Introducing PoshTweet - The PowerShell Twitter Script Library

Joe Pruitt, 2008-30-12

It's probably no surprise from those of you that follow my blog and tech tips here on DevCentral that I'm a fan of Windows PowerShell. I've written a set of Cmdlets that allow you to manage and control your BIG-IP application delivery controllers from within PowerShell and a whole set of articles around those Cmdlets.

I've been a Twitter user for a few years now and over the holidays, I've noticed that Jeffrey Snover from the PowerShell team has hopped aboard the Twitter bandwagon and that got me to thinking...

Since I live so much of my time in the PowerShell command prompt, wouldn't it be great to be able to tweet from there too? Of course it would!

HTTP Requests

So, last night I went ahead and whipped up a first draft of a set of PowerShell functions that allow access to the Twitter services. I implemented the functions based on Twitter's REST based methods so all that was really needed to get things going was to implement the HTTP GET and POST requests needed for the different API methods. Here's what I came up with.

```
function Execute-HTTPGetCommand()
{
  param([string] $url = $null);
  if ( $url )
  {
    [System.Net.WebClient]$webClient = New-Object System.Net.WebClient
    $webClient.Credentials = Get-TwitterCredentials
    [System.IO.Stream]$stream = $webClient.OpenRead($url);
    [System.IO.StreamReader]$sr = New-Object System.IO.StreamReader -argumentList $stream;
    [string]$results = $sr.ReadToEnd();
    $results;
  }
}
function Execute-HTTPPostCommand()
{
  param([string] $url = $null, [string] $data = $null);
```

```
  if ( $url -and $data )
  {
    [System.Net.WebRequest]$webRequest = [System.Net.WebRequest]::Create($url);
    $webRequest.Credentials = Get-TwitterCredentials
    $webRequest.PreAuthenticate = $true;
    $webRequest.ContentType = "application/x-www-form-urlencoded";
    $webRequest.Method = "POST";
    $webRequest.Headers.Add("X-Twitter-Client", "PoshTweet");
    $webRequest.Headers.Add("X-Twitter-Version", "1.0");
    $webRequest.Headers.Add("X-Twitter-URL", "http://devcentral.f5.com/poshtweet");
    [byte[]]$bytes = [System.Text.Encoding]::UTF8.GetBytes($data);
    $webRequest.ContentLength = $bytes.Length;
    [System.IO.Stream]$reqStream = $webRequest.GetRequestStream();
    $reqStream.Write($bytes, 0, $bytes.Length);
    $reqStream.Flush();
    [System.Net.WebResponse]$resp = $webRequest.GetResponse();
    $rs = $resp.GetResponseStream();
    [System.IO.StreamReader]$sr = New-Object System.IO.StreamReader -argumentList $rs;
    [string]$results = $sr.ReadToEnd();
    $results;
  }
}
```

Credentials

Once those were completed, it was relatively simple to get the Status methods for public_timeline, friends_timeline, user_timeline, show, update, replies, and destroy going. But, for several of those services, user credentials were required. I opted to store them in a script scoped variable and provided a few functions to get/set the username/password for Twitter.

```
$script:g_creds = $null;
function Set-TwitterCredentials()
{
  param([string]$user = $null, [string]$pass = $null);
  if ( $user -and $pass )
  {
    $script:g_creds = New-Object System.Net.NetworkCredential -argumentList ($user, $pass);
  }
  else
  {
    $creds = Get-TwitterCredentials;
  }
}
```

```
function Get-TwitterCredentials()
{
 if ( $null -eq $g_creds )
 {
  trap { Write-Error "ERROR: You must enter your Twitter credentials for PoshTweet to work!"; continue; }
  $c = Get-Credential
  if ( $c )
  {
   $user = $c.GetNetworkCredential().Username;
   $pass = $c.GetNetworkCredential().Password;
   $script:g_creds = New-Object System.Net.NetworkCredential -argumentList ($user, $pass);
  }
 }
 $script:g_creds;
}
```

The Status functions

Now that the credentials were out of the way, it was time to tackle the Status methods. These methods are a combination of HTTP GETs and POSTs that return an array of status entries. For those interested in the raw underlying XML that's returned, I've included the $raw parameter, that when set to $true, will not do a user friendly display, but will dump the full XML response. This would be handy, if you want to customize the output beyond what I've done.

```
#--------------------------------------------------------------------------
# public_timeline
#--------------------------------------------------------------------------
function Get-TwitterPublicTimeline()
{
  param([bool]$raw = $false);
  $results = Execute-HTTPGetCommand "http://twitter.com/statuses/public_timeline.xml";
  Process-TwitterStatus $results $raw;
}
#--------------------------------------------------------------------------
# friends_timeline
#--------------------------------------------------------------------------
function Get-TwitterFriendsTimeline()
{
 param([bool]$raw = $false);
 $results = Execute-HTTPGetCommand "http://twitter.com/statuses/friends_timeline.xml";
 Process-TwitterStatus $results $raw
}

#--------------------------------------------------------------------------
#user_timeline
#--------------------------------------------------------------------------
function Get-TwitterUserTimeline()
{
 param([string]$username = $null, [bool]$raw = $false);
 if ( $username )
 {
  $username = "/$username";
 }
 $results = Execute-HTTPGetCommand "http://twitter.com/statuses/user_timeline$username.xml";
 Process-TwitterStatus $results $raw
}
```

```
#-----------------------------------------------------------------------
# show
#-----------------------------------------------------------------------
function Get-TwitterStatus()
{
  param([string]$id, [bool]$raw = $false);
  if ( $id )
  {
    $results = Execute-HTTPGetCommand "http://twitter.com/statuses/show/" + $id + ".xml";
    Process-TwitterStatus $results $raw;
  }
}

#-----------------------------------------------------------------------
# update
#-----------------------------------------------------------------------
function Set-TwitterStatus()
{
  param([string]$status);
  $encstatus = [System.Web.HttpUtility]::UrlEncode("$status");
  $results = Execute-HTTPPostCommand "http://twitter.com/statuses/update.xml" "status=$encstatus";
  Process-TwitterStatus $results $raw;
}

#-----------------------------------------------------------------------
# replies
#-----------------------------------------------------------------------
function Get-TwitterReplies()
{
  param([bool]$raw = $false);
  $results = Execute-HTTPGetCommand "http://twitter.com/statuses/replies.xml";
  Process-TwitterStatus $results $raw;
}

#-----------------------------------------------------------------------
# destroy
#-----------------------------------------------------------------------
function Destroy-TwitterStatus()
{
  param([string]$id = $null);
  if ( $id )
  {
    Execute-HTTPPostCommand "http://twitter.com/statuses/destroy/$id.xml", "id=$id";
  }
}
```

You may notice the Process-TwitterStatus function. Since there was a lot of duplicate code in each of these functions, I went ahead and implemented it in it's own function below:

```
function Process-TwitterStatus()
{
  param([string]$sxml = $null, [bool]$raw = $false);
  if ( $sxml )
  {
    if ( $raw )
    {
      $sxml;
    }
```

```
    else
    {
      [xml]$xml = $sxml;
      if ( $xml.statuses.status )
      {
        $stats = $xml.statuses.status;
      }
      elseif ($xml.status )
      {
        $stats = $xml.status;
      }
      $stats | Foreach-Object -process {
        $info = "by " + $_.user.screen_name + ", " + $_.created_at;
        if ( $_.source ) { $info = $info + " via " + $_.source; }
        if ( $_.in_reply_to_screen_name ) { $info = $info + " in reply to " + $_.in_reply_to_screen_n
        "-------------------------";
        $_.text;
        $info;
      };
      "-------------------------";
    }
  }
}
```

A few hurdles

Nothing goes without a hitch and I found myself pounding my head at why my POST commands were all getting HTTP 417 errors back from Twitter. A quick search brought up this post on Phil Haack's website as well as this Google Group discussing an update in Twitter's services in how they react to the Expect 100 HTTP header. A simple setting in the ServicePointManager at the top of the script was all that was needed to get things working again.

```
[System.Net.ServicePointManager]::Expect100Continue = $false;
```

PoshTweet in Action

So, now it's time to try it out. First you'll need to . source the script and then set your Twitter credentials. This can be done in your Twitter $profile file if you wish. Then you can access all of the included functions. Below, I'll call Set-TwitterStatus to update my current status and then Get-TwitterUserTimeline and Get-TwitterFriendsTimeline to get my current timeline as well as that of my friends.

```
PS> . .\PoshTweet.ps1
PS> Set-TwitterCredentials
PS> Set-TwitterStatus "Hacking away with PoshTweet"
PS> Get-TwitterUserTimeline
-------------------------
Hacking away with PoshTweet
by joepruitt, Tue Dec 30, 12:33:04 +0000 2008 via web
-------------------------
PS> Get-TwitterFriendsTimeline
-------------------------
@astrout Yay, thanks!
by mediaphyter, Tue Dec 30 20:37:15 +0000 2008 via web in reply to astrout
-------------------------
RT @robconery: Headed to a Portland Nerd Dinner tonite - should be fun! http://bit.ly/EUFC
by shanselman, Tue Dec 30 20:37:07 +0000 2008 via TweetDeck
-------------------------
...
```

Things Left Todo

As I said, this was implemented in an hour or so last night so it definitely needs some more work, but I believe I've got the Status methods pretty much covered. Next I'll move on to the other services of User, Direct Message, Friendship, Account, Favorite, Notification, Block, and Help when I've got time.

I'd also like to add support for the "source" field. I'll need to setup a landing page for this library that is public facing so the folks at Twitter will add it to their system. Once I get all the services implemented, I'll more forward in formalizing this as an application and submit it for consideration.

Collaboration

I've posted the source to this set of functions on the DevCentral wiki under PsTwitterApi. You'll need to create an account to get to it, but I promise it will be worth it! Feel free to contribute and add to if you have the time. Everyone is welcome and encouraged to tear my code apart, optimize it, enhance it. Just as long as it get's better in the process. B-).

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com