Applying Data Binding and MVVM

Thomas Claudius Huber

Software Developer

@thomasclaudiush   www.thomasclaudiushuber.com

Module Outline

Learn the data binding basics
- Bind to another element
- Know how the data context works

Apply the MVVM pattern

Notify about property changes

Convert values in a data binding

Demo

Bind to another element

Know How the Data Context Works

ElementName

```
<TextBlock Text="{Binding ElementName=...}"/>
```

Know How the Data Context Works

ElementName   **Source**

```xml
<TextBlock
  Text="{Binding Source={StaticResource myRes}}"/>
```

Know How the Data Context Works

ElementName   **Source**   **RelativeSource**

```xml
<Rectangle Fill="Red" Width="300"
    Height="{Binding RelativeSource=
            {RelativeSource Self},Path=Width}"/>
```

Know How the Data Context Works

ElementName

Source

RelativeSource

DataContext

Know How the Data Context Works

```xml
<Grid DataContext="Pluralsight">
  <StackPanel>
    <TextBlock Text="{Binding}"/>
  </StackPanel>
</Grid>
```

# Know How the Data Context Works

```xml
<Grid DataContext="Pluralsight">
  <StackPanel DataContext="Thomas">
    <TextBlock Text="{Binding}"/>
  </StackPanel>
</Grid>
```
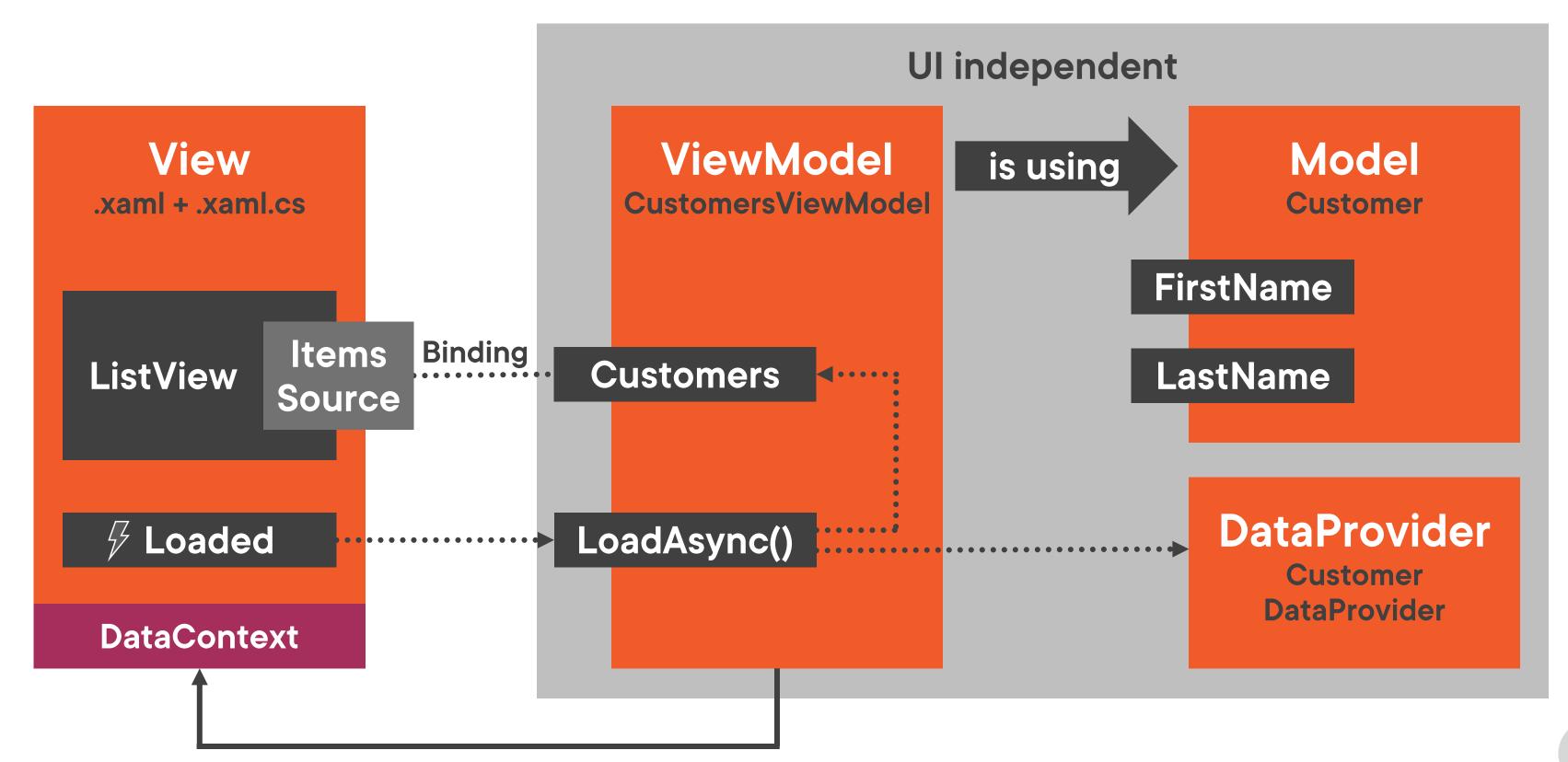
The Model View ViewModel (MVVM) Pattern

View
.xaml + .xaml.cs

ListView

Items Source

⚡ Loaded

DataContext

UI independent

ViewModel
CustomersViewModel

Customers

LoadAsync()

is using

Model
Customer

FirstName

LastName

DataProvider
Customer DataProvider

Binding

Advantages of MVVM

Maintainability

Testability

Demo
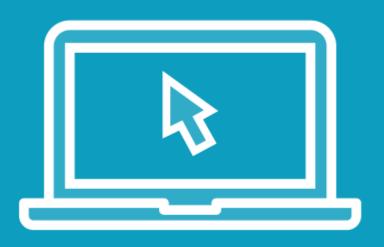
Create a CustomersViewModel

Demo
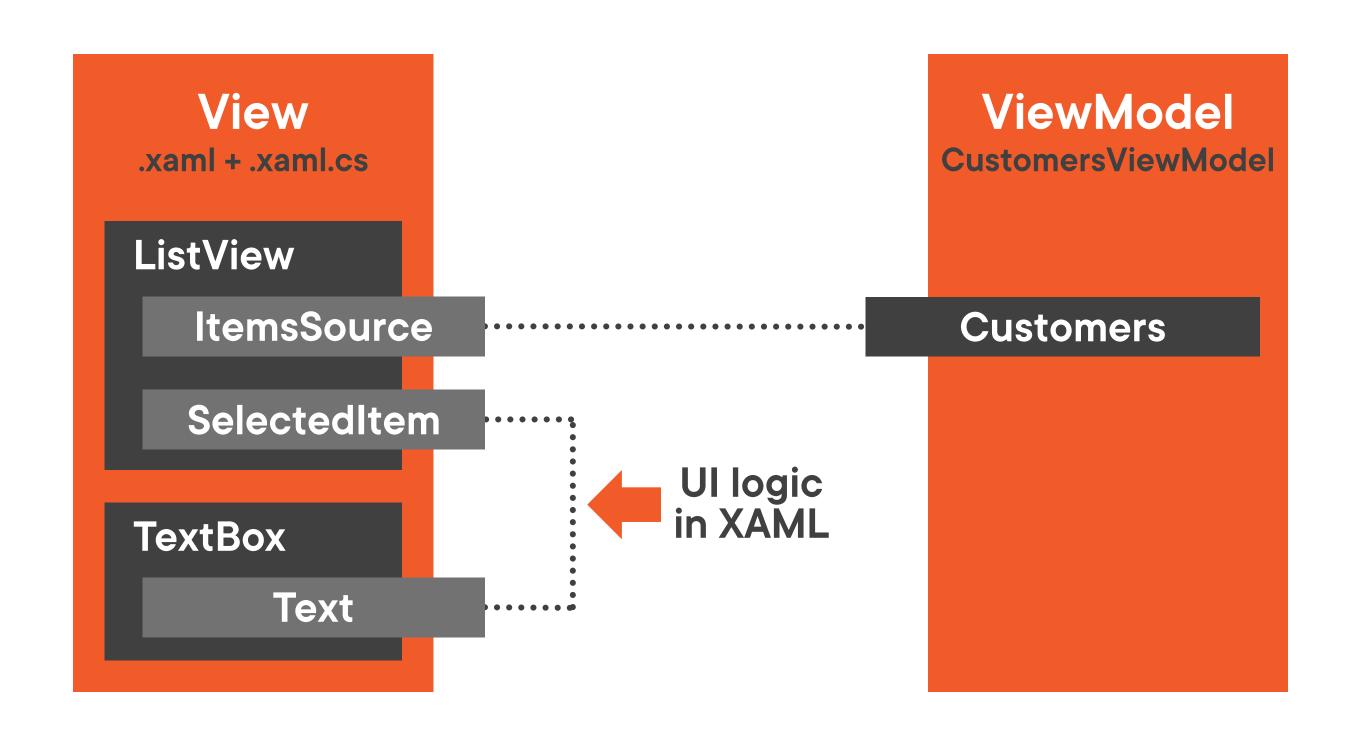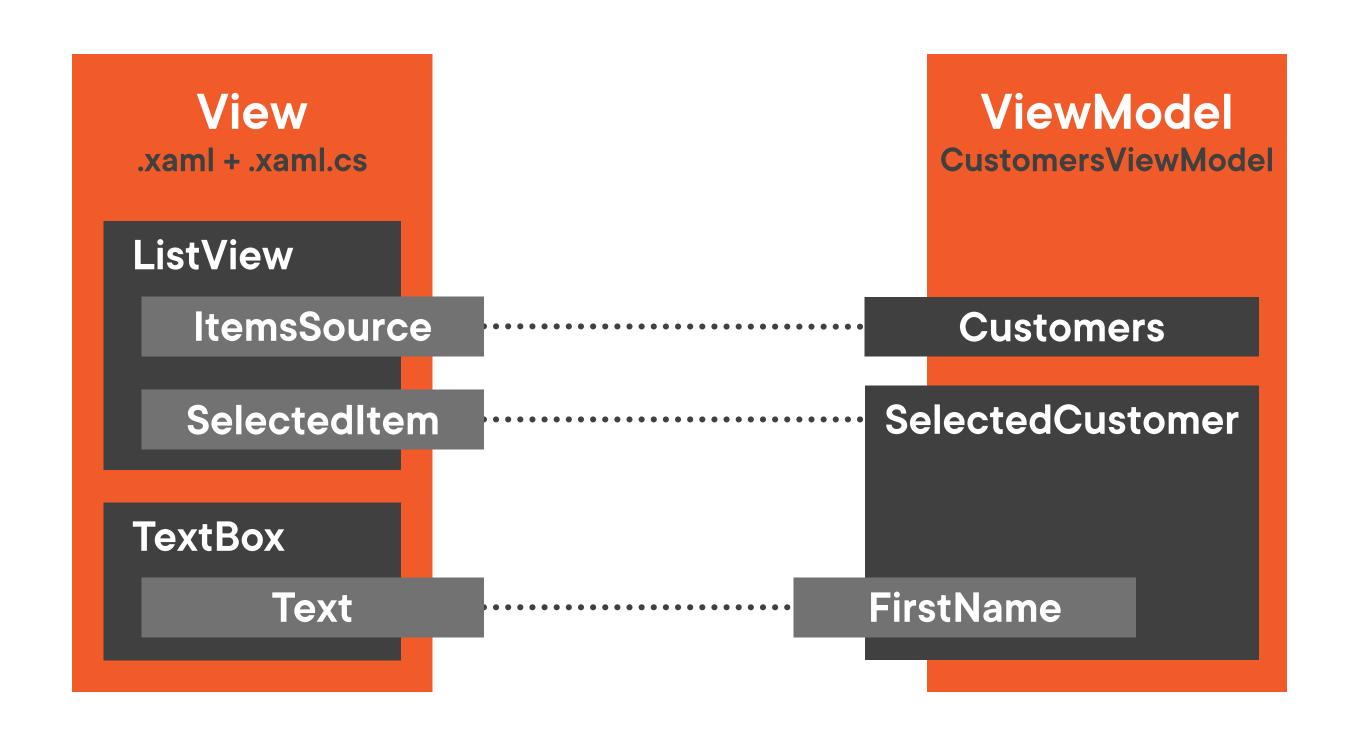
Use the CustomersViewModel for the CustomersView

Add a SelectedCustomer Property

View
.xaml + .xaml.cs

ViewModel
CustomersViewModel

ListView

ItemsSource · · · · · · · · · · · **Customers**

SelectedItem

← **UI logic in XAML**

TextBox

Text

Add a SelectedCustomer Property

View
.xaml + .xaml.cs

ListView

ItemsSource .... Customers

SelectedItem .... SelectedCustomer

TextBox

Text .... FirstName

ViewModel
CustomersViewModel

Demo

Add a SelectedCustomer property

Demo
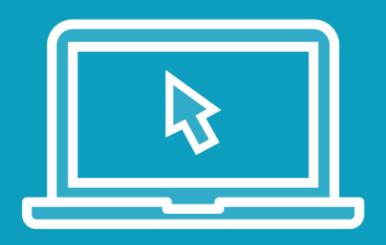
**Implement the logic to add a customer**

Demo

Notify about property changes

Demo

Refactor logic into a ViewModelBase class

Demo

Create a CustomerItemViewModel

Demo

Bind the Grid.Column property

Demo

Convert values with an IValueConverter

Summary

Create data bindings in XAML
- Bind to another element
- Know how the data context works

Apply the MVVM pattern

Work with data bindings
- Notify about property changes
- Use the XAML Binding Failures window
- Convert values with an IValueConverter

Up Next:
Executing Code with Commands