Set Up Omni-Channel Developer Guide

Version 56.0, Spring ’23

@salesforcedocs

Last updated: April 27, 2023

CONTENTS

CHAPTER 1 Omni-Channel Developer Guide

In this chapter ...

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

Set Up Omni-Channel Objects

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

For more information on Salesforce APIs, see Which API Do I Use? in *Salesforce Help*.

AgentWork

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

AgentWorkSkill

Represents a skill used to route a work assignment to an agent. AgentWorkSkill is used for reporting and represents the result of a routing decision. This object is available in API version 42.0 and later.

OmniSupervisorConfig

Represents the Omni-Channel supervisor configuration for an assigned group of supervisors. This object is available in API version 41.0 and later.

OmniSupervisorConfigAction

Represents the actions available to the supervisors of an Omni-Channel supervisor configuration. This object is available in API version 56.0 and later.

OmniSupervisorConfigGroup

Represents the group of agents who are visible to the supervisors of an Omni-Channel supervisor configuration. The group, if visible, appears in the Agents tab of Omni Supervisor. This object is available in API version 41.0 and later.

OmniSupervisorConfigProfile

Represents the supervisor profiles to which an Omni-Channel supervisor configuration applies. User-level configurations override profile-level configurations. This object is available in API version 41.0 and later.

OmniSupervisorConfigQueue

Represents the queues that are visible to the supervisors of an Omni-Channel supervisor configuration. The queue, if visible, appears in the Queues Backlog and Assigned Work tabs of Omni Supervisor. This object is available in API version 53.0 and later.

OmniSupervisorConfigSkill

Represents the skills that are visible to the supervisors of an Omni-Channel supervisor configuration. These skills, if visible, appear in the Skills Backlog tab of Omni Supervisor. This object is available in API version 53.0 and later.

OmniSupervisorConfigUser

Represents the users to which an Omni-Channel supervisor configuration applies. User-level configurations override profile-level configurations. This object is available in API version 41.0 and later.

PendingServiceRouting

Represents the routing details of a work item that's waiting to be routed or assigned. 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.

QueueSobject

Represents the mapping between a queue Group and the sObject types associated with the queue, including custom objects.

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.

ServiceChannelFieldPriority

Represents a secondary routing priority field-value mapping. This object is available in API version 47.0 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.

ServiceResource

Represents a service technician or service crew in Field Service and Salesforce Scheduler, or an agent in Workforce Engagement. This object is available in API version 38.0 and later.

SkillRequirement

Represents a skill that is required to complete a particular task in Field Service, Omni-Channel, Salesforce Scheduler, or Workforce Engagement. Skill requirements can be added to pending service routing objects in Omni-Channel. They can be added to work types, work orders, and work order line items in Field Service and Lightning Scheduler. And they can be added to job profiles in Workforce Engagement. This object is available in API version 38.0 and later. You also can add skill requirements to work items in Omni-Channel skills-based routing using API version 42.0 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()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

To access this object, Omni-Channel must be enabled.

Fields

Field	Details
`AcceptDateTime`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** Indicates when the work item was accepted.
`ActiveTime`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** 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. If After Conversation Work is in use, `ActiveTime` ends when the `AfterConversationActualTime` period ends or the agent closes the work item, whichever occurs first. Note: `ActiveTime` is tracked only for work that is routed using the tab-based capacity model.
`AcwExtensionCount`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The number of times that an agent extended the After Conversation Work (ACW) timer. This field is available in API version 55.0 and later.
`AcwExtensionDuration`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The length of time (in seconds) that the After Conversation Work (ACW) timer was extended each time that the agent extended the timer. This field is available in API version 55.0 and later.

Field	Details
	To find the total extension duration, multiply this field by `AcwExtensionCount` or use `AfterConversationActualTime`.
`AfterConversationActualTime`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The number of seconds an agent spent on After Conversation Work (ACW) after customer contact ended. This field is available in API version 52.0 and later.
`AgentCapacityWhenDeclined`	**Type** double **Properties** Filter, Nillable, Sort **Description** The agent's capacity when declining work, either explicitly or through push timeout.
`AssignedDateTime`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** Indicates when the work item was assigned to an agent. This field is a calculated field.
`BotId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** The ID of the Einstein Bot that performed the work. This field only applies to Enhanced Bots. This is a relationship field. This field is available in API version 52.0 and later. **Relationship Name** Bot **Relationship Type** Lookup **Refers To** BotDefinition
`CancelDateTime`	**Type** dateTime

Field	Details
	Properties Filter, Nillable, Sort **Description** Indicates when the work item was canceled.
`CapacityModel`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** Indicates the capacity model used to determine agent capacity. Valid values are `StatusBased` and `TabBased`. This field is available in API version 50.0 and later. Note: A work item consumes agent capacity only if it was first assigned to the agent by Omni-Channel using queues or skills.
`CapacityPercentage`	**Type** percent **Properties** Create, Filter, Nillable, Sort **Description** 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 doesn't receive new work items until there's 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.
`CapacityWeight`	**Type** double **Properties** Create, Filter, Nillable, Sort **Description** 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.
`CloseDateTime`	**Type** dateTime **Properties** Filter, Nillable, Sort

Field	Details
	Description Indicates when the work item was closed.
`DeclineDateTime`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** Date and time when the agent declined this record.
`DeclineReason`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The provided reason for why an agent declined the work request.
`HandleTime`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The amount of time an agent had the work item open. Calculated by `CloseDateTime – AcceptedDateTime`. If After Conversation Work is in use, `HandleTime` ends when the `AfterConversationActualTime` period ends or the agent closes the work item, whichever occurs first.
`IsOwnerChangeInitiated`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether a work item owner change triggered the direct assignment of the work item to the agent. The default value is `false`. Status-Based Capacity Model has to be turned on to use this field. This field is available in API version 50.0 and later.
`IsPreferredUserRequired`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort

Field	Details
	Description Indicates whether a work item stays with the preferred user even when the user isn't available. The default value is false. This field is available in API version 50.0 and later.
`IsStatusChangeInitiated`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether a work item status change triggered the direct assignment of the work item to the agent. The default value is false. Status-Based Capacity Model has to be turned on to use this field. This field is available in API version 50.0 and later.
`Name`	**Type** string **Properties** Autonumber, Defaulted on create, Filter, idLookup, Sort **Description** An automatically generated ID number that identifies the record.
`OriginalGroupId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the queue that the work assignment was originally routed to. This field is a relationship field. **Relationship Name** OriginalGroup **Relationship Type** Lookup **Refers To** Group
`OriginalQueueId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the queue that the work assignment was originally routed to. Due to API changes, `OriginalQueueId` is no longer recommended. Use `OriginalGroupId` instead.

Field	Details
`OwnerId`	**Type** reference **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** The ID of the owner of the AgentWork. This field is a polymorphic relationship field. This field is available in API version 50.0 and later. **Relationship Name** Owner **Relationship Type** Lookup **Refers To** Group, User
`PendingServiceRoutingId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** The ID of the PendingServiceRouting on page 25 from which the AgentWork was created. This field is a relationship field. This field is available in API version 50.0 and later. **Relationship Name** PendingServiceRouting **Relationship Type** Lookup **Refers To** PendingServiceRouting
`PreferredUserId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the preferred user to handle the work. This field is a relationship field. This field is available in API v46.0 and later. **Relationship Name** PreferredUser **Relationship Type** Lookup **Refers To** User

Field	Details
`PushTimeout`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The time limit set for an agent to respond to an item before it's pushed to another agent. The time limit is measured in seconds. This field is available in API version 36.0 and later. Effective API version 57.0, for inbound Voice calls, this field represents the time limit set for an agent to respond to a call before it's declined. The value must be between 0 and 20. The value is capped at 20, so any number greater than that is treated as 20 seconds. This applies to the following telephony models: • Service Cloud Voice with Amazon Connect • Service Cloud Voice with Partner Telephony from Amazon Connect
`PushTimeoutDateTime`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date and time (in UTC) when the push timeout event occurred. This field is available in API version 36.0 and later.
`RequestDateTime`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** Indicates when the work was requested.
`RoutingModel`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** Determines how incoming work items are routed to agents assigned to a service channel. Possible values are: • `ExternalRouting` • `LeastActive` • `MostAvailable`

Field	Details
`RoutingPriority`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The order in which work items from the queue that are associated with the routing configuration are routed to agents.
`RoutingType`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** The type of Omni-Channel routing. Possible values are: • `QueueBased` • `SkillsBased`
`SecondaryRoutingPriority`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** Indicates the secondary routing priority.
`ServiceChannelId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The ID of the service channel that's associated with the work assignment. This field is a relationship field. **Relationship Name** ServiceChannel **Relationship Type** Lookup **Refers To** ServiceChannel
`ShouldSkipCapacityCheck`	**Type** boolean

Field	Details
	Properties Create, Defaulted on create, Filter, Group, Sort **Description** Indicates whether to skip checking an agent's available capacity (`true`) or not (`false`) 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. When `true`, the receiving agent can exceed their set capacity to accept the item, but they don't receive more Omni-Channel routed work. When `false`, the receiving agent can't exceed their set capacity and must have enough open capacity to accept the item. The default value is `false`.
`SpeedToAnswer`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The amount of time between when the work was requested and when an agent accepted it.
`Status`	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The working status of the work item. Valid values are: • `Assigned` – The item is assigned to the agent but hasn't been opened. • `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. • `Closed` – The item is closed. • `Declined` – The item was assigned to the agent but the agent explicitly declined it. • `DeclinedOnPushTimeout` – The item was declined because push time-out is enabled and the item request timed out with the agent. • `Opened` – The agent opened the item. • `Transferred`–The item was transferred from an agent to another agent, queue, or skill. • `Unavailable` – The item was assigned to the agent but the agent became unavailable (went offline or lost connection).
`UserId`	**Type** reference

Field	Details
	Properties Create, Filter, Group, Sort **Description** The ID of the user that the work item was assigned to. This field is a relationship field. **Relationship Name** User **Relationship Type** Lookup **Refers To** User
`WorkItemId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The ID of the object that's routed to the agent through Omni-Channel. This field is a polymorphic relationship field. **Relationship Name** WorkItem **Relationship Type** Lookup **Refers To** Custom objects and these standard objects: Case, Account, Lead,Contact, Activity, Opportunity, CustomEntityData, SocialPost, Order, ContactRequest, LiveChatTranscript, MessagingSession, VoiceCall, PersonTraining, SwarmMember, Incident, Claim, ClaimRecovery, ClaimCoverage, PaymentRequest, Referral

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.

When `AgentWork` records are created, they have the status Assigned. After a record is created, it's 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`.

Associated Objects

This object has the following associated objects. Unless noted, they're available in the same API version as this object.

AgentWorkOwnerSharingRule
Sharing rules are available for the object.

AgentWorkShare
Sharing is available for the object.

AgentWorkSkill

Represents a skill used to route a work assignment to an agent. AgentWorkSkill is used for reporting and represents the result of a routing decision. This object is available in API version 42.0 and later.

Supported Calls

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

Special Access Rules

To access this object, Omni-Channel must be enabled.

Fields

Field	Details
`AgentWorkId`	**Type** reference **Properties** Filter, Group, Sort **Description** The AgentWork object associated with this skill.
`IsAdditionalSkill`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** After a designated timeout period, a skill marked as additional is dropped from Omni-Channel routing. The case is then routed to the best-matched agent, even if the agent doesn't have all the skills. The default value is false. Available in API version 48.0 and later.
`Name`	**Type** string **Properties** Autonumber, Defaulted on create, Filter, idLookup, Sort **Description** An automatically generated ID number that identifies the record.

Field	Details
`SkillId`	**Type** reference **Properties** Filter, Group, Sort **Description** The skill that is required or additional.
`SkillLevel`	**Type** double **Properties** Filter, Sort **Description** The level of the required or additional skill. Skill levels can range from 1 to 10. Depending on your business needs, you might want the skill level to reflect years of experience, certification levels, or license classes.
`SkillPriority`	**Type** int **Properties** Aggregatable, Filter, Group, Nillable, Sort **Description** For additional skills, specifies the order in which skills are dropped if after the specified timeout no agent with that skill is available. Higher priority-value skills are dropped first. Lower priority-value skills, for example 0, are dropped last. Skills with the same priority value are dropped as a group. You can set skill priority using attribute setup for skills-based routing or Apex code.
`WasDropped`	**Type** boolean **Properties** Filter, Group, Sort **Description** For skills marked as additional, indicates if the skill was dropped from Omni-Channel routing because an agent with this skill was not available. The default value is false. Available in API version 48.0 and later.

OmniSupervisorConfig

Represents the Omni-Channel supervisor configuration for an assigned group of supervisors. This object is available in API version 41.0 and later.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. Because changing terms in our code can break current implementations, we maintained this object's name.

Supported Calls

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

Special Access Rules

To access this object, Omni-Channel must be enabled.

As of Spring '20 and later, only authenticated internal and external users can access this object.

Fields

Field	Details
`DeveloperName`	**Type** string **Properties** Create, Filter, Group, Sort **Description** 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. Note: When creating large sets of data, always specify a unique `DeveloperName` for each record. If no `DeveloperName` is specified, performance slows down while Salesforce generates one for each record.
`IsTimelineHidden`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** If set to `true`, hides the agent timeline from the supervisors assigned to this supervisor configuration. The default value is `false`. This field is available in API version 53.0 and later.
`Language`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort **Description** The language of this supervisor configuration.

Field	Details
`MasterLabel`	**Type** string **Properties** Create, Filter, Group, Sort **Description** A unique label name for this supervisor configuration. The name must begin with a letter. The name can contain alphanumeric characters and underscores. The name can't contain spaces, two consecutive underscores, or end with an underscore. The name appears as Omni Supervisor Configuration Name in the UI.
`SkillVisibility`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Determines which work items based on skills are visible to the supervisors assigned to this supervisor configuration. Possible values are: • `AllSkills` — Show work items with all skill requirements selected in this supervisor configuration. • `AnySkill` — Show work items with at least one skill requirement selected in this supervisor configuration. This field is available in API version 53.0 and later.

OmniSupervisorConfigAction

Represents the actions available to the supervisors of an Omni-Channel supervisor configuration. This object is available in API version 56.0 and later.

Supported Calls

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

Special Access Rules

To access this object, Omni-Channel must be enabled.

Only authenticated internal and external users can access this object.

Fields

Field	Details
`DisplayOrder`	**Type** int **Properties** Create, Filter, Group, Nillable, Sort **Description** The order in which the action is displayed.
`OmniSupervisorActionType`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort **Description** An action that a supervisor can perform. Possible values are: • `AgentDetails.CustomAction` • `AllAgents.AWSDashboard`—All Agents - View Amazon Real-Time Metrics • `AllAgents.AssignLearning` • `AllAgents.ChangeQueues` • `AllAgents.ChangeSkills` • `AllAgents.CustomAction` • `AssignedWork.AWSDashboard`—Assigned Work - View Amazon Real-Time Metrics • `AssignedWork.CustomAction` • `AssignedWorkDetails.CustomAction` • `QueueDetails.CustomAction` • `QueuesBacklog.AWSDashboard`—Queues Backlog - View Amazon Real-Time Metrics • `QueuesBacklog.CustomAction` • `QueuesBacklog.ManageQueues`—Queues Backlog - Assign Agents to Queues • `SkillDetails.CustomAction` • `SkillsBacklog.AWSDashboard`—Skills Backlog - View Amazon Real-Time Metrics • `SkillsBacklog.CustomAction`
`OmniSupervisorConfigId`	**Type** reference **Properties** Create, Filter, Group, Sort

Field	Details
	Description A unique identifier for the Omni-Channel supervisor configuration. This field is a relationship field. **Relationship Name** OmniSupervisorConfig **Relationship Type** Lookup **Refers To** OmniSupervisorConfig

OmniSupervisorConfigGroup

Represents the group of agents who are visible to the supervisors of an Omni-Channel supervisor configuration. The group, if visible, appears in the Agents tab of Omni Supervisor. This object is available in API version 41.0 and later.

Supported Calls

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

Special Access Rules

To access this object, Omni-Channel must be enabled.

As of Spring '20 and later, only authenticated internal and external users can access this object.

Fields

Field	Details
`GroupId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** A unique identifier for the group of agents that's made visible to the supervisors who are assigned to the Omni-Channel supervisor configuration. This is a relationship field. **Relationship Name** Group **Relationship Type** Lookup

Field	Details
	Refers To Group
`OmniSupervisorConfigId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** A unique identifier for the Omni-Channel supervisor configuration. This is a relationship field. **Relationship Name** OmniSupervisorConfig **Relationship Type** Lookup **Refers To** OmniSupervisorConfig

OmniSupervisorConfigProfile

Represents the supervisor profiles to which an Omni-Channel supervisor configuration applies. User-level configurations override profile-level configurations. This object is available in API version 41.0 and later.

Supported Calls

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

Special Access Rules

To access this object, Omni-Channel must be enabled.

As of Spring '20 and later, only authenticated internal and external users can access this object.

Fields

Field	Details
`OmniSupervisorConfigId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** A unique identifier for the Omni-Channel supervisor configuration.

Field	Details
	This is a relationship field. **Relationship Name** OmniSupervisorConfig **Relationship Type** Lookup **Refers To** OmniSupervisorConfig
`ProfileId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** A unique identifier for the profile that's associated with this Omni-Channel supervisor configuration. A profile can be associated with only one Omni-Channel supervisor configuration. This field is unique within your org. This is a relationship field. **Relationship Name** Profile **Relationship Type** Lookup **Refers To** Profile

OmniSupervisorConfigQueue

Represents the queues that are visible to the supervisors of an Omni-Channel supervisor configuration. The queue, if visible, appears in the Queues Backlog and Assigned Work tabs of Omni Supervisor. This object is available in API version 53.0 and later.

Supported Calls

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

Special Access Rules

To access this object, Omni-Channel must be enabled.

Only authenticated internal and external users can access this object.

Fields

Field	Details
`OmniSupervisorConfigId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** om A unique identifier for the Omni-Channel supervisor configuration. This is a relationship field. **Relationship Name** OmniSupervisorConfig **Relationship Type** Lookup **Refers To** OmniSupervisorConfig
`QueueId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** A unique identifier for the queue that's made visible to the supervisors who are assigned to the Omni-Channel supervisor configuration. This is a relationship field. **Relationship Name** Queue **Relationship Type** Lookup **Refers To** Group

OmniSupervisorConfigSkill

Represents the skills that are visible to the supervisors of an Omni-Channel supervisor configuration. These skills, if visible, appear in the Skills Backlog tab of Omni Supervisor. This object is available in API version 53.0 and later.

Supported Calls

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

Special Access Rules

To access this object, Omni-Channel must be enabled.

Only authenticated internal and external users can access this object.

Fields

Field	Details
`OmniSupervisorConfigId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** A unique identifier for the Omni-Channel supervisor configuration. This is a relationship field. **Relationship Name** OmniSupervisorConfig **Relationship Type** Lookup **Refers To** OmniSupervisorConfig
`SkillId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** A unique identifier for the skill that's made visible to the supervisors who are assigned to the Omni-Channel supervisor configuration. This is a relationship field. **Relationship Name** Skill **Relationship Type** Lookup **Refers To** Skill

OmniSupervisorConfigUser

Represents the users to which an Omni-Channel supervisor configuration applies. User-level configurations override profile-level configurations. This object is available in API version 41.0 and later.

Supported Calls

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

Special Access Rules

To access this object, Omni-Channel must be enabled.

As of Spring '20 and later, only authenticated internal and external users can access this object.

Fields

Field	Details
`OmniSupervisorConfigId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** A unique identifier for the Omni-Channel supervisor configuration. This is a relationship field. **Relationship Name** OmniSupervisorConfig **Relationship Type** Lookup **Refers To** OmniSupervisorConfig
`UserId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** A unique identifier for the user associated with this Omni-Channel supervisor configuration. A user can be associated with only one Omni-Channel supervisor configuration. This field is unique within your org. This is a relationship field. **Relationship Name** User **Relationship Type** Lookup **Refers To** User

PendingServiceRouting

Represents the routing details of a work item that's waiting to be routed or assigned. This object is available in API version 40.0 and later.

Supported Calls

`create()`, `delete()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

To access this object, Omni-Channel must be enabled.

Fields

Field	Details
`BotId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** Bot ID the work item is assigned to. You can only use enhanced bots. This field is available in API version 55.0 and later. This field is a relationship field. **Relationship Name** Bot **Relationship Type** Lookup **Refers To** BotDefinition
`CapacityPercentage`	**Type** percent **Properties** Create, Filter, Nillable, Sort, Update **Description** Indicates the amount of work that this work item represents as a percentage. Valid values are from 0 to 100.
`CapacityWeight`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update

Field	Details
	Description Indicates the amount of work that this work item represents as a whole number.
`CustomRequestedDateTime`	**Type** dateTime **Properties** Create, Filter, Nillable, Sort, Update **Description** Retains the datetime of this 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 are rerouted.
`DropAdditionalSkillsTimeout`	**Type** int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Time to wait before a skill marked as additional is dropped from Omni-Channel routing. The case is then routed to the best-matched agent even if they don't have all the skills.
`GroupId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** ID of the Omni-Channel queue.
`IsOwnerChangeInitiated`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether a work item owner change triggered the direct assignment of this work item to the agent. The default value is `false`.
`IsPreferredUserRequired`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort

Field	Details
	Description Indicates whether this work item stays with the preferred user even when the user is not available. The default value is `false`. This field is available in API version 50.0 and later.
`IsPushAttempted`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether a push has been attempted. `true` if this work item was pushed to an agent at least one time and `false` otherwise.
`IsPushed`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether this work item is pushed to an agent.
`IsReadyForRouting`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** Indicates whether this work item is ready to be routed to an agent. If `true`, you can't edit this PendingServiceRouting record.
`IsStatusChangeInitiated`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether a work item status change triggered the direct assignment of this work item to the agent. The default value is `false`. This field is available in API version 50.0 and later.
`IsTransfer`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether this work item routing is a transfer request.

Field	Details
`LastDeclinedAgentSession`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Chat session ID of the agent who last declined this work item.
`Name`	**Type** string **Properties** Autonumber, Defaulted on create, Filter, idLookup, Sort **Description** Name of the PendingServiceRouting record.
`OwnerId`	**Type** reference **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** ID of the owner of this PendingServiceRouting record.
`PreferredUserId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** ID of the preferred user to handle the work item.
`PushTimeout`	**Type** int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The time limit set for an agent to respond to an item before it's pushed to another agent. The time limit is measured in seconds. This field is available in API version 36.0 and later. Effective API version 57.0, for inbound Voice calls, this field represents the time limit set for an agent to respond to a call before it's declined. The value must be between 0 and 20. The value is capped at 20, so any number greater than that is treated as 20 seconds. This applies to the following telephony models: • Service Cloud Voice with Amazon Connect • Service Cloud Voice with Partner Telephony from Amazon Connect

Field	Details
`QueueId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** ID of the Omni-Channel queue. Due to API changes, `QueueId` is no longer recommended. Use `GroupId` instead.
`RoutingModel`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Type of routing model. Possible values are: • `ExternalRouting` • `LeastActive` • `MostAvailable`
`RoutingPriority`	**Type** int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Order in which work items are routed to agents. This field is considered with skills-based routing only. Queue-based routing sets a work item's priority from the routing configuration.
`RoutingType`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort **Description** Indicates whether the work item is routed by queue or by skills-based routing. Possible values are: • `QueueBased` • `SkillsBased`
`SecondaryRoutingPriority`	**Type** int

Field	Details
	Properties Create, Filter, Group, Nillable, Sort, Update **Description** Indicates the secondary routing priority.
`Serial`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** Serial number of the PendingServiceRouting record. The serial number is automatically incremented each time the PendingServiceRouting record is modified.
`ServiceChannelId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** ID of the service channel.
`WorkItemId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** ID of the work item.

Usage

When you use the PendingServiceRouting object for queue-based routing, the object doesn't invoke triggers before or after insert, or any action (trigger, workflow rule, validation) that could interfere with the creation of the record.

Associated Objects

This object has the following associated objects. Unless noted, they are available in the same API version as this object.

PendingServiceRoutingOwnerSharingRule
Sharing rules are available for the object.

PendingServiceRoutingShare
Sharing is available for the object.

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()`, `describeSObjects()`, `update()`, `query()`, `retrieve()`

Special Access Rules

To access this object, Omni-Channel must be enabled.

As of Spring '20 and later, only authenticated internal and external users can access this object.

Fields

Field	Details
`PresenceDeclineReasonId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** The ID of the PresenceDeclineReason record.
`PresenceUserConfigId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The ID of the PresenceUserConfig record where the decline reasons are added.

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.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. Because changing terms in our code can break current implementations, we maintained this object's name.

Supported Calls

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

Special Access Rules

To access this object, Omni-Channel must be enabled.

As of Spring '20 and later, only authenticated internal and external users can access this object.

Fields

Field	Details
`DeveloperName`	**Type** string **Properties** Create, Filter, Group, Sort **Description** 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. Note: When creating large sets of data, always specify a unique `DeveloperName` for each record. If no `DeveloperName` is specified, performance slows down while Salesforce generates one for each record. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`Language`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort **Description** The language of the PresenceDeclineReason.
`MasterLabel`	**Type** string **Properties** Create, Filter, Group, Sort **Description** The label for the PresenceDeclineReason.

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()`, `describeSObjects()`, `update()`, `query()`, `retrieve()`

Special Access Rules

To access this object, Omni-Channel must be enabled.

As of Spring '20 and later, only authenticated internal and external users can access this object.

Fields

Field	Details
`AcwExtensionDuration`	**Type** int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The maximum length of time, measured in seconds, an agent can spend on After Conversation Work (ACW) each time they extend the timer. You must set this field if `HasAcwExtensionEnabled` is set to `true`. This field is available in API version 56.0 and later.
`AfterConvoWorkMaxTime`	**Type** int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The maximum length of time, measured in seconds, an agent has to complete After Conversation Work (ACW). You must set this field if `HasAfterConvoWorkTimer` is set to `true`. Specify a value from 36 through 3600. This field is available in API version 56.0 and later.
`Capacity`	**Type** int **Properties** Create, Filter, Group, Sort **Description** The maximum number of work assignments that can be pushed to an agent at a time.
`DeveloperName`	**Type** string **Properties** Create, Filter, Group, Sort

Field	Details
	Description 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. Note: When creating large sets of data, always specify a unique `DeveloperName` for each record. If no `DeveloperName` is specified, performance may slow while Salesforce generates one for each record. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`HasAcwExtensionEnabled`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** If set to `true`, agents can extend their After Conversation Work (ACW) time. Available only if `HasAfterConvoWorkTimer` is set to `true`. If set to `true`, you must also set the `AcwExtensionDuration` and `MaxExtensions` fields. The default value is `true`. This field is available in API version 56.0 and later.
`HasAfterConvoWorkTimer`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** If set to `true`, After Conversation Work (ACW) time can be configured for the channel. If set to `true`, you must also set the `AfterConvoWorkMaxTime` field. The default value is `false`. This field is available in API version 56.0 and later.
`Language`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort **Description** The language of the presence configuration.
`MasterLabel`	**Type** string

Field	Details
	Properties Create, Filter, Group, Sort **Description** The label of the presence configuration.
`MaxExtensions`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The maximum number of times an agent can extend their After Work Conversation (ACW) time. Specify a value from 1 through 10. You must set this field if `HasAcwExtensionEnabled` is set to `true`. This field is available in API version 56.0 and later.
`OptionsIsAutoAcceptEnabled`	**Type** boolean **Properties** Create, Filter **Description** Indicates whether work items that are routed to agents are automatically accepted (`true`) or not (`false`). Available only if `OptionsIsDeclineEnabled` is set to `false`.
`OptionsIsDeclineEnabled`	**Type** boolean **Properties** Create, Filter **Description** Indicates whether agents can decline work items that are routed to them (`true`) or not (`false`). Available only if `OptionsIsAutoAcceptEnabled` is set to `false`.
`OptionsIsDeclineReasonEnabled`	**Type** boolean **Properties** Create, Filter **Description** Indicates whether agents can select a reason for declining work requests (`true`) or not (`false`). This can be selected only if decline reasons are enabled.
`OptionsIsDisconnectSoundEnabled`	**Type** boolean

Field	Details
	Properties Create, Filter **Description** Indicates whether a sound is played when agents are disconnected from Omni-Channel (`true`) or not (`false`).
`OptionsIsRequestSoundEnabled`	**Type** boolean **Properties** Create, Filter **Description** Indicates whether a sound plays with incoming work requests (`true`) or not (`false`). Set to `true` by default.
`PresenceStatusOnDeclineId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** The ID of the presence status that's automatically assigned to the agent when the agent declines a work item. Available only if `OptionsIsDeclineEnabled` is set to `true`.
`PresenceStatusOnPushTimeoutId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** 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.

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()`, `update()`, `retrieve()`

Special Access Rules

To access this object, Omni-Channel must be enabled.

As of Spring '20 and later, only authenticated internal and external users can access this object.

Fields

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

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()`, `update()`, `retrieve()`

Special Access Rules

To access this object, Omni-Channel must be enabled.

As of Spring '20 and later, only authenticated internal and external users can access this object.

Fields

Field	Details
`PresenceUserConfigId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The ID of the presence configuration.
`UserId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The ID of the user who's associated with this presence configuration. A user can be associated with only one presence configuration.

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()`

Special Access Rules

To access this object, Omni-Channel must be enabled.

As of Spring '20 and later, only authenticated internal and external users can access this object.

Fields

Field	Details
`CapacityPercentage`	**Type** percent **Properties** Create, Filter, Nillable, Sort, Update **Description** The percentage of an agent's capacity for work items that's consumed by a specific type of work item from this service channel.

Field	Details
	For example, you might give phone calls a capacity percentage of `100`. 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.
`CapacityWeight`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The amount of an agent's capacity for work items that's consumed by a work item from this service channel. 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. This field is available in API version 33.0 and later.
`DeveloperName`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** 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. Note: When creating large sets of data, always specify a unique `DeveloperName` for each record. If no `DeveloperName` is specified, performance slows down while Salesforce generates one for each record. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`DropAdditionalSkillsTimeout`	**Type** int **Properties** Create, Filter, Group Nillable, Sort, Update **Description** The number of seconds to wait before a skill marked as **Additional Skill** is dropped from Omni-Channel routing. The case is then routed to the best-matched agent even if they don't have all the skills.

Field	Details
`IsAttributeBased`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** Indicates whether this routing is attribute-based. Available in API version 45.0 and later.
`Language`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The language of the presence status.
`MasterLabel`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** The label of the presence status.
`OverflowAssigneeId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The ID of the user or queue that's set as the Overflow Assignee.
`PushTimeout`	**Type** int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The number of seconds set for push timeout. **0** is returned when push timeout isn't enabled. Available in API version 36.0 and later.
`RoutingModel`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update

Field	Details
	Description The routing type that determines how work items are routed (pushed) to agents. Possible values are `Least Active` and `Most Available`.
`RoutingPriority`	**Type** int **Properties** Create, Filter, Group, Sort, Update **Description** 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, *0*) are routed to agents first.
`ServiceChannelId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The ID of the service channel that's associated with this configuration. This field is available in API version 32.0 and earlier.

QueueSobject

Represents the mapping between a queue Group and the sObject types associated with the queue, including custom objects.

Supported Calls

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

Special Access Rules

As of Summer '20 and later, only authenticated internal and external users can access this object.

A queue is a Group whose `Type` is `Queue`. To create a Group, you must have the Manage Users permission.

Fields

Field	Details
`QueueId`	**Type** reference **Properties** Create, Filter, Group, Sort

Field	Details
	Description The ID of a queue. This is a relationship field. **Relationship Name** Queue **Relationship Type** Lookup **Refers To** Group
`SobjectType`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort **Description** A list of object types that can be associated with the queue specified by the `QueueId`.

Usage

Use this object to associate a queue with the sObject that can be associated with the queue, including custom objects.

Warning: You can't update or insert more than 18 queues at once when using the Bulk API.

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

Special Access Rules

To access this object, Omni-Channel must be enabled.

As of Spring '20 and later, only authenticated internal and external users can access this object.

Fields

Field	Details
`AcwExtensionDuration`	**Type** int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The maximum length of time, measured in seconds, an agent can spend on After Conversation Work (ACW) each time they extend the timer. You must set this field if `HasAcwExtensionEnabled` is set to `true`. Available only for service channels of type Messaging or Voice.
`AfterConvoWorkMaxTime`	**Type** int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The maximum length of time, measured in seconds, an agent has to complete After Conversation Work (ACW). You must set this field if `HasAfterConvoWorkTimer` is set to `true`. Specify a value from 36 through 3600. Available only for service channels of type Messaging or Voice. For service channels of type Voice, this field is available in API version 52.0 and later. For service channels of type Messaging, this field is available in API version 56.0 and later.
`CapacityModel`	**Type** picklist **Properties** Create, Filter, Group, Nillable, RestrictedPicklist, Sort, Update **Description** The method that determines when an agent's capacity for a work item is released. With the Status-Based capacity routing model, work remains assigned and applied to an agent's capacity until the work is completed or reassigned to a different agent. In contrast, the tab-based capacity model releases an agent's capacity when a work tab is closed in the service console. Possible values are StatusBased and TabBased.
`CapacityPercentage`	**Type** percent **Properties** Create, Filter, Nillable, Sort **Description** The percentage of an agent's capacity for work items that's consumed by a specific type of work item from this service channel.

Field	Details
	For example, you might give phone calls a capacity percentage of *100*. 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 32.0 and earlier. For later API versions, you can set the capacity percentage of work items on the QueueRoutingConfig object. The CapacityPercentage field was removed in API version 33.0.
`CapacityWeight`	**Type** double **Properties** Create, Filter, Nillable, Sort **Description** The amount of an agent's capacity for work items that's consumed by a work item from this service channel. 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. 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 QueueRoutingConfig object. The CapacityWeight field was removed in API version 33.0.
`DeveloperName`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** 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. Note: When creating large sets of data, always specify a unique `DeveloperName` for each record. If no `DeveloperName` is specified, performance slows down while Salesforce generates one for each record.
`DoesCheckCapOnOwnerChange`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update

Field	Details
	Description In the Status-Based capacity routing model, when work is reassigned to a specific agent, you can choose to override the capacity check and keep the work assigned to the agent. The default value is `false`.
`DoesCheckCapOnStatusChange`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** In the Status-Based capacity routing model, when work is reopened, you can choose to override the capacity check keep the work assigned to a specific agent. The default value is `false`.
`DoesMinimizeWidgetOnAccept`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** Automatically minimizes the Omni-Channel widget when an agent accepts work. The default value is `false`.
`HasAcwExtensionEnabled`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** If set to `true`, agents can extend their After Conversation Work (ACW) time. Available only if `HasAfterConvoWorkTimer` is set to `true`. If set to `true`, you must also set the `AcwExtensionDuration` and `MaxExtensions` fields. The default value is `false`. Available only for service channels of type Messaging or Voice. This field is available in API version 56.0 and later.
`HasAfterConvoWorkTimer`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** If set to `true`, After Conversation Work (ACW) time can be configured for the channel. If set to `true`, you must also set the `AfterConvoWorkMaxTime` field. The default value is `false`. Available only for service channels of type Messaging or Voice.

Field	Details
	For service channels of type Voice, this field is available in API version 52.0 and later. For service channels of type messaging, this field is available in API version 56.0 and later.
`Language`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The language of the service channel.
`MasterLabel`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** The label of the service channel.
`MaxExtensions`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The maximum number of times an agent can extend their After Work Conversation (ACW) time. Specify a value from 1 through 10. You must set this field if `HasAcwExtensionEnabled` is set to `true`. Available only for service channels of type Messaging or Voice. This field is available in API version 56.0 and later.
`RelatedEntity`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The type of object that's associated with this service channel. This field is unique within your organization.
`SecRoutingPriorityField`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update

Field	Details
	Description The name of the standard field or the id of the custom field that is used for secondary routing priority. This field is unique within your organization.
`StatusField`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The picklist field that you use to track work status in the Status-Based capacity routing model. Use ServiceChannelStatusField to specify the values that indicate completed and in-progress work-item status.

ServiceChannelFieldPriority

Represents a secondary routing priority field-value mapping. This object is available in API version 47.0 and later.

Supported Calls

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

Special Access Rules

To access this object, Omni-Channel must be enabled.

Fields

Field	Details
`Priority`	**Type** int **Properties** Create, Filter, Group, Sort, Update **Description** The priority number assigned to the mapped field value.
`ServiceChannelId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The ID of the service channel.

Field	Details
`Value`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** The value of the SecRoutingPriorityField field defined in parent ServiceChannel.

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()`, `update()`, `retrieve()`

Special Access Rules

To access this object, Omni-Channel must be enabled.

As of Spring '20 and later, only authenticated internal and external users can access this object.

Fields

Field	Details
`ServiceChannelId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The ID of the service channel.
`ServicePresenceStatusId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** The ID of the presence status that's associated with the service channel that's specified by the `ServicePresenceChannelId`.

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()`, `update()`, `retrieve()`

Special Access Rules

To access this object, Omni-Channel must be enabled.

As of Spring '20 and later, only authenticated internal and external users can access this object.

Fields

Field	Details
`DeveloperName`	**Type** string **Properties** Create, Filter, Group, Sort **Description** 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. Note: When creating large sets of data, always specify a unique `DeveloperName` for each record. If no `DeveloperName` is specified, performance slows down while Salesforce generates one for each record. Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.
`Language`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort **Description** The language of the presence status.
`MasterLabel`	**Type** string

Field	Details
	Properties Create, Filter, Group, Sort **Description** The label of the presence status.

ServiceResource

Represents a service technician or service crew in Field Service and Salesforce Scheduler, or an agent in Workforce Engagement. This object is available in API version 38.0 and later.

Supported Calls

`create()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `update()`, `upsert()`

Special Access Rules

Field Service or Workforce Engagement must be enabled.

Fields

Field Name	Details
`Description`	**Type** textarea **Properties** Create, Nillable, Update **Description** The description of the resource.
`IsActive`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** When selected, this option means that the resource can be assigned to work orders. For service tracking purposes, resources can't be deleted, so deactivating a resource is the best way to send them into retirement. Deactivating a user doesn't deactivate the related service resource. You can't create a service resource that is linked to an inactive user.

Field Name	Details
`IsCapacityBased`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** Capacity-based resources are limited to a certain number of hours or appointments in a specified time period. Tip: The Capacities related list shows a resource's capacity.
`IsOptimizationCapable`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** This field is reserved for Field Service and the managed package. Create a custom field instead of using this field to include a service resource in optimization.
`LastKnownLatitude`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The latitude of the last known location.
`LastKnownLongitude`	**Type** double **Properties** Create, Filter, Nillable, Sort, Update **Description** The longitude of the last known location.
`LastKnownLocation`	**Type** location **Properties** Nillable **Description** The service resource's last known location. You can configure this field to display data collected from a custom mobile app. This field isn't visible in the user interface, but you can expose it on service resource page layouts or set up field tracking to be able to view a resource's location history.

Field Name	Details
`LastKnownLocationDate`	**Type** dateTime **Properties** Filter, Nillable, Sort, Update **Description** The date and time of the last known location.
`LastReferencedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date when the service resource was last modified. Its label in the user interface is `Last Modified Date`.
`LastViewedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date when the service resource was last viewed.
`LocationId`	**Type** reference **Properties** Create, Filter, Group, Sort, Nillable, Update **Description** The location associated with the service resource. For example, a service vehicle driven by the service resource. LocationId is a relationship field. **Relationship Name** Location **Relationship Type** Lookup **Refers To** Location
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update

Field Name	Details
	Description The resource's name, for example the name or title of the associated user or service crew.
`OwnerId`	**Type** reference **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** The owner of the service resource. OwnerId is a polymorphic relationship field. **Relationship Name** Owner **Relationship Type** Lookup **Refers To** Group, User
`RelatedRecordId`	**Type** reference **Properties** Create, Filter, Group, Sort, Nillable, Update **Description** The associated user. Its label in the UI is `User`. If the service resource represents a service crew rather than a user, leave the `User` field blank and select the related crew in the `ServiceCrewId` field. RelatedRecordId is a relationship field. **Relationship Name** RelatedRecord **Relationship Type** Lookup **Refers To** User
`ResourceType`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Indicates whether the resource is a Technician (T), Dispatcher (D), Crew (C), Asset (S), Agent (A), or Planner (P). The default value is Technician (T). Resources who

Field Name	Details
	are dispatchers can't be capacity-based or included in scheduling optimization. Only users with the Field Service Dispatcher permission-set license can be dispatchers. You can't add additional resource types. To create a dependent lookup filter with ServiceResource.ResourceType, use only the first letter of the picklist value, for example T for Technician.
`ServiceCrewId`	**Type** reference **Properties** Create, Filter, Group, Sort, Nillable, Update **Description** The associated service crew. If the service resource represents a crew, select the crew. Note: This field is hidden for all users by default. To use it, update its field-level security settings in Setup and add it to your service resource page layouts.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

ServiceResourceChangeEvent (API version 48.0)
Change events are available for the object.

ServiceResourceFeed
Feed tracking is available for the object.

ServiceResourceHistory
History is available for tracked fields of the object.

ServiceResourceOwnerSharingRule
Sharing rules are available for the object.

ServiceResourceShare
Sharing is available for the object.

SkillRequirement

Represents a skill that is required to complete a particular task in Field Service, Omni-Channel, Salesforce Scheduler, or Workforce Engagement. Skill requirements can be added to pending service routing objects in Omni-Channel. They can be added to work types, work orders, and work order line items in Field Service and Lightning Scheduler. And they can be added to job profiles in Workforce Engagement. This object is available in API version 38.0 and later. You also can add skill requirements to work items in Omni-Channel skills-based routing using API version 42.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `update()`, `upsert()`

Special Access Rules

If you want to use SkillRequirement for Field Service use cases, then Field Service must be enabled.

If you want to use SkillRequirement only for Omni-Channel skills-based routing use cases, then you don't need Field Service to be enabled.

If you want to use SkillRequirement for Workforce Engagement use cases, then Workforce Engagement must be enabled.

Fields

Field Name	Details
`IsAdditionalSkill`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** Indicates that a skill is additional. After a designated timeout period, a skill marked as additional is dropped from Omni-Channel routing. The case is then routed to the best-matched agent even if they don’t have all the skills.
`LastReferencedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp for when the current user last viewed a record related to this record.
`LastViewedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the current user last viewed this record. If this value is null, this record might only have been referenced (`LastReferencedDate`) and not viewed.
`RelatedRecordId`	**Type** reference **Properties** Create, Filter, Group, Sort

Field Name	Details
	Description The record that the skill is required for. The related record can be a work order, work order line item, work type, or pending service routing record. This is a polymorphic relationship field. **Relationship Name** RelatedRecord **Relationship Type** Lookup **Refers To** WorkOrder, WorkOrderLineItem, WorkType
`SkillId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The skill that is required. This is a relationship field. **Relationship Name** Skill **Relationship Type** Lookup **Refers To** Skill
`SkillLevel`	**Type** double **Properties** Create, Defaulted on create, Filter, Nillable, Sort, Update **Description** The level of the skill required. Skill levels can range from zero to 99.99. Depending on your business needs, you can have the skill level to reflect years of experience, certification levels, or license classes.
`SkillNumber`	**Type** string **Properties** Autonumber, Defaulted on create, Filter, idLookup, Sort **Description** An auto-generated number identifying the skill requirement.

Field Name	Details
`SkillPriority`	**Type** int **Properties** Aggregatable, Create, Filter, Group, Nillable, Sort, Update **Description** For additional skills, specify the order in which skills are dropped if after the specified timeout no agent with that skill is available. Higher priority-value skills are dropped first. Lower priority-value skills, for example 0, are dropped last. Skills with the same priority value are dropped as a group. You can set skill priority using skills-based routing rules or Apex code.

Usage

Field Service

Skill requirements help dispatchers assign work orders to service resources with the proper expertise. You can still assign a work order, work order line item, or related service appointment to a service resource that does *not* have the specified skills, so skill requirements serve more as a suggestion than a rule.

Note: If you're using the Field Service managed package, use matching rules to ensure that appointments are only assigned to service resources who possess the skills listed on the parent work order.

If many of your work orders require the same skills, add skill requirements to work types to save time and keep your processes consistent. When you add a skill requirement to a work type, work orders and work order line items that use that type automatically inherit the skill requirement. For example, if all annual maintenance visits for your Classic Refrigerator product require a Refrigerator Maintenance skill level of at least 50, add that skill requirement to the Annual Maintenance Visit work type. When you create a work order for a customer's annual fridge maintenance, applying that work type adds the skill requirement as well.

Omni-Channel

We recommend that you use Omni-Channel flow or skills-based routing rules to create skills-based routing requests. When you do so, work items are routed by creating a PendingServiceRouting object. The PendingServiceRouting object can have multiple SkillRequirements objects associated with it. When a work item requires multiple skills, it's routed to an agent who has all of the required skills. The PendingServiceRouting object adds attributes to the work item that represent the skill (skill id), priority, skill proficiency, and timestamp.

Workforce Engagement

Workforce Engagement uses skill requirements to assign shifts to agents who have the right skills. You can still assign shifts to service resources if they don't have those skills.

In a non-Omni workflow, create a scheduling rule that matches agents to shifts based on their skills and the job profile's skill requirements. Shift scheduling tools can then assign agents with the right skills.

Associated Objects

This object has the following associated objects. Unless noted, they're available in the same API version as this object.

SkillRequirementFeed
Feed tracking is available for the object.

SkillRequirementHistory
History is available for tracked fields of the object.

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()`

Special Access Rules

To access this object, Omni-Channel must be enabled.

Fields

Field	Details
`AtCapacityDuration`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The duration that the user is at full capacity. This field is updated when the agent's capacity changes, such as when the agent is assigned, declines, or closes a work item. Available in API versions 34.0 and later.
`AverageCapacity`	**Type** double **Properties** Filter, Nillable, Sort **Description** The user's average capacity. This field is updated when the agent's capacity changes, such as when the agent is assigned, declines, or closes a work item. Available in API versions 34.0 and later.
`ConfiguredCapacity`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The user's total configured capacity.
`IdleDuration`	**Type** int **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description The duration that the user is idle. This field is updated when the agent's capacity changes, such as when the agent is assigned, declines, or closes a work item. Available in API versions 34.0 and later.
`IsAway`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the user's status is `Away`. The default value is `false`.
`IsCurrentState`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether a presence status is the user's current state. If `true`, the agent is in that presence status. Available in API versions 34.0 and later. The default value is `false`.
`Name`	**Type** string **Properties** Create, Filter, Nillable, Sort, Update **Description** An automatically generated ID number that identifies the record.
`OwnerId`	**Type** reference **Properties** Filter, Group, Sort **Description** The ID of the owner of the `UserServicePresence` entity. For external routing, allows the entity to be used in the Streaming API to listen to events whenever a `UserServicePresence` record is created, modified, or deleted.
`ServicePresenceStatusId`	**Type** reference **Properties** Create, Filter, Nillable, Sort, Update

Field	Details
	Description The ID of the presence status that's associated with the presence user that's specified by the `UserId`.
`StatusDuration`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The duration of the user service presence status. This field is set only when the current user service presence status ends, such as when the agent changes to another presence status or logs out. Available in API versions 34.0 and later. This field is a calculated field: StatusEndDate - StatusStartDate.
`StatusEndDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The end date of the user service presence status. This field is set only when the current user service presence status ends, such as when the agent changes to another presence status or logs out. Available in API versions 34.0 and later.
`StatusStartDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The start date of the user service presence status. Available in API versions 34.0 and later.
`UserId`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** The ID of the Omni-Channel user.

Usage

Apex triggers aren't supported with `UserServicePresence`.

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

Associated Objects

This object has the following associated objects. Unless noted, they are available in the same API version as this object.

UserServicePresenceOwnerSharingRule
Sharing rules are available for the object.

UserServicePresenceShare
Sharing is available for the object.

Omni-Channel Metadata API Types

The Metadata API lets you access Set Up Omni-Channel feature settings and metadata information.

The following types are available with the Metadata API.

- Flow
- OmniChannelSettings
- OmniSupervisorConfig
- PresenceDeclineReason
- PresenceUserConfig
- Queue
- QueueRoutingConfig
- ServiceChannel
- ServicePresenceStatus
- WorkSkillRouting
- WorkSkillRoutingAttribute

If you need more information on the Salesforce Metadata API, see the *Metadata API Developer Guide*.

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 Set Up 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/57.0/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/57.0/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');
                             }
                         });
                     } 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/57.0/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/57.0/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	`true` if retrieving the agent's work items was successful; `false` if retrieving the agent's work items wasn't successful.
`works`	JSON string of `work` objects	A JSON string of `work` 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 `workId` 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 (`true`) or not (`false`).

`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/57.0/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	`true` if retrieving the agent's work items was successful; `false` 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/57.0/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	`true` if retrieving the current presence status channels was successful; `false` 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/57.0/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	`true` if retrieving the presence status ID was successful; `false` 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 `statusId`.

Sample Code–Visualforce

```
<apex:page>
    <apex:includeScript value="/support/console/57.0/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>
```

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/57.0/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/57.0/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
`success`	Boolean	`true` if setting the agent's status was successful; `false` if setting the agent's status 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.
`channels`	JSON string of `channel` 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 Set Up Omni-Channel. These events apply to Salesforce Classic only.

Standard Console Events

Event	Description	Payload
`sforce.console.ConsoleEvent.OPEN_TAB`	Fired when a primary tab or subtab is opened. Available in API version 30.0 or later.	• `id`—the ID of the opened tab • `objectId`—the object ID of the opened tab, if available
`sforce.console.ConsoleEvent.CLOSE_TAB`	Fired when a primary tab or subtab with a specified ID in the `additionalParams` 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.	• `id`—the ID of the closed tab • `objectID`—the object ID of the closed tab, if available
`sforce.console.ConsoleEvent.CONSOLE_LOGOUT`	Delays the execution of logging out of a console when a user clicks **Logout**. When **Logout** is clicked: **1.** An overlay appears, which tells a user that logout is in progress. **2.** Callbacks are executed that have been registered by using `sforce.console.ConsoleEvent.CONSOLE_LOGOUT`. **3.** Console logout logic is executed. 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.	None

Event	Description	Payload
	Available in API version 31.0 or later.	

Set Up Omni-Channel Console Events

Event	Description	Payload
`sforce.console.ConsoleEvent.PRESENCE.LOGIN_SUCCESS`	Fired when an Set Up Omni-Channel user logs into Set Up Omni-Channel successfully. Available in API version 32.0 or later.	• `statusId`—the ID of the agent's current presence status.
`sforce.console.ConsoleEvent.PRESENCE.STATUS_CHANGED`	Fired when a user changes his or her presence status. Available in API version 32.0 or later.	• `statusId`—the ID of the agent's current presence status. • `channels`—channelJSON string of channel objects. • `statusName`—the name of the agent's current presence status. • `statusApiName`—the API name of the agent's current presence status.
`sforce.console.ConsoleEvent.PRESENCE.LOGOUT`	Fired when a user logs out of Salesforce. Available in API version 32.0 or later.	None
`sforce.console.ConsoleEvent.PRESENCE.WORK_ASSIGNED`	Fired when a user is assigned a new work item. Available in API version 32.0 or later.	• `workItemId`—the ID of the object that's routed through Set Up Omni-Channel. This object becomes a work assignment with a `workId` when it's assigned to an agent. • `workId`—the ID of a work assignment that's routed to an agent.
`sforce.console.ConsoleEvent.PRESENCE.WORK_ACCEPTED`	Fired when a user accepts a work assignment, or when a work assignment is automatically accepted. Available in API version 32.0 or later.	• `workItemId`—the ID of the object that's routed through Set Up Omni-Channel. This object becomes a work assignment with a `workId` when it's assigned to an agent. • `workId`—the ID of a work assignment that's routed to an agent.
`sforce.console.ConsoleEvent.PRESENCE.WORK_DECLINED`	Fired when a user declines a work assignment. Available in API version 32.0 or later.	• `workItemId`—the ID of the object that's routed through Set Up Omni-Channel. This object becomes a work assignment with a `workId` when it's assigned to an agent. • `workId`—the ID of a work assignment that's routed to an agent.

Event	Description	Payload
`sforce.console.ConsoleEvent.PRESENCE.WORK_CLOSED`	Fired when the status of an AgentWork object is changed to Closed. Available in API version 32.0 or later.	• `workItemId`—the ID of the object that's routed through Set Up Omni-Channel. This object becomes a work assignment with a `workId` when it's assigned to an agent. • `workId`—the ID of a work assignment that's routed to an agent.
`sforce.console.ConsoleEvent.PRESENCE.WORKLOAD_CHANGED`	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 Set Up Omni-Channel widget.	• `ConfiguredCapacity`—the configured capacity for the agent. • `PreviousWorkload`—the agent's workload before the change. • `NewWorkload`—the agent's new workload after the change.

`channel`

The `channel` object contains the following properties:

Name	Type	Description
`channelId`	String	Retrieves the ID of a service channel that's associated with a presence status.
`developerName`	String	Retrieves the developer name of the the service channel that's associated with the `channelId`.

Methods for Console Events

Method	Description
addEventListener()	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.
fireEvent()	Fires a custom event. This method is only available in API version 25.0 or later.
removeEventListener()	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.

`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
`workId`	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
`workItemId`	String	The ID of the object that's routed through Omni-Channel. This object becomes a work assignment with a `workId` 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 (`true`) or not (`false`).

`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
`channelId`	String	The ID of the channel.
`developerName`	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
`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` 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.

`lowerAgentWorkFlag` for Lightning Experience

Lowers a flag for this agent work item.

Arguments

Name	Type	Description
`workId`	string	The ID of the work item to lower the flag on.

Sample Code

Component code:

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

Controller code:

```
({
lowerFlag: 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.lowerAgentWorkFlag({workId: work.workId}).then(function(res) {
            if (res) {
                console.log("Flag lowered successfully");
            } else {
                console.log("Flag lower failed");
            }
        }).catch(function(error) {
            console.log(error);
        });
    });
}
})
```

Response

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

`raiseAgentWorkFlag` for Lightning Experience

Raises a flag for this agent work item.

Arguments

Name	Type	Description
`workId`	string	The ID of the work item to raise the flag on.
`message`	string	Optional. The message associated with this flag.

Sample Code

Component code:

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

Controller code:

```
({
raiseFlag: 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.raiseAgentWorkFlag({workId: work.workId, message: "Raise Flag
Message"}).then(function(res) {
            if (res) {
                console.log("Flag raised successfully");
            } else {
                console.log("Flag raise 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
`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.
`channels`	JSON string of `channel` 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:omniChannelConnectionError
Indicates that a network connection issue occurred.

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 Omni-Channel.

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 the status of an AgentWork object is changed to Closed.

lightning:omniChannelWorkFlagUpdated
Indicates that an agent's work item flag has been raised or lowered.

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:omniChannelConnectionError`

Indicates that a network connection issue occurred.

Response

Name	Type	Description
`error`	object	The network connection error message.

Example: This example prints a line to the browser's developer console when a network connection error occurs.

Component code:

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

Controller code:

```
({
    onConnectionError : function(component, event, helper) {
        console.log("Network Connection Error.");
        var error = event.getParam('error');
        console.log(error);
    },
})
```

`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
`channelId`	string	Retrieves the ID of the service channel that's associated with a presence status.

Name	Type	Description
`developerName`	string	Retrieves the developer name of the service channel that's associated with the `channelId`.

`lightning:omniChannelLogout`

Indicates that an agent has logged out of Omni-Channel.

Response

None

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

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 `workId` 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 `workId` 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 `workId` 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 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 the status of an AgentWork object is changed 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 `workId` 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:omniChannelWorkFlagUpdated`

Indicates that an agent's work item flag has been raised or lowered.

Response

Name	Type	Description
`workId`	string	The ID of a work item with the updated flag.
`isFlagged`	Boolean	Specifies whether the flag is raised or not.

Name	Type	Description
`message`	string	Optional. A message associated with changing the flag.
`roleUpdatedBy`	string	The role of the user who triggered this flag change. The value is `AGENT` or `SUPERVISOR`.
`updatedBy`	string	The ID of the user who triggered this flag change.

Example: This example prints a line to the browser's developer console when an agent's work item flag is raised or lowered.

Component code:

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

Controller code:

```
({
onChannelWorkFlagUpdated : function(cmp, evt, hlp) {
    var workId = evt.getParam('workId');
    var message = evt.getParam('message');
    var isFlagged = evt.getParam('isFlagged');
    console.log("WorkFlag  event");
    console.log("  workId    : "+ workId);
    console.log("  isFlagged : "+ isFlagged);
    console.log("  message   : "+ message);
}
})
```

`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.

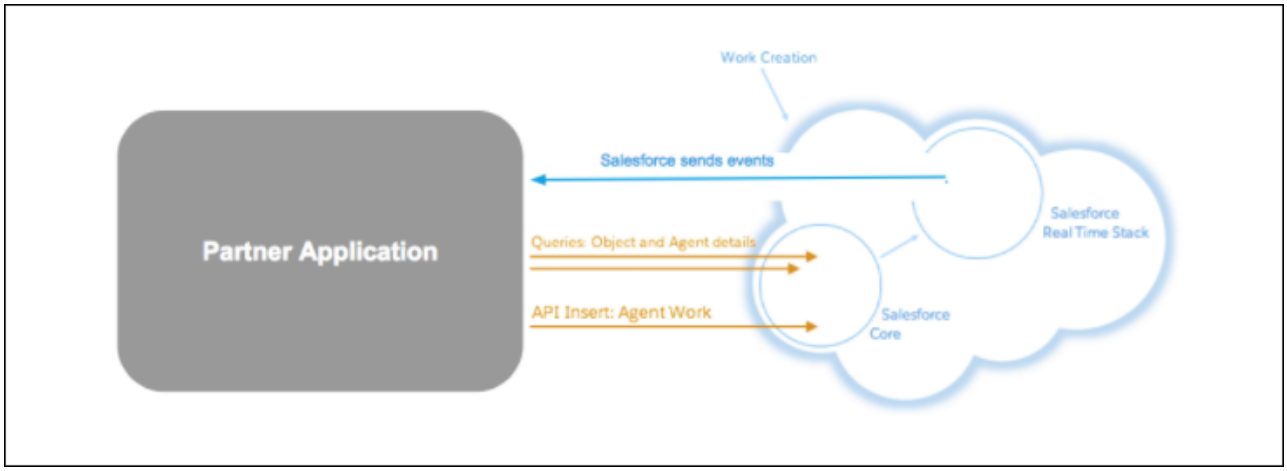
Before setting up external routing, make sure that you have a working implementation of Omni-Channel. You'll use version 41.0 or later of Salesforce standard APIs and streaming APIs to connect Salesforce with an external routing implementation in your routing configuration. Then you can create queues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

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 25
- 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 = 41.0;
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 Chat (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 recommend using both external routing and Omni-Channel queue-based routing in the same implementation. If the same agent is in both queues, the agent's capacity could be exceeded. We don't have control over an agent's capacity in external routing. 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