# SolarWinds Virtual Agent Sample

## Overview

The SolarWinds Virtual Agent SDK sample includes two Virtual Agents that automatically create and update ticket records. The Pre-Conversation Virtual Agent looks for an existing ticket (from field in the Web Client – see below) or creates a ticket in SolarWinds and fills out the following fields:

* Client – guest record (looked up by their email address)
* Room Number – field from Web Client
* Extension – field from Web Client
* Asset Number – field from Web Client
* Request Type – field from Web Client
* Detail – the guest’s question
* Location – the guest’s location from their Client record

The Post-Conversation Virtual Agent updates the pre-existing ticket record by adding a note with the conversation history that happened between the guest and the Chime agent.

## Chime Web Client customizations for Virtual Agent sample

This Virtual Agent sample uses additional information from Web Client. The Web Client should be changed to have the following fields:

* Room
* Extension
* Asset Number
* Request Type
* Open Ticket ID

## References

The Virtual Agent sample uses the following references:

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Reflection;

using System.Text;

## Pre-Conversation

In the IncomingSeekerOffer method, the Pre-Conversation Virtual Agent caches the guest’s question, e-mail address, and session ID.

public bool IncomingSeekerOffer(int sessionId, string queueName, int queueID, string email, string question, PreviousSessionState? prevState)

{

 var seekerInfo = new SeekerInfo(email, question);

 if (!\_openSessions.ContainsKey(sessionId))

 {

 \_openSessions.Add(sessionId, seekerInfo);

 }

 else

 {

 \_openSessions[sessionId] = seekerInfo;

 }

 return true;

}

In the SeekerConnected method, the Virtual Agent extracts the custom Web Client fields Room Number, Asset, Extension, and Request Type, then calls to the SolarWinds\_API class to do the pre-conversation work of creating the ticket and assigning these fields. Finally, the Virtual Agent sends the URL to the newly created ticket as a Custom Tab object for the agent’s Context Window Extension.

const string RoomNumber = "RoomNumber";

const string Extension = "Extension";

const string AssetNumber = "AssetNumber";

const string RequestType = "RequestType";

public bool SeekerConnected(int sessionId)

{

 if (\_openSessions.ContainsKey(sessionId))

 {

 var seekerInfo = \_openSessions[sessionId];

 var data = \_pluginManager.GetSeekerDictionary(sessionId);

 var roomNumber = "";

 var extension = "";

 var assetNumber = "";

 var requestType = -1;

 if (data.ContainsKey(RoomNumber))

 {

 roomNumber = data[RoomNumber];

 }

 if (data.ContainsKey(Extension))

 {

 extension = data[Extension];

 }

 if (data.ContainsKey(AssetNumber))

 {

 assetNumber = data[AssetNumber];

 }

 if (data.ContainsKey(RequestType))

 {

 try

 {

 requestType = int.Parse(data[RequestType]);

 }

 catch (Exception ex)

 {

 \_pluginManager.LogMessageInChime(LoggingLevel.Error, "Exception parsing RequestType from seeker data. This field is required to create a ticket in SolarWinds.");

 }

 }

 var ticket\_url = \_api.PreConversation(sessionId, requestType, seekerInfo, roomNumber, extension, assetNumber);

 var custom\_tabs = new List<CustomTab>() { new CustomTab(ticket\_url, "SolarWinds Ticket", false) };

 \_pluginManager.SendCustomTabToAgent(sessionId, custom\_tabs);

 }

 else

 {

 \_pluginManager.LogMessageInChime(LoggingLevel.Error, "No seeker info found for session ID");

 }

 \_pluginManager.DisconnectVirtualAgent(sessionId, true);

 return true;

}

## Post-Conversation

In the SeekerConnected method, the Post-Conversation Virtual Agent retrieves the chat messages between the guest and agent and passes those to the PostConversation method of the SolarWinds\_API class.

public bool SeekerConnected(int sessionId)

{

 \_va\_state = VirtualAgentState.Busy;

 \_api.PostConversation(sessionId, \_pluginManager.GetChatMessagesClean(sessionId));

 \_pluginManager.DisconnectVirtualAgent(sessionId, false);

 \_va\_state = VirtualAgentState.Online;

 return true;

}

In the PostConversation method, the SolarWinds\_API class retrieves the cached ticked ID for the appropriate Chime session and formats the chat messages. Then the Note is created and associated with the existing Ticket in SolarWinds.

public void PostConversation(int sessionId, List<string> chatMessages)

{

 var chatHistory = "";

 if (chatMessages.Any())

 {

 chatHistory = chatMessages.Aggregate((next, workingSentence) => next + Environment.NewLine + workingSentence);

 }

 if (\_sessionIdToTicket.ContainsKey(sessionId))

 {

 var ticket = \_sessionIdToTicket[sessionId];

 var note = new Note(chatHistory, ticket.id);

 CreateNote(note);

 RemoveSeeker(sessionId);

 }

 else

 {

 \_pluginManager.LogMessageInChime(LoggingLevel.Error, "Chime session ID was not found in virtual agent cache of session Tickets in SolarWinds");

 }

}