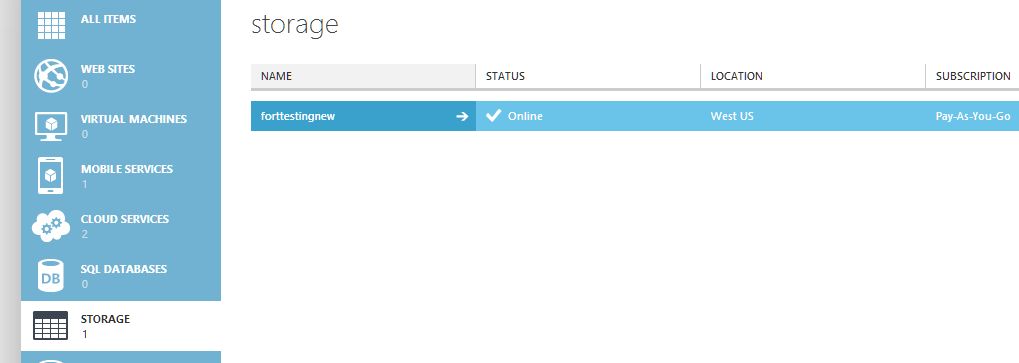
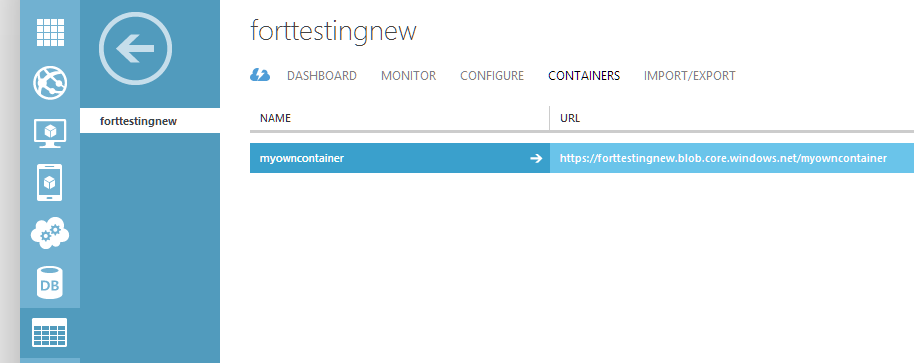
**Upload Images into Blob using Windows 8 App**

Step 1: Create storage.

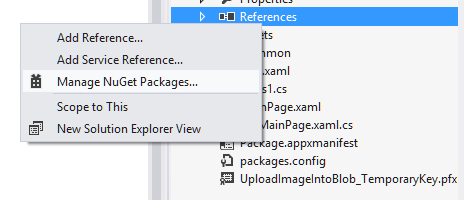


Step 2: Create Container. Set to Public Container.

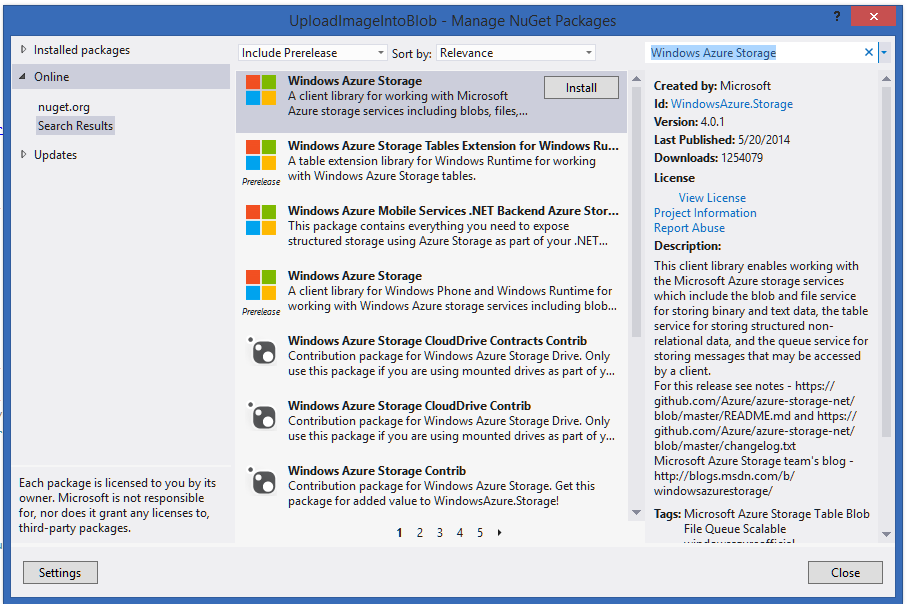


Step 3: Start Visual Studio Create Windows 8 New XAML/ C# App.

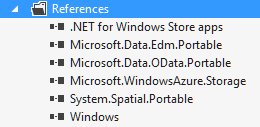
Right click on “References” & select “Manage NuGet Packages…”



Enter the term for search “Windows Azure Storage”



Step 4: after installation Windows Azure Storage Package or DLL added to the App.



Step 5: Open MainPage.xaml file

<Grid Background="{StaticResource ApplicationPageBackgroundThemeBrush}">

<StackPanel>

<StackPanel Width="200" HorizontalAlignment="Left" Margin="100,100,0,0">

<TextBlock Text="Photo" Margin="0,20,0,0"/>

<StackPanel Orientation="Horizontal">

<Button x:Name="btnCamera" Content="Camera" Click="btnCamera\_Click\_1"/>

<Button x:Name="btnGallery" Content="Gallery" Click="btnGallery\_Click\_1" Margin="30,0,0,0"/>

</StackPanel>

</StackPanel>

</StackPanel>

</Grid>

Step 6: Then open MainPage.xaml.cs file

Add Windows Azure Storage reference at top of the page.

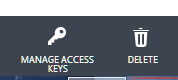
using Microsoft.WindowsAzure.Storage;

using Microsoft.WindowsAzure.Storage.Auth;

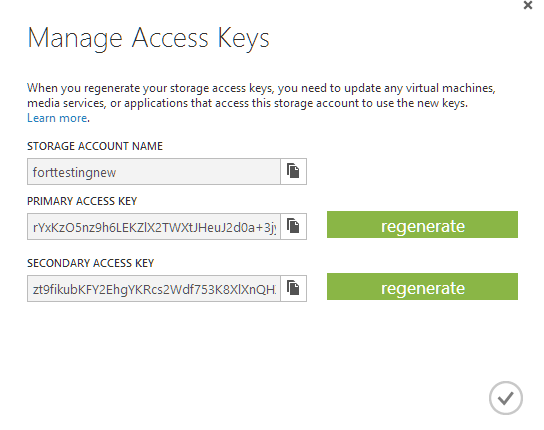
using Microsoft.WindowsAzure.Storage.Blob;

There are two options for user to upload the image from Gallery collection or from camera.

Open Windows Azure Management Portal and select storage. Click on the “Manage Access Key”



First copy storage name & then copy primary access key



Also copy the Container & copy into the Code.

All code are as below:

using Microsoft.WindowsAzure.Storage;

using Microsoft.WindowsAzure.Storage.Auth;

using Microsoft.WindowsAzure.Storage.Blob;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using Windows.Foundation;

using Windows.Foundation.Collections;

using Windows.Media.Capture;

using Windows.Storage;

using Windows.Storage.Pickers;

using Windows.Storage.Streams;

using Windows.UI.Xaml;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Controls.Primitives;

using Windows.UI.Xaml.Data;

using Windows.UI.Xaml.Input;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Media.Imaging;

using Windows.UI.Xaml.Navigation;

namespace UploadImageIntoBlob

{

/// <summary>

/// An empty page that can be used on its own or navigated to within a Frame.

/// </summary>

public sealed partial class MainPage : Page

{

public MainPage()

{

this.InitializeComponent();

}

/// <summary>

/// Invoked when this page is about to be displayed in a Frame.

/// </summary>

/// <param name="e">Event data that describes how this page was reached. The Parameter

/// property is typically used to configure the page.</param>

protected override void OnNavigatedTo(NavigationEventArgs e)

{

}

BitmapImage bitmapCamera = new BitmapImage();

String imageUri = "";

private async void btnCamera\_Click\_1(object sender, RoutedEventArgs e)

{

CameraCaptureUI cameraUI = new CameraCaptureUI();

Size aspectRatio = new Size(15, 10);

cameraUI.PhotoSettings.CroppedAspectRatio = aspectRatio;

cameraUI.PhotoSettings.MaxResolution = CameraCaptureUIMaxPhotoResolution.MediumXga;

//StorageFile capturedMedia;

StorageFile capturedMedia = await cameraUI.CaptureFileAsync(CameraCaptureUIMode.Photo);

// Retrieve storage account from connection string

CloudStorageAccount storageAccount = new CloudStorageAccount(

new StorageCredentials("forttestingnew",

"rYxKzO5nz9h6LEKZlX2TWXtJHeuJ2d0a+3jytW9QWX6JmBcgut1g8GhFAmvfV5SiEEHdKCI0nGG1IdCSTw9UVg=="), true);

// Create the blob client

CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();

// Retrieve reference to a previously created container

CloudBlobContainer container = blobClient.GetContainerReference("myowncontainer");

await container.CreateIfNotExistsAsync();

//// Retrieve reference to a blob named "myblob"

//CloudBlob blob = container.GetBlobReference("myblob");

var containercloudname = string.Format("picture-{0}", capturedMedia.Name.ToLowerInvariant());

CloudBlockBlob blob = null;

// Create or overwrite the "myblob" blob with contents from a local file

//foreach (StorageFile sFile in capturedMedia)

//{

using (var fileStream = await capturedMedia.OpenStreamForReadAsync())

{

blob = container.GetBlockBlobReference(containercloudname);

await blob.UploadFromStreamAsync(fileStream.AsInputStream());

}

}

private async void btnGallery\_Click\_1(object sender, RoutedEventArgs e)

{

FileOpenPicker openPicker = new FileOpenPicker();

openPicker.SuggestedStartLocation = PickerLocationId.PicturesLibrary;

openPicker.FileTypeFilter.Add(".jpg");

openPicker.FileTypeFilter.Add(".png");

openPicker.FileTypeFilter.Add(".jpeg");

StorageFile file = await openPicker.PickSingleFileAsync();

if (file != null)

{

using (var fileStream = await file.OpenSequentialReadAsync())

{

try

{

CloudStorageAccount storageAccount = new CloudStorageAccount(

new StorageCredentials("forttestingnew",

"rYxKzO5nz9h6LEKZlX2TWXtJHeuJ2d0a+3jytW9QWX6JmBcgut1g8GhFAmvfV5SiEEHdKCI0nGG1IdCSTw9UVg=="), true);

// Create the blob client

CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();

CloudBlobContainer container = blobClient.GetContainerReference("myowncontainer");

await container.CreateIfNotExistsAsync();

var blob = container.GetBlockBlobReference(file.Name);

await blob.DeleteIfExistsAsync();

await blob.UploadFromStreamAsync(fileStream);

//IbState.Text += DateTime.Now.ToString() + ": Save picture '" + file.Name + "' successfully!\n";

}

catch (Exception ex)

{

//IbState.Text += (ex.Message + "\n");

}

}

}

}

}

}

Then run the app. uploading the image open container in Windows Azure Management Portal.

