

```
1 using GalaSoft.MvvmLight;
2 using System.Windows.Input;
3 using System.Collections;
4 using GalaSoft.MvvmLight.Command;
5 using Assignment6.Model;
6 using System.Drawing;
7 using System.Collections.ObjectModel;
8 using System.Net;
9 using System.Threading.Tasks;
10 using System;
11 using System.Xml.Linq;
12 using System.Linq;
13 using System.Collections.Generic;
14 using System.Windows;
15 using System.IO;
16
17 namespace Assignment6.ViewModel
18 {
19     /// <summary>
20     /// This class contains properties that the main View can data bind to.
21     /// <para>
22     /// Use the <strong>mvvmminpc</strong> snippet to add bindable properties to ↗
23     /// this ViewModel.
24     /// </para>
25     /// <para>
26     /// You can also use Blend to data bind with the tool's support.
27     /// </para>
28     /// <para>
29     /// See http://www.galasoft.ch/mvvm
30     /// </para>
31     /// </summary>
32     public class MainViewModel : ViewModelBase
33     {
34         public ICommand SearchCommand { get; private set; }
35         public ICommand ShowImageCommand { get; private set; }
36         private ObservableCollection<string> photoList;
37         private ObservableCollection<string> titleList;
38         private string selectedImage;
39         private string searchText;
40         private int selectedIndex;
41         public WebClient flickrClient = new WebClient();
42         Task<String> flickrTask = null;
43         private const string KEY = "92df9b38a25b11ca3dab168760478b1c";
44         /// <summary>
45         /// Initializes a new instance of the MainViewModel class.
46         /// </summary>
47         public MainViewModel()
48         {
49             SearchCommand = new RelayCommand(SearchImagesMethodAsync);
```

```
49         ShowImageCommand = new RelayCommand(DisplayImage);
50         titleList = new ObservableCollection<string>();
51         photoList = new ObservableCollection<string>();
52     }
53
54     public async void SearchImagesMethodAsync()
55     {
56         if (flickrTask != null && flickrTask.Status != TaskStatus.RanToCompletion)
57         {
58             var result = MessageBox.Show(
59                 "Cancel the current Flickr search?",
60                 "Are you sure?",
61                 MessageBoxButton.YesNo,
62                 MessageBoxImage.Question);
63             if (result == MessageBoxResult.No)
64                 return;
65             else
66                 flickrClient.CancelAsync();
67         }
68         else if (searchText == null || String.IsNullOrEmpty(searchText))
69         {
70             MessageBox.Show("You must enter search text!",
71                 "Search Empty",
72                 MessageBoxButton.OK,
73                 MessageBoxImage.Error);
74         }
75     }
76     else
77     {
78         var flickrURL = string.Format("https://api.flickr.com/services/
79             rest/?method=flickr.photos.search&api_key={0}&tags={1}
80             &tag_mode=all&per_page=500&privacy_filter=1&format=rest",
81             KEY,
82             searchText.Replace(" ", ","));
83         titleList.Clear();
84         photoList.Clear();
85         selectedImage = null;
86         RaisePropertyChanged("SelectedImage");
87         RaisePropertyChanged("PhotoList");
88         RaisePropertyChanged("TitleList");
89         titleList.Add("Loading...");
90         RaisePropertyChanged("TitleList");
91         try
92         {
93             flickrTask = flickrClient.DownloadStringTaskAsync
94                 (flickrURL);
95             XDocument flickrXML = XDocument.Parse(await flickrTask);
96             var flickrPhotos =
```

```
94         from photo in flickrXML.Descendants("photo")
95         let id = photo.Attribute("id").Value
96         let title = photo.Attribute("title").Value
97         let secret = photo.Attribute("secret").Value
98         let server = photo.Attribute("server").Value
99         let farm = photo.Attribute("farm").Value
100         select new FlickrResult
101         {
102             Title = title,
103             URL = string.Format(
104                 "https://{0}.staticflickr.com/{1}/{2}_{3}.jpg",
105                 farm, server, id, secret)
106         };
107     titleList.Clear();
108     RaisePropertyChanged("TitleList");
109     if (flickrPhotos.Any())
110     {
111         var flickrPhotoList = flickrPhotos.ToList();
112         foreach (var p in flickrPhotoList)
113         {
114             photoList.Add(p.URL);
115         }
116         foreach (var t in flickrPhotoList)
117         {
118             titleList.Add(t.Title);
119         }
120         RaisePropertyChanged("TitleList");
121         RaisePropertyChanged("PhotoList");
122     }
123     else
124     {
125         titleList.Add("No matches");
126         RaisePropertyChanged("TitleList");
127     }
128 }
129 catch (WebException)
130 {
131     if (flickrTask.Status == TaskStatus.Faulted)
132     {
133         MessageBox.Show("Unable to get results from Flickr",
134             "Flickr Error",
135             MessageBoxButton.OK,
136             MessageBoxImage.Error);
137     }
138 }
139 }
140 }
141 }
```

```
142
143     public void DisplayImage()
144     {
145         if (titleList.Count() == 1 && titleList.ElementAt(0).Equals      ↗
            ("Loading..."))
146         {
147             //Do nothing
148         }
149         else if(selectedIndex >= 0 && selectedIndex < photoList.Count())
150         {
151             selectedImage = photoList.ElementAt(selectedIndex);
152             RaisePropertyChanged("SelectedImage");
153         }
154     }
155
156     public ObservableCollection<string> TitleList
157     {
158         get { return titleList; }
159         set
160         {
161             titleList = value;
162             RaisePropertyChanged("TitleList");
163         }
164     }
165
166     public ObservableCollection<string> PhotoList
167     {
168         get { return photoList; }
169         set
170         {
171             photoList = value;
172             RaisePropertyChanged("PhotoList");
173         }
174     }
175
176
177     public string SearchText
178     {
179         get { return searchText; }
180         set
181         {
182             searchText = value;
183             RaisePropertyChanged("SearchText");
184         }
185     }
186
187     public string SelectedImage
188     {
189         get { return selectedImage; }
```

```
190         set
191         {
192             selectedImage = value;
193             RaisePropertyChanged("SelectedImage");
194         }
195     }
196
197     public int SelectedIndex
198     {
199         get { return selectedIndex; }
200         set
201         {
202             selectedIndex = value;
203             RaisePropertyChanged("SelectedIndex");
204         }
205     }
206 }
207 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Windows;
7 using System.Windows.Controls;
8 using System.Windows.Data;
9 using System.Windows.Documents;
10 using System.Windows.Input;
11 using System.Windows.Media;
12 using System.Windows.Media.Imaging;
13 using System.Windows.Navigation;
14 using System.Windows.Shapes;
15
16 namespace Assignment6
17 {
18     /// <summary>
19     /// Interaction logic for MainWindow.xaml
20     /// </summary>
21     public partial class MainWindow : Window
22     {
23         public MainWindow()
24         {
25             InitializeComponent();
26         }
27     }
28 }
29
```

```

1 <Window x:Class="Assignment6.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6     xmlns:i="http://schemas.microsoft.com/expression/2010/interactivity"
7     xmlns:ei="http://schemas.microsoft.com/expression/2010/interactions"
8     xmlns:local="clr-namespace:Assignment6"
9     mc:Ignorable="d"
10    Title="Flickr Viewer" Height="600" Width="750" ResizeMode="NoResize">
11    <Grid DataContext="{Binding MainViewModel, Source={StaticResource
12        ViewModelLocator}}" Background="LightGray">
13        <ListBox ItemsSource="{Binding TitleList}" SelectedItem="{Binding
14            SelectedImage}" SelectedIndex="{Binding SelectedIndex}"
15            HorizontalAlignment="Left" Height="429" Margin="25,93,0,0"
16            VerticalAlignment="Top" Width="180">
17            <i:Interaction.Triggers>
18                <i:EventTrigger eventName="MouseDoubleClick">
19                    <i:InvokeCommandAction Command="{Binding ShowImageCommand}"/>
20                </i:EventTrigger>
21            </i:Interaction.Triggers>
22        </ListBox>
23        <TextBox HorizontalAlignment="Left" Height="23" Margin="25,27,0,0"
24            TextWrapping="Wrap" Text="{Binding SearchText}"
25            VerticalAlignment="Top" Width="437"/>
26        <Button Content="Search" Command="{Binding SearchCommand}"
27            HorizontalAlignment="Left" Margin="495,27,0,0" VerticalAlignment="Top"
28            Width="90" Height="23"/>
29        <Border BorderThickness="1">
30            <Image Source="{Binding SelectedImage}" HorizontalAlignment="Left"
31                Height="429" Margin="220.2,92.2,0,0" VerticalAlignment="Top"
32                Width="495"/>
33        </Border>
34    </Grid>
35</Window>

```

```
1  /*
2   In App.xaml:
3   <Application.Resources>
4       <vm:ViewModelLocator xmlns:vm="clr-namespace:Assignment6"
5                               x:Key="Locator" />
6   </Application.Resources>
7
8   In the View:
9   DataContext="{Binding Source={StaticResource Locator}, Path=ViewModelName}"
10
11   You can also use Blend to do all this with the tool's support.
12   See http://www.galasoft.ch/mvvm
13  */
14
15  using GalaSoft.MvvmLight;
16  using GalaSoft.MvvmLight.Ioc;
17  using Microsoft.Practices.ServiceLocation;
18
19  namespace Assignment6.ViewModel
20  {
21      /// <summary>
22      /// This class contains static references to all the view models in the
23      /// application and provides an entry point for the bindings.
24      /// </summary>
25      public class ViewModelLocator
26      {
27          /// <summary>
28          /// Initializes a new instance of the ViewModelLocator class.
29          /// </summary>
30          public ViewModelLocator()
31          {
32              ServiceLocator.SetLocatorProvider(() => SimpleIoc.Default);
33
34              ////if (ViewModelBase.IsInDesignModeStatic)
35              ////{
36              ////    // Create design time view services and models
37              ////    SimpleIoc.Default.Register<IDataService, DesignDataService>
38              ////    ();
39              ////}
40              ////else
41              ////{
42              ////    // Create run time view services and models
43              ////    SimpleIoc.Default.Register<IDataService, DataService>();
44              ////}
45
46              SimpleIoc.Default.Register<MainViewModel>();
47          }
48
49          public MainViewModel MainViewModel
```



```
49     {
50         get
51         {
52             return ServiceLocator.Current.GetInstance<MainViewModel>();
53         }
54     }
55
56     public static void Cleanup()
57     {
58         // TODO Clear the ViewModels
59     }
60 }
61 }
```

```
... Rojas\Documents\CECS475\Assignment6\Assignment6\App.xaml 1
1 <Application x:Class="Assignment6.App" xmlns="http://schemas.microsoft.com/  ↗
  winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/  ↗
  xaml" xmlns:local="clr-namespace:Assignment6" StartupUri="MainWindow.xaml"  ↗
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008" d1p1:Ignorable="d"  ↗
  xmlns:d1p1="http://schemas.openxmlformats.org/markup-compatibility/2006">
2   <Application.Resources>
3     <ResourceDictionary>
4       <vm:ViewModelLocator x:Key="ViewModelLocator" d:IsDataSource="True"  ↗
        xmlns:vm="clr-namespace:Assignment6.ViewModel" />
5     </ResourceDictionary>
6   </Application.Resources>
7 </Application>
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Assignment6.Model
8 {
9     class FlickrResult
10    {
11        public string Title { get; set; }
12        public string URL { get; set; }
13    }
14 }
15
```