# Virtual Agent Integration with ServiceNow REST API

## Note about references

The following examples use the following references. These examples use Newtonsoft Json library to serialize the objects we send to the ServiceNow API in json.

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;

## Authenticate with ServiceNow (create a basic authentication header for subsequent HTTP requests)

This code snippet shows you how to use your ServiceNow credentials to construct an authentication header for making basically authenticated HTTP requests to the ServiceNow REST API. The ServiceNow REST API supports [basic authentication](http://www.w3.org/Protocols/HTTP/1.0/draft-ietf-http-spec.html#BasicAA) and OAuth. You can then use a dummy query to ServiceNow to verify that your authentication header is valid. The dummy query in this case asks for the first ten records from the Incident table, this is a default ServiceNow table.

authHeaders = new AuthenticationHeaderValue("Basic", Convert.ToBase64String(Encoding.UTF8.GetBytes(username + ":" + password)));

public void ConnectToServiceNow()

{

using (var client = new HttpClient())

{

client.DefaultRequestHeaders.Authorization = authHeaders;

string connectURL = instance\_url + @"incident?sysparm\_limit=10";

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

if (response.IsSuccessStatusCode)

{

isConnectedToServiceNow = true;

Console.WriteLine("Success connecting");

}

else

{

isConnectedToServiceNow = false;

}

}

}

## Create a ticket in ServiceNow

This example shows you how to create an incident record (ticket) in ServiceNow via their REST API, populate some fields, and then construct a URL to the newly created ticket. We use the Newtonsoft Json library to serialize a local Incident object for making the Post request. For Post requests, ServiceNow requires the accept header and the content-type header to be “application/json”.

public class Incident

{

public string short\_description;

public string work\_notes;

}

public void CreateTicket(string email, string question)

{

string createIncidentURL = instance\_url + @"incident";

Incident newIncident = new Incident() { short\_description = question, work\_notes = "Incident reported by " + email};

var jsonString = JsonConvert.SerializeObject(newIncident);

using (var client = new HttpClient())

{

client.DefaultRequestHeaders.Authorization = authHeaders;

client.DefaultRequestHeaders.Add("Accept", "application/json");

HttpRequestMessage request = new HttpRequestMessage(HttpMethod.Post, createIncidentURL);

request.Content = new StringContent(jsonString, Encoding.UTF8, "application/json");

var response = client.SendAsync(request).Result;

if (response.IsSuccessStatusCode)

{

var newIncidentURL = response.Headers.Location.OriginalString; }

}

}

## Update a ServiceNow ticket

This example shows how you can update a record in ServiceNow, given that record’s sys\_id. This sample will update the comment field of an Incident record, given that record’s sys\_id.

public class UpdateIncident

{

public string comments;

}

public void UpdateTicket(string comment, string sys\_id)

{

UpdateIncident updateIncident = new UpdateIncident(){ comments = comment};

var jsonString = JsonConvert.SerializeObject(updateIncident);

string updateIncidentURL = instance\_url + @"incident/" + sys\_id + "?sysparm\_exclude\_ref\_link=true";

using (var client = new HttpClient())

{

client.DefaultRequestHeaders.Authorization = authHeaders;

client.DefaultRequestHeaders.Add("Accept", "application/json");

HttpRequestMessage request = new HttpRequestMessage(HttpMethod.Put, updateIncidentURL);

request.Content = new StringContent(jsonString, Encoding.UTF8, "application/json");

var response = client.SendAsync(request).Result;

if (response.IsSuccessStatusCode)

{

Console.WriteLine("Success updating ticket");

}

else

{

Console.WriteLine("Unsuccessfully updated ticket");

}

}

}