Como usar Cimbalino Windows Phone Toolkit Media Library - ScreenshotService

Este artigo mostra como usar o Cimbalino Windows Phone Toolkit Media Library - ScreenshotService.

Introdução

Cimbalino Windows Phone Toolkit é um conjunto de itens uteis e poderosos para ajudar na implementação de aplicações Windows Phone. Projeto base do toolkit contendo serviços para implementação do padrão de MVVM, conversores, classes auxiliares, métodos de extensões;

- Cimbalino.Phone.Toolkit.MediaLibrary - Projeto do toolkit contendo serviços para implementação do padrão de MVVM com acesso à biblioteca de imagens, músicas e vídeos.

IScreenshotService - Representa a interface do servilço com a capacidade de tirar "screenshots". A implementação é ScreenshotService.

Note: é preciso definir a capacidade ID_CAP_MEDIALIB_PHOTO no ficheiro manifest.

Contruindo o exemplo

O código fonte pode ser obtido em Exemplo ScreenshotService (github).

Os pacotes estão disponíveis em Nuget Package Manager (para ambos os "targets") e podem ser instalados, para mais detalhes consultar: Como instalar os pacotes Cimbalino Windows Phone Toolkit. O exemplo deste artigo usa MVVM Light para ajudar na implementação do padrão MVVM, para mais detalhes consultar:

- Como instalar os pacotes de MVVM Light Toolkit para Windows Phone
- Como usar MVVM Light Toolkit para Windows Phone

Registando os serviços

Devemos começar por registar cada serviço no ViewModelLocator, como podemos ver de seguida:

```
using Cimbalino.Phone.Toolkit.Services;
```

```
using GalaSoft.MvvmLight;
```

```
using GalaSoft.MvvmLight.Ioc;
```

```
using Microsoft.Practices.ServiceLocation;
```

```

```

```
/// This class contains static references to all the view models in the
```

```
/// application and provides an entry point for the bindings.
```

```
public class ViewModelLocator
```


(C) Copyright Nokia 2014. All rights reserved.

```
    {
        /// Initializes a new instance of the ViewModelLocator class.
        public ViewModelLocator()
        {
            ServiceLocator.SetLocatorProvider(() => SimpleIoc.Default);
            if (!SimpleIoc.Default.IsRegistered<IScreenshotService>())
            {
                SimpleIoc.Default.Register<IScreenshotService, ScreenshotService>();
            }
            SimpleIoc.Default.Register<MainViewModel>();
        }

        /// Gets the main view model.
        public MainViewModel MainViewModel
        {
            get
            {
                return ServiceLocator.Current.GetInstance<MainViewModel>();
```


(C) Copyright Nokia 2014. All rights reserved.

```
            }

        }

        public static void Cleanup()

        {

            // TODO Clear the ViewModels

            var viewModelLocator = (ViewModelLocator)App.Current.Resources["Locator"];

            viewModelLocator.MainViewModel.Cleanup();

        }

    }
```

Implementando a ViewModel

Em seguinda devemos implementar o MainViewModel, como podemos ver de seguida:

```
    using System.Windows;

    using System.Windows.Input;

    using System.Windows.Threading;

    using Cimbalino.Phone.Toolkit.Services;

```


(C) Copyright Nokia 2014. All rights reserved.

```
using GalaSoft.MvvmLight.Command;

using GalaSoft.MvvmLight;


/// This class contains properties that the main View can data bind to.

public class MainViewModel : ViewModelBase

{

    /// The screenshot service

    private readonly IScreenshotService _screenshotService;


    /// Initializes a new instance of the MainViewModel class.

    public MainViewModel(IScreenshotService screenshotService)

    {

        _screenshotService = screenshotService;

        TakeScreenshotCommand = new RelayCommand(TakeScreenshot);

    }


    /// Gets the take screenshot command.

    public ICommand TakeScreenshotCommand { get; private set; }
```


(C) Copyright Nokia 2014. All rights reserved.

```
25.
26.         /// Command to take screenshot.
27.         private void TakeScreenshot()
28.         {
29.             _screenshotService.TakeScreenshot("CimbalinoScreenshot");
30.         }
31.     }
```

Implementando a View

A **MainPage.xaml** será algo do género:

```
1. <phone:PhoneApplicationPage x:Class="CimbalinoSample.MainPage"
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:phone="clr-
namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
7.                             xmlns:shell="clr-
namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
8.                             DataContext="{Binding MainViewModel,
9.                                                   Source={StaticResource
```


(C) Copyright Nokia 2014. All rights reserved.

```
Locator}}"
                                FontFamily="{StaticResource PhoneFontFamilyNormal}"
                                FontSize="{StaticResource PhoneFontSizeNormal}"
                                Foreground="{StaticResource PhoneForegroundBrush}"
                                Orientation="Portrait"
                                SupportedOrientations="Portrait"
                                shell:SystemTray.IsVisible="True"
                                mc:Ignorable="d">

    <!--  LayoutRoot is the root grid where all page content is placed  -->
    <Grid x:Name="LayoutRoot" Background="Transparent">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>

        <!--  TitlePanel contains the name of the application and page title  -->
        <StackPanel x:Name="TitlePanel"
```


(C) Copyright Nokia 2014. All rights reserved.

```
                    Grid.Row="0"

                    Margin="12,17,0,28">

            <TextBlock Margin="12,0"

                       Style="{StaticResource PhoneTextTitle2Style}"

                       Text="Cimbalino Sample" />

            <TextBlock Margin="9,-7,0,0"

                       Style="{StaticResource PhoneTextTitle2Style}"

                       Text="ScreenshotService" />

        </StackPanel>


        <!--  ContentPanel - place additional content here  -->

        <Grid x:Name="ContentPanel"

              Grid.Row="1"

              Margin="12,0,12,0">

            <TextBlock TextWrapping="Wrap">This samples has the goal to show how
to use Cimbalino Windows Phone Toolkit Media Library -
ScreenshotService</TextBlock>


            <Button Margin="0,219,0,293"

                    Command="{Binding TakeScreenshotCommand}"
```


(C) Copyright Nokia 2014. All rights reserved.

```
                    Content="Take Screenshot" />
```

```
        </Grid>
```

```
    </Grid>
```

```
</phone:PhoneApplicationPage>
```


(C) Copyright Nokia 2014. All rights reserved.