ServiceNow Integration with Twitter

Twitter Setup

Create new Application

1. Login to Twitter Developer [Twitter App](link).
2. Click "Create an app" and fill out mandatory fields to create a new Application.

App details

The following app details will be visible to app users and are required to generate the API keys needed to authenticate Twitter developer products.

App name (required) ⓘ

Maximum characters: **32**

Application description (required)

Share a description of your app. This description will be visible to users so this is a good place to tell them what your app does.

Please be detailed.

3. Go to "Keys and tokens" in your application and take note of the "API Key" and "API secret key" as they will be needed during the ServiceNow Instance Setup – Oauth.

egration

Keys and tokens   Permissions

Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys

16GvfAHErJGeA1YYWO1tRYnW1 (API key)

IK9lf5tyTIxVBvfui6mZynMUm7odUvabh9Fnd2FO72pJ(

(Regenerate)

ServiceNow Instance Setup

Oauth

1. Log into you ServiceNow instance, for example, https://<yourinstancename>.service-now.com/

1. In the ServiceNow application navigator, search "Rest Message". Click "Rest Message" module under Outbound section.

2. Select the "Get Twitter OAuth Token" option. Click "post" option under the HTTP Methods.

3. Under the Authentication tab, you will need to fill out the fields (API key & API secret key) provided in your Twitter setup.

Authentication	HTTP Request

REST Message HTTP Methods support the following Authentication types:
Basic authentication
Mutual (two-way authentication)
OAuth 2.0

Authentication configured on the HTTP Method will override the parent REST Message configuration.

More info

Authentication type	Basic (simple)
Basic authentication user ID (simple)	16GvfAHErJGeA1YYWO1tRYnW1
Basic authentication password (simple)	••••••••

4. When done, go to the Relation Links sections, and click "test'. Take note of the access token as it will be needed during the "Get Twitter Search" section.

Response {"token_type":"bearer","access_token":"AAAAAAAAAAAAAAAAAAAAAAHij%2BAAAAAAu33OhintFt99t5C3fEmvO7F%2Fy2w%3DTvC3y2DCGvaTE5zK1Iacz88tSUTk4gMvGacZHXRFbKI OTn9zwD"}

Get Twitter Search

1. In the ServiceNow application navigator, search "Rest Message". Click "Rest Message" module under Outbound section.

2. Select the "Get Twitter Search" option. Click "get" option under the HTTP Methods.

3. In the Variable Substitutions section, create new. Fill out the "test value" field using the token access recorded in the previous step. Append the word "bearer" before the access token.

Name	token		Method	get	🔍
Escape type	No escaping ⬍		Application	Global	
Test value	bearer AAAAAAAAAAAAAAAAAAAAAAHij%2BAAAAAAAu33OhintFt99t5C3fEmvO7F%2Fy2w%3DTvC3y2DCGvaTE5zK1Iacz88tSUTk4gMvGacZHXRFbKIOTn9zwD				

4. When done, go to the HTTP Request section, and replace the value of "q" with your desired hashtag.
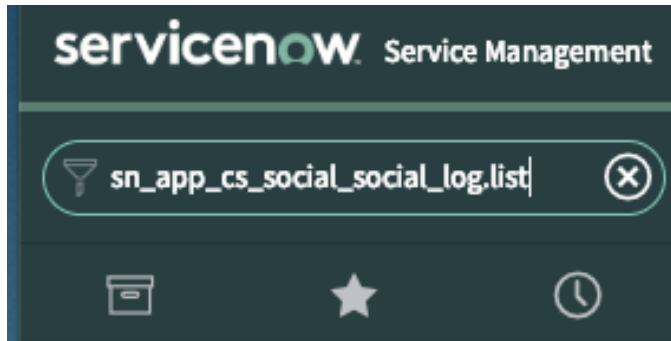
⚙	☰ Name	☰ Value
✕	ⓘ q	#Mynowfeedback

Scheduled Job

1. In the ServiceNow application navigator, search "Scheduled Jobs". Select "Twitter Scheduled Job Script" option and click "Execute Now" to run script that will get all tweets associated with the provided hashtag.

✳ Name | name	Twitter Scheduled Job Script
Active | active	✔
Run | run_type	Daily | daily
Time | run_time	Hours 00
Conditional | conditional	☐

Run this script \| script 🖫

```javascript
1    var twitterCall = new TwitterScriptInclude();
2    var jsonObject = twitterCall.getTweets();
3    var getSocialProfile;
4
5
6    var channelGr = new GlideRecord('sn_app_cs_social_social_channel');
7    channelGr.addQuery('name', 'Twitter');
8    channelGr.query();
9    channelGr.next();
10   var channelUniqueID = channelGr.getUniqueValue();
11
12   jsonObject.statuses.forEach(function(key){
13       var gr = new GlideRecord('sn_app_cs_social_social_log');
14       var url = 'https://www.twitter.com/statuses/' + key.id_str;
15       var currDate = new GlideTime(Date.parse(key.created_at));
16       if(!gr.get('external_id', key.id_str)){
17           gr.initialize();
18           gr.setValue('external_id', key.id_str);
19           gr.setValue('message', key.text);
20           gr.setValue('social_url', url);
21           if(!gr.social_profile){
22               var grProfile = new GlideRecord('sn_app_cs_social_social_profile');
23               if(!grProfile.get('social_id', key.user.screen_name)){
24                   grProfile.initialize();
25                   grProfile.setValue('channel' , channelUniqueID );
26                   grProfile.setValue('social_id', key.user.screen_name);
27                   grProfile.insert();
```

2. Check the Social Log List to see the new records created.