Agent Interaction SDK Services Developer Guide

The Outbound Service

5/3/2026

Contents

The Outbound Service

The outbound service is the `IExtendedOutboundService` interface defined in the`com.genesyslab.ail.ws.outbound` namespace. It covers the management of outbound campaigns.
The `IExtendedOutboundService` replaces the deprecated `IOutboundService`.

> **Important**
>
> You must still use the `IOutboundService` if your Agent Interaction Service Configuration Layer `enable-chain-75api` option (in the `outbound` section) is set to `false`.

For detailed information on the `IOutboundService`, see the 7.2 version of this *Developer's Guide.*

Introduction

The following subsections provide an overview of the outbound service and its constituent campaigns, outbound chains, records, and interactions.

Outbound Campaigns

An *outbound campaign* is a flexible master plan that organizes calling lists for generating calls to customers, handling customer data, and managing call results.
The Outbound Contact Server (OCS) manages outbound campaigns and automates outbound call dialing. For further information, refer to the *Outbound Contact 7.6* documentation.

Outbound Records

An *outbound record* is an element of a calling list for an outbound campaign. It contains the information required to enable an agent to call a contact—for example, a phone number, the contact name, and how to process the record (type and status).
Chained records are multiple records for the same contact in a calling list. These occur when a contact has several available phone numbers. Each record of the chain can have different time boundaries, as well as different values stored in its business data field.

Outbound Chains

Each *outbound chain* instance contains customer outbound data, provided as a collection of outbound record objects. For example, in the context of a voice outbound call, each record of the chain contains a phone number associated with the customer to be called. If the call with a given record fails, the agent can get a chained record to attempt a new call for this customer.
Regardless of the campaign mode, your application can receive events for new or modified outbound

chains that the agent should process. To determine which outbound chain is associated with an interaction, check the `interaction:outboundChainId` attribute.

The Outbound Service

The outbound service interfaces with outbound campaigns and lists of records. Your application should use this service to let an agent participate in an outbound campaign by processing the records of a calling list.

Features

Your application can use the outbound service to:

- Stay informed about a campaign's progress once the campaign is started.
- Modify a campaign:
  - Add new records to a campaign.
  - Change the campaign dialing mode (if possible).
- Process an agent's outbound voice interactions of an agent:
  - Dial a record.
  - Stay informed about an outbound call's progress.
  - Change the active record for an outbound voice interaction.
  - Cancel, mark as `Do Not Call`, or reject a record or a chain of records.

Mandatory Services

Your application cannot use the outbound service independently from the following services:

- The agent service:
  - Before dialing calls, your application first must use the agent service to log in an agent on a DN. See also The Agent Service.
- The voice interaction service:
  - While the agent is logged on a DN, your application can use the `IInteractionVoiceService` interface to process the calls corresponding to records.
    See also Voice Interactions.
- The interaction service:
  - To access the attributes of your voice interactions, your application uses the `IInteractionService` DTO methods.
  - All the voice interaction attributes are published in events of type `InteractionEvent`. This type is described in the `IInteractionService` interface. Outbound chain information is accessible from this service.
    See also The Interaction Service.

Outbound Campaigns

An agent who participates in an active or running outbound campaign, should be informed of those campaigns' characteristics and also of campaign changes when they occur.
The `IExtendedOutboundService` interface lets your application access campaign information, as detailed in the following.

Campaign Attributes

For each outbound campaign, the `IExtendedOutboundService` interface defines a set of attributes in the `outbound.campaign` domain. Your application can use these attributes for information purposes:

- `outbound.campaign:name`—The name of the outbound campaign.
- `outbound.campaign:description`—The description of the outbound campaign; for example, its purpose.
- `outbound.campaign:mode`—The current mode of the campaign.

For each campaign, your application uses the following attributes for management:

- `outbound.campaign:campaignId`—The system ID of the campaign.
- `outbound.campaign:state`—The status of the campaign.
- `outbound.campaign:eventType`—When present, this event indicates the type of campaign.
- `outbound.campaign:actionsPossible`—The possible actions that the agent can perform on the campaign at a certain point in time.

The following sections detail how to use these attributes.
Use the `IExtendedOutboundService.getCampaignsDTO()` method to read attributes having the `read` property. A call to this method returns an array of `CampaignDTO` objects. The `CampaignDTO` class is a container that associates a campaign ID with a key-value list. For further information about DTO, see Data Transfer Object.
The `IExtendedOutboundService` interface has no writable campaign attributes. An agent using your application cannot change the campaign attribute values. However, actions and external events can modify the attribute values. Changes are propagated in events. See Campaign Events.

Campaign Dialing Modes

The *campaign dialing modes* determine how an agent, or a group of agents, participates in an outbound campaign. The table below presents the possible dialing modes of an outbound campaign.

Campaign Dialing Modes

CampaignMode	Description
PREVIEW	In this dialing mode, an agent requests one or several records from the OCS, previews the record(s), and decides to process one of them.
PUSH_PREVIEW (also called PROACTIVE)	In this mode, the agent receives the record, and does not need to request it in order to preview it.
PROGRESSIVE	The OCS dials a record in the list as soon as an agent is available.
ENGAGED_PROGRESSIVE	The OCS creates a voice interaction to dial a record in the list when an agent is available and engaged.
PREDICTIVE	The OCS predicts agent availability and dials a record. Your agent application gets a dialing voice interaction and an outbound chain for this record.
ENGAGED_PREDICTIVE	The OCS predicts when an agent is available and engaged, and creates a voice interaction to dial a record in the list.
UNKNOWN	Unknown campaign mode.

> **Important**
>
> In a mode that is not ENGAGED_, the contact may be online before the agent. For further information, refer to the Outbound Contact 7.6 documentation.

Campaign Actions

The `CampaignAction` enumeration lists the possible actions that your agent application can perform on a campaign using certain `IExtendedOutboundService` methods. `CampaignAction` usually pertains to campaigns in preview dialing mode.
Test the `outbound.campaign:actionsPossible` attribute of the `IExtendedOutboundService` interface to determine which actions are possible at a certain point in time. When possible actions on a campaign change, a `CampaignOutboundEvent` event may propagate the new value of the `outbound.campaign:actionsPossible` attribute for this campaign. See Campaign Events.

Campaign Status

The current status of a campaign is available as the value of the interface's `outbound.campaign:state` attribute. The `CampaignStatus` enumeration lists the possible status values of an outbound campaign. For further information, refer to the *Agent Interaction SDK 7.6 Services API Reference*.
The status of an outbound campaign can change due to the OCS management of campaigns. Status changes are propagated with `CampaignOutboundEvent` events. See Campaign Events.

Campaign Events

When changes occur on an outbound campaign, the event service of your application can receive `CampaignOutboundEvent` events only for agents who take part in the outbound campaign.
To properly take into account those events, your application must subscribe to them with the event service and must retrieve at least the following published attributes:

- `outbound.campaign:campaignId`—Determines which campaign is related to the event.
- `outbound.campaign:eventType`—When present, this event indicates the type of campaign.
- `outbound.campaign:actionsPossible`—Updated possible actions for the related campaign.

The published `outbound.campaign:eventType` attribute points out which value changes are propagated in an `CampaignOutboundEvent` event. The `CampaignEventType` enumeration lists the possible values of this attribute. The table below shows which published attributes to read according to the `CampaignEventType` value.

CampaignEventType and Published Attribute

Campaign-EventType	Associated Attribute	Description
`CAMPAIGN_ADDED`	`outbound.campaign:*`	A campaign was added in the outbound service.
`CAMPAIGN_REMOVED`	`outbound.campaign:*`	A campaign was removed from the outbound service.
`CAMPAIGN_MODE_CHANGED`	`outbound.campaign:mode`	The mode of a campaign has changed.
`CAMPAIGN_STATE_CHANGED`	`outbound.campaign:state`	The status of a campaign has changed.
`UNKNOWN`	`outbound.campaign:*`	Unknown type of event on an outbound campaign.

For further information about events and subscribing, see The Event Service.

Outbound Chain Events

When changes occur on an outbound chain, the event service of your application can receive`OutboundChainEvent` events only for agents who take part in the outbound campaign. To properly take into account those events, your application must subscribe to them with the event service, and must retrieve at least the published attributes listed in the following table.

Outbound Chain Events

Attribute	Description
`outbound.chain:activeRecordId`	The active record of the chain.
`outbound.chain:recordIds`	The list of outbound record identifiers in this chain. The number of records can change during the life cycle of the interaction. The interaction has one record at initialization (the initial record), and at least one record after calling `requestChainedRecords`.
`outbound.chain:eventType`	The type of the event.[#pgfId-1025870 1]
`outbound.chain:outboundChainId`	The interaction identifier.a
`outbound.chain:reason`	The reason.a
`outbound.chain:campaignMode`	The campaign mode related to this outbound chain.a
`outbound.chain:interactionIds`	The interaction identifers related to this chain.

Outbound Chains and Records

The Introduction and Outbound Campaigns sections introduced you to the design of the outbound feature and noted that outbound data does not interfere with interaction management. Now, to handle campaign information and outbound records, you need to add some code and make certain modifications to your agent application.
These changes you need to make require that you address two main issues:

- Subscribe to the correct events.
- Identify if a given interaction has outbound information.

Subscribe to Outbound and Chain Events

In order to get notified of changes to active outbound campaigns and chains, you need to subscribe to `CampaignOutboundEvents` and `OutboundChainEvents`. Use the following code snippets as guidelines for how to subscribe:

```
/// Creating topic objects for the outbound service
TopicsService [] topicServices = new TopicsService[1];
topicServices[0] = new TopicsService();
topicServices[0].serviceName = "IExtendedOutboundService";
topicServices[0].topicsEvents = new TopicsEvent[2];
topicServices[0].topicsEvents[0] = new TopicsEvent();

// Creating a topic event for campaign outbound events
topicServices[0].topicsEvents[0].eventName = "CampaignOutboundEvent";
topicServices[0].topicsEvents[0].attributes =
new String[]{"outbound.campaign:*"};
topicServices[0].topicsEvents[0].triggers = new Topic[1];
topicServices[0].topicsEvents[0].triggers[0] = new Topic();
topicServices[0].topicsEvents[0].triggers[0].key = "PLACE";
topicServices[0].topicsEvents[0].triggers[0].value = mPlaceId;
topicServices[0].topicsEvents[1] = new TopicsEvent();

// Creating a topic event for outbound chain events
topicServices[0].topicsEvents[1].eventName = "OutboundChainEvent";
topicServices[0].topicsEvents[1].attributes =
new String[]{"outbound.chain:*"};
topicServices[0].topicsEvents[1].triggers = new Topic[1];
topicServices[0].topicsEvents[1].triggers[0] = new Topic();
topicServices[0].topicsEvents[1].triggers[0].key = "PLACE";
topicServices[0].topicsEvents[1].triggers[0].value = mPlaceId;
```

Check Interactions for Outbound Information

When an interaction arrives at your agent application while an outbound campaign is active, depending on the mode of the campaign, you may need to check to see if interactions with `NEW` and `DIALING` status are associated with outbound information. If that is the case, add the following code to your application so that it handles the appropriate `InteractionEvents` for such outbound interactions:

```
InteractionDTO[] myInteractionDTO =
   myInteractionService.getInteractionsDTO(
      new string[]{myInteractionId},
      new string[]{"interaction.outboundChainId"});
KeyValue myAttrKeyValue = myInteractionDTO[0].data[0];
String myOutboundChainId = (String) myAttrKeyValue.value;
```

Outbound Attributes

The `IExtendedOutboundService` interface exposes two types of attributes used to process an outbound record, one type each, respectively, in the `outbound.chain` and`outbound.record` domains.

Outbound Chain Attributes

The `outbound.chain` domain defines a subset of outbound data for an outbound chain. See Outbound Chain Events for details. These outbound chain attributes make the links between interactions and outbound records.

Record Attributes

For each outbound record, the outbound.record attributes define a set of data characteristics related to an outbound record. The following tables list representative (but non exhaustive) record attributes. The first list contains information attributes and the second, attributes for management purposes.

Record Attributes for Information (Selected)

Attribute	Description
outbound.record:callingListName	The name of the calling list to which a record belongs.
outbound.record:phone	The phone number to dial to process a record.
outbound.record:campaignId	The ID of the campaign to which a record belongs; this ID can be used to retrieve information about the corresponding campaign. See Outbound Campaigns for details.
`outbound.record:outboundChainId`	The `OutboundChain` to which this record belongs, or `null`.

Record Attributes for Management (Selected)

Attribute	Description
`outbound.record:recordId`	The ID of a record.
`outbound.record:status`	The status of a record.
`outbound.record:actionsPossible`	The possible actions on a record.

Attribute	Description
`outbound.record:callResult`	The call result of a record.

From the agent point of view, a record does not exist independent of an outbound voice interaction. So your application needs an interaction ID to access a given record's attribute values.
Using the attribute properties, your application can read the attributes of the domain with the`IExtendedOutboundService.getRecordsDTO()` method. It can then write those attributes with the `IExtendedOutboundService.setRecordsDTO()` method. These methods use `OutboundRecordDTO` objects. The `OutboundRecordDTO` class associates a record ID with a key-value list of attributes. For further information, see Handling Interaction DTOs.

> **Important**
>
> When your application has set new values for a record with the `IExtendedOutboundService.setRecordsDTO()` method, your application must call the`IExtendedOutboundService.updateRecord()` method to commit all modifications in the OCS.

Outbound Actions

The `IExtendedOutboundService` interfaces provide you with outbound methods that the agent calls to perform outbound actions. These include `cancel`, `do not call`, and`reschedule`. Such actions are independent from the interaction used to process the outbound record or the outbound chain. They manage record information on the Outbound Server.

Outbound Campaign in Preview Mode

In preview and predictive dialing modes, an agent may have to use the `START_PREVIEW_MODE`, `GET_PREVIEW_RECORD`, and `STOP_PREVIEW_MODE` actions according to the options set in the OCS and Configuration Layer. For further details, refer to the Outbound Contact 7.6 Reference Manual.
When an agent explicitly initiates preview mode with the outbound service, he or she notifies the OCS that he or she is ready to participate in a specific campaign.
In this specific mode, an agent requests one or several records from the OCS, previews each record, and decides whether or not to dial a call. For an agent to stop preview mode with the outbound service, he must notify the OCS that he no longer is participating in the specified campaign. The `IExtendedOutboundService` interface provides your application with three methods to work with preview mode.

Outbound Service Methods for Preview Mode

Method	Description
`startPreviewMode()`	An agent is ready to start participating in a campaign.
`getPreviewRecordDTO()`	An agent retrieves a preview record from a campaign, and this method returns the record data in a DTO. Your application receives an interaction event for the outbound voice interaction associated with this outbound record.
`stopPreviewMode()`	An agent stops participating in a campaign.

> **Important**
>
> If your campaign mode is push preview, your application does not need to request the preview record.

The following code snippet shows how to use the startPreviewMode(), getPreviewRecordDTO(), and stopPreviewMode() methods:

```
/// agent0 is ready to participate in a campaign identified by
/// myCampaignID
myOutboundService.startPreviewMode(“agent0”,myCampaignID);
///...
///Retrieving a preview record for agent0
OutboundRecordDTO myRecordDTO =
myOutboundService.getPreviewRecordDTO("agent0", myCampaignId,
new string[]{"outbound.record:*"}); /// all record attributes
/// Displaying the DTO content
foreach(KeyValue myPair in myRecordDTO.data)
{
   System.Console.WriteLine("Key="+ myPair.key
   +" value="+myPair.value.ToString());
}
/// agent0 no longer participates in a campaign identified by
/// myCampaignID.
myOutboundService.stopPreviewMode(“agent0”,myCampaignID);
```

Then, if a record is available, you get two events:

- `OutboundChainEvent`
- `InteractionEvent`

Use the `OutboundChainEvent` event to inform the user that a new outbound record should be processed.

The corresponding InteractionEvent references an interaction object with the status NEW. Use this interaction to retrieve the outbound chain that contains the preview record. To get this record, get the outbound.chain:activeRecordId attribute using OutboundChainDTO, as follows:

```
myOutbondChainDTO = myOutboundService.getOutboundChainDTO(myPlaceId, myOutboundChainId,
   new string[]{"outbound.chain:activeRecordId"}
);
```

Then, use the Interaction instance as you normally would to process the outbound record. Get outbound record data to fill in Interaction data and the parameters for your methods. After that, you need to create an interaction to process a voice call and use the record data to dial the call.
If you deal with multimedia interactions, multimedia information is available in the record's custom fields (outbound.record:customFields). When the interaction is processed, you mark the corresponding outbound chain as processed:

```
myOutboundService.markChainProcessed(myPlaceId,myOutboundChainId);
```

Outbound Campaign in Predictive Mode

Handling a predictive outbound interaction is simpler than handling a preview outbound interaction. The setup is the same, but you do not have to request records since Outbound Server is in charge of distributing records. Refer to outbound documentation for further details. For a predictive outbound campaign, your application just waits for RINGING interactions and outbound chains.

Active Campaigns

To determine if your agent is to participate in a predictive outbound campaign, test whether the outbound.campaign:mode is PREDICTIVE. To do this, get information in CampaignOutboundEvent events.

Handling a Predictive Outbound Interaction

If your agent is to participate in a predictive campaign, your application should get interactions in DIALING or TALKING status, which you process as usual. For further details, see The Interaction Service, and other interaction-handling chapters.
When your application gets these predictive interactions through interaction events, identify the outbound data. Each received outbound interaction is associated with an outbound chain that contains the record. Use this data to fill in interaction data. To get this record, get the outbound.chain:activeRecordId attribute using OutboundChainDTO:

```
myOutbondChainDTO = myOutboundService.getOutboundChainDTO(myPlaceId,
   myOutboundChainId, new string[]{"outbound.chain:activeRecordId"});
```

When the agent has processed the outbound record, he or she must specify the processing result using the IExtendedOutboundService.setRecordDTO method. When the interaction is processed, you mark the corresponding outbound chain as processed:

```
myOutboundService.markChainProcessed(myPlaceId,myOutboundChainId);
```

To determine whether the agent must get a record or wait for a new interaction, refer to Outbound Solution Documentation.