For <http://addchime.com/salesforce-click-to-chat.html>

# Virtual Agent Integration with Salesforce REST API

## Note about references

The following examples use the following references. These examples use Newtonsoft Json library to deserialize the responses from Salesforce, and Microsoft Visual Studio Web API library to make a Patch HTTP request.

using Microsoft.VisualStudio.Services.WebApi;

using Newtonsoft.Json;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net.Http;

using System.Net.Http.Headers;

using System.Text;

using System.Xml.Linq;

## Connect to Salesforce (create an authentication header for subsequent HTTP requests)

This code snippet shows you how to use your salesforce credentials to retrieve an access token and construct an authentication header for making subsequent HTTP requests to the Salesforce REST API.

public class SFResponse

{

 public string access\_token = "";

 public string instance\_url = "";

 public string token\_type = "";

}

public static async void ConnectToSalesForce()

 {

 StringBuilder body = new StringBuilder();

 body.Append("?grant\_type=password&");

 body.Append("client\_id=" + client\_id + "&");

 body.Append("client\_secret=" + client\_secret + "&");

 body.Append("username=" + username + "&");

 body.Append("password=" + password + security\_token);

 using (var client = new HttpClient())

 {

 AuthURL = "https://login.salesforce.com/services/oauth2/token";

 var values = new Dictionary<string, string>();

 var content = new FormUrlEncodedContent(values);

 var response = client.PostAsync(AuthURL + body, content).Result;

 if (response.IsSuccessStatusCode)

 {

 var jsonString = await response.Content.ReadAsStringAsync();

 SFResponse dataObjects = JsonConvert.DeserializeObject<SFResponse>(jsonString);

 authHeaders = new AuthenticationHeaderValue(dataObjects.token\_type, dataObjects.access\_token);

 isConnectedToSF = true;

 return;

 }

 Console.WriteLine("{0} ({1})", (int)response.StatusCode, response.ReasonPhrase);

 isConnectedToSF = false;

 }

 }

## Query a Salesforce table (look for a record with a matching e-mail address)

This example shows you how to query the Salesforce Contact table for records that match the provided e-mail address, and receive the Department and Title fields from those record. We’re using the authentication header created in the first example.

public class QueryResponse

{

 public List<Contact> records;

}

public class Contact

{

 public string Department = "";

 public string Title = "";

}

public static async void QuerySalesforce(string email)

 {

 using (var client = new HttpClient())

 {

 client.DefaultRequestHeaders.Authorization = authHeaders;

 var query = "SELECT Department , Title FROM Contact WHERE Email = \'" + email + "\'";

 var requestURL = instance\_url + "/services/data/v20.0/query?q=" + query.Replace(" ", "+");

 var response = client.GetAsync(requestURL).Result;

 if (response.IsSuccessStatusCode)

 {

 var jsonString = await response.Content.ReadAsStringAsync();

 QueryResponse data = JsonConvert.DeserializeObject<QueryResponse>(jsonString);

 Contact seeker = data.records.FirstOrDefault();

 if (seeker != null)

 {

 recordID = seeker.Id;

 }

 }

 }

 }

## Update a Salesforce record

This example shows how you can update a record field in Salesforce, given that record’s ID. This sample will update the description field of a Contact record.

 public static void UpdateSalesforce(string recordId, string description)

 {

 using (var client = new HttpClient())

 {

 client.DefaultRequestHeaders.Authorization = authHeaders;

 var values = new Dictionary<string, string>();

 values.Add("Description", description);

 var stringPayload = JsonConvert.SerializeObject(values);

 var content = new StringContent(stringPayload, Encoding.UTF8, "application/json");

 var requestUrl = instance\_url + "/services/data/v20.0/sobjects/Contact/" + recordId;

 var response = client.PatchAsync(requestUrl, content).Result;

 if (response.IsSuccessStatusCode)

 {

 Console.WriteLine("Success patching Contact object in salesforce");

 didUpdateSF = true;

 }

 else

 {

 Console.WriteLine(response.StatusCode + " " + response.ReasonPhrase);

 }

 }

 }

## Post a Chatter feed item to a Contact record

This example will show you how, given a record ID, you can post a Chatter feed item directly to that Contact record’s page. The feed item will show up as being posted by the entity you have authenticated yourself as in step 1.

public class FeedItem

{

 public Body body = new Body();

 public string feedElementType = "FeedItem";

 public string subjectId = Program.recordID;

 public FeedItem(string descr)

 {

 body.messageSegments.Add(new MessageSegment() { text = descr });

 }

 public class Body

 {

 public List<MessageSegment> messageSegments = new List<MessageSegment>();

 }

 public class MessageSegment

 {

 public string type = "Text";

 public string text;

 }

}

 public static void PostChatterFeed(string recordID, string description)

 {

 string chatterUrl = instance\_url + "/services/data/v35.0/chatter/feed-elements?feedElementType=FeedItem&subjectId="+recordID+"&text=New+post";

 using (var client = new HttpClient())

 {

 client.DefaultRequestHeaders.Authorization = authHeaders;

 FeedItem values = new FeedItem(description);

 var stringPayload = JsonConvert.SerializeObject(values);

 var content = new StringContent(stringPayload, Encoding.UTF8, "application/json");

 var response = client.PostAsync(chatterUrl, content).Result;

 if (response.IsSuccessStatusCode)

 {

 Console.WriteLine("Success posting Chatter Feed object to salesforce");

 didUpdateSF = true;

 }

 else

 {

 Console.WriteLine(response.StatusCode + " " + response.ReasonPhrase);

 }

 }

 }