction has been pulled from a workbin.</li> <li>• REVOKED — The open media interaction has been revoked.</li> <li>• COMPLETED — The open media interaction has been completed (Mark as done).</li> <li>• ERROR — The open media interaction has an error.</li> <li>• SAVED — The open media interaction has been saved.</li> <li>• TRANSFERRING — The open media interaction is being transferred.</li> </ul>							

Description	Represents the JSON structure of an interaction. Attributes specific to voice interactions are: <code>callUuid</code> , <code>direction</code> , <code>callType</code> , <code>ani</code> , <code>dnis</code> and <code>recordingState</code> .								
	<table border="1"> <thead> <tr> <th data-bbox="818 390 1023 436">Name</th> <th data-bbox="1023 390 1221 436">Type</th> <th data-bbox="1221 390 1425 436">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="818 436 1023 1839"></td> <td data-bbox="1023 436 1221 1839"></td> <td data-bbox="1221 436 1425 1839"> <ul style="list-style-type: none"> <li>• <b>TRANSFER_COMPLETED</b> — The open media interaction has been transferred and the transfer has been completed.</li> <li>• <b>INVITED_CONFERENCE</b> — The open media interaction receives a conference invitation.</li> <li>• <b>LEFT_CONFERENCE</b> — The open media interaction has left the conference.</li> <li>• <b>USER_DATA_ATTACHED</b> — Data has been attached to the interaction.</li> <li>• <b>USER_DATA_UPDATED</b> — The attached data has changed in the interaction.</li> <li>• <b>JOIN_PENDING</b> — Trying to join the chat session.</li> <li>• <b>JOIN_FAILED</b> — The</li> </ul> </td> </tr> </tbody> </table>			Name	Type	Description			<ul style="list-style-type: none"> <li>• <b>TRANSFER_COMPLETED</b> — The open media interaction has been transferred and the transfer has been completed.</li> <li>• <b>INVITED_CONFERENCE</b> — The open media interaction receives a conference invitation.</li> <li>• <b>LEFT_CONFERENCE</b> — The open media interaction has left the conference.</li> <li>• <b>USER_DATA_ATTACHED</b> — Data has been attached to the interaction.</li> <li>• <b>USER_DATA_UPDATED</b> — The attached data has changed in the interaction.</li> <li>• <b>JOIN_PENDING</b> — Trying to join the chat session.</li> <li>• <b>JOIN_FAILED</b> — The</li> </ul>
Name	Type	Description							
		<ul style="list-style-type: none"> <li>• <b>TRANSFER_COMPLETED</b> — The open media interaction has been transferred and the transfer has been completed.</li> <li>• <b>INVITED_CONFERENCE</b> — The open media interaction receives a conference invitation.</li> <li>• <b>LEFT_CONFERENCE</b> — The open media interaction has left the conference.</li> <li>• <b>USER_DATA_ATTACHED</b> — Data has been attached to the interaction.</li> <li>• <b>USER_DATA_UPDATED</b> — The attached data has changed in the interaction.</li> <li>• <b>JOIN_PENDING</b> — Trying to join the chat session.</li> <li>• <b>JOIN_FAILED</b> — The</li> </ul>							



Description	Represents the JSON structure of an interaction. Attributes specific to voice interactions are: <code>callUuid</code> , <code>direction</code> , <code>callType</code> , <code>ani</code> , <code>dnis</code> and <code>recordingState</code> .											
	<table border="1"> <thead> <tr> <th data-bbox="818 390 1019 436">Name</th> <th data-bbox="1019 390 1221 436">Type</th> <th data-bbox="1221 390 1422 436">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="818 436 1019 1713"></td> <td data-bbox="1019 436 1221 1713"></td> <td data-bbox="1221 436 1422 1713">                     connection with the chat server failed.                     <ul style="list-style-type: none"> <li>• HISTORY_IN_PROGRESS — Loading the content of the chat interaction.</li> <li>• HISTORY_DONE — The content of the chat interaction has been loaded.</li> <li>• CANCELLED — The outbound email is cancelled.</li> <li>• SENT — The outbound email is sent.</li> <li>• READY — The call preview is ready.</li> <li>• CANCELED — The call preview is cancelled.</li> <li>• REJECTED — The call preview is rejected.</li> </ul> </td> </tr> <tr> <td data-bbox="818 1713 1019 1814">previousState</td> <td data-bbox="1019 1713 1221 1814">string</td> <td data-bbox="1221 1713 1422 1814">The previous state of the interaction.</td> </tr> </tbody> </table>			Name	Type	Description			connection with the chat server failed. <ul style="list-style-type: none"> <li>• HISTORY_IN_PROGRESS — Loading the content of the chat interaction.</li> <li>• HISTORY_DONE — The content of the chat interaction has been loaded.</li> <li>• CANCELLED — The outbound email is cancelled.</li> <li>• SENT — The outbound email is sent.</li> <li>• READY — The call preview is ready.</li> <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.
Name	Type	Description										
		connection with the chat server failed. <ul style="list-style-type: none"> <li>• HISTORY_IN_PROGRESS — Loading the content of the chat interaction.</li> <li>• HISTORY_DONE — The content of the chat interaction has been loaded.</li> <li>• CANCELLED — The outbound email is cancelled.</li> <li>• SENT — The outbound email is sent.</li> <li>• READY — The call preview is ready.</li> <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.										

Description	Represents the JSON structure of an interaction. Attributes specific to voice interactions are: <code>callUuid</code> , <code>direction</code> , <code>callType</code> , <code>ani</code> , <code>dnis</code> and <code>recordingState</code> .																							
	<table border="1"> <thead> <tr> <th data-bbox="818 390 1019 436">Name</th> <th data-bbox="1019 390 1221 436">Type</th> <th data-bbox="1221 390 1422 436">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="818 436 1019 562"><code>parties</code></td> <td data-bbox="1019 436 1221 562">Array.</td> <td data-bbox="1221 436 1422 562">A collection of all the parties involved in the interaction.</td> </tr> <tr> <td data-bbox="818 562 1019 747"><code>isConsultation</code></td> <td data-bbox="1019 562 1221 747">boolean</td> <td data-bbox="1221 562 1422 747">This property is true if the interaction is a consultation; otherwise, it's false.</td> </tr> <tr> <td data-bbox="818 747 1019 1310"><code>isMainCaseInteraction</code></td> <td data-bbox="1019 747 1221 1310">boolean</td> <td data-bbox="1221 747 1422 1310">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 on.</td> </tr> <tr> <td data-bbox="818 1310 1019 1461"><code>callUuid</code></td> <td data-bbox="1019 1310 1221 1461">string</td> <td data-bbox="1221 1310 1422 1461">The UUID of the call. This attribute is only on voice interactions.</td> </tr> <tr> <td data-bbox="818 1461 1019 1709"><code>direction</code></td> <td data-bbox="1019 1461 1221 1709">string</td> <td data-bbox="1221 1461 1422 1709">The call direction. Possible values are: IN, OUT or UNKNOWN. This attribute is only on voice interactions.</td> </tr> <tr> <td data-bbox="818 1709 1019 1835"><code>callType</code></td> <td data-bbox="1019 1709 1221 1835">string</td> <td data-bbox="1221 1709 1422 1835">The call type. Possible values are: INTERNAL,</td> </tr> </tbody> </table>			Name	Type	Description	<code>parties</code>	Array.	A collection of all the parties involved in the interaction.	<code>isConsultation</code>	boolean	This property is true if the interaction is a consultation; otherwise, it's false.	<code>isMainCaseInteraction</code>	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 on.	<code>callUuid</code>	string	The UUID of the call. This attribute is only on voice interactions.	<code>direction</code>	string	The call direction. Possible values are: IN, OUT or UNKNOWN. This attribute is only on voice interactions.	<code>callType</code>	string	The call type. Possible values are: INTERNAL,
Name	Type	Description																						
<code>parties</code>	Array.	A collection of all the parties involved in the interaction.																						
<code>isConsultation</code>	boolean	This property is true if the interaction is a consultation; otherwise, it's false.																						
<code>isMainCaseInteraction</code>	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 on.																						
<code>callUuid</code>	string	The UUID of the call. This attribute is only on voice interactions.																						
<code>direction</code>	string	The call direction. Possible values are: IN, OUT or UNKNOWN. This attribute is only on voice interactions.																						
<code>callType</code>	string	The call type. Possible values are: INTERNAL,																						

Description	Represents the JSON structure of an interaction. Attributes specific to voice interactions are: <code>callUuid</code> , <code>direction</code> , <code>callType</code> , <code>ani</code> , <code>dnis</code> and <code>recordingState</code> .																										
	<table border="1"> <thead> <tr> <th data-bbox="815 394 1019 436">Name</th> <th data-bbox="1019 394 1221 436">Type</th> <th data-bbox="1221 394 1442 436">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="815 436 1019 646"></td> <td data-bbox="1019 436 1221 646"></td> <td data-bbox="1221 436 1442 646">                             INBOUND, OUTBOUND, CONSULT or UNKNOWN. This attribute is only on voice interactions.                         </td> </tr> <tr> <td data-bbox="815 646 1019 865"> <code>ani</code> </td> <td data-bbox="1019 646 1221 865"> <code>string</code> </td> <td data-bbox="1221 646 1442 865">                             The Automatic Number Identification service. This attribute is only on voice interactions.                         </td> </tr> <tr> <td data-bbox="815 865 1019 1075"> <code>dnis</code> </td> <td data-bbox="1019 865 1221 1075"> <code>string</code> </td> <td data-bbox="1221 865 1442 1075">                             The Dialed Number Identification Service. This attribute is only on voice interactions.                         </td> </tr> <tr> <td data-bbox="815 1075 1019 1348"> <code>recordingState</code> </td> <td data-bbox="1019 1075 1221 1348"> <code>string</code> </td> <td data-bbox="1221 1075 1442 1348">                             The call recording state. Possible values are: STOPPED, RECORDING or PAUSED. This attribute is only on voice interactions.                         </td> </tr> <tr> <td data-bbox="815 1348 1019 1537"> <code>isCaseSelected</code> </td> <td data-bbox="1019 1348 1221 1537"> <code>boolean</code> </td> <td data-bbox="1221 1348 1442 1537">                             Is true if the case containing this interaction is selected, otherwise is false.                         </td> </tr> <tr> <td data-bbox="815 1537 1019 1726"> <code>isCaseExpanded</code> </td> <td data-bbox="1019 1537 1221 1726"> <code>boolean</code> </td> <td data-bbox="1221 1537 1442 1726">                             Is true if the case containing this interaction is expanded, otherwise is false.                         </td> </tr> <tr> <td data-bbox="815 1726 1019 1831"> <code>interactionUUID</code> </td> <td data-bbox="1019 1726 1221 1831"> <code>string</code> </td> <td data-bbox="1221 1726 1442 1831">                             The <code>attr_itx_id</code> for a multimedia                         </td> </tr> </tbody> </table>			Name	Type	Description			INBOUND, OUTBOUND, CONSULT or UNKNOWN. This attribute is only on voice interactions.	<code>ani</code>	<code>string</code>	The Automatic Number Identification service. This attribute is only on voice interactions.	<code>dnis</code>	<code>string</code>	The Dialed Number Identification Service. This attribute is only on voice interactions.	<code>recordingState</code>	<code>string</code>	The call recording state. Possible values are: STOPPED, RECORDING or PAUSED. This attribute is only on voice interactions.	<code>isCaseSelected</code>	<code>boolean</code>	Is true if the case containing this interaction is selected, otherwise is false.	<code>isCaseExpanded</code>	<code>boolean</code>	Is true if the case containing this interaction is expanded, otherwise is false.	<code>interactionUUID</code>	<code>string</code>	The <code>attr_itx_id</code> for a multimedia
Name	Type	Description																									
		INBOUND, OUTBOUND, CONSULT or UNKNOWN. This attribute is only on voice interactions.																									
<code>ani</code>	<code>string</code>	The Automatic Number Identification service. This attribute is only on voice interactions.																									
<code>dnis</code>	<code>string</code>	The Dialed Number Identification Service. This attribute is only on voice interactions.																									
<code>recordingState</code>	<code>string</code>	The call recording state. Possible values are: STOPPED, RECORDING or PAUSED. This attribute is only on voice interactions.																									
<code>isCaseSelected</code>	<code>boolean</code>	Is true if the case containing this interaction is selected, otherwise is false.																									
<code>isCaseExpanded</code>	<code>boolean</code>	Is true if the case containing this interaction is expanded, otherwise is false.																									
<code>interactionUUID</code>	<code>string</code>	The <code>attr_itx_id</code> for a multimedia																									

<b>Description</b>	Represents the JSON structure of an interaction. Attributes specific to voice interactions are: <code>callUuid</code> , <code>direction</code> , <code>callType</code> , <code>ani</code> , <code>dnis</code> and <code>recordingState</code> .		
	<b>Name</b>	<b>Type</b>	<b>Description</b>
			interaction or the <code>callUuid</code> 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>
	<code>name</code>	string	The name of the party.

# Media namespace

## Contents

- [1 Methods](#)
  - [1.1 getMediaList](#)
  - [1.2 getMediaByName](#)
  - [1.3 setState](#)
- [2 Type definitions](#)
  - [2.1 Media](#)
  - [2.2 State](#)
  - [2.3 Device](#)

Learn about the Media namespace methods and type definitions in the Service Client API.

## Methods

The Media namespace includes the following methods:

- `getMediaList`
- `getMediaByName`
- `setState`

### getMediaList

<b>Signature</b>	<code>getMediaList() → {Array.}</code>
<b>Description</b>	Get the list of media with attributes.
<b>Returns</b>	Array.

### getMediaByName

<b>Signature</b>	<code>getMediaByName(<i>name</i>)</code>								
<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><code>name</code></td> <td>string</td> <td>The media name.</td> </tr> </tbody> </table>	Name	Type	Description	<code>name</code>	string	The media name.		
Name	Type	Description							
<code>name</code>	string	The media name.							

### setState

<b>Signature</b>	<code>setState(<i>name</i>, <i>stateOperationName</i>)</code>											
<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><code>name</code></td> <td>string</td> <td>The media name.</td> </tr> <tr> <td><code>stateOperationName</code></td> <td>string</td> <td>An <code>operationName</code> from the agent</td> </tr> </tbody> </table>	Name	Type	Description	<code>name</code>	string	The media name.	<code>stateOperationName</code>	string	An <code>operationName</code> from the agent		
Name	Type	Description										
<code>name</code>	string	The media name.										
<code>stateOperationName</code>	string	An <code>operationName</code> from the agent										

Signature	setState( <i>name</i> , <i>stateOperationName</i> )								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td>states list. See State.</td> </tr> </tbody> </table>	Name	Type	Description			states list. See State.		
Name	Type	Description							
		states list. See State.							

## Type definitions

The Media namespace includes the following object types:

- Media
- State
- Device

### Media

Description	Represents the JSON structure of a media.											
Type	Object											
<p><b>Properties</b></p>	<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>state</td> <td>media.State</td> <td>The media state object.</td> </tr> </tbody> </table>	Name	Type	Description	name	string	The media name.	state	media.State	The media state object.		
Name	Type	Description										
name	string	The media name.										
state	media.State	The media state object.										

### State

Description	Represents the JSON structure of a media state.								
Type	Object								
<p><b>Properties</b></p>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>type</td> <td>string</td> <td>The type of operation. Possible values are: <ul style="list-style-type: none"> <li>• LOGOUT</li> <li>• READY</li> <li>• PARTIAL_READY *</li> </ul> </td> </tr> </tbody> </table>	Name	Type	Description	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> </ul>		
Name	Type	Description							
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> </ul>							

Description	Represents the JSON structure of a media state.														
	<table border="1"> <thead> <tr> <th data-bbox="818 302 1021 344">Name</th> <th data-bbox="1021 302 1224 344">Type</th> <th data-bbox="1224 302 1427 344">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="818 344 1021 848"></td> <td data-bbox="1021 344 1224 848"></td> <td data-bbox="1224 344 1427 848"> <ul style="list-style-type: none"> <li>• NOT_READY</li> <li>• NOT_READY_ACTION_CODE</li> <li>• NOT_READY_AFTER_CALLWO</li> <li>• NOT_READY_AFTER_CALLWO</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="818 848 1021 947">displayName</td> <td data-bbox="1021 848 1224 947">string</td> <td data-bbox="1224 848 1427 947">The display name of the state.</td> </tr> <tr> <td data-bbox="818 947 1021 1136">operationName</td> <td data-bbox="1021 947 1224 1136">string</td> <td data-bbox="1224 947 1427 1136">The operation name to use with agent.setState and media.setState.</td> </tr> </tbody> </table>			Name	Type	Description			<ul style="list-style-type: none"> <li>• NOT_READY</li> <li>• NOT_READY_ACTION_CODE</li> <li>• NOT_READY_AFTER_CALLWO</li> <li>• NOT_READY_AFTER_CALLWO</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 agent.setState and media.setState.
Name	Type	Description													
		<ul style="list-style-type: none"> <li>• NOT_READY</li> <li>• NOT_READY_ACTION_CODE</li> <li>• NOT_READY_AFTER_CALLWO</li> <li>• NOT_READY_AFTER_CALLWO</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 agent.setState and media.setState.													

\* 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 style="text-align: center;"><b>Properties</b></p>	<table border="1"> <thead> <tr> <th data-bbox="818 1430 1021 1472">Name</th> <th data-bbox="1021 1430 1224 1472">Type</th> <th data-bbox="1224 1430 1427 1472">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="818 1472 1021 1745">number</td> <td data-bbox="1021 1472 1224 1745">string</td> <td data-bbox="1224 1472 1427 1745">                     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="818 1745 1021 1808">dynamicPhoneNumber</td> <td data-bbox="1021 1745 1224 1808">string</td> <td data-bbox="1224 1745 1427 1808">The dynamic phone number</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
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										



Description	Represents the JSON structure of a media.		
	<b>Name</b>	<b>Type</b>	<b>Description</b> configured for the agent for 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

## Contents

- [1 Methods](#)
  - [1.1 getAllowedServices](#)
  - [1.2 triggerActivity](#)
  - [1.3 popupToast](#)
  - [1.4 updateToast](#)
  - [1.5 closeToast](#)

Learn about the System namespace methods in the Service Client API.

## Methods

The System namespace includes the following methods:

- `getAllowedServices`
- `triggerActivity`
- `closeToast`
- `popupToast`
- `updateToast`

### getAllowedServices

<b>Signature</b>	<code>getAllowedServices() → {Array.}</code>
<b>Description</b>	Gets the list of allowed services, as determined by the security configuration for Agent Desktop. If the domain of the web application that calls this method isn't listed in the <code>service-client-api.accepted-web-content-origins</code> option, then this method fails. Contact your Genesys representative for details about this configuration.
<b>Returns</b>	Array.

### triggerActivity

<b>Signature</b>	<code>triggerActivity()</code>
<b>Description</b>	Triggers a fake activity to prevent the inactivity timer from closing the agent session.

### popupToast

<b>Signature</b>	<code>popupToast(parameters) → {string}</code>						
<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><code>title</code></td><td>string</td><td>The title</td></tr></tbody></table>	Name	Type	Description	<code>title</code>	string	The title
Name	Type	Description					
<code>title</code>	string	The title					

Signature	popupToast(parameters) → {string}		
	<b>Name</b>	<b>Type</b>	<b>Description</b>
	iconUrl	string	The URL of the icon you want to display in the title bar of the custom 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: <pre>{ "key1" ; "value one", "key2" ; "value two", "key3" ; "value three" }.</pre>
	buttons	Array.	Optional. Each character string in this array becomes a button. All buttons are displayed as buttons, not hyperlinks, in the following order: [Button 2] [Button 3] ... [Button N] [Button 1].
	buttonShowDismiss	boolean	Optional. If set to <code>true</code> , 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 <code>false</code> ,

<b>Signature</b>	popupToast(parameters) → {string}		
	<b>Name</b>	<b>Type</b>	<b>Description</b>
			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.
	sendToMyMessage	object	Optional. If set to <code>true</code> , 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>	updateToast(id, parameters) → {boolean}										
<b>Description</b>	Updates the specified toast.										
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>								
	id	string	The identifier of the toast to update. The identifier is returned by the popupToast method.								
	parameters	object	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Name</th> <th style="text-align: center;">Type</th> <th style="text-align: center;">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</td> </tr> </tbody> </table>	Name	Type	Description	title	string	The title	iconUrl	string
Name	Type	Description									
title	string	The title									
iconUrl	string	The URL of									

Signature	updateToast(id, parameters) → {boolean}																				
	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 data-bbox="1354 415 1435 844">the icon you want to display in the title bar of the custom toast popup.</td> </tr> <tr> <td>subject</td> <td>string</td> <td data-bbox="1354 856 1451 949">Optional. The subject.</td> </tr> <tr> <td>message</td> <td>string</td> <td data-bbox="1354 961 1451 1054">Optional. The message.</td> </tr> <tr> <td>keyValues</td> <td>Object</td> <td data-bbox="1354 1066 1549 1642">Optional. JSON object used to fill the key value pair list. For example: {"key1" : "value one", "key2" : "value two", "key3" : "value three"}.</td> </tr> <tr> <td>buttons</td> <td>Array&lt;string&gt;</td> <td data-bbox="1354 1654 1451 1801">Each character in this</td> </tr> </tbody> </table>	Name	Type	Description			the icon you want to display in the title bar of the custom toast popup.	subject	string	Optional. The subject.	message	string	Optional. The message.	keyValues	Object	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>	Each character in this
Name	Type	Description																			
		the icon you want to display in the title bar of the custom toast popup.																			
subject	string	Optional. The subject.																			
message	string	Optional. The message.																			
keyValues	Object	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>	Each character in this																			

Signature	updateToast(id, parameters) → {boolean}								
	Name	Type	Description						
			<table border="1"> <thead> <tr> <th data-bbox="1224 363 1289 405">Name</th> <th data-bbox="1289 363 1354 405">Type</th> <th data-bbox="1354 363 1422 405">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="1224 405 1289 1843"></td> <td data-bbox="1289 405 1354 1843">array</td> <td data-bbox="1354 405 1422 1843">                     becomes a button. All buttons are displayed as buttons, not hyperlinks, in the following order: [Button 2] [Button 3] ... [Button N] [Button 1].                       If set to true, displays <b>Show</b> and <b>Dismiss</b> buttons and pops <b>Show</b> and <b>Dismiss</b> buttons from the current iframe if the <b>Show</b> button is pushed. If set                 </td> </tr> </tbody> </table>	Name	Type	Description		array	becomes a button. All buttons are displayed as buttons, not hyperlinks, in the following order: [Button 2] [Button 3] ... [Button N] [Button 1].  If set to true, displays <b>Show</b> and <b>Dismiss</b> buttons and pops <b>Show</b> and <b>Dismiss</b> buttons from the current iframe if the <b>Show</b> button is pushed. If set
	Name	Type	Description						
	array	becomes a button. All buttons are displayed as buttons, not hyperlinks, in the following order: [Button 2] [Button 3] ... [Button N] [Button 1].  If set to true, displays <b>Show</b> and <b>Dismiss</b> buttons and pops <b>Show</b> and <b>Dismiss</b> buttons from the current iframe if the <b>Show</b> button is pushed. If set							

<b>Signature</b>	updateToast(id, parameters) → {boolean}		
	<b>Name</b>	<b>Type</b>	<b>Description</b>
			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>	closeToast(id) → {boolean}		
<b>Description</b>	Closes the specified toast.		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
	id	string	The identifier of the toast to close. The identifier is returned by the popupToast method.
<b>Returns</b>	true if the toast has been updated; false if the toast identifier has not been found.		



# Voice namespace

## Contents

- **1 Methods**
  - 1.1 answer
  - 1.2 dial
  - 1.3 hangUp
  - 1.4 hold
  - 1.5 resume
  - 1.6 pauseCallRecording
  - 1.7 resumeCallRecording
  - 1.8 startCallRecording
  - 1.9 stopCallRecording
  - 1.10 isMicrophoneMute
  - 1.11 muteMicrophone
  - 1.12 unmuteMicrophone
  - 1.13 isSpeakerMute
  - 1.14 muteSpeaker
  - 1.15 unmuteSpeaker

Learn about the Voice namespace methods in the Service Client API.

## 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

<b>Signature</b>	hangUp('interactionId')										
<b>Description</b>	Releases the incoming call.										
<b>Parameters</b>	<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

<b>Signature</b>	hold('interactionId')										
<b>Description</b>	Holds the incoming call.										
<b>Parameters</b>	<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

<b>Signature</b>	resume('interactionId')										
<b>Description</b>	Resumes the held call.										
<b>Parameters</b>	<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.