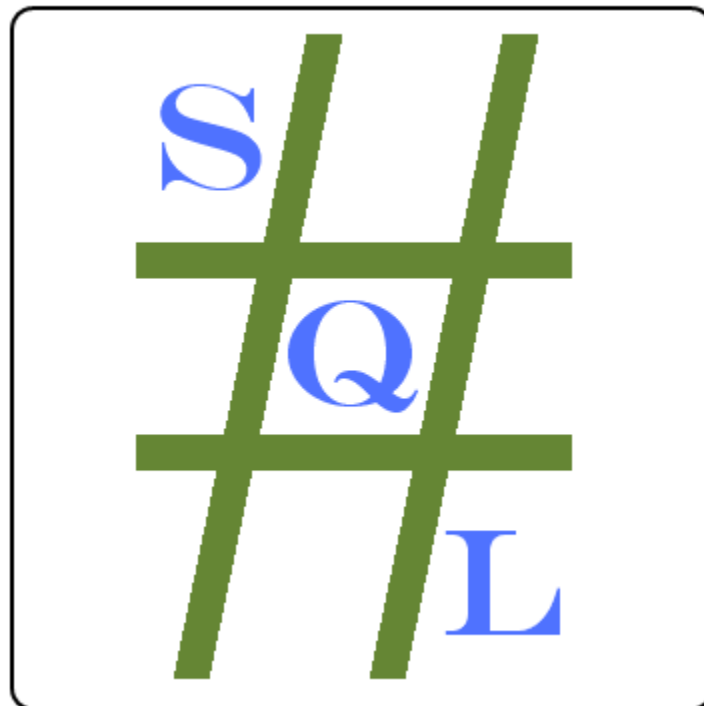


SQL#

EXPANDING THE CAPABILITIES OF T-SQL



HOW TO SET UP TWITTER FOR SQL#

March 13th, 2011

Now that Twitter (as of August 31st, 2010) requires OAuth authentication, applications can no longer pass in Username's and Password. The goal of OAuth is to increase security by having the user's login credentials only known to the main site – Twitter in this case. In order to communicate with Twitter now you must create a Twitter Application which will give you a ConsumerKey and ConsumerSecret to identify your Application by. Twitter Users now have credentials specific to each application known as the AccessToken (similar to Username) and AccessTokenSecret (similar to Password). The Twitter User that creates an Application will automatically have their AccessToken and AccessTokenSecret generated and available in the Twitter Application Details page. All four values – ConsumerKey, ConsumerSecret, AccessToken, and AccessTokenSecret – need to be sent to Twitter in place of what used to be the Username and Password values.

Most users of SQL# Twitter functions only access Twitter for their own account. In this case the above information is all you need to know. However, some people need to send in their customers' UserNames and Passwords and this no longer works as described above since the AccessToken and AccessTokenSecret are only for the User that created the Application. In this case you need to use xAuth which will allow you to translate other peoples' UserNames and Passwords for your ConsumerKey and ConsumerSecret into the AccessToken and AccessTokenSecret for your Application. While SQL# does have an xAuth function, be aware that Twitter grants xAuth access on a per-Application basis and is very careful regarding whom they give xAuth access to. If you need this level of access, please contact Twitter. According to the Twitter xAuth page:

xAuth provides a way for desktop and mobile applications to exchange a username and password for an OAuth access token. Once the access token is retrieved, xAuth-enabled developers should dispose of the login and password corresponding to the user.

xAuth access is restricted to approved applications. If your application is a desktop or mobile application and the standard web OAuth flow or PIN-code out-of-band flow is not right for you, send a detailed message to api@twitter.com to request xAuth privileges. Include the name of your application, the consumer key, the application ID (if available), and a summary of how xAuth is best-suited for your application.

If you do request xAuth access from Twitter, be sure to explain that you are using SQL# which runs inside of a database call that has no possibility of user interaction.



Setting up your Twitter Application:

- 1) Go to:
<http://dev.twitter.com/>
- 2) If you are not logged in, use the “Sign in” link at the top:
<http://dev.twitter.com/login>
- 3) Click on the “Your apps” link at the top:
<http://dev.twitter.com/apps>
- 4) Click on the “Register a new app” button on the right:
<http://dev.twitter.com/apps/new>
- 5) Fill in:
 - a. Application Name: This is what shows up when posting updates just underneath the text of the update, as in:
Date and Time via Application Name
This name needs to be unique within Twitter and if it is not you will be notified when you try to save the Application.
 - b. Description: You need to have at least 10 characters here.
 - c. Application Website: This cannot be blank
 - d. Application Type: Set this to “Client”
 - e. Default Access Type: Set this to “Read & Write”
- 6) Fill in the captcha words and Click the “Register Application” at the bottom. NOTE: If there is ANY error on your form, you will need to re-set the value of the “Application Type” back to “Client” as it will be set to “Browser” for some reason.
- 7) You are now directed to your application page with the URL being:
http://dev.twitter.com/apps/{your_application_id}
If you need to request xAuth access, this is the ApplicationID they will want.
- 8) Scroll down to the “OAuth 1.0a Settings” section. The first two items are the “ConsumerKey” and “ConsumerSecret”. These two values identify your application and are needed for ALL SQL# Twitter functions (whether you use xAuth or not).
- 9) Scroll up and click on the “My Access Token” button on the right. The two values shown here – “Access Token (oauth_token)” and “Access Token Secret



(oath_token_secret)” – are the other two values you need for the SQL# Twitter functions (for most users).

SQL# Twitter Examples:

- In either case, run once: `EXEC SQL#.SQLsharp_SetSecurity 2, 'SQL#.Twitterizer'`
- Typical Usage:

```
DECLARE      @ConsumerKey NVARCHAR(100),
             @ConsumerSecret NVARCHAR(100),
             @AccessToken NVARCHAR(100),
             @AccessTokenSecret NVARCHAR(100)

SELECT      @ConsumerKey = 'aaaaaaaaaaaa',
            @ConsumerSecret = 'bbbbbbbbbbbbbb',
            @AccessToken = '9999999-cccccccccccc',
            @AccessTokenSecret = 'ddddddddddddddddd'

DECLARE @StatusID BIGINT

SET @StatusID = SQL#.Twitter_Update(@ConsumerKey, @ConsumerSecret,
@AccessToken, @AccessTokenSecret, 'test!!!!!!', NULL, NULL, NULL)

SELECT @StatusID
```

- Usage with xAuth:

```
DECLARE      @ConsumerKey NVARCHAR(100),
             @ConsumerSecret NVARCHAR(100),
             @AccessToken NVARCHAR(100),
             @AccessTokenSecret NVARCHAR(100),
             @UserName NVARCHAR(100),
             @Password NVARCHAR(100)

SELECT      @ConsumerKey = 'aaaaaaaaaaaa',
            @ConsumerSecret = 'bbbbbbbbbbbbbb'

SELECT      @UserName = 'myusername',
            @Password = 'mypassword'

SELECT      @AccessToken = xauth.AccessToken,
            @AccessTokenSecret = xauth.AccessTokenSecret
FROM        SQL#.Twitter_xAuth(@ConsumerKey, @ConsumerSecret, @UserName,
@Password) xauth

DECLARE @StatusID BIGINT

SET @StatusID = SQL#.Twitter_Update(@ConsumerKey, @ConsumerSecret,
@AccessToken, @AccessTokenSecret, 'test again!!!!!!', null, null, null)

SELECT @StatusID
```

