Como usar Cimbalino Windows Phone Toolkit - ApplicationManifestService

Este artigo tem como objetivo mostrar como usar Cimbalino Windows Phone Toolkit Background – ApplicationManifestService.

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.Background - Projeto do toolkit contendo serviços para implementação do padrão de MVVM compatível com “background agents”.

- IApplicationManifestService - Representa a interface para o serviço com a capacidade de ler o ficheiro manifest da aplicação. A implementação é ApplicationManifestService.

Página resultante

Contruindo o exemplo

O código fonte pode ser obtido em Exemplo ApplicationManifestService (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 serviços

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

```
    /// 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<IApplicationManifestService>())
            {
                SimpleIoc.Default.Register<IApplicationManifestService,
ApplicationManifestService>();
            }
            SimpleIoc.Default.Register<MainViewModel>();
        }

        public MainViewModel MainViewModel
        {
            get
            {
                return ServiceLocator.Current.GetInstance<MainViewModel>();
            }
        }

        public static void Cleanup()
        {
            // TODO Clear the ViewModels
        }
    }
```

Implementando a ViewModel

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

```
    /// This class contains properties that the main View can data bind to.
    public class MainViewModel : ViewModelBase
    {
        /// The public application url.
        private readonly string _appUrl;

        /// The application manifest.
        private readonly ApplicationManifest _applicationManifest;

        /// Initializes a new instance of the application Manifest Service.
        public MainViewModel(IApplicationManifestService applicationManifestService)
        {
            _applicationManifest = applicationManifestService.GetApplicationManifest();
            _appUrl = string.Concat("http://windowsphone.com/s?appid=",
_applicationManifest.App.ProductId);
        }

        /// Gets the title.
        public string Title
        {
            get
            {
                return _applicationManifest.App.Title;
            }
```

 (C) Copyright Nokia 2014. All rights reserved.

```
        }

        /// Gets the author.
        public string Author
        {
            get
            {
                return _applicationManifest.App.Author;
            }
        }

        /// Gets the version.
        public string Version
        {
            get
            {
                return _applicationManifest.App.Version;
            }
        }

        /// Gets the description.
        public string Description
        {
            get
            {
                return _applicationManifest.App.Description;
            }
        }

        /// Gets the product ID.
        public string ProductID
        {
            get
            {
                return _applicationManifest.App.ProductId;
            }
        }

        /// Gets the publisher.
        public string Publisher
        {
            get
            {
                return _applicationManifest.App.Publisher;
            }
        }

        /// Gets the capabilities.
        public IList Capabilities
        {
            get
            {
                return _applicationManifest.App.Capabilities.ToList();
            }
        }
    }
```

 (C) Copyright Nokia 2014. All rights reserved.

Implementando a View

Para conetar a view model com a página devemos adicionar a `ViewModelLocator` no **App.xaml**:

```
<vm:ViewModelLocator x:Key="Locator" d:IsDataSource="True"  />
```

e adicionar o binding na página principal:

```
 DataContext="{Binding MainViewModel,
                       Source={StaticResource Locator}}"
```

A MainPage.xaml será algo do género:

```
<phone:PhoneApplicationPage x:Class="CimbalinoSample.MainPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                            xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                            xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                            xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
                            xmlns:phone="clr-
namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
                            xmlns:shell="clr-
namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
                            DataContext="{Binding MainViewModel,
                                                  Source={StaticResource 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"
                    Grid.Row="0"
                    Margin="12,17,0,28">
            <TextBlock Margin="12,0"
                       Style="{StaticResource PhoneTextTitle2Style}"
                       Text="Cimbalino Sample" />
        </StackPanel>

        <!--  ContentPanel - place additional content here  -->
        <Grid x:Name="ContentPanel"
              Grid.Row="1"
```

 (C) Copyright Nokia 2014. All rights reserved.

```
                Margin="12,0,12,0" >
            <TextBlock TextWrapping="Wrap"
                       VerticalAlignment="Top" Height="50">
                Title: <Run Text="{Binding Title}"/>
                </TextBlock>
            <TextBlock TextWrapping="Wrap"
                       VerticalAlignment="Top" Margin="0,50,0,0">
                Author: <Run Text="{Binding Author}"/>
            </TextBlock>
            <TextBlock TextWrapping="Wrap"
                       VerticalAlignment="Top" Margin="0,100,0,0">
                Version: <Run Text="{Binding Version}"/>
            </TextBlock>
            <TextBlock TextWrapping="Wrap"
                       VerticalAlignment="Top" Margin="0,150,0,0">
                Description: <Run Text="{Binding Description}"/>
            </TextBlock>
            <TextBlock TextWrapping="Wrap"
                       VerticalAlignment="Top" Margin="0,300,0,0">
                ProductID: <Run Text="{Binding ProductID}"/>
            </TextBlock>
            <TextBlock TextWrapping="Wrap"
                       VerticalAlignment="Top" Margin="0,350,0,0">
                Publisher: <Run Text="{Binding Publisher}"/>
            </TextBlock>
            <TextBlock TextWrapping="Wrap" Text="Capabilities:"
                       VerticalAlignment="Top" Margin="0,400,0,0"/>
            <ListBox DisplayMemberPath="Name" Margin="0,450,0,0"
                     ItemsSource="{Binding Capabilities}"/>
        </Grid>

    </Grid>

</phone:PhoneApplicationPage>
```

Exemplo Relacionado

- Criando uma página "Sobre" para aplicações em Windows Phone

Código Fonte

O código fonte pode ser obtido aqui

 (C) Copyright Nokia 2014. All rights reserved.

 (C) Copyright Nokia 2014. All rights reserved.