

Exploring Your First Xamarin.Android Project



Gill Cleeren

CTO XPIRIT BELGIUM

@gillcleeren www.snowball.be



Overview



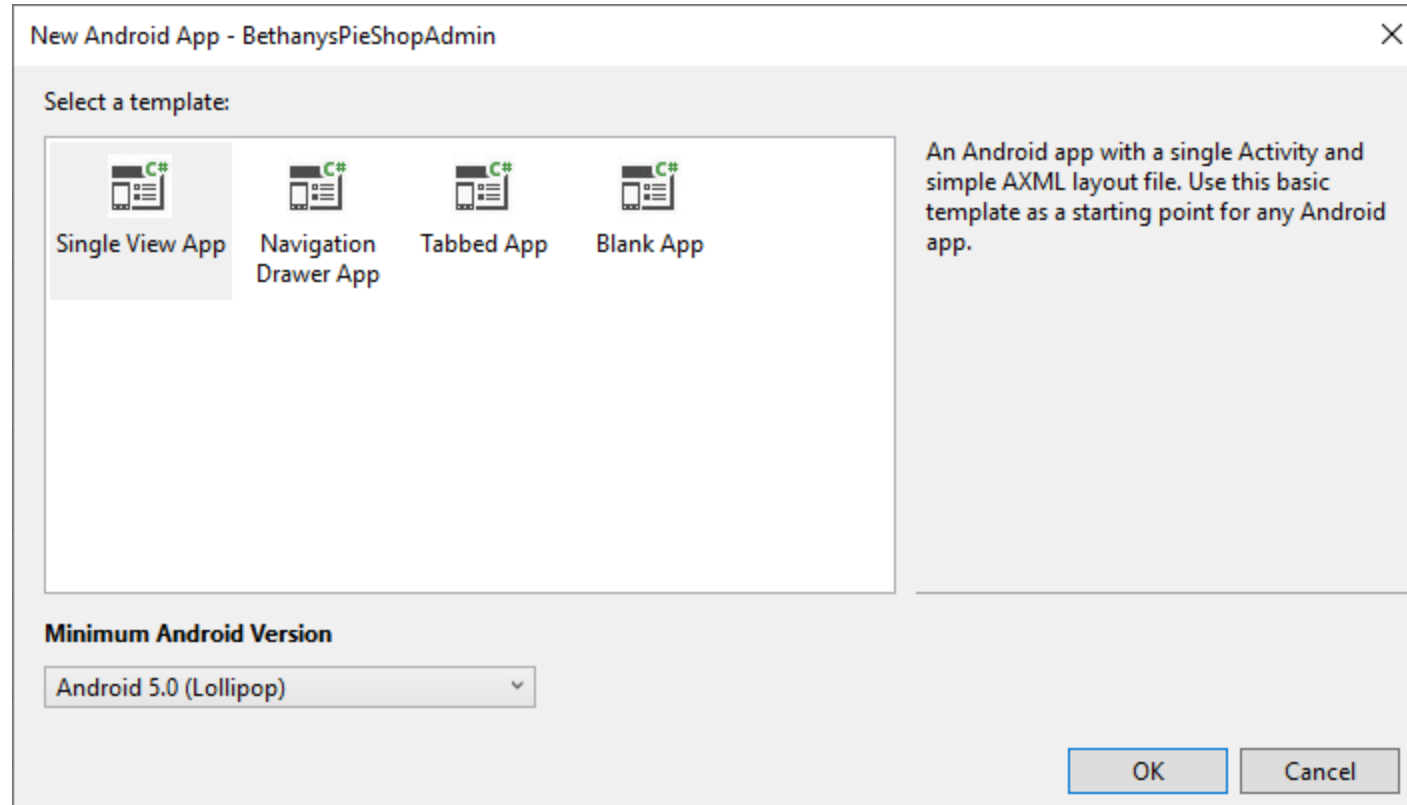
Fundamentals of Xamarin.Android
Code-sharing with Xamarin



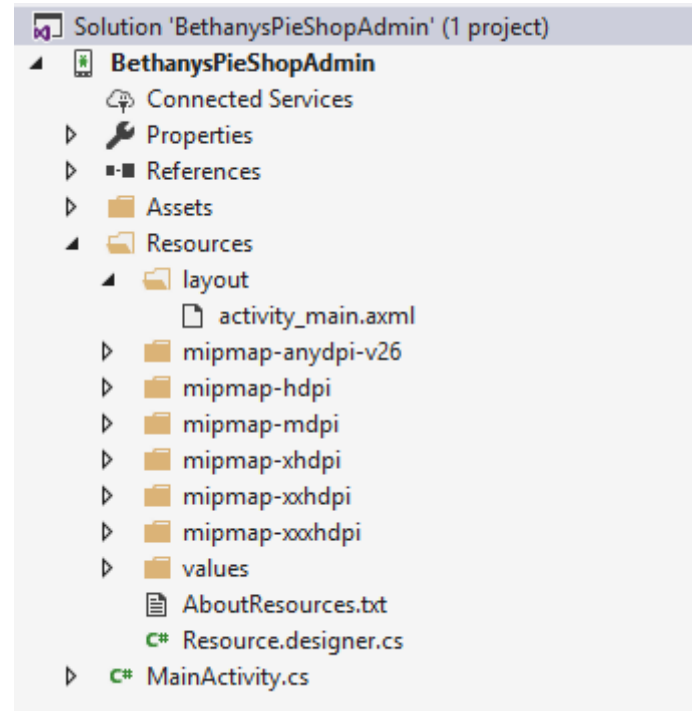
Fundamentals of Xamarin.Android



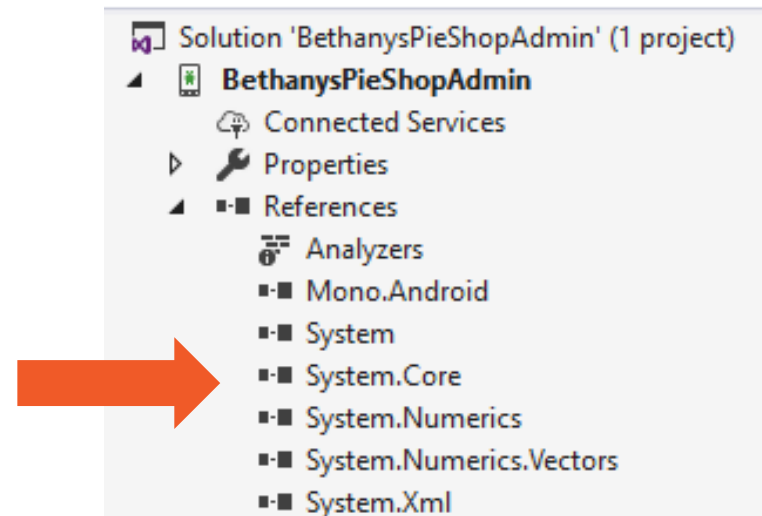
Project Templates



File → New Project



Package References



Fundamental Concepts in Android



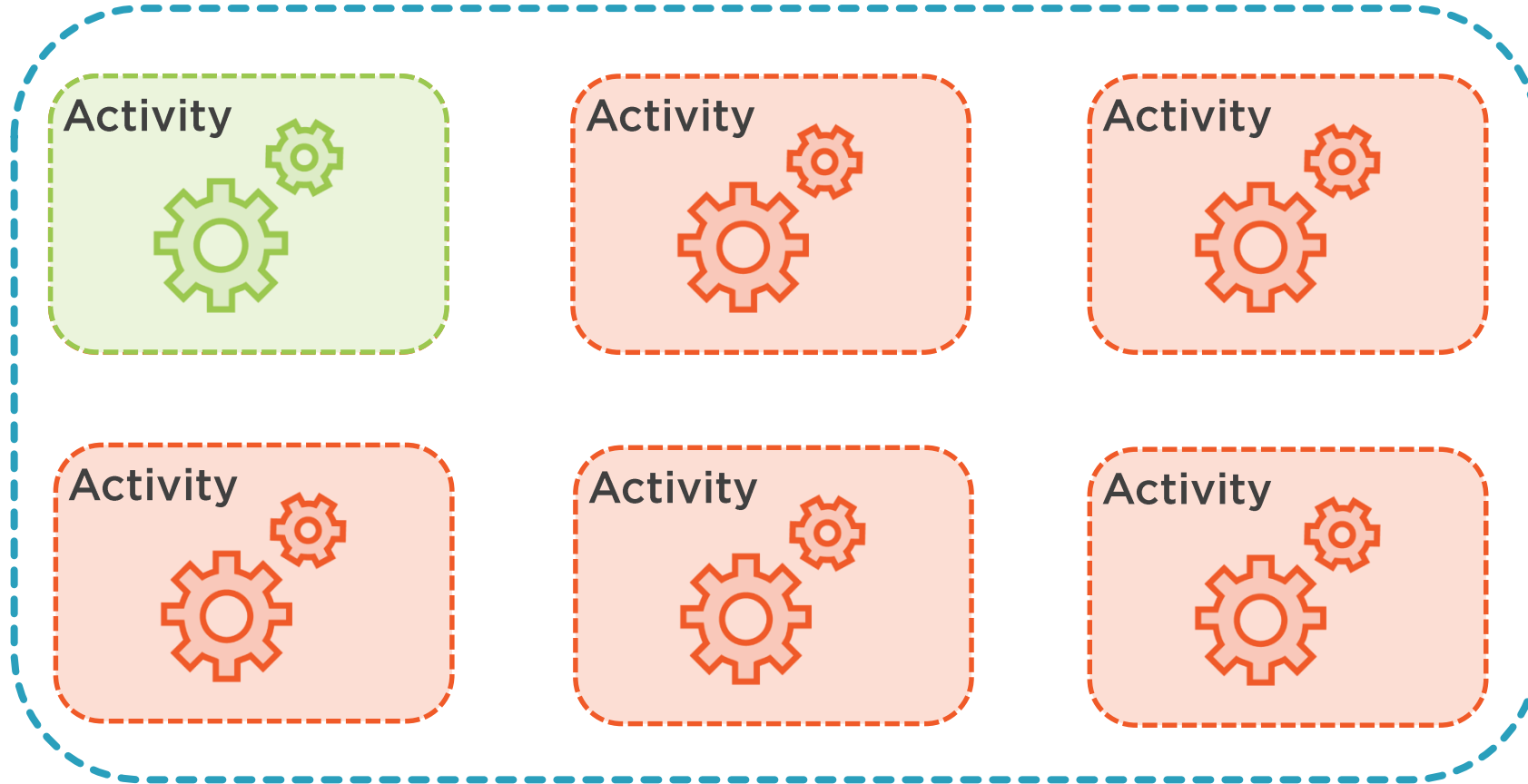
Activity



View

Android Applications

App





Inherit from base Activity

[Activity]

Can start another activity

- Same or other app

Have a lifecycle

```
[Activity(Label = "@string/app_name",  
Theme = "@style/AppTheme", MainLauncher = true)]  
public class MainActivity : AppCompatActivity  
{  
  
}
```

Default Generated Activity



Demo



Creating a new project

Looking at the default Activity



OnCreate

```
[Activity(Label = "@string/app_name", Theme =
"@style/AppTheme", MainLauncher = true)]
public class MainActivity : AppCompatActivity
{
    protected override void OnCreate
        (Bundle savedInstanceState)
    {
        base.OnCreate(savedInstanceState);
        SetContentView(Resource.Layout.activity_main);
    }
}
```



Views



Visual part



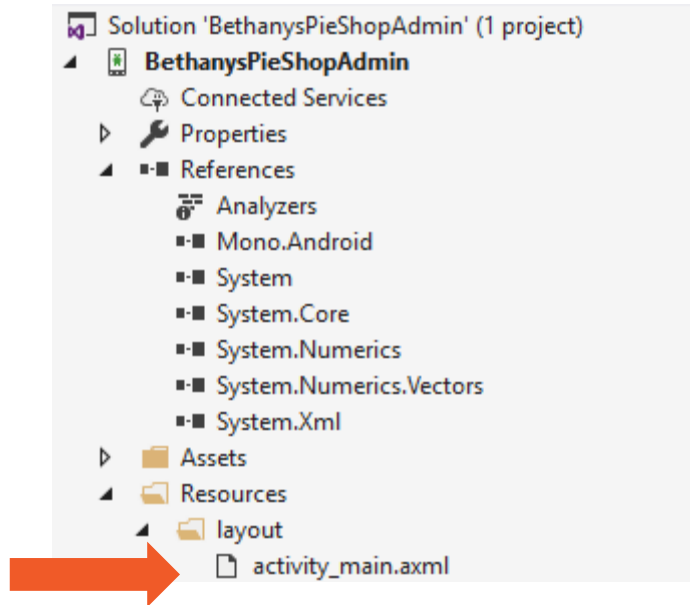
*.xml



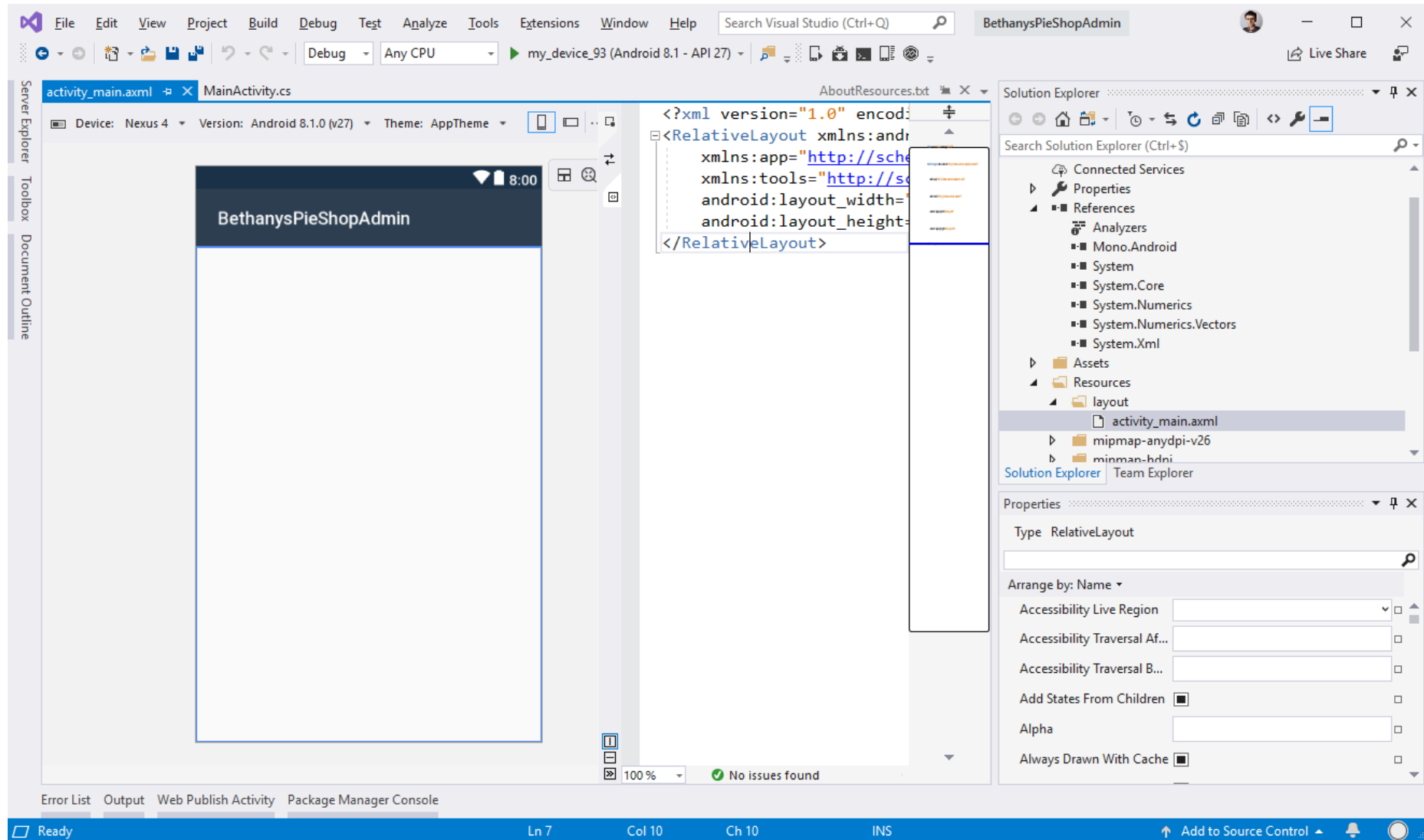
Live in
Resources\Layout



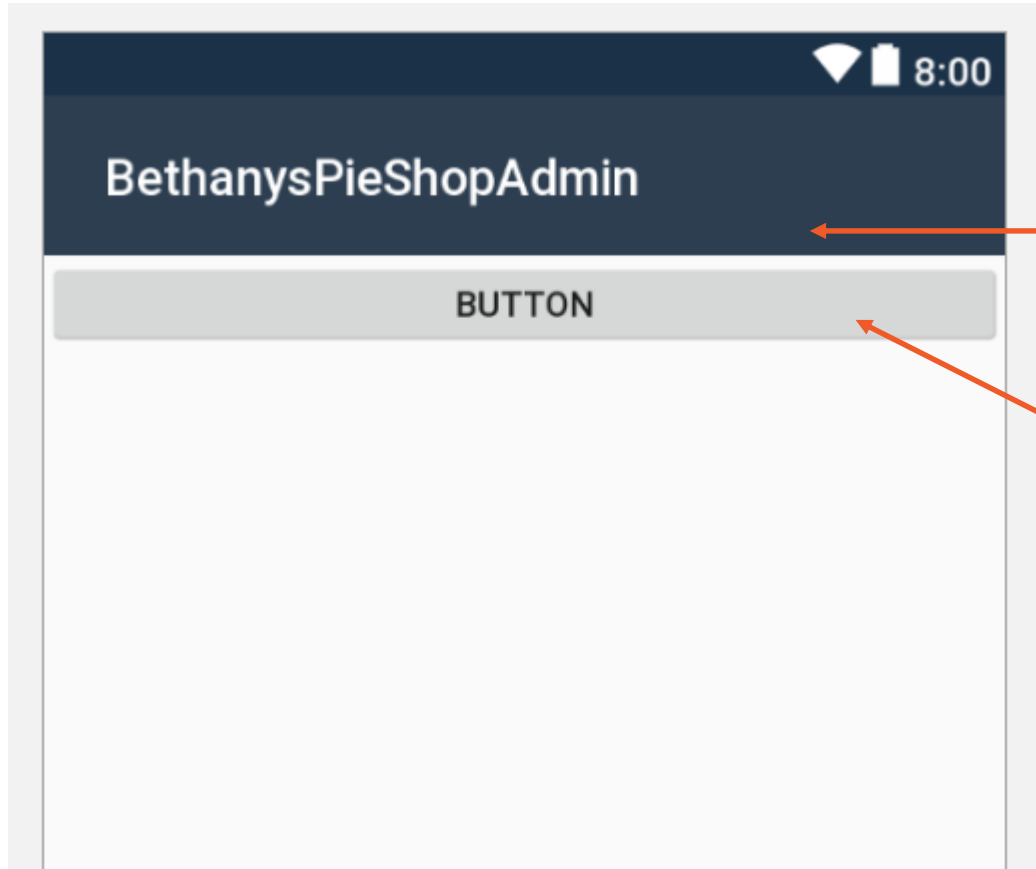
A Default View



Visual Studio Designer



AXML Views

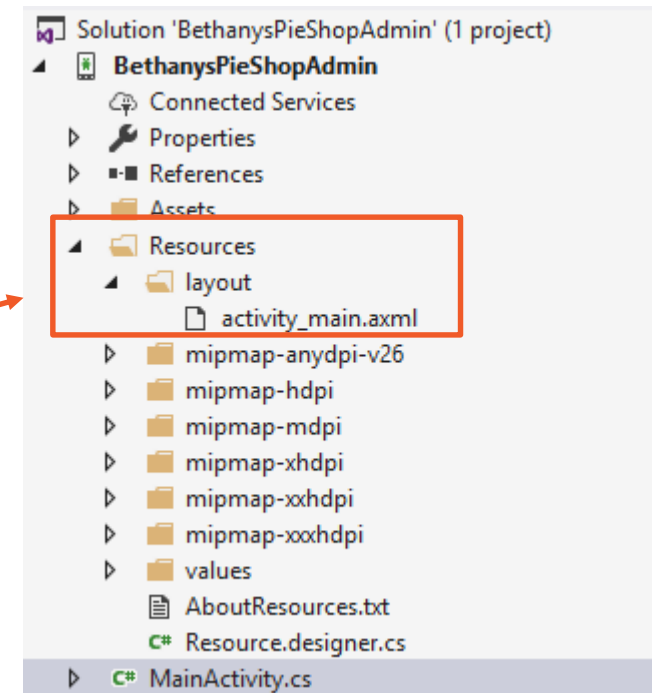


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:text="Button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minWidth="25px"
        android:minHeight="25px"
        android:id="@+id/MyButton" />
</LinearLayout>
```



Linking the View and the Activity

```
protected override void OnCreate(Bundle savedInstanceState)
{
    base.OnCreate(savedInstanceState);
    // Set our view from the "main" layout resource
    SetContentView(Resource.Layout.activity_main);
}
```



The Missing Link: Resource.Designer.cs

```
[System.CodeDom.Compiler.GeneratedCodeAttribute(tool: "Xamarin.Android.Build.Tasks", version: "1.0.0.0")]
public partial class Resource
{
    static Resource()...

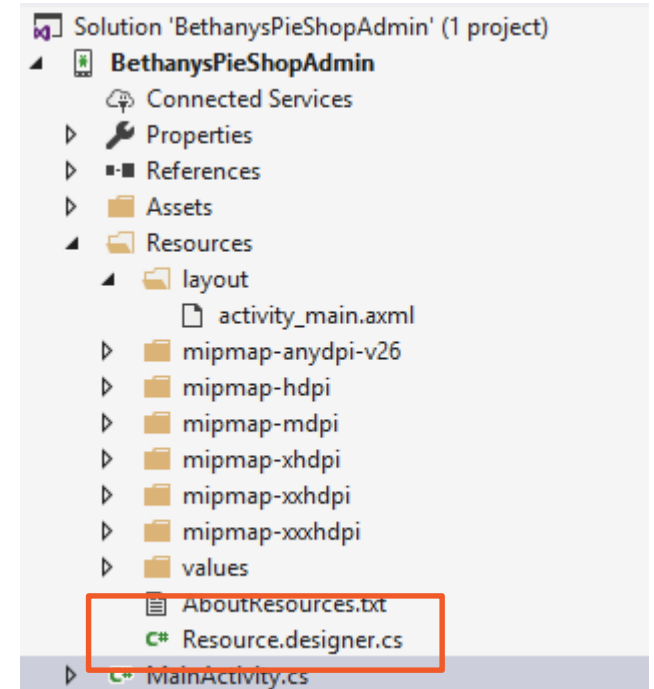
    public static void UpdateIdValues()...

    public partial class Animation...
    public partial class Animator...
    public partial class Attribute...
    public partial class Boolean...
    public partial class Color...
    public partial class Dimension...
    public partial class Drawable...
    public partial class Id...
    public partial class Integer...
    public partial class Layout
    {
        /// ...

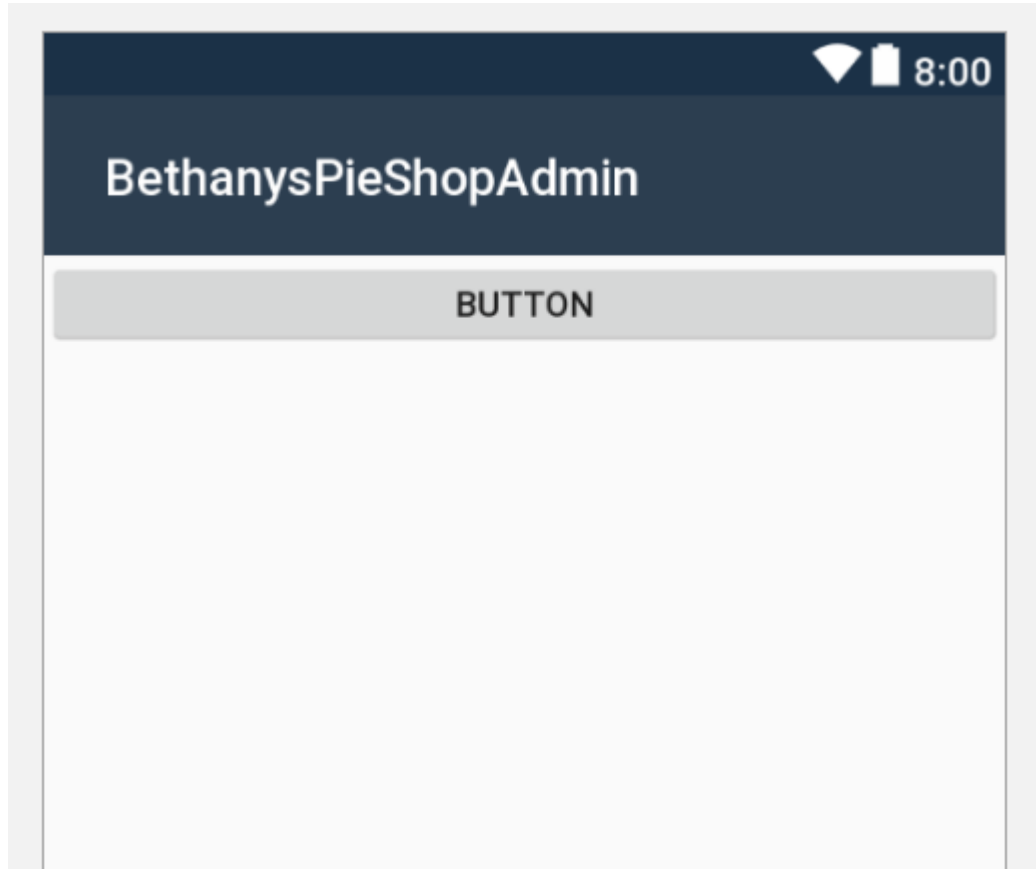
        // aapt resource value: 0x7f04001a
        public const int activity_main = 2130968602;

        static Layout()...

        private Layout()...
    }
}
```



Resource Values



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:text="Button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minWidth="25px"
        android:minHeight="25px"
        android:id="@+id/MyButton" />
</LinearLayout>
```

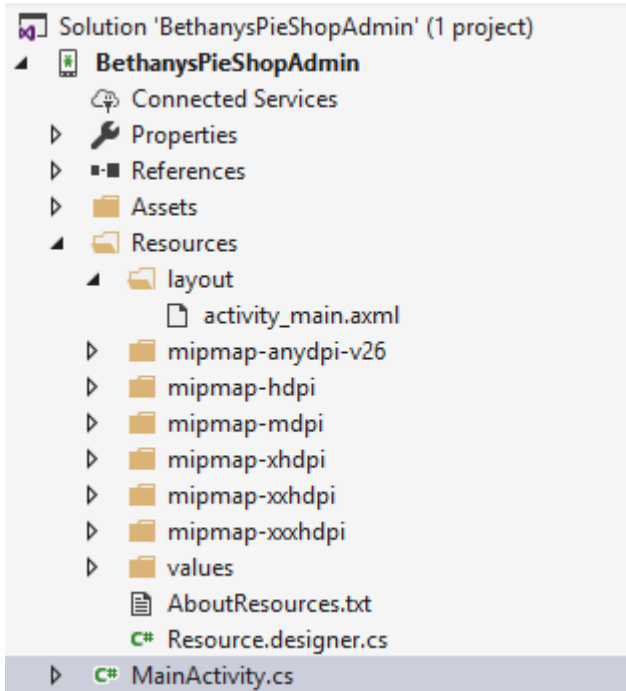


Accessing Controls from Code

```
protected override void OnCreate(Bundle savedInstanceState)
{
    base.OnCreate(savedInstanceState);
    SetContentView(Resource.Layout.activity_main);
    Button myButton = FindViewById<Button>(Resource.Id.MyButton);
    myButton.Click += MyButton_Click;
}

private void MyButton_Click(object sender, EventArgs e)
{
    //handle button click here
}
```





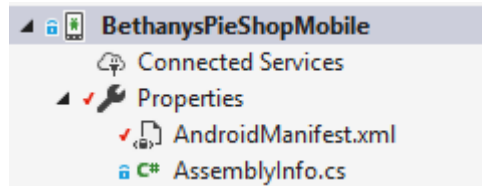
Mipmap

- App icons
- Launch screen images

Drawable

- All images

Will get listed in Resources class



Android Manifest file

- List application contents
- Activities, resources...
- Application permissions

Working with Permissions

**Access
functionalities**

**Application
manifest**

Consent



Demo



Looking at the View

Adding a widget

Accessing the widget from code

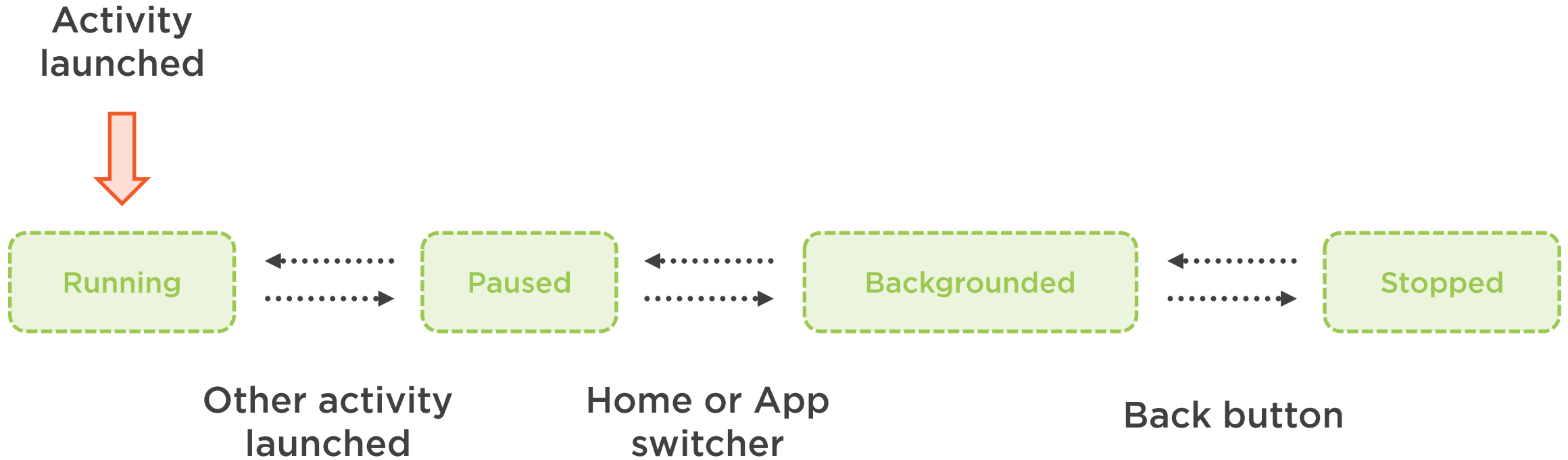


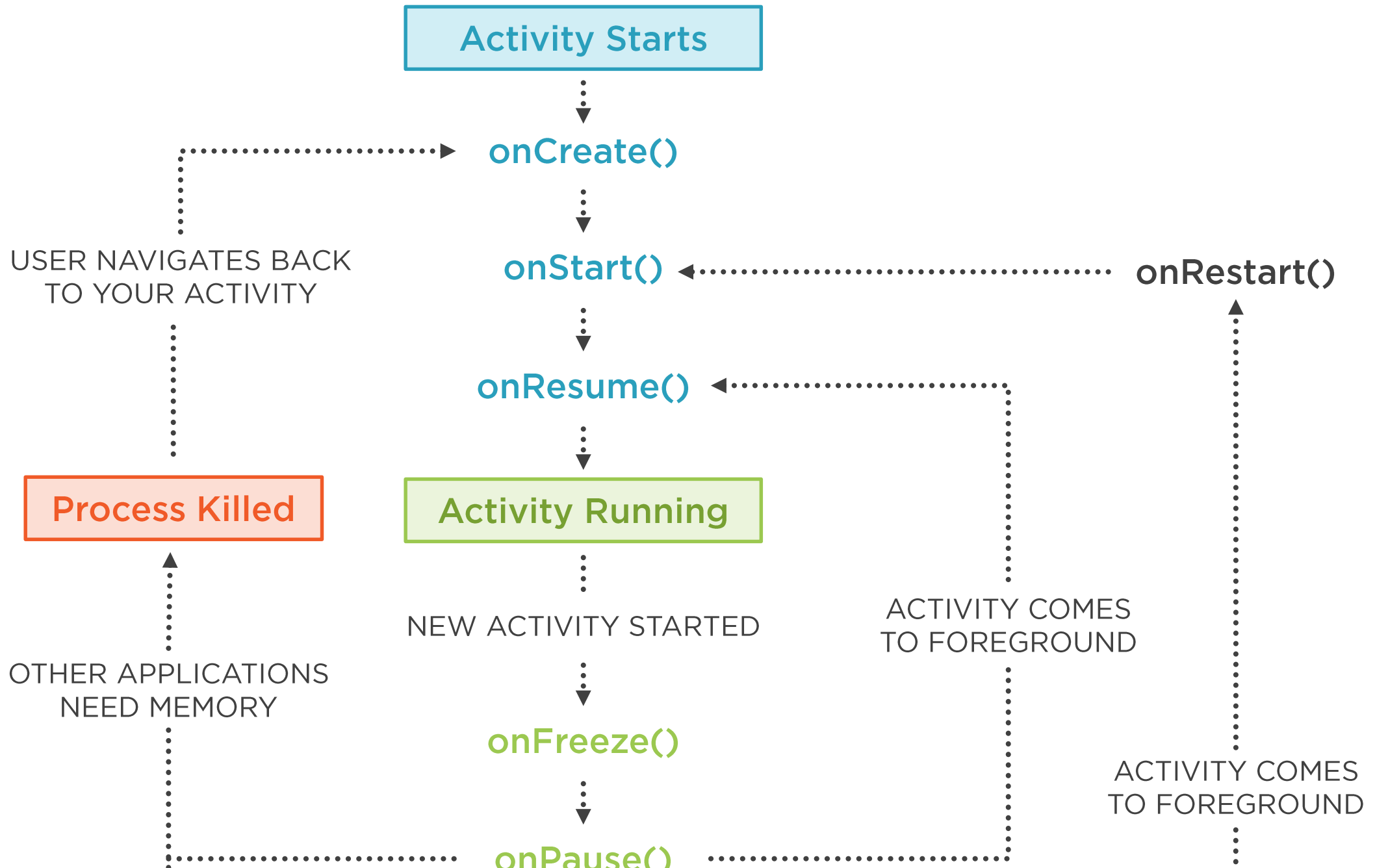


Activity lifecycle

- Number of states
- Corresponding events

Activity States





Important Lifecycle Methods

OnCreate

OnStart

OnResume

OnPause

OnStop



Demo



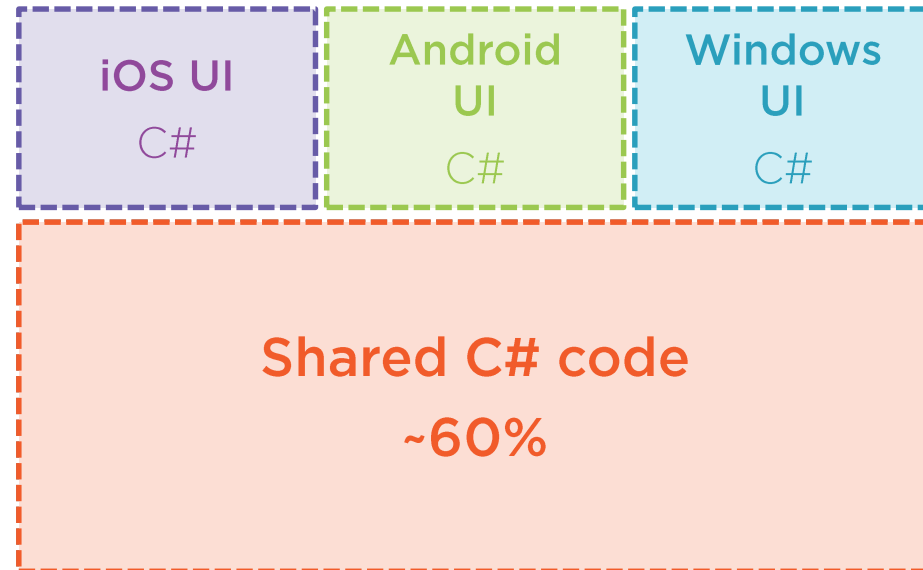
Looking at the lifecycle methods



Code-sharing with Xamarin



Remember This?



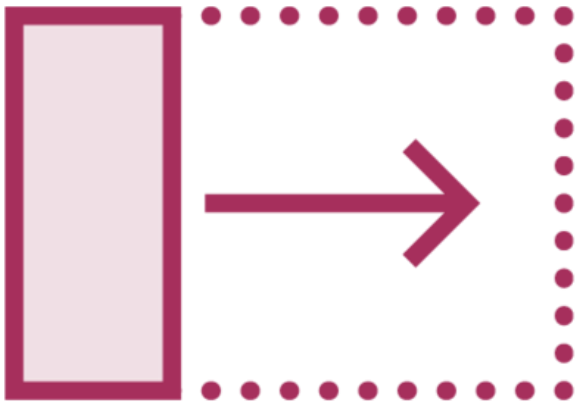
Code-sharing Options

.NET Standard

Shared Projects

**Portable Class
Library**





Models

Data Access

Services

Service Access

Demo



Creating a .NET Standard Project

Adding shared code



Summary



Activity is core of Android development

Concept of resources is vital

Code-sharing is a key aspect for Xamarin development





Up next:
Creating our first real screen

