

// Fig. 28.4: FickrViewerForm.cs

// Invoking a web service asynchronously with class WebClient

using System;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Net;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Xml.Linq;

namespace FlickrViewer

{

public partial class FlickrViewerForm : Form

{

// Use your Flickr API key here--you can get one at:

// http://www.flickr.com/services/apps/create/apply

private const string KEY = "b30227ff47eb8cc60a4fc0b1f2a40516";

// object used to invoke Flickr web service

private WebClient flickrClient = new WebClient();

Task<string> flickrTask = null; // Task<string> that queries Flickr

public FlickrViewerForm()

{

InitializeComponent();

} // end constructor

// initiate asynchronous Flickr search query;

// display results when query completes

private async void searchButton\_Click( object sender, EventArgs e )

{

// if flickrTask already running, prompt user

if ( flickrTask != null &&

flickrTask.Status != TaskStatus.RanToCompletion )

{

var result = MessageBox.Show(

"Cancel the current Flickr search?",

"Are you sure?", MessageBoxButtons.YesNo,

MessageBoxIcon.Question );

// determine whether user wants to cancel prior search

if ( result == DialogResult.No )

return;

else

flickrClient.CancelAsync(); // cancel current search

} // end if

// Flickr's web service URL for searches

var flickrURL = "https://api.flickr.com/services/rest/?method=" +

$"flickr.photos.search&api\_key={KEY}&tags={inputTextBox.Text.Replace(" ", ",")}" +

$"&tag\_mode=all&per\_page=500&privacy\_filter=1";

imagesListBox.DataSource = null; // remove prior data source

imagesListBox.Items.Clear(); // clear imagesListBox

pictureBox.Image = null; // clear pictureBox

imagesListBox.Items.Add( "Loading..." ); // display Loading...

try

{

// invoke Flickr web service to search Flick with user's tags

flickrTask =

flickrClient.DownloadStringTaskAsync( flickrURL );

// await flickrTask then parse results with XDocument and LINQ

XDocument flickrXML = XDocument.Parse( await flickrTask );

// gather information on all photos

var flickrPhotos =

from photo in flickrXML.Descendants( "photo" )

let id = photo.Attribute( "id" ).Value

let title = photo.Attribute( "title" ).Value

let secret = photo.Attribute( "secret" ).Value

let server = photo.Attribute( "server" ).Value

let farm = photo.Attribute( "farm" ).Value

select new FlickrResult

{

Title = title,

URL = string.Format(

"http://farm{0}.staticflickr.com/{1}/{2}\_{3}.jpg",

farm, server, id, secret )

};

imagesListBox.Items.Clear(); // clear imagesListBox

// set ListBox properties only if results were found

if ( flickrPhotos.Any() )

{

imagesListBox.DataSource = flickrPhotos.ToList();

imagesListBox.DisplayMember = "Title";

} // end if

else // no matches were found

imagesListBox.Items.Add( "No matches" );

} // end try

catch ( WebException )

{

// check whether Task failed

if ( flickrTask.Status == TaskStatus.Faulted )

MessageBox.Show( "Unable to get results from Flickr",

"Flickr Error", MessageBoxButtons.OK,

MessageBoxIcon.Error );

imagesListBox.Items.Clear(); // clear imagesListBox

imagesListBox.Items.Add( "Error occurred" );

} // end catch

} // end method searchButton\_Click

// display selected image

private async void imagesListBox\_SelectedIndexChanged(

object sender, EventArgs e )

{

if ( imagesListBox.SelectedItem != null )

{

string selectedURL =

( ( FlickrResult ) imagesListBox.SelectedItem ).URL;

// use WebClient to get selected image's bytes asynchronously

WebClient imageClient = new WebClient();

byte[] imageBytes = await imageClient.DownloadDataTaskAsync(

selectedURL );

// display downloaded image in pictureBox

MemoryStream memoryStream = new MemoryStream( imageBytes );

pictureBox.Image = Image.FromStream( memoryStream );

} // end if

} // end method imagesListBox\_SelectedIndexChanged

} // end class FlickrViewerForm

} // end namespace FlickrViewer

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* (C) Copyright 1992-2014 by Deitel & Associates, Inc. and \*

\* Pearson Education, Inc. All Rights Reserved. \*

\* \*

\* DISCLAIMER: The authors and publisher of this book have used their \*

\* best efforts in preparing the book. These efforts include the \*

\* development, research, and testing of the theories and programs \*

\* to determine their effectiveness. The authors and publisher make \*

\* no warranty of any kind, expressed or implied, with regard to these \*

\* programs or to the documentation contained in these books. The authors \*

\* and publisher shall not be liable in any event for incidental or \*

\* consequential damages in connection with, or arising out of, the \*

\* furnishing, performance, or use of these programs. \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/