

# Xamarin Hands-on

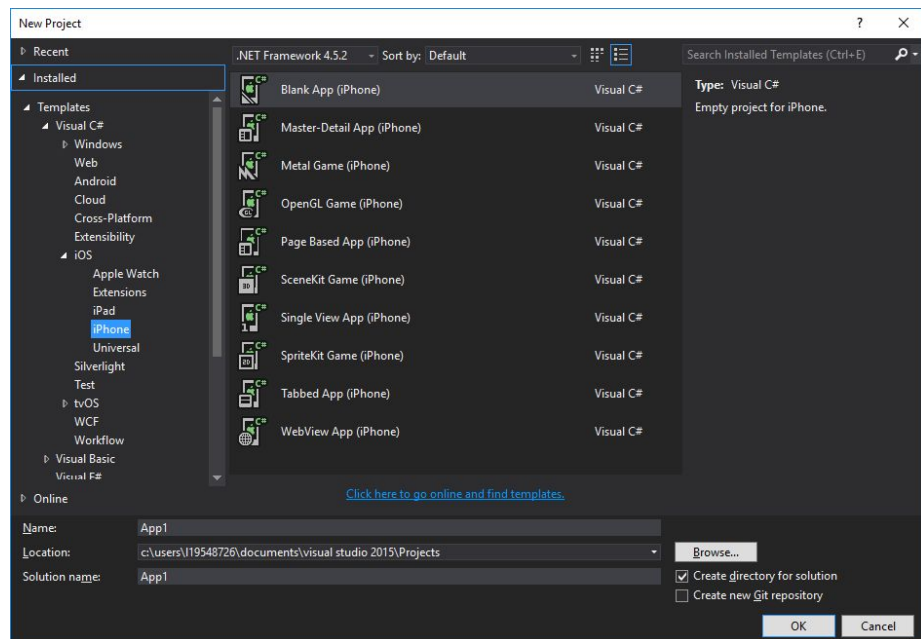
A guide through Xamarin Forms and Xamarin Native

# Why using Xamarin Native instead of Forms?

- Smaller application size
- Improved performance
- Easier to create a more “direct” native user interface
- Code syntax more similar to “true” native syntax



# Adding a new Xamarin iOS Native project



# Application setup (iOS)

- References & NuGets
- Application Bootstrapper - Registration of platform specific concretizations



# View Types

iOS	Android
UIViewController	Activity
UINavigationController	Activity with FragmentManager
UIViewController	Fragment
...	...



# Control naming iOS vs Android

iOS	Android
UIButton	Button
UILabel	TextView
UITextField	EditText
UISwitch	CheckBox
UIImageView	ImageView

# Creating the first UIViewController

- Create UI controls programmatically in ViewDidLoad
- Remember to add created views to the View
- “No” logic in the actual View, just delegate to the view model
- Responding to notifications from the ViewModel
- Displaying the view controller as the first view from AppDelegate



# Opening a view from a ViewModel

- Implement `OpenPage()`

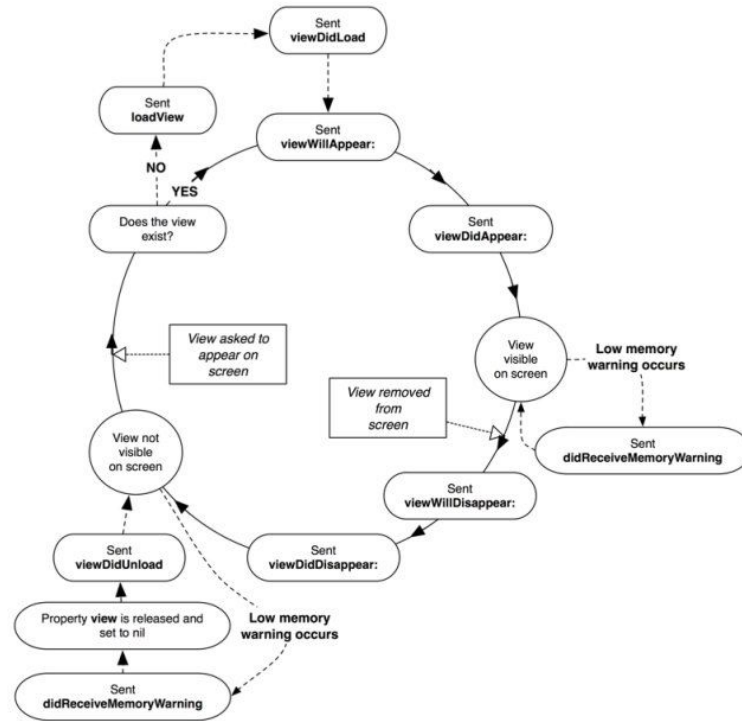




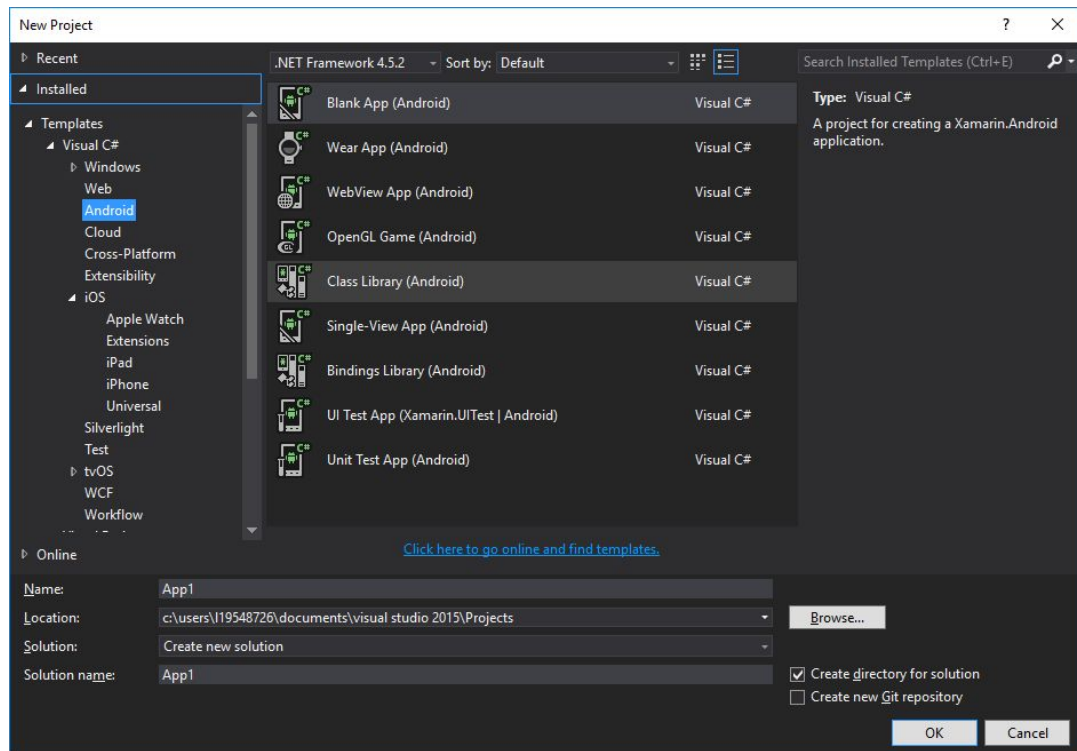
# UITableView vs RecyclerView (ListView)

iOS	Android
UITableView	RecyclerView
UITableViewSource	Adapter
UITableViewCell	ViewHolder & Cell layout

# iOS view lifecycle



# Adding a new Xamarin Android native project



# Application setup (Android)

- References & NuGets
- Application Bootstrapper - Registration of platform specific concretizations
- Directory structure



# View Types

iOS	Android
UIViewController	Activity
UINavigationController	Activity with FragmentManager
UIViewController	Fragment
...	...



# Control naming iOS vs Android

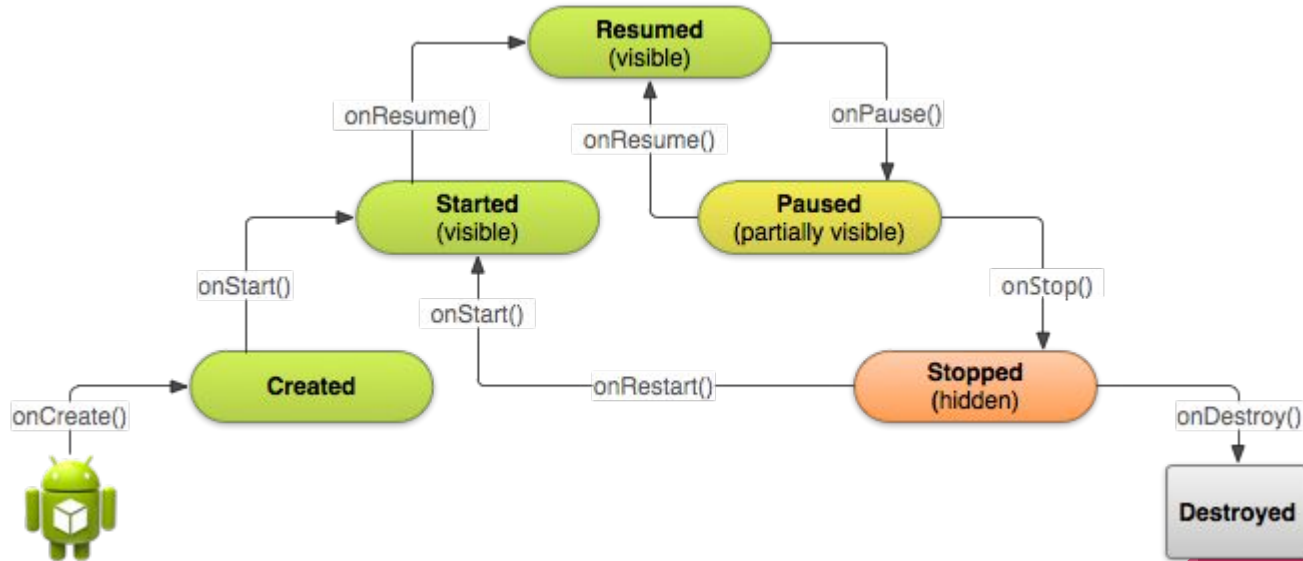
iOS	Android
UIButton	Button
UILabel	TextView
UITextField	EditText
UISwitch	CheckBox
UIImageView	ImageView

# Creating the first Activity

- Create UI controls via the designer (XML)
- Connecting controls from the layout to the code
- “No” logic in the actual View, just delegate to the view model



# Android Activity lifecycle





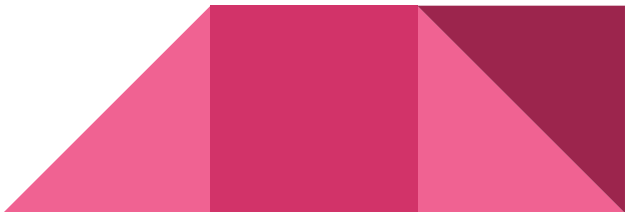
# Opening a view from a ViewModel

- Implement `OpenPage()`
- Create an Activity using `Class.FromType`, not `Activator`



# UITableView vs RecyclerView (ListView)

iOS	Android
UITableView	RecyclerView
UITableViewSource	Adapter
UITableViewCell	ViewHolder & Cell layout



# Can we extend this even further?

Let's try to convert it to WPF!



# Download the code & contact

<https://github.com/MikaelStalvik/SwenugDemo>

mikael.stalvik@lindex.com

