# CREATE A WINDOWS PHONE 8.1 PROJECT

1.

Install-Package Caliburn.Micro –pre

2. BRING IN APP.XAML AND APP.XAML.CS

**SNIPPET 1**

**CHANGE THE NAMESPACE FROM App1**

**CHANGE TO APPLICATION DEFINITION ON BUILD TYPE**

3. CREATE VIEW + VIEW MODEL

 public class MainViewModel : **Screen**

 {

 **public ObservableCollection<NewsItemViewModel> Items { get; set; }**

 public MainViewModel()

 {

 Items = new ObservableCollection<NewsItemViewModel>()

 {

 ITEMS GO HERE

 }

 }

 }

4. CREATE NEWS ITEM VIEW MODEL

 public class NewsItemViewModel

 {

 public string Heading { get; set; }

 public string Uri { get; set; }

 }

5. IN APP.XAML.CS FIX

// TODO: container.PerRequest<T>();

// TODO: DisplayRootView<MainView>();

6. FIX THE VIEW

<StackPanel Margin="20">

 **<TextBlock FontSize="32">**

 **Awesome News Reader**

 **</TextBlock>**

 **<ListView** **ItemsSource**="{Binding Items}">

 <ListView.ItemTemplate>

 <DataTemplate>

 **<TextBlock FontSize="24" Margin="5" Text="{Binding Heading}">**

**</TextBlock>**

 </DataTemplate>

 </ListView.ItemTemplate>

 </ListView>

</StackPanel>

7. TEST THE APP

8. ADD CLICK HANDLER FOR THE ITEM

**micro:Message.Attach="[Event Tapped] = [Action Tapped($dataContext)]"**

9. INJECT NAVIGATION SERVICE

public MainViewModel(**INavigationService navigationService**)

10. SETUP TAPPED

public void Tapped(NewsItemViewModel item)

{

 \_navigationService.
**NavigateToViewModel**<DetailsViewModel>**(item);**

}

11. SETUP DETAILS VIEW MODEL

public class DetailsViewModel : **Screen**

{

 public NewsItemViewModel **Parameter { get; set; }**

}

12. SETUP DETAILS VIEW

 <StackPanel Margin="20">

 <**TextBlock FontSize="32" Text="{Binding Parameter.Heading}">**

 **</TextBlock>**

 <WebView **Source="{Binding Parameter.Uri}"** Margin="0,0,0,0" Height="500">

 </WebView>

 </StackPanel>

13. TEST THE APP

14. SETUP BACK HANDLER

protected override void OnNavigatedTo

(NavigationEventArgs e)

{

 **Windows.Phone.UI.Input.HardwareButtons.BackPressed += HardwareButtons\_BackPressed;**

}

void **HardwareButtons\_BackPressed**(object sender, Windows.Phone.UI.Input.BackPressedEventArgs e)

{

 e.Handled = true;

 **Windows.Phone.UI.Input.HardwareButtons.BackPressed -= HardwareButtons\_BackPressed;**

 **Frame.GoBack();**

}

15. TEST THE APP

14. ADD WINDOWS 8.1 PROJECT

15. MOVE SHARED FILES

16. ADD **#if WINDOWS\_PHONE\_APP**

17. EXPLAIN THERE IS NO BACK BUTTON ON WINDOWS TABLETS

18. MOVE DETAILS VIEW TO EACH PROJECT

19. ADD BOTTOM BAR

 <Page.BottomAppBar>

 <CommandBar>

 <AppBarButton Icon="Back" Label="Back" micro:Message.Attach="[Event Tapped] = [Action Back()]"/>

 </CommandBar>

 </Page.BottomAppBar>

20. INJECT NAVIGATION SERVICE AND ADD BACK TO DETAILS VIEW MODEL

 private readonly INavigationService \_navigationService;

 public DetailsViewModel**(INavigationService navigationService)**

 {

 \_navigationService = navigationService;

 }

  **public void Back()**

 **{**

 **\_navigationService.GoBack();**

 **}**

21. TEST THE APP

22. CREATE A NEW AZURE MOBILE SERVICE

23. ADD A TABLE – **NEWS ITEM**

24. CREATE A NEW PORTABLE UNIVERSAL PROJECT

25. INSTALL MOBILE SERVICES

**Install-Package WindowsAzure.MobileServices –pre**

26. CREATE INTERFACE

 **public interface IRepository**

 **{**

 **Task CreateDefaultRecords();**

 **Task<IEnumerable<NewsItem>> GetAll();**

 **}**

27. CREATE REPOSITORY CLASS

**SNIPPET 2**

28. ADD THE MOBILE SERVICE CODE FROM THE PORTAL

29. ADD CREATE DEFAULT RECORDS

**SNIPPET 2**

30. INSTALL MOBILE SERVICE INTO **EACH PROJECT!!**

31. INJECT REPO INTO MAIN VIEW MODEL

32. CALL LOAD DATA

  **public async Task LoadData()**

 **{**

 **await \_repository.CreateDefaultRecords();**

 **var items = await \_repository.GetAll();**

 **foreach (var item in items.Select(x => new NewsItemViewModel {Heading = x.Heading, Uri = x.Uri}))**

 **{**

 **Items.Add(item);**

 **}**

 **}**