Session 1: Universal FlickrSearch Demo

## Preparation

1. In Visual Studio – Tools – Code Snippets Manager
   1. Select XAML in the Language dropdown
   2. Click Add…
   3. Navigate to <JumpStart sample code folder>\Code Snippets\XAML JumpStart. Click Select Folder.
   4. Select Visual C# in the Language dropdown
   5. Click Add…
   6. Navigate to <JumpStart sample code folder>\Code Snippets\C# JumpStart. Click Select Folder.
   7. Click OK
2. Get a FlickrAPI Key:
   1. Get an API Key from the flickr developer website. See <https://www.flickr.com/services/apps/create/> . Get a non-commercial key.
   2. When you have a key, edit Begin\FlickrSearcher.cs and insert your key in the apiKey variable at the start of class FlickrSearchUrl.

## Create Project

1. Create new Visual C# - Store Apps – Universal Apps Blank App project: **FlickrSearch**
2. Explain the project structure
3. Right-click on **Shared** folder, click **Add existing item…**
   1. Navigate to \Begin and select FlickrSearcher.cs and TemplateUtility.cs
   2. Walk through FlickrSearcher.cs to explain what it does

## Build Windows UI

1. In the Windows Store project, open MainPage.xaml
2. Select the Grid control on the page
   1. In the designer, divide the Grid into **3 rows**, Height **100px**, **40px**, and 1**\***
3. CTRL+K,CTRL+X Insert snippet: **JumpStartFlickrSearchUI**
   1. This adds the heading TextBlock and the main body GridView on rows 0 and 1 respectively
4. Inside the GridView element: <GridView>…</GridView>:
   1. CTRL+K,CTRL+X Insert snippet **JumpStartFlickrSearchDataTemplate**
5. Open MainPage.xaml.cs
   1. After MainPage constructor, Insert snippet **JSFlickrLogic**
   2. Briefly explain that the OnNavigatedTo method we just pasted in will cause a search on Flickr for a searchterm “flowers”, and other methods will update the tile for the app, and send a toast.
6. Copy contents of \Begin\Assets-Windows and
   1. paste into Assets folder of Store project
   2. also paste contents of \Begin\Assets-Phone into Assets folder of Phone project (to save time later)
7. Open App Manifest. Explain that we just pasted in the app artwork which is configured on the Visual Assets tab.
   1. While there, go to Visual Assets – Splash Screen and set the background color to #FFFFFF
8. Set Startup project to be the Windows 8.1 project.
9. Run

## Build Phone UI

1. Copy the XAML from inside the Grid of the Windows project
   1. Paste into the Content Grid of the Phone project
   2. Make following changes:

<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">

<Grid.RowDefinitions>

<RowDefinition Height="80"/>

<RowDefinition Height="20"/>

<RowDefinition/>

</Grid.RowDefinitions>

<TextBlock HorizontalAlignment="Left" VerticalAlignment="Bottom"

Margin="20,0,0,0" TextWrapping="Wrap" Text="flickR"

Style="{StaticResource HeaderTextBlockStyle}" />

<GridView x:Name="gridView" Grid.Row="2" Margin="10,0,0,0"

ItemsSource="{Binding}">

<GridView.ItemTemplate>

<DataTemplate>

<Grid HorizontalAlignment="Left" Width="460" Height="300">

<Border Background="{ThemeResource ListViewItemPlaceholderBackgroundThemeBrush}">

<Image Source="{Binding ImageUrl}" Stretch="UniformToFill" AutomationProperties.Name="{Binding Title}"/>

</Border>

<StackPanel VerticalAlignment="Bottom" Background="{ThemeResource ListViewItemOverlayBackgroundThemeBrush}">

<TextBlock Text="{Binding Title}" Foreground="{ThemeResource ListViewItemOverlayForegroundThemeBrush}" Style="{StaticResource TitleTextBlockStyle}" Height="60" Margin="15,0,15,0"/>

</StackPanel>

</Grid>

</DataTemplate>

</GridView.ItemTemplate>

</GridView>

</Grid>

1. In MainPage.xaml.cs:
   1. Delete **OnNavigatedTo**
   2. Insert **JSFlickrLogic** snippet – exactly the same as the Windows app
2. **Optional**:
   1. Create a new class in the Shared project called MainPage.xaml.cs
   2. Copy the contents of MainPage.xaml.cs from the Windows project and paste inside MainPage.xaml.cs in the Shared project
   3. Delete everything apart from the constructor in MainPage.xaml.cs in the Windows Project
   4. Delete everything apart from the constructor in MainPage.xaml.cs in the Windows Phone Project
   5. Delete the constructor in MainPage.xaml.cs in the Shared project, and change the class
   6. Explain that this is just one way to share common code associated with our XAML pages
3. Run
4. In manifest on both Store and Phone projects, set **Toast Capable = Yes**, then deploy both again.
   1. **Pin Tile** on both Windows and on the Phone. Expand the tile on the Start screen to be a wide tile.
   2. Run both again by using the tile (or from Visual Studio). Show that the tile gets updated, and a Toast notification comes up on both platforms.