

salesforce

---

# Omni-Channel Developer Guide

Version 44.0, Winter '19



 @salesforcedocs

Last updated: December 4, 2018



# CONTENTS

<b>Chapter 1: Omni-Channel Developer Guide</b> .....	<b>1</b>
Omni-Channel SOAP API Objects .....	<b>2</b>
AgentWork .....	<b>3</b>
OmniSupervisorConfig .....	<b>9</b>
OmniSupervisorConfigGroup .....	<b>10</b>
OmniSupervisorConfigProfile .....	<b>11</b>
OmniSupervisorConfigUser .....	<b>11</b>
PendingServiceRouting .....	<b>12</b>
PresenceConfigDeclineReason .....	<b>15</b>
PresenceDeclineReason .....	<b>16</b>
PresenceUserConfig .....	<b>17</b>
PresenceUserConfigProfile .....	<b>20</b>
PresenceUserConfigUser .....	<b>20</b>
QueueRoutingConfig .....	<b>21</b>
ServiceChannel .....	<b>23</b>
ServiceChannelStatus .....	<b>25</b>
ServicePresenceStatus .....	<b>26</b>
UserServicePresence .....	<b>27</b>
Omni-Channel Objects for the Salesforce Console .....	<b>28</b>
Omni-Channel Objects for the Salesforce Console Integration Toolkit .....	<b>29</b>
Omni-Channel Objects for the Lightning Console JavaScript API .....	<b>44</b>
External Routing for Omni-Channel .....	<b>60</b>
Technical Architecture and Process .....	<b>60</b>
Integrate External Routing .....	<b>61</b>
Expected Behavior .....	<b>63</b>
Troubleshooting .....	<b>64</b>
<b>INDEX</b> .....	<b>65</b>



# CHAPTER 1 Omni-Channel Developer Guide

## In this chapter ...

- [Omni-Channel SOAP API Objects](#)
- [Omni-Channel Objects for the Salesforce Console](#)
- [External Routing for Omni-Channel](#)

Customize your Omni-Channel records and console integration with Omni-Channel API objects and console methods.

# Omni-Channel SOAP API Objects

---

Use SOAP API to create, retrieve, update or delete records, such as accounts, leads, and custom objects. The SOAP API includes several objects that let you control and customize your Omni-Channel records, including Omni-Channel users, routing configurations, and statuses.

If you need more information on Salesforce's SOAP API, see the [SOAP API Developer's Guide](#).

## [AgentWork](#)

Represents a work assignment that's been routed to an agent. This object is available in API version 32.0 and later.

## [OmniSupervisorConfig](#)

Represents a configuration that determines a user's Omni-Channel Supervisor settings. This object is available in API version 41.0 and later.

## [OmniSupervisorConfigGroup](#)

Represents a public group assigned to the Omni-Channel Supervisor configuration that's visible to a supervisor user. This object is available in API version 41.0 and later.

## [OmniSupervisorConfigProfile](#)

Represents a configuration that determines the Omni-Channel Supervisor settings that are assigned to users who are assigned to a specific profile. User-level configurations override profile-level configurations. This object is available in API version 41.0 and later.

## [OmniSupervisorConfigUser](#)

Represents a configuration that determines the Omni-Channel Supervisor settings that are assigned to a user. User-level configurations override profile-level configurations. This object is available in API version 41.0 and later.

## [PendingServiceRouting](#)

Represents a work assignment that's waiting to be routed. This object is available in API version 40.0 and later.

## [PresenceConfigDeclineReason](#)

Represents the settings for a decline reason that a presence user provides when declining work. This object is available in API version 37.0 and later.

## [PresenceDeclineReason](#)

Represents an Omni-Channel decline reason that agents can select when declining work requests. This object is available in API version 37.0 and later.

## [PresenceUserConfig](#)

Represents a configuration that determines a presence user's settings. This object is available in API version 32.0 and later.

## [PresenceUserConfigProfile](#)

Represents a configuration that determines the settings that are assigned to presence users who are assigned to a specific profile. User-level configurations override profile-level configurations. This object is available in API version 32.0 and later.

## [PresenceUserConfigUser](#)

Represents a configuration that determines the settings that are assigned to a presence user. These user-level configurations override profile-level configurations. This object is available in API version 32.0 and later.

## [QueueRoutingConfig](#)

Represents the settings that determine how work items are routed to agents. This object is available in API version 32.0 and later.

## [ServiceChannel](#)

Represents a channel of work items that are received from your organization—for example, cases, chats, or leads. This object is available in API version 32.0 and later.

[ServiceChannelStatus](#)

Represents the status that's associated with a specific service channel. This object is available in API version 32.0 and later.

[ServicePresenceStatus](#)

Represents a presence status that can be assigned to a service channel. This object is available in API version 32.0 and later.

[UserServicePresence](#)

Represents a presence user's real-time presence status. This object is available in API version 32.0 and later.

## AgentWork

Represents a work assignment that's been routed to an agent. This object is available in API version 32.0 and later.

## Supported Calls

`create()`, `delete()`, `query()`, `getDeleted()`, `getUpdated()`, `retrieve()`, `undelete()`

## Fields

Field	Details
<code>AcceptDatetime</code>	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable, Sort</p> <p><b>Description</b> Indicates when the work item was accepted.</p>
<code>ActiveTime</code>	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The amount of time an agent actively worked on the work item. Tracks when the item is open and in focus in the agent's console.</p>
<code>AgentCapacityWhenDeclined</code>	<p><b>Type</b> double</p> <p><b>Properties</b> Filter, Nillable, Sort</p> <p><b>Description</b> The agent's capacity when declining work, either explicitly or through push timeout.</p>
<code>AssignedDateTime</code>	<p><b>Type</b> dateTime</p>

Field	Details
	<p><b>Properties</b> Filter, Nillable, Sort</p> <p><b>Description</b> Indicates when the work item was assigned to an agent,</p>
CancelDateTime	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable, Sort</p> <p><b>Description</b> Indicates when the work item was canceled.</p>
CapacityPercentage	<p><b>Type</b> percent</p> <p><b>Properties</b> Create, Filter, Nillable, Sort</p> <p><b>Description</b> The percentage of an agent's capacity for work items that's consumed by a specific type of work item from this service channel.  When an agent's combined work items reach 100%, the agent won't receive new work items until there is enough open capacity for more work. For example, if you give phone calls a capacity percentage of 100, an agent on a call doesn't receive new work items until the call ends.</p>
CapacityWeight	<p><b>Type</b> double</p> <p><b>Properties</b> Create, Filter, Nillable, Sort</p> <p><b>Description</b> The amount of an agent's capacity for work items that's consumed by a work item from this service channel.  For example, if cases are assigned a capacity weight of 2, an agent with a capacity of 6 can accept up to 3 cases before the agent is at capacity and can't receive new work items.</p>
CloseDateTime	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable, Sort</p> <p><b>Description</b> Indicates when the work item was closed.</p>



Field	Details
CreatedById	<p><b>Type</b> reference</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> ID of the User who created this record.</p>
CreatedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Defaulted on createFilter, Sort</p> <p><b>Description</b> Date and time when this record was created.</p>
DeclineDatetime	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable, Sort</p> <p><b>Description</b> Date and time when the agent declined this record.</p>
DeclineReason	<p><b>Type</b> string</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The provided reason for why an agent declined the work request.</p>
HandleTime	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The amount of time an agent had the work item open. Calculated by <code>Close Time - Accepted Time</code>.</p>
Id	<p><b>Type</b> ID</p> <p><b>Properties</b> Defaulted on create, Filter, Group, idLookup, Sort</p>

Field	Details
	<p><b>Description</b> The ID of the AgentWork object.</p>
IsDeleted	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> Indicates whether the record has been moved to the Recycle Bin (<code>true</code>) or not (<code>false</code>).</p>
LastModifiedById	<p><b>Type</b> reference</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> The ID of the user who last modified this record.</p>
LastModifiedDate	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Defaulted on create, Filter, Sort</p> <p><b>Description</b> Date and time when a user last modified this record.</p>
Name	<p><b>Type</b> string</p> <p><b>Properties</b> Autonumber, Defaulted on create, Filter, idLookup, Sort</p> <p><b>Description</b> An automatically generated ID number that identifies the record.</p>
OriginalQueueId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The ID of the queue that the work assignment was originally routed to.</p>
PushTimeout	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p>

Field	Details
	<p><b>Description</b></p> <p>The number of seconds set for push timeout. <code>0</code> is returned when push timeout isn't enabled. Available in API version 36.0 and later.</p>
PushTimeoutDateTime	<p><b>Type</b></p> <p>dateTime</p> <p><b>Properties</b></p> <p>Filter, Nillable, Sort</p> <p><b>Description</b></p> <p>Indicates when the push timeout event occurred. Available in API version 36.0 and later.</p>
RequestDateTime	<p><b>Type</b></p> <p>dateTime</p> <p><b>Properties</b></p> <p>Filter, Nillable, Sort</p> <p><b>Description</b></p> <p>Indicates when the work was requested.</p>
ServiceChannelId	<p><b>Type</b></p> <p>reference</p> <p><b>Properties</b></p> <p>Create, Filter, Group, Sort</p> <p><b>Description</b></p> <p>The ID of the service channel that's associated with the work assignment.</p>
ShouldSkipCapacityCheck	<p><b>Type</b></p> <p>boolean</p> <p><b>Properties</b></p> <p>Create, Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b></p> <p>Indicates whether to skip checking an agent's available capacity (<code>true</code>) or not (<code>false</code>) when an externally routed work item is created. This field is used when agents can simultaneously handle work from both Omni-Channel queues and queues using external routing.</p> <p>When <code>true</code>, the receiving agent can exceed their set capacity to accept the item, but they don't receive more Omni-Channel routed work. When <code>false</code>, the receiving agent can't exceed their set capacity and must have enough open capacity to accept the item.</p>
SpeedToAnswer	<p><b>Type</b></p> <p>int</p> <p><b>Properties</b></p> <p>Filter, Group, Nillable, Sort</p>

Field	Details
	<p><b>Description</b></p> <p>The amount of time between when the work was requested and when an agent accepted it.</p>
Status	<p><b>Type</b></p> <p>picklist</p> <p><b>Properties</b></p> <p>Filter, Group, Restricted picklist, Sort</p> <p><b>Description</b></p> <p>The working status of the work item. Valid values are:</p> <ul style="list-style-type: none"> <li>• Assigned – The item is assigned to the agent but hasn't been opened.</li> <li>• Opened – The agent opened the item.</li> <li>• Unavailable – The item was assigned to the agent but the agent became unavailable (went offline or lost connection).</li> <li>• Declined – The item was assigned to the agent but the agent explicitly declined it.</li> <li>• DeclinedOnPushTimeout – The item was declined because push time-out is enabled and the item request timed out with the agent.</li> <li>• Closed – The item is closed.</li> <li>• Canceled – The item no longer needs to be routed. For example: a chat visitor cancels their Omni-Channel routed chat request before it reaches an agent.</li> <li>• Transferred – The item was transferred from an agent to another agent, queue, or skill.</li> </ul>
SystemModstamp	<p><b>Type</b></p> <p>dateTime</p> <p><b>Properties</b></p> <p>Defaulted on create, Filter, Sort</p> <p><b>Description</b></p> <p>Date and time when a user or automated process (such as a trigger) last modified this record.</p>
UserId	<p><b>Type</b></p> <p>reference</p> <p><b>Properties</b></p> <p>Create, Filter, Group, Sort</p> <p><b>Description</b></p> <p>The ID of the user that the work item was assigned to.</p>
WorkItemId	<p><b>Type</b></p> <p>reference</p> <p><b>Properties</b></p> <p>Create, Filter, Group, Sort</p>

Field	Details
	<p><b>Description</b></p> <p>The ID of the object that's routed to the agent through Omni-Channel.</p>

## Usage

`AgentWork` records can only be deleted if they have the status Closed, Declined, or Unavailable. They can't be deleted if their status is Assigned or Opened because they're active in Omni-Channel.

`AgentWork` records have the status Assigned when they're created. Once created, the record is automatically pushed to the assigned agent.

While the metadata for `AgentWork` indicates support for `upsert()` and `update()`, these calls aren't used with `AgentWork` because none of its fields can be updated.

Apex triggers are supported with `AgentWork`.


## OmniSupervisorConfig

Represents a configuration that determines a user's Omni-Channel Supervisor settings. This object is available in API version 41.0 and later.

## Supported Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`, `upsert()`

## Fields

Field	Details
<code>DeveloperName</code>	<p><b>Type</b></p> <p>string</p> <p><b>Properties</b></p> <p>Create, Filter, Group, Sort, Update</p> <p><b>Description</b></p> <p>The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization.</p> <p> <b>Note:</b> When creating large sets of data, always specify a unique <code>DeveloperName</code> for each record. If no <code>DeveloperName</code> is specified, performance may slow while Salesforce generates one for each record.</p>

Field	Details
Language	<p><b>Type</b> picklist</p> <p><b>Properties</b> Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update</p> <p><b>Description</b> The language of the Omni-Channel Supervisor configuration.</p>
MasterLabel	<p><b>Type</b> string</p> <p><b>Properties</b> Create, Filter, Group, Sort, Update</p> <p><b>Description</b> The master label for the the <code>OmniSupervisorConfig</code>.</p>

## OmniSupervisorConfigGroup

Represents a public group assigned to the Omni-Channel Supervisor configuration that's visible to a supervisor user. This object is available in API version 41.0 and later.

### Supported Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`, `upsert()`

### Fields

Field	Details
GroupId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Sort</p> <p><b>Description</b> The ID of the group that's made visible to the supervisor user in the Omni-Channel Supervisor configuration.</p>
OmniSupervisorConfigId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Sort</p> <p><b>Description</b> The ID of the Omni-Channel Supervisor configuration.</p>

## OmniSupervisorConfigProfile

Represents a configuration that determines the Omni-Channel Supervisor settings that are assigned to users who are assigned to a specific profile. User-level configurations override profile-level configurations. This object is available in API version 41.0 and later.

### Supported Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`, `upsert()`

### Fields

Field	Details
OmniSupervisorConfigId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Sort</p> <p><b>Description</b> The ID of the Omni-Channel Supervisor configuration.</p>
ProfileId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Sort</p> <p><b>Description</b> The ID of the profile associated with this Omni-Channel Supervisor configuration. A profile can be associated with only one Omni-Channel Supervisor configuration.</p>

## OmniSupervisorConfigUser

Represents a configuration that determines the Omni-Channel Supervisor settings that are assigned to a user. User-level configurations override profile-level configurations. This object is available in API version 41.0 and later.

### Supported Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`, `upsert()`

### Fields

Field	Details
OmniSupervisorConfigId	<p><b>Type</b> reference</p>

Field	Details
	<p><b>Properties</b> Create, Filter, Group, Sort</p> <p><b>Description</b> The ID of the Omni-Channel Supervisor configuration.</p>
UserId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Sort</p> <p><b>Description</b> ID of the user associated with this Omni-Channel Supervisor configuration. A user can be associated with only one Omni-Channel Supervisor configuration.</p>

## PendingServiceRouting

Represents a work assignment that's waiting to be routed. This object is available in API version 40.0 and later.

### Supported Calls

`query()`, `getDeleted()`, `getUpdated()`, `retrieve()`

### Fields

Field	Details
CapacityPercentage	<p><b>Type</b> double</p> <p><b>Properties</b> Filter, Nillable, Sort, Update</p> <p><b>Description</b> Indicates the amount of work that this item represents as a percentage. Valid values are 0–100.</p>
CapacityWeight	<p><b>Type</b> double</p> <p><b>Properties</b> Filter, Nillable, Sort, Update</p> <p><b>Description</b> Indicates the amount of work that this work item represents as a whole number.</p>



Field	Details
CustomRequestedDatetime	<p><b>Type</b> dateTime</p> <p><b>Properties</b> Filter, Nillable, Sort, Update</p> <p><b>Description</b> Retains the datetime of a work item's initial request, so work items are rerouted using the datetime of the initial work request. When left blank, work items are rerouted using the datetime when they were rerouted.</p>
IsPushAttempted	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> Indicates whether a push has been attempted. <code>true</code> if an agent was chosen at least once to route this <code>PendingServiceRouting</code>; <code>false</code> otherwise.</p>
IsPushed	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> Indicates whether the <code>PendingServiceRouting</code> is pushed to an agent.</p>
IsReadyforRouting	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> Indicates whether the work item is ready to be routed to an agent. You can't edit a <code>PendingServiceRouting</code> object that is set to <code>True</code>.</p>
IsTransfer	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> Indicates whether this <code>PendingServiceRouting</code> is a transfer request.</p>
LastDeclinedAgentSession	<p><b>Type</b> string</p>

Field	Details
	<p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> Live Agent Session ID of the agent who last declined this work item.</p>
Name	<p><b>Type</b> string</p> <p><b>Properties</b> Defaulted on create, Filter, Sort</p> <p><b>Description</b> Name of the <code>PendingServiceRouting</code>.</p>
OwnerId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The ID of the owner of this <code>PendingServiceRouting</code>.</p>
QueueId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The ID of the Omni-Channel queue.</p>
RoutingModel	<p><b>Type</b> picklist</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> Type of routing model. For a queue configured with a queue routing configuration, the routing model is <code>ExternalRouting</code> for all external routing <code>PendingServiceRouting</code>.</p>
RoutingPriority	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The order in which work items are routed to agents.</p>

Field	Details
RoutingType	<p><b>Type</b> string</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> Indicates whether the work item is routed by queue or by skills-based routing. Valid values are <i>SkillsBased</i>   <i>Queue</i>.</p>
Serial	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> Serial number of the <code>PendingServiceRouting</code>. The serial number is automatically incremented each time the <code>PendingServiceRouting</code> is modified.</p>
ServiceChannelId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> ID of the Service Channel.</p>
WorkItemId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> ID of the work item.</p>

## Usage

When you use `PendingServiceRouting` objects for queue-based routing, the `PendingServiceRouting` objects don't invoke before insert triggers.

Sharing rules aren't supported with `PendingServiceRouting` even if the `ownerId` field is enabled.

## PresenceConfigDeclineReason

Represents the settings for a decline reason that a presence user provides when declining work. This object is available in API version 37.0 and later.

## Supported Calls

`create()`, `delete()`, `describeObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

## Fields

Field	Details
PresenceDeclineReasonId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Sort</p> <p><b>Description</b> The ID of the <a href="#">PresenceDeclineReason</a> record.</p>
PresenceUserConfigId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Sort</p> <p><b>Description</b> The ID of the <a href="#">PresenceUserConfig</a> record where the decline reasons are added.</p>

## PresenceDeclineReason


Represents an Omni-Channel decline reason that agents can select when declining work requests. This object is available in API version 37.0 and later.

## Supported Calls

`create()`, `delete()`, `describeObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

## Fields

Field	Details
DeveloperName	<p><b>Type</b> string</p> <p><b>Properties</b> Create, Filter, Group, Sort, Update</p> <p><b>Description</b> The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With</p>

Field	Details
	<p>this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization.</p> <p> <b>Note:</b> When creating large sets of data, always specify a unique <code>DeveloperName</code> for each record. If no <code>DeveloperName</code> is specified, performance may slow while Salesforce generates one for each record.</p>
Language	<p><b>Type</b> picklist</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Restricted picklist, Sort, Update</p> <p><b>Description</b> The language of the <a href="#">PresenceDeclineReason</a>.</p>
MasterLabel	<p><b>Type</b> string</p> <p><b>Properties</b> Create, Filter, Group, Sort, Update</p> <p><b>Description</b> The master label for the <a href="#">PresenceDeclineReason</a>.</p>

## PresenceUserConfig


Represents a configuration that determines a presence user's settings. This object is available in API version 32.0 and later.

### Supported Calls

`create()`, `delete()`, `describeObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

### Fields

Field	Details
Capacity	<p><b>Type</b> int</p> <p><b>Properties</b> Create, Filter, Group, Sort, Update</p> <p><b>Description</b> The maximum number of work assignments that can be pushed to an agent at a time.</p>
DeveloperName	<p><b>Type</b> string</p>

Field	Details
	<p><b>Properties</b> Create, Filter, Group, Sort, Update</p> <p><b>Description</b> The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization.</p> <p> <b>Note:</b> When creating large sets of data, always specify a unique <code>DeveloperName</code> for each record. If no <code>DeveloperName</code> is specified, performance may slow while Salesforce generates one for each record.</p>
Language	<p><b>Type</b> picklist</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Restricted picklist, Sort, Update</p> <p><b>Description</b> The language of the presence configuration.</p>
MasterLabel	<p><b>Type</b> string</p> <p><b>Properties</b> Create, Filter, Group, Sort, Update</p> <p><b>Description</b> The label of the presence configuration.</p>
OptionsIsAutoAcceptEnabled	<p><b>Type</b> boolean</p> <p><b>Properties</b> Create, Filter, Update</p> <p><b>Description</b> Indicates whether work items that are routed to agents are automatically accepted (<code>true</code>) or not (<code>false</code>). Available only if <code>OptionsIsDeclineEnabled</code> is set to <code>false</code>.</p>
OptionsIsDeclineEnabled	<p><b>Type</b> boolean</p> <p><b>Properties</b> Create, Filter, Update</p> <p><b>Description</b> Indicates whether agents can decline work items that are routed to them (<code>true</code>) or not (<code>false</code>). Available only if <code>OptionsIsAutoAcceptEnabled</code> is set to <code>false</code>.</p>

Field	Details
OptionsIsDeclineReasonEnabled	<p><b>Type</b> boolean</p> <p><b>Properties</b> Create, Filter, Update</p> <p><b>Description</b> Indicates whether agents can select a reason for declining work requests (<code>true</code>) or not (<code>false</code>). This can be selected only if decline reasons are enabled.</p>
OptionsIsDisconnectSoundEnabled	<p><b>Type</b> boolean</p> <p><b>Properties</b> Create, Filter, Update</p> <p><b>Description</b> Indicates whether a sound is played when agents are disconnected from Omni-Channel (<code>true</code>) or not (<code>false</code>).</p>
OptionsIsRequestSoundEnabled	<p><b>Type</b> boolean</p> <p><b>Properties</b> Create, Filter, Update</p> <p><b>Description</b> Indicates whether a sound plays with incoming work requests (<code>true</code>) or not (<code>false</code>). Set to <code>true</code> by default.</p>
PresenceStatusOnDeclineId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Sort, Update</p> <p><b>Description</b> The ID of the presence status that's automatically assigned to the agent when the agent declines a work item. Available only if <code>OptionsIsDeclineEnabled</code> is set to <code>true</code>.</p>
PresenceStatusOnPushTimeoutId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Sort, Update</p> <p><b>Description</b> The ID of the presence status that's automatically assigned to the agent when the agent doesn't respond to a work item before push timeout occurs. Available in API version 36.0 and later.</p>

## PresenceUserConfigProfile

Represents a configuration that determines the settings that are assigned to presence users who are assigned to a specific profile. User-level configurations override profile-level configurations. This object is available in API version 32.0 and later.

### Supported Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`

### Fields

Field	Details
PresenceUserConfigId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Sort</p> <p><b>Description</b> If an individual user is also assigned a presence configuration through the PresenceUserConfigProfile, this configuration will override that.</p>
ProfileId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Sort</p> <p><b>Description</b> The ID of the profile that's associated with this presence configuration. A profile can be associated with only one presence configuration.</p>

## PresenceUserConfigUser

Represents a configuration that determines the settings that are assigned to a presence user. These user-level configurations override profile-level configurations. This object is available in API version 32.0 and later.

### Supported Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`

### Fields

Field	Details
PresenceUserConfigId	<p><b>Type</b> reference</p>



Field	Details
	<p><b>Properties</b> Create, Filter, Group, Sort</p> <p><b>Description</b> The ID of the presence configuration.</p>
UserId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Sort</p> <p><b>Description</b> The ID of the user who's associated with this presence configuration. A user can be associated with only one presence configuration.</p>

## QueueRoutingConfig


Represents the settings that determine how work items are routed to agents. This object is available in API version 32.0 and later.

### Supported Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`

### Fields

Field	Details
CapacityPercentage	<p><b>Type</b> percent</p> <p><b>Properties</b> Create, Filter, Nillable, Sort, Update</p> <p><b>Description</b> The percentage of an agent's capacity for work items that's consumed by a specific type of work item from this service channel.  For example, you might give phone calls a capacity percentage of <i>100</i>. If an agent receives a phone call, the agent won't receive new work items until the call ends, because at that point the agent's capacity will have reached 100%.  This field is available in API version 33.0 and later.</p>
CapacityWeight	<p><b>Type</b> double</p> <p><b>Properties</b> Create, Filter, Nillable, Sort, Update</p>

Field	Details
	<p><b>Description</b></p> <p>The amount of an agent's capacity for work items that's consumed by a work item from this service channel.</p> <p>For example, if an agent has a capacity of 6, and cases are assigned a capacity weight of 2, an agent can be assigned up to 3 cases before the agent is at capacity and can't receive new work items.</p> <p>This field is available in API version 33.0 and later.</p>
DeveloperName	<p><b>Type</b></p> <p>string</p> <p><b>Properties</b></p> <p>Create, Filter, Group, Sort, Update</p> <p><b>Description</b></p> <p>The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization.</p> <p> <b>Note:</b> When creating large sets of data, always specify a unique <code>DeveloperName</code> for each record. If no <code>DeveloperName</code> is specified, performance may slow while Salesforce generates one for each record.</p>
Language	<p><b>Type</b></p> <p>picklist</p> <p><b>Properties</b></p> <p>Create, Filter, Group, Nillable, Restricted picklist, Sort, Update</p> <p><b>Description</b></p> <p>The language of the presence status.</p>
MasterLabel	<p><b>Type</b></p> <p>string</p> <p><b>Properties</b></p> <p>Create, Filter, Group, Sort, Update</p> <p><b>Description</b></p> <p>The label of the presence status.</p>
OverflowAssigneeId	<p><b>Type</b></p> <p>reference</p> <p><b>Properties</b></p> <p>Create, Filter, Group, Nillable, Sort, Update</p>

Field	Details
	<p><b>Description</b> The ID of the user or queue that's set as the Overflow Assignee.</p>
PushTimeout	<p><b>Type</b> int</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Sort, Update</p> <p><b>Description</b> The number of seconds set for push timeout. <b>0</b> is returned when push timeout isn't enabled. Available in API version 36.0 and later.</p>
RoutingModel	<p><b>Type</b> picklist</p> <p><b>Properties</b> Create, Filter, Group, Restricted picklist, Sort, Update</p> <p><b>Description</b> The routing type that determines how work items are routed (pushed) to agents. Possible values are <code>Least Active</code> and <code>Most Available</code>.</p>
RoutingPriority	<p><b>Type</b> int</p> <p><b>Properties</b> Create, Filter, Group, Sort, Update</p> <p><b>Description</b> The priority in which work items from the service channels that are related to this routing configuration are routed to agents. Work items from routing configurations that have lower priority values (for example, <code>0</code>) are routed to agents first.</p>
ServiceChannelId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Sort, Update</p> <p><b>Description</b> The ID of the service channel that's associated with this configuration.</p>

## ServiceChannel


Represents a channel of work items that are received from your organization—for example, cases, chats, or leads. This object is available in API version 32.0 and later.

## Supported Calls

`create()`, `query()`, `retrieve()`, `update()`, `upsert()`

## Fields

Field	Details
CapacityPercentage	<p><b>Type</b> percent</p> <p><b>Properties</b> Create, Filter, Nillable, Sort, Update</p> <p><b>Description</b> The percentage of an agent's capacity for work items that's consumed by a specific type of work item from this service channel.</p> <p>For example, you might give phone calls a capacity percentage of <code>100</code>. If an agent receives a phone call, the agent won't receive new work items until the call ends, because at that point the agent's capacity will have reached 100%.</p> <p>This field is available in API version 32.0 and earlier. For later API versions, you can set the capacity percentage of work items on the <a href="#">QueueRoutingConfig</a> object.</p>
CapacityWeight	<p><b>Type</b> double</p> <p><b>Properties</b> Create, Filter, Nillable, Sort, Update</p> <p><b>Description</b> The amount of an agent's capacity for work items that's consumed by a work item from this service channel.</p> <p>For example, if an agent has a capacity of <code>6</code>, and cases are assigned a capacity weight of <code>2</code>, an agent can be assigned up to 3 cases before the agent is at capacity and can't receive new work items.</p> <p>This field is available in API version 32.0 and earlier. For later API versions, you can set the capacity weight of work items on the <a href="#">QueueRoutingConfig</a> object.</p>
DeveloperName	<p><b>Type</b> string</p> <p><b>Properties</b> Create, Filter, Group, Sort, Update</p> <p><b>Description</b> The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization.</p>

Field	Details
	 <b>Note:</b> When creating large sets of data, always specify a unique <code>DeveloperName</code> for each record. If no <code>DeveloperName</code> is specified, performance may slow while Salesforce generates one for each record.
Language	<p><b>Type</b> picklist</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Restricted picklist, Sort, Update</p> <p><b>Description</b> The language of the service channel.</p>
MasterLabel	<p><b>Type</b> string</p> <p><b>Properties</b> Create, Filter, Group, Sort, Update</p> <p><b>Description</b> The label of the service channel.</p>
RelatedEntity	<p><b>Type</b> picklist</p> <p><b>Properties</b> Create, Filter, Group, Restricted picklist, Sort, Unique, Update</p> <p><b>Description</b> The type of object that's associated with this service channel.</p>

## ServiceChannelStatus

Represents the status that's associated with a specific service channel. This object is available in API version 32.0 and later.

### Supported Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`

### Fields

Field	Details
ServiceChannelId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Sort</p>

Field	Details
	<p><b>Description</b> The ID of the service channel.</p>
ServicePresenceStatusId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Sort</p> <p><b>Description</b> The ID of the presence status that's associated with the service channel that's specified by the <code>ServicePresenceChannelId</code>.</p>


## ServicePresenceStatus

Represents a presence status that can be assigned to a service channel. This object is available in API version 32.0 and later.

### Supported Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`, `upsert()`

### Fields

Field	Details
DeveloperName	<p><b>Type</b> string</p> <p><b>Properties</b> Create, Filter, Group, Sort, Update</p> <p><b>Description</b> The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization.</p> <p> <b>Note:</b> When creating large sets of data, always specify a unique <code>DeveloperName</code> for each record. If no <code>DeveloperName</code> is specified, performance may slow while Salesforce generates one for each record.</p>
Language	<p><b>Type</b> picklist</p> <p><b>Properties</b> Create, Filter, Group, Nillable, Restricted picklist, Sort, Update</p>

Field	Details
	<p><b>Description</b> The language of the presence status.</p>
MasterLabel	<p><b>Type</b> string</p> <p><b>Properties</b> Create, Filter, Group, Sort, Update</p> <p><b>Description</b> The label of the presence status.</p>

## UserServicePresence

Represents a presence user's real-time presence status. This object is available in API version 32.0 and later.

### Supported Calls

`delete()`, `query()`, `getDeleted()`, `getUpdated()`, `retrieve()`, `undelete()`

### Fields

Field	Details
ConfiguredCapacity	<p><b>Type</b> int</p> <p><b>Properties</b> Filter, Group, Nillable, Sort</p> <p><b>Description</b> The user's total configured capacity.</p>
IsAway	<p><b>Type</b> boolean</p> <p><b>Properties</b> Defaulted on create, Filter, Group, Sort</p> <p><b>Description</b> Indicates whether the user's status is <code>Away</code>.</p>
Name	<p><b>Type</b> string</p> <p><b>Properties</b> Create, Filter, Nillable, Sort, Update</p>

Field	Details
	<p><b>Description</b> An automatically generated ID number that identifies the record.</p>
OwnerId	<p><b>Type</b> reference</p> <p><b>Properties</b> Filter, Group, Sort</p> <p><b>Description</b> The ID of the owner of the <code>UserServicePresence</code> entity. For external routing, allows the entity to be used in the Streaming API to listen to events whenever a <code>UserServicePresence</code> record is created, modified, or deleted.</p>
ServicePresenceStatusId	<p><b>Type</b> reference</p> <p><b>Properties</b> Create, Filter, Nillable, Sort, Update</p> <p><b>Description</b> The ID of the presence status that's associated with the presence user that's specified by the <code>UserId</code>.</p>
UserId	<p><b>Type</b> string</p> <p><b>Properties</b> Create, Filter, Group, Sort, Update</p> <p><b>Description</b> The ID of the Omni-Channel user.</p>

## Usage

Apex triggers aren't supported with `UserServicePresence`.

Sharing rules aren't supported with `UserServicePresence` even if the `OwnerId` field is enabled.

In API version 41.0 or later, `UserServicePresence` records can be deleted programmatically. The "Customize Application" permission is required.

## Omni-Channel Objects for the Salesforce Console

Omni-Channel lets your call center route any type of incoming work item to the most qualified, available agents.

### [Omni-Channel Objects for the Salesforce Console Integration Toolkit](#)

The Salesforce Console Integration Toolkit includes several objects that let you control how Omni-Channel works within the Salesforce console for your organization.



### [Omni-Channel Objects for the Lightning Console JavaScript API](#)

Omni-Channel lets your call center route any type of incoming work item to the most qualified, available agents. The Lightning Console JavaScript API for Lightning Experience includes several objects that let you control how Omni-Channel works within the Lightning Service Console for your organization.

## Omni-Channel Objects for the Salesforce Console Integration Toolkit

The Salesforce Console Integration Toolkit includes several objects that let you control how Omni-Channel works within the Salesforce console for your organization.

If you need more information on the Salesforce Console Integration Toolkit, see [Salesforce Console Integration Toolkit for Salesforce Classic](#).

### [acceptAgentWork](#)

Accepts a work item that's assigned to an agent. Available in API versions 32.0 and later.

### [closeAgentWork](#)

Changes the status of a work item to "Closed" and removes it from the list of work items in the Omni-Channel widget. Available in API versions 32.0 and later.

### [declineAgentWork](#)

Declines a work item that's assigned to an agent. Available in API versions 32.0 and later.

### [getAgentWorks](#)

Returns a list of work items that are currently assigned to an agent and open in the agent's workspace. Available in API versions 32.0 and later.

### [getAgentWorkload](#)

In API version 35.0 and later, we can retrieve an agent's currently assigned workload. Use this method for rerouting work to available agents.

### [getServicePresenceStatusChannels](#)

Retrieves the service channels that are associated with an Omni-Channel user's current presence status. Available in API versions 32.0 and later.

### [getServicePresenceStatusId](#)

Retrieves an agent's current presence status. Available in API versions 32.0 and later.

### [login](#)

Logs an agent into Omni-Channel with a specific presence status. You also can use this method to reconnect to Omni-Channel after a connection error. Available in API versions 32.0 and later.

### [logout](#)

Logs an agent out of Omni-Channel. Available in API versions 32.0 and later.

### [setServicePresenceStatus](#)

Sets an agent's presence status to a status with a particular ID. In API version 35.0 and later, we log the user into presence if that user is not already logged in, so you don't have to make additional calls. You also can use this method to reconnect to Omni-Channel after a connection error.

### [Methods for Console Events](#)

JavaScript can be executed when certain types of events occur in a console, such as when a user closes a tab. In addition to the standard methods for console events, there are a few events that are specific to Omni-Channel. These events apply to Salesforce Classic only.

## acceptAgentWork

Accepts a work item that's assigned to an agent. Available in API versions 32.0 and later.

### Syntax

```
sforce.console.presence.acceptAgentWork(workId:String, (optional) callback:function)
```

### Arguments

Name	Type	Description
workId	String	The ID of the work item the agent accepts.
callback	function	JavaScript method to call when an agent accepts the work item associated with the workId.

### Sample Code–Visualforce

```
<apex:page>
  <apex:includeScript value="/support/console/44.0/integration.js"/>
  <a href="#" onClick="testAcceptWork();return false;">Accept Assigned Work Item</a>

  <script type="text/javascript">
    function testAcceptWork() {
      //First get the ID of the assigned work item to accept it
      sforce.console.presence.getAgentWorks(function(result) {
        if (result.success) {
          var works = JSON.parse(result.works);
          var work = works[0];
          if (!work.isEngaged) {
            //Now that we have the assigned work item ID, we can accept it
            sforce.console.presence.acceptAgentWork(work.workId,
function(result) {
          if (result.success) {
            alert('Accepted work successfully');
          } else {
            alert('Accept work failed');
          }
        });
        } else {
          alert('The work item has already been accepted');
        }
      });
    }
  </script>
</apex:page>
```

## Response

This method is asynchronous so it returns its response in an object in a callback method. The response object contains the following properties:

Name	Type	Description
success	Boolean	true if accepting the work item was successful; false if accepting the work item wasn't successful.

## closeAgentWork

Changes the status of a work item to "Closed" and removes it from the list of work items in the Omni-Channel widget. Available in API versions 32.0 and later.

## Syntax

```
sforce.console.presence.closeAgentWork(workId:String, (optional) callback:function)
```

## Arguments

Name	Type	Description
workId	String	The ID of the work item that's closed.
callback	function	JavaScript method to call when the work item associated with the workId is closed.

## Sample Code–Visualforce

```
<apex:page>
  <apex:includeScript value="/support/console/44.0/integration.js"/>
  <a href="#" onClick="testCloseWork();return false;">Close Engaged Work Item</a>
  <script type="text/javascript">
    function testCloseWork() {
      //First get the ID of the engaged work item to close it
      sforce.console.presence.getAgentWorks(function(result) {
        if (result.success) {
          var works = JSON.parse(result.works);
          var work = works[0];
          if (work.isEngaged) {
            //Now that we have the engaged work item ID, we can close it
            sforce.console.presence.closeAgentWork(work.workId,function(result)
            {
              if (result.success) {
                alert('Closed work successfully');
              } else {
                alert('Close work failed');
              }
            }
            ));
          }
        }
      });
    }
  </script>
</apex:page>
```

```

        } else {
            alert('The work item should be accepted first');
        }
    }
});
}
</script>
</apex:page>

```

## Response

This method is asynchronous so it returns its response in an object in a callback method. The response object contains the following properties:

Name	Type	Description
success	Boolean	true if closing the work item was successful; false if closing the work item wasn't successful.

## declineAgentWork

Declines a work item that's assigned to an agent. Available in API versions 32.0 and later.

## Syntax

```

sforce.console.presence.declineAgentWork(workId:String, (optional) declineReason:String,
(optional) callback:function)

```

## Arguments

Name	Type	Description
workId	String	The ID of the work item that the agent declines.
declineReason	String	The provided reason for why the agent declined the work request.
callback	function	JavaScript method to call when an agent declines the work item associated with the workId.

## Sample Code–Visualforce

```

<apex:page >
    <apex:includeScript value="/support/console/44.0/integration.js"/>
    <a href="#" onClick="testDeclineWork();return false;">Decline Assigned Work Item</a>

    <script type="text/javascript">
        function testDeclineWork() {
            //First, get the ID of the assigned work item to accept it
            sforce.console.presence.getAgentWorks(function(result) {

```

```

        if (result.success) {
            var works = JSON.parse(result.works);
            var work = works[0];
            sforce.console.presence.declineAgentWork(work.workId, function(result)
        {
            if (result.success) {
                alert('Declined work successfully');
            } else {
                alert('Decline work failed');
            }
        });
    });
}
</script>
</apex:page>

```

## Response

This method is asynchronous so it returns its response in an object in a callback method. The response object contains the following properties:

Name	Type	Description
success	Boolean	true if declining the work item was successful; false if declining the work item wasn't successful.

## getAgentWorks

Returns a list of work items that are currently assigned to an agent and open in the agent's workspace. Available in API versions 32.0 and later.

## Syntax

```
sforce.console.presence.getAgentWorks (callback: function)
```

## Arguments

Name	Type	Description
callback	function	JavaScript method to call when the list of an agent's work items is retrieved.

## Sample Code–Visualforce

```

<apex:page>
    <apex:includeScript value="/support/console/44.0/integration.js"/>
    <a href="#" onClick="testGetWorks();return false;">Get Agent's Current Work Items</a>

```

```

<script type="text/javascript">
  function testGetWorks() {
    //These values are for example purposes only.
    sforce.console.presence.getAgentWorks(function(result) {
      if (result.success) {
        alert('Get work items successful');
        var works = JSON.parse(result.works);
        alert('First Agent Work ID is: ' + works[0].workId);
        alert('Assigned Entity Id of the first Agent Work is: ' +
works[0].workItemId);
        alert('Is first Agent Work Engaged: ' + works[0].isEngaged);
      } else {
        alert('Get work items failed');
      }
    });
  }
</script>
</apex:page>

```

## Response

This method is asynchronous so it returns its response in an object in a callback method. The response object contains the following properties:

Name	Type	Description
success	Boolean	<code>true</code> if retrieving the agent's work items was successful; <code>false</code> if retrieving the agent's work items wasn't successful.
works	JSON string of <a href="#">work objects</a>	A JSON string of <code>work</code> objects that represents the work items assigned to the agent that are open in the agent's workspace.

## work

The `work` object contains the following properties:

Name	Type	Description
workItemId	String	The ID of the object that's routed through Omni-Channel. This object becomes a work assignment with a <code>workId</code> when it's assigned to an agent.
workId	String	The ID of a work assignment that's routed to an agent.
isEngaged	Boolean	Indicates whether an agent is working on a work item that's been assigned to them ( <code>true</code> ) or not ( <code>false</code> ).

## getAgentWorkload

In API version 35.0 and later, we can retrieve an agent's currently assigned workload. Use this method for rerouting work to available agents.

## Syntax

```
sforce.console.presence.getAgentWorkload(callback: function)
```

## Arguments

Name	Type	Description
callback	function	JavaScript method to call when the agent's configured capacity and work retrieved.

## Sample Code–Visualforce

```
<apex:page>
  <apex:includeScript value="/support/console/44.0/integration.js"/>
  <a href="#" onClick="testGetAgentWorkload();return false;">
    Get Agent's configured capacity and current workload
  </a>

  <script type="text/javascript">
    function testGetAgentWorkload() {
      sforce.console.presence.getAgentWorkload(function(result) {
        if (result.success) {
          alert('Retrieved Agent Configured Capacity and Current Workload
successfully');
          alert('Agent\'s configured capacity is: ' + result.configuredCapacity);

          alert('Agent\'s currently assigned workload is: ' +
result.currentWorkload);
        } else {
          alert('Get Agent Workload failed');
        }
      });
    }
  </script>
</apex:page>
```

## Response

This method is asynchronous so it returns its response in an object in a callback method. The response object contains the following properties:

Name	Type	Description
success	Boolean	<code>true</code> if retrieving the agent's work items was successful; <code>false</code> if retrieving the agent's work items wasn't successful.
configuredCapacity	Number	Indicates the agent's configured capacity (work that's assigned to the current user) through Presence Configuration.
currentWorkload	Number	Indicates the agent's currently assigned workload.

## getServicePresenceStatusChannels

Retrieves the service channels that are associated with an Omni-Channel user's current presence status. Available in API versions 32.0 and later.

### Syntax

```
sforce.console.presence.getServicePresenceStatusChannels (callback: function)
```

### Arguments

Name	Type	Description
callback	function	JavaScript method to call when the channels associated with a presence status are retrieved.

### Sample Code—Visualforce

```
<apex:page>
  <apex:includeScript value="/support/console/44.0/integration.js"/>
  <a href="#" onClick="testGetChannels();return false;">
    Get Channels Associated with a Presence Status
  </a>

  <script type="text/javascript">
    function testGetChannels() {
      //These values are for example purposes only.
      sforce.console.presence.getServicePresenceStatusChannels (function (result) {
        if (result.success) {
          alert('Retrieved Service Presence Status Channels successfully');
          var channels = JSON.parse(result.channels);
          //For example purposes, just retrieve the first channel
          alert('First channel ID is: ' + channels[0].channelId);
          alert('First channel developer name is: ' + channels[0].developerName);
        } else {
          alert('Get Service Presence Status Channels failed');
        }
      });
    }
  </script>
</apex:page>
```

### Response

This method is asynchronous so it returns its response in an object in a callback method. The response object contains the following properties:



Name	Type	Description
success	Boolean	<code>true</code> if retrieving the current presence status channels was successful; <code>false</code> if the retrieving the current presence status channels wasn't successful.
channels	JSON string of channel objects	Returns the IDs and API names of the channels associated with the presence status.

## getServicePresenceStatusId

Retrieves an agent's current presence status. Available in API versions 32.0 and later.

### Syntax

```
sforce.console.presence.getServicePresenceStatusId(callback: function)
```

### Arguments

Name	Type	Description
callback	function	JavaScript method to call when the agent's presence status is retrieved.

### Sample Code—Visualforce

```
<apex:page>
  <apex:includeScript value="/support/console/44.0/integration.js"/>
  <a href="#" onClick="testGetStatusId();return false;">Get Omni-Channel Status ID</a>

  <script type="text/javascript">
    function testGetStatusId() {
      sforce.console.presence.getServicePresenceStatusId(function(result) {
        if (result.success) {
          alert('Get Status Id successful');
          alert('Status Id is: ' + result.statusId);
        } else {
          alert('Get Status Id failed');
        }
      });
    }
  </script>
</apex:page>
```

### Response

This method is asynchronous so it returns its response in an object in a callback method. The response object contains the following properties:

Name	Type	Description
success	Boolean	<code>true</code> if retrieving the presence status ID was successful; <code>false</code> if the retrieving the presence status ID wasn't successful.
statusName	String	The name of the agent's current presence status.
statusApiName	String	The API name of the agent's current presence status.
statusId	String	The ID of the agent's current presence status.

## login

Logs an agent into Omni-Channel with a specific presence status. You also can use this method to reconnect to Omni-Channel after a connection error. Available in API versions 32.0 and later.

## Syntax

```
sforce.console.presence.login(statusId:String, (optional) callback:function)
```

## Arguments

Name	Type	Description
statusId	String	The ID of the presence status. Agents must be given access to this presence status through their associated profile or permission set.
callback	function	JavaScript method to call when the agent is logged in with the presence status associated with <code>statusId</code> .

## Sample Code–Visualforce

```
<apex:page>
  <apex:includeScript value="/support/console/44.0/integration.js"/>
  <a href="#" onClick="testLogin('0N5xx00000000081');return false;">Log In to
  Omni-Channel</a>

  <script type="text/javascript">
    function testLogin(statusId) {
      //Gets the Salesforce ID of the presence status entity which the current user
      has been assigned through their permission set or profile.
      //These values are for example purposes only.
      sforce.console.presence.login(statusId, function(result) {
        if (result.success) {
          alert('Login successful');
        } else {
          alert('Login failed');
        }
      });
    }
  </script>
</apex:page>
```

```
</script>
</apex:page>
```

## Response

This method is asynchronous so it returns its response in an object in a callback method. The response object contains the following properties:

Name	Type	Description
success	Boolean	true if the login was successful; false if the login wasn't successful.

## Logout

Logs an agent out of Omni-Channel. Available in API versions 32.0 and later.

## Syntax

```
sforce.console.presence.logout((optional) callback: function)
```

## Arguments

Name	Type	Description
callback	function	JavaScript method to call when the agent is logged out of Omni-Channel.

## Sample Code—Visualforce

```
<apex:page>
  <apex:includeScript value="/support/console/44.0/integration.js"/>
  <a href="#" onClick="testLogout();return false;">Log out of Omni-Channel</a>

  <script type="text/javascript">
    function testLogout() {
      sforce.console.presence.logout(function(result) {
        if (result.success) {
          alert('Logout successfully');
        } else {
          alert('Logout failed');
        }
      });
    }
  </script>
</apex:page>
```

## Response

This method is asynchronous so it returns its response in an object in a callback method. The response object contains the following properties:

Name	Type	Description
success	Boolean	true if the logout was successful; false if the logout wasn't successful.

## setServicePresenceStatus

Sets an agent's presence status to a status with a particular ID. In API version 35.0 and later, we log the user into presence if that user is not already logged in, so you don't have to make additional calls. You also can use this method to reconnect to Omni-Channel after a connection error.

### Syntax

```
sforce.console.presence.setServicePresenceStatus(statusId:String,
  (optional) callback:function)
```

### Arguments

Name	Type	Description
statusId	String	The ID of the presence status you want to set the agent to. Agents must be given access to this presence status through their associated profile or permission set.
callback	function	JavaScript method to call when the agent's status is changed to the presence status associated with statusId.

### Sample Code–Visualforce

```
<apex:page>
  <apex:includeScript value="/support/console/44.0/integration.js"/>
  <a href="#" onClick="testSetStatus('0N5xx00000000081');return false;">Set Presence
  Status</a>

  <script type="text/javascript">
    function testSetStatus(statusId) {

      //Sets the user's presence status to statusID. Assumes that the user was
      assigned this presence status through Setup.
      //These values are for example purposes only
      sforce.console.presence.setServicePresenceStatus(statusId, function(result) {

        if (result.success) {
          alert('Set status successful');
          alert('Current statusId is: ' + result.statusId);
          alert('Channel list attached to this status is: ' + result.channels);
          //printout in console for lists
        } else {
          alert('Set status failed');
        }
      });
    }
  }
```

```
</script>
</apex:page>
```

## Response

This method is asynchronous so it returns its response in an object in a callback method. The response object contains the following properties:

Name	Type	Description
<code>success</code>	Boolean	<code>true</code> if setting the agent's status was successful; <code>false</code> if setting the agent's status wasn't successful.
<code>statusName</code>	String	The name of the agent's current presence status.
<code>statusApiName</code>	String	The API name of the agent's current presence status.
<code>statusId</code>	String	The ID of the agent's current presence status.
<code>channels</code>	JSON string of <code>channel</code> objects	Returns the IDs and API names of the channels associated with the presence status.

## Methods for Console Events

JavaScript can be executed when certain types of events occur in a console, such as when a user closes a tab. In addition to the standard methods for console events, there are a few events that are specific to Omni-Channel. These events apply to Salesforce Classic only.

### Standard Console Events

Event	Description	Payload
<code>sforce.console.ConsoleEvent.OPEN_TAB</code>	Fired when a primary tab or subtab is opened. Available in API version 30.0 or later.	<ul style="list-style-type: none"> <li><code>id</code>—the ID of the opened tab</li> <li><code>objectId</code>—the object ID of the opened tab, if available</li> </ul>
<code>sforce.console.ConsoleEvent.CLOSE_TAB</code>	Fired when a primary tab or subtab with a specified ID in the <code>additionalParams</code> argument is closed. Or, fired when a primary tab or subtab with no specified ID is closed. Available in API version 30.0 or later.	<ul style="list-style-type: none"> <li><code>id</code>—the ID of the closed tab</li> <li><code>objectId</code>—the object ID of the closed tab, if available</li> </ul>
<code>sforce.console.ConsoleEvent.CONSOLE_LOGOUT</code>	Delays the execution of logging out of a console when a user clicks <b>Logout</b> . When <b>Logout</b> is clicked: <ol style="list-style-type: none"> <li>An overlay appears, which tells a user that logout is in progress.</li> </ol>	None

Event	Description	Payload
	<p>2. Callbacks are executed that have been registered by using <code>sfconsole.ConsoleEvent.CONSOLE_LOGOUT</code></p> <p>3. Console logout logic is executed.</p> <p>If the callback contains synchronous blocking code, the console logout code isn't executed until the blocking code is executed. As a best practice, avoid synchronous blocking code or long code execution during logout.</p> <p>Available in API version 31.0 or later.</p>	

## Omni-Channel Console Events

Event	Description	Payload
<code>sfconsole.ConsoleEvent.PRESENCE.LOGIN_SUCCESS</code>	<p>Fired when an Omni-Channel user logs into Omni-Channel successfully.</p> <p>Available in API version 32.0 or later.</p>	<ul style="list-style-type: none"> <li><code>statusId</code>—the ID of the agent's current presence status.</li> </ul>
<code>sfconsole.ConsoleEvent.PRESENCE.STATUS_CHANGED</code>	<p>Fired when a user changes his or her presence status.</p> <p>Available in API version 32.0 or later.</p>	<ul style="list-style-type: none"> <li><code>statusId</code>—the ID of the agent's current presence status.</li> <li><code>channels</code>—<a href="#">channel</a> JSON string of channel objects.</li> <li><code>statusName</code>—the name of the agent's current presence status.</li> <li><code>statusApiName</code>—the API name of the agent's current presence status.</li> </ul>
<code>sfconsole.ConsoleEvent.PRESENCE.LOGOUT</code>	<p>Fired when a user logs out of Salesforce.</p> <p>Available in API version 32.0 or later.</p>	None
<code>sfconsole.ConsoleEvent.PRESENCE.WORK_ASSIGNED</code>	<p>Fired when a user is assigned a new work item.</p> <p>Available in API version 32.0 or later.</p>	<ul style="list-style-type: none"> <li><code>workItemId</code>—the ID of the object that's routed through Omni-Channel. This object becomes a work assignment with a <code>workId</code> when it's assigned to an agent.</li> <li><code>workId</code>—the ID of a work assignment that's routed to an agent.</li> </ul>
<code>sfconsole.ConsoleEvent.PRESENCE.WORK_ACCEPTED</code>	<p>Fired when a user accepts a work assignment, or</p>	<ul style="list-style-type: none"> <li><code>workItemId</code>—the ID of the object that's routed through Omni-Channel. This object becomes a work assignment with a <code>workId</code> when it's assigned to an agent.</li> </ul>

Event	Description	Payload
	<p>when a work assignment is automatically accepted.</p> <p>Available in API version 32.0 or later.</p>	<ul style="list-style-type: none"> <li><code>workId</code>—the ID of a work assignment that's routed to an agent.</li> </ul>
<code>sforce.console.ConsoleEvent.PRESENCE.WORK_DECLINED</code>	<p>Fired when a user declines a work assignment.</p> <p>Available in API version 32.0 or later.</p>	<ul style="list-style-type: none"> <li><code>workItemId</code>—the ID of the object that's routed through Omni-Channel. This object becomes a work assignment with a <code>workId</code> when it's assigned to an agent.</li> <li><code>workId</code>—the ID of a work assignment that's routed to an agent.</li> </ul>
<code>sforce.console.ConsoleEvent.PRESENCE.WORK_CLOSED</code>	<p>Fired when a user closes a tab in the console that's associated with a work item. When the tab for that work item is closed, the status of the AgentWork object associated with it automatically changes to "Closed."</p> <p>Available in API version 32.0 or later.</p>	<ul style="list-style-type: none"> <li><code>workItemId</code>—the ID of the object that's routed through Omni-Channel. This object becomes a work assignment with a <code>workId</code> when it's assigned to an agent.</li> <li><code>workId</code>—the ID of a work assignment that's routed to an agent.</li> </ul>
<code>sforce.console.ConsoleEvent.PRESENCE.WORKLOAD_CHANGED</code>	<p>Fired when an agent's workload changes. This includes receiving new work items, declining work items, and closing items in the console. It's also fired when there's a change to an agent's capacity or Presence Configuration or when the agent goes offline in the Omni-Channel widget.</p>	<ul style="list-style-type: none"> <li><code>ConfiguredCapacity</code>—the configured capacity for the agent.</li> <li><code>PreviousWorkload</code>—the agent's workload before the change.</li> <li><code>NewWorkload</code>—the agent's new workload after the change.</li> </ul>

## channel

The `channel` object contains the following properties:

Name	Type	Description
<code>channelId</code>	String	Retrieves the ID of a service channel that's associated with a presence status.

Name	Type	Description
<code>developerName</code>	String	Retrieves the developer name of the the service channel that's associated with the <code>channelId</code> .

## Methods for Console Events

Method	Description
<a href="#">addEventListener()</a>	Adds a listener for a custom event type or a standard event type when the event is fired. This method adds a listener for custom event types in API version 25.0 or later; it adds a listener for standard event types in API version 30.0 or later.
<a href="#">fireEvent()</a>	Fires a custom event. This method is only available in API version 25.0 or later.
<a href="#">removeEventListener()</a>	Removes a listener for a custom event type or a standard event type. This method removes a listener for custom event types in API version 25.0 or later; it removes a listener for standard event types in API version 30.0 or later.

## Omni-Channel Objects for the Lightning Console JavaScript API

Omni-Channel lets your call center route any type of incoming work item to the most qualified, available agents. The Lightning Console JavaScript API for Lightning Experience includes several objects that let you control how Omni-Channel works within the Lightning Service Console for your organization.

If you need more information on the Lightning Console JavaScript API, see [Lightning Console JavaScript API for Lightning Experience](#).

### [acceptAgentWork for Lightning Experience](#)

Accepts a work item that's assigned to an agent.

### [closeAgentWork for Lightning Experience](#)

Changes the status of a work item to *Closed* and removes it from the list of work items in the Omni-Channel utility.

### [declineAgentWork for Lightning Experience](#)

Declines a work item that's assigned to an agent.

### [getAgentWorks for Lightning Experience](#)

Returns a list of work items that are assigned to an agent and open in the agent's workspace.

### [getAgentWorkload for Lightning Experience](#)

Retrieves an agent's currently assigned workload. Use this method to reroute work to available agents.

### [getServicePresenceStatusChannels for Lightning Experience](#)

Retrieves the service channels that are associated with an Omni-Channel user's current presence status.

### [getServicePresenceStatusId for Lightning Experience](#)

Retrieves an agent's current presence status.

### [login for Lightning Experience](#)

Logs an agent in to Omni-Channel with a specific presence status.



[logout for Lightning Experience](#)

Logs an agent out of Omni-Channel.

[setServicePresenceStatus for Lightning Experience](#)

Sets an agent's presence status to a status with a particular ID. If the specified agent is not already logged in, we log in the agent with the presence status. This method removes the need for you to make more calls.

[Events for Omni-Channel](#)

JavaScript can be executed when certain types of events occur in a console, such as when a user closes a tab. There are a few events that are specific to Omni-Channel. These events apply to Lightning Experience only.

## acceptAgentWork for Lightning Experience

Accepts a work item that's assigned to an agent.

### Arguments

Name	Type	Description
workId	string	The ID of the work item the agent accepts.

### Sample Code

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <lightning:button label="Accept" onclick="{! c.acceptWork }" />
</aura:component>
```

Controller code:

```
({
  acceptWork: function(cmp, evt, hlp) {
    var omniAPI = cmp.find("omniToolkit");
    omniAPI.getAgentWorks().then(function(result) {
      var works = JSON.parse(result.works);
      var work = works[0];
      omniAPI.acceptAgentWork({workId: work.workId}).then(function(res) {
        if (res) {
          console.log("Accepted work successfully");
        } else {
          console.log("Accept work failed");
        }
      }).catch(function(error) {
        console.log(error);
      });
    });
  }
})
```

## Response

This method returns a promise that, upon success, resolves to `true` and is rejected on error.

## `closeAgentWork` for Lightning Experience

Changes the status of a work item to *Closed* and removes it from the list of work items in the Omni-Channel utility.

## Arguments

Name	Type	Description
<code>workId</code>	string	The ID of the work item that's closed.

## Sample Code

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <lightning:button label="Close" onclick="{! c.closeWork }" />
</aura:component>
```

Controller code:

```
((
  closeWork: function(cmp, evt, hlp) {
    var omniAPI = cmp.find("omniToolkit");
    omniAPI.getAgentWorks().then(function(result) {
      var works = JSON.parse(result.works);
      var work = works[0];
      omniAPI.closeAgentWork({workId: work.workId}).then(function(res) {
        if (res) {
          console.log("Closed work successfully");
        } else {
          console.log("Close work failed");
        }
      }).catch(function(error) {
        console.log(error);
      });
    });
  }
}))
```

## Response

This method returns a promise that, upon success, resolves to `true` and is rejected on error.

## `declineAgentWork` for Lightning Experience

Declines a work item that's assigned to an agent.

## Arguments

Name	Type	Description
workId	string	The ID of the work item that the agent declines.
declineReason	string	The reason that the agent declined the work request.

## Sample Code

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <lightning:button label="Decline" onclick="{! c.declineWork }" />
</aura:component>
```

Controller code:

```
((
  declineWork: function(cmp, evt, hlp) {
    var omniAPI = cmp.find("omniToolkit");
    omniAPI.getAgentWorks().then(function(result) {
      var works = JSON.parse(result.works);
      var work = works[0];
      omniAPI.declineAgentWork({workId: work.workId}).then(function(res) {
        if (res) {
          console.log("Declined work successfully");
        } else {
          console.log("Decline work failed");
        }
      }).catch(function(error) {
        console.log(error);
      });
    });
  }
}))
```

## Response

This method returns a promise that, upon success, resolves to `true` and is rejected on error.

## `getAgentWorks` for Lightning Experience

Returns a list of work items that are assigned to an agent and open in the agent's workspace.

## Sample Code

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
```

```
<lightning:button label="Get Agent works" onclick="{! c.getAgentWorks }" />
</aura:component>
```

Controller code:

```
((
  getAgentWorks: function(cmp, evt, hlp) {
    var omniAPI = cmp.find("omniToolkit");
    omniAPI.getAgentWorks().then(function(result) {
      var works = JSON.parse(result.works);
      console.log('First Agent Work ID is: ' + works[0].workId);
      console.log('Assigned Entity Id of the first Agent Work is: ' +
works[0].workItemId);
      console.log('Is first Agent Work Engaged: ' + works[0].isEngaged);
    }).catch(function(error) {
      console.log(error);
    });
  }
})
```

## Response

This method returns a promise that, upon success, resolves to an array of `work` objects, containing the following fields.

Name	Type	Description
<code>workItemId</code>	String	The ID of the object that's routed through Omni-Channel. This object becomes a work assignment with a <code>workId</code> when it's assigned to an agent.
<code>workId</code>	String	The ID of a work assignment that's routed to an agent.
<code>isEngaged</code>	Boolean	Indicates whether an agent is working on a work item that's been assigned to them ( <code>true</code> ) or not ( <code>false</code> ).

## getAgentWorkload for Lightning Experience

Retrieves an agent's currently assigned workload. Use this method to reroute work to available agents.

### Sample Code

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <lightning:button label="Get workload" onclick="{! c.getAgentWorkload }" />
</aura:component>
```

Controller code:

```
((
  getAgentWorkload: function(cmp, evt, hlp) {
    var omniAPI = cmp.find("omniToolkit");
    omniAPI.getAgentWorkload().then(function(result) {
      console.log('Retrieved Agent Configured Capacity and Current Workload
```

```

successfully');
        console.log('Agent\'s configured capacity is: ' + result.configuredCapacity);

        console.log('Agent\'s currently assigned workload is: ' +
result.currentWorkload);
    }).catch(function(error) {
        console.log(error);
    });
}
})

```

## Response

This method returns a promise that, upon success, resolves to an object, containing the following fields.

Name	Type	Description
configuredCapacity	number	The agent's configured capacity (work that's assigned to the current user) through Presence Configuration.
currentWorkload	number	The agent's currently assigned workload.

## getServicePresenceStatusChannels for Lightning Experience

Retrieves the service channels that are associated with an Omni-Channel user's current presence status.

### Sample Code

Component code:

```

<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
    <lightning:omniToolkitAPI aura:id="omniToolkit" />
    <lightning:button label="Get Status Channels" onclick="{! c.getStatusChannels }" />
</aura:component>

```

Controller code:

```

({
    getStatusChannels: function(cmp, evt, hlp) {
        var omniAPI = cmp.find("omniToolkit");
        omniAPI.getServicePresenceStatusChannels().then(function(result) {
            var channels = JSON.parse(result.channels);
            //For example purposes, just retrieve the first channel
            console.log('First channel ID is: ' + channels[0].channelId);
            console.log('First channel developer name is: ' + channels[0].developerName);

        }).catch(function(error) {
            console.log(error);
        });
    }
})

```

## Response

This method returns a promise that, upon success, resolves to an array of `channel` objects, containing the following fields.

Name	Type	Description
<code>channelId</code>	String	The ID of the channel.
<code>developerName</code>	String	The name of the channel.

## `getServicePresenceStatusId` for Lightning Experience

Retrieves an agent's current presence status.

### Sample Code

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <lightning:button label="Get Status" onclick="{! c.getStatus }" />
</aura:component>
```

Controller code:

```
((
  getStatus: function(cmp, evt, hlp) {
    var omniAPI = cmp.find("omniToolkit");
    omniAPI.getServicePresenceStatusId().then(function(result) {
      console.log('Status Id is: ' + result.statusId);
    }).catch(function(error) {
      console.log(error);
    });
  }
})
```

## Response

This method returns a promise that, upon success, resolves to an object, containing the following fields.

Name	Type	Description
<code>statusName</code>	string	The name of the agent's current presence status.
<code>statusApiName</code>	string	The API name of the agent's current presence status.
<code>statusId</code>	string	The ID of the agent's current presence status.

## `login` for Lightning Experience

Logs an agent in to Omni-Channel with a specific presence status.

## Arguments

Name	Type	Description
statusId	string	The ID of the presence status. Agents must be given access to this presence status through their associated profile or permission set.

## Sample Code

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <lightning:button label="Login" onclick="{! c.login }" />
</aura:component>
```

Controller code:

```
((
  login: function(cmp, evt, hlp) {
    var omniAPI = cmp.find("omniToolkit");
    omniAPI.login({statusId: "0N5xx0000000001"}).then(function(result) {
      if (result) {
        console.log("Login successful");
      } else {
        console.log("Login failed");
      }
    }).catch(function(error) {
      console.log(error);
    });
  }
})
```

## Response

This method returns a promise that, upon success, resolves to `true` and is rejected on error.

## logout for Lightning Experience

Logs an agent out of Omni-Channel.

## Sample Code

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <lightning:button label="Logout" onclick="{! c.logout }" />
</aura:component>
```

Controller code:

```
({
  logout: function(cmp, evt, hlp) {
    var omniAPI = cmp.find("omniToolkit");
    omniAPI.logout().then(function(result) {
      if (result) {
        console.log("Logout successful");
      } else {
        console.log("Logout failed");
      }
    }).catch(function(error) {
      console.log(error);
    });
  }
})
```

## Response

This method returns a promise that, upon success, resolves to `true` and is rejected on error.

## setServicePresenceStatus for Lightning Experience

Sets an agent's presence status to a status with a particular ID. If the specified agent is not already logged in, we log in the agent with the presence status. This method removes the need for you to make more calls.

## Arguments

Name	Type	Description
statusId	string	The ID of the presence status to which you want to set the agent. Agents must be given access to this presence status through their associated profile or permission set.

## Sample Code

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <lightning:button label="Set Status" onclick="{! c.setStatus }" />
</aura:component>
```

Controller code:

```
({
  setStatus: function(cmp, evt, hlp) {
    var omniAPI = cmp.find("omniToolkit");
    omniAPI.setServicePresenceStatus({statusId: "0N5xx0000000002"}).then(function(result)
    {
      console.log('Current statusId is: ' + result.statusId);
      console.log('Channel list attached to this status is: ' + result.channels);
    });
  }
})
```



```

        }).catch(function(error) {
            console.log(error);
        });
    }
})

```

## Response

This method returns a promise that, upon success, resolves to an object containing the following fields.

Name	Type	Description
<code>statusName</code>	string	The name of the agent's current presence status.
<code>statusApiName</code>	string	The API name of the agent's current presence status.
<code>statusId</code>	string	The ID of the agent's current presence status.
<code>channels</code>	JSON string of <code>channel</code> objects	Returns the IDs and API names of the channels associated with the presence status.

## Events for Omni-Channel

JavaScript can be executed when certain types of events occur in a console, such as when a user closes a tab. There are a few events that are specific to Omni-Channel. These events apply to Lightning Experience only.

### [lightning:omniChannelLoginSuccess](#)

Indicates that an agent has been logged into Omni-Channel successfully.

### [lightning:omniChannelStatusChanged](#)

Indicates that an agent has changed his or her presence status in Omni-Channel.

### [lightning:omniChannelLogout](#)

Indicates that an agent has logged out of Salesforce.

### [lightning:omniChannelWorkAssigned](#)

Indicates that an agent has been assigned a new work item.

### [lightning:omniChannelWorkAccepted](#)

Indicates that an agent has accepted a work assignment, or that a work assignment has been automatically accepted.

### [lightning:omniChannelWorkDeclined](#)

Indicates that an agent has declined a work assignment.

### [lightning:omniChannelWorkClosed](#)

Indicates that an agent has closed a tab in the console that's associated with a work item. When the tab is closed, the status of the `AgentWork` object associated with it automatically changes to `Closed`.

### [lightning:omniChannelWorkloadChanged](#)

Indicates that an agent's workload has changed. This includes receiving new work items, declining work items, and closing items in the console. It also indicates that there has been a change to an agent's capacity or presence configuration, or that the agent has gone offline in the Omni-Channel utility.

## lightning:omniChannelLoginSuccess

Indicates that an agent has been logged into Omni-Channel successfully.

### Response

Name	Type	Description
statusId	string	The ID of the agent's current presence status.

 **Example:** This example prints a line to the browser's developer console when an Omni-Channel user logs into Omni-Channel successfully.

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <aura:handler event="lightning:omniChannelLoginSuccess" action="{! c.onLoginSuccess
  }"/>
</aura:component>
```

Controller code:

```
((
  onLoginSuccess : function(component, event, helper) {
    console.log("Login success.");
    var statusId = event.getParam('statusId');
    console.log(statusId);
  },
  ))
```

## lightning:omniChannelStatusChanged

Indicates that an agent has changed his or her presence status in Omni-Channel.

### Response

Name	Type	Description
statusId	string	The ID of the agent's current presence status.
channels	string	JSON string of channel objects.
statusName	string	The name of the agent's current presence status.
statusApiName	string	The API name of the agent's current presence status.

 **Example:** This example prints status details to the browser's developer console when an Omni-Channel user's presence status is changed.

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <aura:handler event="lightning:omniChannelStatusChanged" action="{! c.onStatusChanged
  }"/>
</aura:component>
```

Controller code:

```
((
  onStatusChanged : function(component, event, helper) {
    console.log("Status changed.");
    var statusId = event.getParam('statusId');
    var channels = event.getParam('channels');
    var statusName = event.getParam('statusName');
    var statusApiName = event.getParam('statusApiName');
    console.log(statusId);
    console.log(channels);
    console.log(statusName);
    console.log(statusApiName);
  },
  ))
```

## channel

The `channel` object contains the following properties:

Name	Type	Description
<code>channelId</code>	string	Retrieves the ID of the service channel that's associated with a presence status.
<code>developerName</code>	string	Retrieves the developer name of the service channel that's associated with the <code>channelId</code> .

## lightning:omniChannelLogout

Indicates that an agent has logged out of Salesforce.

### Response

None

 **Example:** This example prints a line to the browser's developer console when an Omni-Channel user logs out of Salesforce.

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
```

```
<aura:handler event="lightning:omniChannelLogout" action="{! c.onLogout }"/>
</aura:component>
```

Controller code:

```
((
  onLogout : function(component, event, helper) {
    console.log("Logout success.");
  },
}))
```

## lightning:omniChannelWorkAssigned

Indicates that an agent has been assigned a new work item.

### Response

Name	Type	Description
workItemId	string	The ID of the object that's routed through Omni-Channel. This object becomes a work assignment with a <code>workId</code> when it's assigned to an agent.
workId	string	The ID of a work assignment that's routed to an agent.



**Example:** This example prints work details to the browser's developer console when an Omni-Channel user is assigned a new work item.

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <aura:handler event="lightning:omniChannelWorkAssigned" action="{! c.onWorkAssigned
  }"/>
</aura:component>
```

Controller code:


```
((
  onWorkAssigned : function(component, event, helper) {
    console.log("Work assigned.");
    var workItemId = event.getParam('workItemId');
    var workId = event.getParam('workId');
    console.log(workItemId);
    console.log(workId);
  },
}))
```

## lightning:omniChannelWorkAccepted

Indicates that an agent has accepted a work assignment, or that a work assignment has been automatically accepted.

### Response

Name	Type	Description
workItemId	string	The ID of the object that's routed through Omni-Channel. This object becomes a work assignment with a <code>workId</code> when it's assigned to an agent.
workId	string	The ID of a work assignment that's routed to an agent.

 **Example:** This example prints work details to the browser's developer console when an Omni-Channel user accepts a work assignment, or when a work assignment is automatically accepted.

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <aura:handler event="lightning:omniChannelWorkAccepted" action="{! c.onWorkAccepted
  }"/>
</aura:component>
```

Controller code:

```
((
  onWorkAccepted : function(component, event, helper) {
    console.log("Work accepted.");
    var workItemId = event.getParam('workItemId');
    var workId = event.getParam('workId');
    console.log(workItemId);
    console.log(workId);
  },
  })
```


## lightning:omniChannelWorkDeclined

Indicates that an agent has declined a work assignment.

### Response

Name	Type	Description
workItemId	string	The ID of the object that's routed through Omni-Channel. This object becomes a work assignment with a <code>workId</code> when it's assigned to an agent.

Name	Type	Description
workId	string	The ID of a work assignment that's routed to an agent.

 **Example:** This example prints work details to the browser's developer console when an Omni-Channel user declines a work assignment.

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <aura:handler event="lightning:omniChannelWorkDeclined" action="{! c.onWorkDeclined
  }"/>
</aura:component>
```

Controller code:

```
{{
  onWorkDeclined : function(component, event, helper) {
    console.log("Work declined.");
    var workItemId = event.getParam('workItemId');
    var workId = event.getParam('workId');
    console.log(workItemId);
    console.log(workId);
  },
}}
```

## lightning:omniChannelWorkClosed

Indicates that an agent has closed a tab in the console that's associated with a work item. When the tab is closed, the status of the AgentWork object associated with it automatically changes to Closed.

### Response

Name	Type	Description
workItemId	string	The ID of the object that's routed through Omni-Channel. This object becomes a work assignment with a <code>workId</code> when it's assigned to an agent.
workId	string	The ID of a work assignment that's routed to an agent.

 **Example:** This example prints work details to the browser's developer console when an Omni-Channel user closes a tab in the console that's associated with a work item.

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <aura:handler event="lightning:omniChannelWorkClosed" action="{! c.onWorkClosed
  }"/>
</aura:component>
```

Controller code:

```
{{
  onWorkClosed : function(component, event, helper) {
    console.log("Work closed.");
    var workItemId = event.getParam('workItemId');
    var workId = event.getParam('workId');
    console.log(workItemId);
    console.log(workId);
  },
}}
```

## lightning:omniChannelWorkloadChanged

Indicates that an agent's workload has changed. This includes receiving new work items, declining work items, and closing items in the console. It also indicates that there has been a change to an agent's capacity or presence configuration, or that the agent has gone offline in the Omni-Channel utility.

### Response

Name	Type	Description
configuredCapacity	number	The configured capacity for the agent.
previousWorkload	number	The agent's workload before the change.
newWorkload	number	The agent's new workload after the change.

 **Example:** This example prints workload details to the browser's developer console when an agent's workload changes.

Component code:

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global" >
  <lightning:omniToolkitAPI aura:id="omniToolkit" />
  <aura:handler event="lightning:omniChannelWorkloadChanged" action="{!
  c.onWorkloadChanged }"/>
</aura:component>
```

Controller code:

```
{{
  onWorkloadChanged : function(component, event, helper) {
    console.log("Workload changed.");
  }
}}
```

```

    var configuredCapacity = event.getParam('configuredCapacity');
    var previousWorkload = event.getParam('previousWorkload');
    var newWorkload = event.getParam('newWorkload');
    console.log(configuredCapacity);
    console.log(previousWorkload);
    console.log(newWorkload);
  },
})

```

## External Routing for Omni-Channel

Multiple routing options, one console. Integrate third-party routing with Omni-Channel to give your support team more routing options for their work.

Before setting up external routing, make sure that you have a working implementation of Omni-Channel. You'll use version 41.0 or later of Salesforce standard APIs and streaming APIs to connect Salesforce with an external routing implementation in your routing configuration. Then you can create queues that use either Omni-Channel routing or your external routing implementation.

Are you ready to set up and use external routing? Let's get started.

### [External Routing Technical Architecture and Process](#)

See an overview of how external routing works to connect Salesforce with your external routing implementation.

### [Expected Behavior for External Routing for Omni-Channel](#)

Verify that the behavior you observe while testing and using your implementation of external routing matches the following expected behavior scenarios.

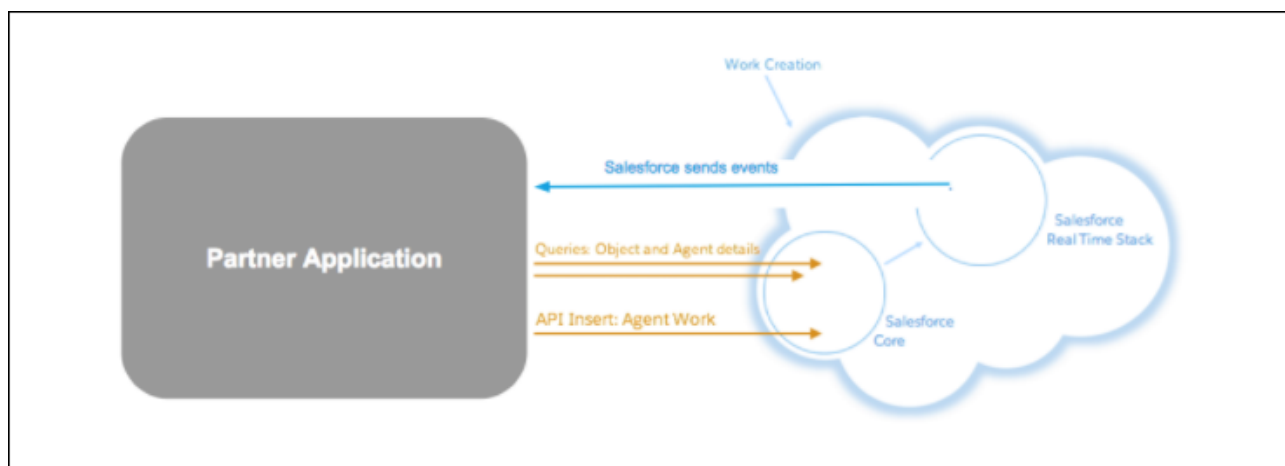
### [Troubleshooting External Routing for Omni-Channel](#)

If you encounter issues with your implementation of External Routing for Omni-Channel, try the following troubleshooting steps.

## External Routing Technical Architecture and Process

See an overview of how external routing works to connect Salesforce with your external routing implementation.

The following image provides a visual description of how information is shared between Salesforce and your partner application.





Information is shared using Salesforce APIs and the APIs for your partner application using the following process.

1. Salesforce sends events using Salesforce Streaming APIs for the `PendingServiceRouting` object.
2. Partner application creates a session to authenticate.
3. Partner application queries Salesforce for work details, agent availability, and more.
4. Partner application determines the routing decision.
5. AgentWork is created and pushed to the specified agent in Salesforce.

#### [Salesforce API Resources for External Routing](#)

Use the following resources as you integrate your partner application with Omni-Channel.

## Salesforce API Resources for External Routing

Use the following resources as you integrate your partner application with Omni-Channel.

- [AgentWork](#)
- [PendingServiceRouting](#) on page 12
- [UserServicePresence](#)
- [Streaming API Developer Guide](#)
- [Streaming API Message Durability](#)
- [Streaming API Limits](#)

## Integrate External Routing for Omni-Channel

Use the following steps to integrate your external routing implementation with Omni-Channel.

- [Step 1: Create a Routing Configuration and Queue for External Routing](#)
- [Step 2: Create a PushTopic for PendingServiceRouting](#)
- [Step 3: Listen to the PushTopic](#)
- [Step 4: Create AgentWork](#)

### Step 1: Create a Routing Configuration and Queue for External Routing

External routing requires a separate routing configuration and queue in Omni-Channel. These separate objects define routing behavior and assign work to agents.

1. In Setup, enter *Routing Configurations* in the Quick Find box, then select **Routing Configurations**.
2. Create a routing configuration and select **External Routing** for the routing model.
3. Enter *Queues* in the Quick Find box, then select **Queues**.
4. Create a queue and connect it to the routing configuration you created.

### Step 2: Create a PushTopic for PendingServiceRouting

A PushTopic is a query that is the basis for notifying listeners of changes to records in an organization. Create a PushTopic for `PendingServiceRouting` so partners can receive event notifications.

Reference the following code sample to create a PushTopic using Apex code. If you need help creating a PushTopic, see [Create a PushTopic](#) in the *Streaming API Developer Guide*.

PushTopic events use the following database values to indicate work status:

- ASSIGNED (0, "Assigned")
- UNAVAILABLE (1, "Unavailable")
- DECLINED (2, "Declined")
- OPENED (3, "Opened")
- CLOSED (4, "Closed")
- DECLINED\_ON\_PUSH\_TIMEOUT (5, "DeclinedOnPushTimeout")
- CANCELLED (6, "Canceled")
- TRANSFERRED (7, "Transferred")

```
PushTopic pushTopic = new PushTopic();
pushTopic.Name = PSRPushTopic;
pushTopic.Query = 'Select Id, Serial, QueueId, WorkItemId, IsPushed, ServiceChannelId,
LastDeclinedAgentSession, CreatedDate from PendingServiceRouting where RoutingModel =
'ExternalRouting'';
pushTopic.ApiVersion = 39.0;
pushTopic.NotifyForOperationCreate = true;
pushTopic.NotifyForOperationUpdate = true;
pushTopic.NotifyForOperationDelete = true;
pushTopic.NotifyForFields = 'Referenced';
insert pushTopic;
```

### Step 3: Listen to the PushTopic

Give your event notifications somewhere to go by setting up a listener.

Salesforce's Streaming API uses the HTTP/1.1 request-response model and the Bayeux protocol (CometD implementation). The easiest way to connect to the Streaming API is to use [java sdk](#) and OAuth flow to connect to the PushTopic you created.

For reference and a code sample, see [Use the Connector with OAuth Bearer Token Login](#) from the *Streaming API Developer Guide*.

### Step 4: Create AgentWork

Your partner application must create an AgentWork record to route the work to an agent in Omni-Channel.

When the external routing implementation receives new PendingServiceRouting creation events (where the routing type equals *External*), it uses the standard Salesforce SOAP API to fetch further information based on the PendingServiceRouting. It makes a routing decision by creating AgentWork records. This API is existing functionality that partners use to query Salesforce.

Reference the following code sample to create AgentWork using Apex.

```
AgentWork work = new AgentWork();
work.ServiceChannelId = '<ServiceChannelId>';
work.WorkItemId = '<WorkItemId>';
work.UserId = '<UserId>';
work.PendingServiceRoutingId = '<PendingServiceRoutingId>';
insert work;
```

## Expected Behavior for External Routing for Omni-Channel

Verify that the behavior you observe while testing and using your implementation of external routing matches the following expected behavior scenarios.

### Agent accepts the work:

1. Chat visitor initiates a chat request from the external routing button.
2. `PendingServiceRouting` is created.
3. Partner is notified by a `pushTopic` event (`EventType=Create`, `isPushed=false`).
4. Partner creates `AgentWork` using the PSR.
5. Agent is routed the chat request (`AgentWork Status = Assigned`).
6. Agent accepts the chat request (`AgentWork Status = Accept`).
7. Omni-Channel deletes the `PendingServiceRouting` after Agent accepts the work.
8. Partner is notified by a `pushTopic` event (`EventType=Delete`).

### Agent declines the work through Omni-Channel:

1. Agent declines the assigned `AgentWork`.
2. Salesforce updates the `PendingServiceRouting`.
3. Partner is notified by a `pushTopic` event (`EventType=Update`, `LastDeclinedAgentSession=agent's session id in LiveAgent` (not the Salesforce session), `isPushed=false`).
4. Partner creates a new `AgentWork` using the updated `PendingServiceRouting` for rerouting.

### Agent doesn't accept the work due to push time-out:

1. Existing `PendingServiceRouting` is updated.
2. Partner is notified by a `pushTopic` event (`EventType=Update`, PSR Fields updated: `isPushed=false`, `LastDeclinedAgentSession=agent's liveagent session id`).
3. Partner creates a new `AgentWork` for rerouting.

### Agent transfers the work to an external routing queue:

1. New `PendingServiceRouting` for the transfer is created.
2. Partner is notified by a `pushTopic` event (`EventType=Create`, `isTransfer=true`, `isPushed=false`).
3. The routing process is repeated.

### Agent transfers the work to another agent :

1. The `PendingServiceRouting` from the original chat request is deleted.
2. A new `PendingServiceRouting` isn't created when the work is transferred. Subscribe to `AgentWork` and `LiveChatTranscript` to determine whether the work was transferred to an agent.
3. Two `AgentWorks` are created for the `LiveChatTranscript`:
  - a. First `AgentWork` with the `Status = Opened`
  - b. Second `AgentWork` with the `Status = Assigned`
4. The `LiveChatTranscript` is updated with the `Status = In Progress` and the `Owner = second Agent`.
5. To determine if the Transfer to Agent has occurred, check that the second `AgentWork` isn't inserted into the same `LiveChatTranscript` as the first `AgentWork`.

**!** **Important:** We don't support using both external routing and Omni-Channel queue-based routing in the same implementation. Transfer scenarios between external routing queues and Omni-Channel queues also aren't supported. If you attempt this combination, there can be unknown issues.

## Troubleshooting External Routing for Omni-Channel

If you encounter issues with your implementation of External Routing for Omni-Channel, try the following troubleshooting steps.

### Recover from an External Routing Adaptor Restart

When the third-party adaptor recovers from restarting, it should leverage the durability feature of the Streaming API (since version 37.0) and replay from the last successful position of the PSR topic.

Reference the following code sample in Java.

```
// Register streaming extension
var replayExtension = new cometdReplayExtension();
replayExtension.setChannel(***<Streaming Channel to Subscribe to>***);
replayExtension.setReplay(<Event Replay Option>);
cometd.registerExtension('myReplayExtensionName', replayExtension);
```

For more information, see [Message Durability](#) in the *Streaming API Developer Guide*.

### Recover from a Salesforce Data Recovery Instance

An org instance can be recovered from a Salesforce data center switch. The recovery process involves downtime, so all online agents must be logged out. All states maintained by the third-party adaptor, such as agent presence, aren't applicable and must be reset. The third-party adaptor should reinitialize as when it first subscribed to the topic.

### Test the Client Solution

You can use the Streaming API to listen to CRUD events for `UserServicePresence` and `PendingServiceRouting`. For examples, see [Code Examples](#) in the *Streaming API Developer Guide*.

# INDEX

## A

acceptAgentWork() 30, 45  
AgentWork object 3

## C

closeAgentWork() 31, 46

## D

declineAgentWork() 32, 46

## E

events 41

Events

- lightning:omniChannelLoginSuccess 54
- lightning:omniChannelLogout 55
- lightning:omniChannelStatusChanged 54
- lightning:omniChannelWorkAccepted 57
- lightning:omniChannelWorkAssigned 56
- lightning:omniChannelWorkClosed 58
- lightning:omniChannelWorkDeclined 57
- lightning:omniChannelWorkloadChanged 59

## G

getAgentWorks() 33–34, 47–48  
getServicePresenceStatusChannels() 36, 49  
getServicePresenceStatusId() 37, 50

## L

lightning:omniChannelLoginSuccess 54  
lightning:omniChannelLogout 55  
lightning:omniChannelStatusChanged 54  
lightning:omniChannelWorkAccepted 57  
lightning:omniChannelWorkAssigned 56  
lightning:omniChannelWorkClosed 58  
lightning:omniChannelWorkDeclined 57  
lightning:omniChannelWorkloadChanged 59  
login() 38, 50  
logout() 39, 51

## M

Methods

- acceptAgentWork() 30, 45
- closeAgentWork() 31, 46
- console events 41
- declineAgentWork() 32, 46

Methods (*continued*)

- events 41
- getAgentWorks() 33–34, 47–48
- getServicePresenceStatusChannels() 36, 49
- getServicePresenceStatusId() 37, 50
- login() 38, 50
- logout() 39, 51
- Omni-Channel 53
- setServicePresenceStatus() 40, 52

## O

Objects

- AgentWork 3
- OmniSupervisorConfig 9
- OmniSupervisorConfigGroup 10
- OmniSupervisorConfigProfile 11
- OmniSupervisorConfigUser 11
- PendingServiceRouting 12
- PresenceConfigDeclineReason 15
- PresenceDeclineReason 16
- PresenceUserConfig 17
- PresenceUserConfigProfile 20
- PresenceUserConfigUser 20
- QueueRoutingConfig 21
- ServiceChannel 23
- ServiceChannelStatus 25
- ServicePresenceStatus 26
- UserServicePresence 27

Omni-Channel

- Salesforce Console Integration 28–29, 44
- SOAP API objects 2

OmniSupervisorConfig object 9

OmniSupervisorConfigGroup object 10

OmniSupervisorConfigProfile object 11

OmniSupervisorConfigUser object 11

## P

PendingServiceRouting object 12

PresenceConfigDeclineReason object 15

PresenceDeclineReason object 16

PresenceUserConfig object 17

PresenceUserConfigProfile object 20

## Q

QueueRoutingConfig object 21

## Index

### S

ServiceChannel object [23](#)  
ServiceChannelStatus object [25](#)  
ServicePresenceStatus object [26](#)

setServicePresenceStatus() [40](#), [52](#)

### U

UserServicePresence object [27](#)