Getting hands on with XAML and Xamarin.Forms

Mitch "Rez" Muenster

@MobileRez

mobilerez.tumbler.com

Xamarin Certified Developer

Objectives

- Understanding Xamarin.Forms
- XAML Syntax & Behavior
- Advanced XAML

Understanding Xamarin.Forms

What is Xamarin.Forms?

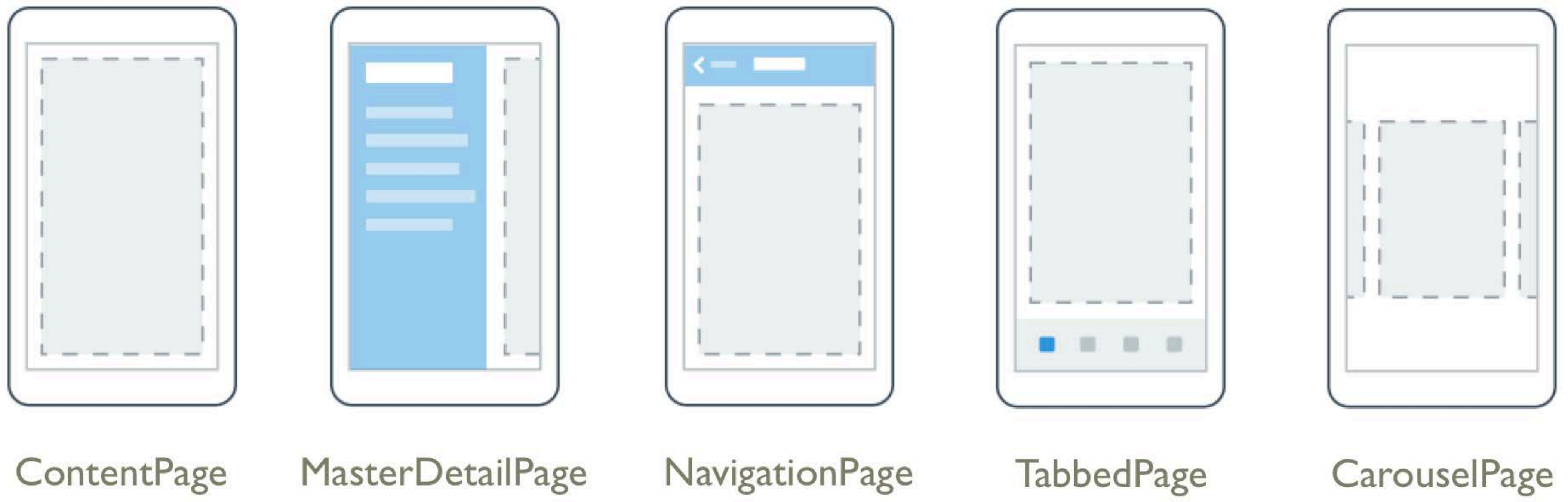
- Xamarin.Forms allows you to rapidly create a cross platform app with a native UI.
- Can be created in either a Shared Class Library or Portable Class Library
- Great for Prototyping or Data-Driven apps.
- Can also use dependency service to access platform specific features.

Understanding Xamarin.Forms UI

- Xamarin.Forms UI is defined in 4 different ways; Pages, Layouts, Cells, and Views.

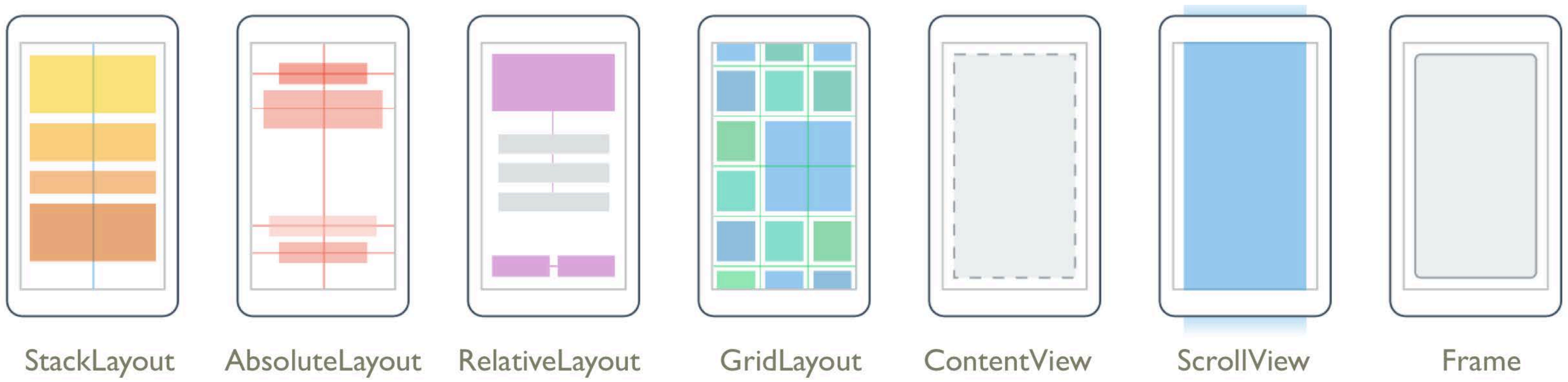
What is a Page?

- A page is used to define a single screen that contains most or all of the screen.



What is a Layout?

- A layout is a special type of view that acts as a container for other views or layouts.



What is a View?

- A View is the term Xamarin.Forms uses for all its basic controls from Buttons to Progress Bars.
- Some of the Views Xamarin.Forms contains are
  - Button
  - Date Picker
  - Entry (*input box*)
  - Label
  - Picker (*The phones form of dropdown list*)
  - Progress Bar

A full list of Views at https://developer.xamarin.com/guides/cross-platform/xamarin-forms/controls/views/

What is a Cell?

- A Cell is a special element that is used inside tables and defines how each item in a list is rendered.
- An example of Cells Xamarin.Forms supports:
  - Entry Cell
  - Switch Cell
  - Text Cell
  - Image Cell

A full list of Cells at https://developer.xamarin.com/guides/cross-platform/xamarin-forms/controls/cells/

Traditional way to build Forms apps

- Xamarin.Forms apps are commonly built using all using C# and not XAML.
- A new Xamarin.Forms app is usually created with a dummy app in a cs file

```
public App()
{
    // The root page of your application
    MainPage = new ContentPage {
        Content = new StackLayout {
            VerticalOptions = LayoutOptions.Center,
            Children = {
                new Label {
                    XAlign = TextAlignment.Center,
                    Text = "Welcome to Xamarin Forms!“
                }
            }
        }
    };
}
```

XAML Syntax & Behavior

What is XAML?

- XAML stands for E<u>x</u>tensible <u>M</u>arkup <u>L</u>anguage and was created by Microsoft specifically for working with the UI
- A XAML file is always associated with a C# code file.

Why use XAML over all code in a .cs file?

- Designer can create UI while coder focuses on code in the code file
- XAML allows for features like DataBinding Animations, Custom behaviors, value converters & more.
- Easier to work with for those who like to have a more visual representation of their layouts
- Helps keep a separation between UI and app logic

Breakdown Of a XAML File

XAML Syntax

Building a layout in XAML

Using OnPlatform

Attached properties

Advanced XAML

Using Resource Dictonary

Resource Dictionary hierarchy

Data Binding + XAML + Forms

Hands on Lab

What’s Next?

- Data Binding with Xamarin.Forms & XAML
- List views and collections with Data Binding, XAML, & Xamarin.Forms

Questions?

mitchmuenster@gmail.com

@MobileRez

mobilerez.tumbler.com