NOKIA Developer

Nokia TweetTalk application for Windows Phone

This article explains how to make use of Twitter API service and get Tweets and other information from twitter in Windows Phone 7 application.

Introduction

This article shows how to use Twitter API in Windows Phone application which is demonstrated by creating a **Nokia TweetTalk** application which includes all the Twitter Accounts of Nokia related Services like Nokia Twitter, Nokia Developer Twitter, Nokia Care Twitter accounts etc. It has also added "Champions" section where one can get the tweets from Champions and Nokia Experts around the world.

How to make use of Twitter API in Windows Phone

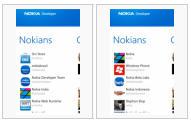
- To make Twitter client we obviously need to use Twitter API to get various information and tweets from Twitter account. Here is the Twitter API site contains various API. (https://dev.twitter.com/docs/api/)
- Here is the nice article from Kipp and Wildbug (Following how to build simple windows phone 7 twitter applications to search Twitter accounts. It also shows how to create Hello world application for Windows phone.

About NOKIA TweetTalk Application

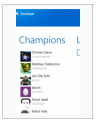
Nokia TweetTalk application contains two sections. First Section "Nokians" contains list of various Twitter accounts of Nokia services like Nokia, Nokia developers, Ovi Store, Nokia Care etc. Second section "Champions" contains list of Nokia champions and Nokia Experts Twitter account. This application is developed specially for Nokia Developers and Nokia Fans to get latest information about NOKIA and Nokia development.

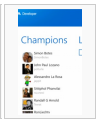
Screenshots

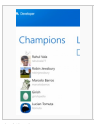
By Screenshots of the application, you can get better idea about this Application.

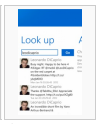

List of Nokia Account

List of Nokia Account


List of Champions Account


List of Champions Account


List of Champions Account


Lookup

Info of Nokiadev Twitter account

Tweets of NokiaDev

Tweets of Windows phone

How to develop

Here we split the code in two section

1. How to List the Twitter accounts with Live Info
2. How to Show live tweets of selected Twitter account

How to List the Twitter accounts with Live Info

First we need to store information coming from twitter API, so we have created a call TwitterItem.cs and created some properties in that class related to Twitter account like below.

```
public class TwitterItems
{
        public string username { get; set; }
        public string tweet { get; set; }
        public string imageURL { get; set; }
        public string datetime { get; set; }

        public string name { get; set; }
        public string location { get; set; }
        public string description { get; set; }
        public string followers_count { get; set; }
        public string url { get; set; }
        public string kincedatetime { get; set; }
}
```

Now We need to list the twitter account that we want to include in our application. To get twitter information we will use various twitter API that returns output data in XML format and will use `WebClient` networking class to asynchronously download a Twitter feed. This takes three lines of code in total:

1. declaring and creating the `WebClient`.

```
WebClient twittername = new WebClient();
```

2. attaching an event handler and then

```
twittername.DownloadStringCompleted += new
DownloadStringCompletedEventHandler(twittername_DownloadStringCompleted);
```

3. calling the asynchronous `DownloadStringAsync` method.

```
twittername.DownloadStringAsync(new
Uri("https://api.twitter.com/1/users/lookup.xml?
screen_name=twitteraccountID1,twitteraccountID2,twitteraccountID3,twitteraccountID4&include_entities=true"));
```

In the `DownloadStringAsync` call, we'll pass a Twitter Url with twitter username as seen above. This feed will pull down the respective user various information like profile picture, Name, follower etc. in an XML format. When the call completes, the `DownloadStringCompleted` event is fired and our generated event handler `twitter_DownloadStringCompleted` will be called.

Here is the full source code that explained above step by step.

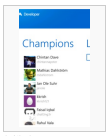
```
private void LoadNokiaTwitterAccount()
{
        WebClient twittername = new WebClient();
        twittername.DownloadStringCompleted += new
DownloadStringCompletedEventHandler(twittername_DownloadStringCompleted);
        twittername.DownloadStringAsync(new
Uri("https://api.twitter.com/1/users/lookup.xml?
screen_name=nokiabrasil,NokiaMexico,NokiaUK,NokiaIndonesia,NokiaPoland,nokiacanada,NokiaNSN,NokiaEastAfrica,NokiaIndia,NokiaUS,Nokia_UK,NokiaPush,thenokiablog,wtnokia,Nseries,NokiaCareers,nokiaaccess,Nokia_Connects,NokiaInnovation,NokiaHelps,Nokia_Presse,nokcnv,mydailyapp,nokiabetalabs,ideasproject,ovibynokia,OviStore,qtbynokia,Symbian,MeetbExperts,selup,windowsphone,nokiadeveloper,nokia,&include_entities=true"));

        progressBar1.IsIndeterminate = true;
        progressBar1.Visibility = Visibility.Visible;
}

void twitter_DownloadStringCompleted(object sender, DownloadStringCompletedEventArgs e)
{
        if (e.Error != null)
            return;

        XElement xmlTweets = XElement.Parse(e.Result);

        listBox1.ItemSource = from tweet in xmlTweets.Descendants("status")
                              select new TwitterItems
                              {
                                  imageURL =
tweet.Element("user").Element("profile_image_url").Value,
                                  tweets = tweet.Element("text").Value,
                                  username =
tweet.Element("user").Element("name").Value,
                                  datetime =
Convert.ToString(tweet.Element("created_at").Value).Remove(20, 5)
                              };

        progressBar1.IsIndeterminate = true;
        progressBar1.Visibility = Visibility.Visible;
}
```

To list the twitter account, we bind the twitter information in List box control and we format list box control such way that we can display twitter accounts Profile image on left, Username and Full name on right as shown in screen shot.

```
<ListBox Tag="ListBox_Tag" Name="listbox2" Height="468" Width="388">
        <ListBox.ItemTemplate>
            <DataTemplate>
                <StackPanel Orientation="Horizontal">
                    <Image Source="{Binding imageURL}" Height="79" Width="79"
VerticalAlignment="Top" Margin="8,10,8,8" />
                    <StackPanel Width="288">
                        <TextBlock Text="{Binding username}" Foreground="Gray"
FontSize="25" />
                        <TextBlock Text="{Binding tweets}" Foreground="Black"
TextWrapping="Wrap" FontSize="21" />
                    </StackPanel>
                </StackPanel>
            </DataTemplate>
        </ListBox.ItemTemplate>
 </ListBox>
```

Output Screenshot

List of Nokia Account


List of Champions Account

How to show live tweets of selected Twitter account

On `Tap` event of ListBox, we will redirect the navigation to the second page using `NavigationService.Navigate` Method in input URI parameter. To get the selected twitter id in second page we have store on selected id in `IsolatedStorageSettings`. So we have created `AppSetting` Class for that.

```
using System;
using System.IO.IsolatedStorage;
using System.Diagnostics;

namespace NokiaTweetTalk
{
public class AppSettings
{
        // Our isolated storage settings
        IsolatedStorageSettings isolatedStore;

        // The isolated storage key names of our settings
        const string DisplayNameSettingKeyName = "DisplayNameSetting";
        const string ScreenNameSettingKeyName = "ScreenNameSetting";

        // The default value of our settings
        const string DisplayNameSettingDefault = "";
        const string ScreenNameSettingDefault = "";

        /// <summary>
        /// Constructor that gets the application settings.
        /// </summary>
        public AppSettings()
        {
            try
            {
                // Get the settings for this application.
                isolatedStore = IsolatedStorageSettings.ApplicationSettings;

            }
            catch (Exception e)
            {
                Debug.WriteLine("Exception while using IsolatedStorageSettings: " +
e.ToString());
            }
        }

        /// <summary>
        /// Update a setting value for our application. If the setting does not
        /// exist, then add the setting.
```


(C) Copyright Nokia 2014. All rights reserved.

```
        /// </summary>
        /// <param name="key"></param>
        /// <param name="value"></param>
        /// <returns></returns>
        public bool AddOrUpdateValue(string Key, Object value)
        {
            bool valueChanged = false;

            // If the key exists
            if (isolatedStore.Contains(Key))
            {
                // If the value has changed
                if (isolatedStore[Key] != value)
                {
                    // Store the new value
                    isolatedStore[Key] = value;
                    valueChanged = true;
                }
            }
            // Otherwise create the key.
            else
            {
                isolatedStore.Add(Key, value);
                valueChanged = true;
            }

            return valueChanged;
        }


/// <summary>
/// Get the current value of the setting, or if it is not found, set the
/// setting to the default setting.
/// </summary>
/// <typeparam name="valueType"></typeparam>
/// <param name="Key"></param>
/// <param name="defaultValue"></param>
/// <returns></returns>
 public valueType GetValueOrDefault<valueType>(string Key, valueType defaultValue)
 {
            valueType value;

            // If the key exists, retrieve the value.
            if (isolatedStore.Contains(Key))
            {
                value = (valueType)isolatedStore[Key];
            }
            // Otherwise, use the default value.
            else
            {
                value = defaultValue;
            }

            return value;
 }


        /// <summary>
        /// Save the settings.
        /// </summary>
        public void Save()
        {
            isolatedStore.Save();
        }


        /// <summary>
        /// Property to get and set a DisplayName Setting Key.
        /// </summary>
        public string DisplayNameSetting
        {
            get
            {
                return GetValueOrDefault<string>(DisplayNameSettingKeyName,
DisplayNameSettingDefault);
            }
            set
            {
                AddOrUpdateValue(DisplayNameSettingKeyName, value);
                Save();
            }
        }

        /// <summary>
        /// Property to get and set a ScreenName Setting Key.
        /// </summary>
        public string ScreenNameSetting
        {
            get
            {
                return GetValueOrDefault<string>(ScreenNameSettingKeyName,
ScreenNameSettingDefault);
            }
            set
            {
                AddOrUpdateValue(ScreenNameSettingKeyName, value);
                Save();
            }
        }
    }
}
```

Here is the code that will run on ListBox's Tap Event; we are storing Username and Displayname of the selected twitter account on Listbox Tap event and then navigate focus to the second page where we are going to list the tweets.

```
private void ListBox2_Tap(object sender, GestureEventArgs e)
{
            try
            {
                ListBoxItem lst = 
this.listbox3.ItemContainerGenerator.ContainerFromItem(this.listbox3.SelectedItem) as
ListBoxItem;
                TwitterItems attemViewModel = this.listbox3.SelectedItem as
TwitterItems;
                string acc = attemViewModel.Username;
                string acc = attemViewModel.Tweets;
                AppSettings settings = new AppSettings();
                settings.DisplayNameSetting = acc;
                settings.ScreenNameSetting = acc;
                NavigationService.Navigate(new Uri("/PanoramaPage1.xaml",
UriKind.Relative));
            }
            catch (Exception ex)
            {
                string err = ex.Message.ToString();
            }
}
```

On second page, first we need get the selected twitter id that we stored in `AppSetting`. To get the value from `AppSetting` we need to create the object and get the value by `settings.DisplayNameSetting` that contains Name and `settings.ScreenNameSetting` Username of twitter account.

After get those value on second page, we use it in page's title and pass it to the other function `BindTweet` as input parameter which fetch the tweets from particular twitter id.

```
 public PanoramaPage1()
{
            InitializeComponent();

    // Start the Progress bar to show loading.

            progressBar1.IsIndeterminate = true;
            progressBar1.Visibility = Visibility.Visible;

            string DisplayName = settings.DisplayNameSetting;
            string ScreenName = settings.ScreenNameSetting;
            if (DisplayName.Length <= 12)
                PageTitle.Header = DisplayName;
            else
                PageTitle.Header = ScreenName;

            BindTweet(DisplayName, ScreenName);

    // Stop Progress bar to show loading.
            progressBar1.IsIndeterminate = false;
            progressBar1.Visibility = Visibility.Collapsed;
 }

// Fetch the tweets and bind it to Listbox
  private void BindTweet(string DisplayName, string ScreenName)
{
            //progressBar1.IsIndeterminate = true;
            //progressBar1.Visibility = Visibility.Visible;

            WebClient twitter = new WebClient();

            twitter.DownloadStringCompleted += new
DownloadStringCompletedEventHandler(twitter_DownloadStringCompleted);
            twitter.DownloadStringAsync(new
Uri("http://api.twitter.com/1/statuses/user_timeline.xml?screen_name=" + ScreenName +
"&count=100"));

            WebClient twitterInfo = new WebClient();

            twitterInfo.DownloadStringCompleted += new
DownloadStringCompletedEventHandler(twitterInfo_DownloadStringCompleted);
            twitterInfo.DownloadStringAsync(new
Uri("https://api.twitter.com/1/user/show.xml?screen_name=" + ScreenName +
"&include_entities=true"));

 }

 void twitter_DownloadStringCompleted(object sender, DownloadStringCompletedEventArgs e)
 {
            if (e.Error != null)
                return;

            XElement xmlTweets = XElement.Parse(e.Result);

            listBox2.ItemsSource = from tweet in xmlTweets.Descendants("status")
                                   select new TwitterItems
                                   {
                                       imageURL =
Tweet.Element("user").Element("profile_image_url").Value,
                                       tweets = tweet.Element("text").Value,
                                       username =
Tweet.Element("user").Element("name").Value,
                                       datetime =
Convert.ToString(Tweet.Element("created_at").Value).Remove(20, 5)
                                   };
}
```

To show the information about the selected twitter id we use a API function of Twitter that gives only information of twitter account. https://api.twitter.com/1/users/show.xml?screen_name=name

```
void twitterInfo_DownloadStringCompleted(object sender, DownloadStringCompletedEventArgs
e)
{
            if (e.Error != null)
                return;

            XDocument xmlTweets = XDocument.Parse(e.Result);
            XElement xmlChildf = xmlTweets.Root;

            List<TwitterItems> lst = new List<TwitterItems>();

            TwitterItems items = new TwitterItems();
            items.imageURL = xmlChildf.Element("profile_image_url").Value;
            items.name = xmlChildf.Element("name").Value;
            items.username = "@" + xmlChildf.Element("screen_name").Value;
            if(xmlChildf.Element("location").Value != "")
                items.location = "Location: " + xmlChildf.Element("location").Value;
            items.url = xmlChildf.Element("url").Value;
            items.followers_count = "Followers Count: " +
xmlChildf.Element("followers_count").Value;
            items.SinceDatetime = "Joining Date: " +
Convert.ToString(xmlChildf.Element("created_at").Value).Remove(20, 5);

            if (xmlChildf.Element("description").Value != "")
                items.description = "Description: " +
xmlChildf.Element("description").Value;

            lst.Add(items);
            lstInfo.ItemsSource = lst;
}
```

Output Screenshot

Info of nokiadeveloper twitter account

tweets from nokiadeveloper twitter account

Installation File

If your phone is unlocked or in other word if it is developer device, you can install .xap file in your phone. To register your phone for development follow steps written in MSDN article on below link.

How to Register Your Phone for Development

You can find the installation file of Nokia TweetTalk Application on below link,

Media:NokiaTweetTalk.zip

Source Code - Nokia Project

Source code of this application is attached in below link. You can download source code zip file from below link.

File:NewsTweetTalk.zip

Celebrity TweetTalk - Replica Application

Celebrity TweetTalk Application which is replica of Nokia TweetTalk is now available for download on Windows Phone Marketplace on below link.

Windows Phone Marketplace

List of Celebrity

List of Celebrity

Selected Celebrity Tweet

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.