

Chapter 16: Data Binding

Section 16.1: Basic Binding to ViewModel

EntryPoint.xaml:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:vm="clr-namespace:MyAssembly.ViewModel;assembly=MyAssembly"
             x:Class="MyAssembly.EntryPage">
  <ContentPage.BindingContext>
    <vm:MyViewModel />
  </ContentPage.BindingContext>
  <ContentPage.Content>
    <StackLayout VerticalOptions="FillAndExpand"
                 HorizontalOptions="FillAndExpand"
                 Orientation="Vertical"
                 Spacing="15">
      <Label Text="Name:" />
      <Entry Text="{Binding Name}" />
      <Label Text="Phone:" />
      <Entry Text="{Binding Phone}" />
      <Button Text="Save" Command="{Binding SaveCommand}" />
    </StackLayout>
  </ContentPage.Content>
</ContentPage>
```

MyViewModel.cs:

```
using System;
using System.ComponentModel;

namespace MyAssembly.ViewModel
{
  public class MyViewModel : INotifyPropertyChanged
  {
    private string _name = String.Empty;
    private string _phone = String.Empty;

    public string Name
    {
      get { return _name; }
      set
      {
        if (_name != value)
        {
          _name = value;
          OnPropertyChanged(nameof(Name));
        }
      }
    }

    public string Phone
    {
      get { return _phone; }
      set
      {
        if (_phone != value)
```

```
        {
            _phone = value;
            OnPropertyChanged(nameof(Phone));
        }
    }

    public ICommand SaveCommand { get; private set; }

    public MyViewModel()
    {
        SaveCommand = new Command(SaveCommandExecute);
    }

    private void SaveCommandExecute()
    {
    }

    public event PropertyChangedEventHandler PropertyChanged;

    protected virtual void OnPropertyChanged(string propertyName)
    {
        PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
    }
}
```