Xamarin.Forms

Mike James
Let us know what you think

Click ‘engage’ to rate session.
Rate 12 sessions to get the supercool GOTO reward

Join the conversation gotoldn
Who is this guy?
Cross platform developer
Architecting Mobile Apps
Build Apps Multiple Times

- iOS App
  - Objective-C
  - XCode

- Android App
  - Java
  - Eclipse

- Windows App
  - C#
  - Visual Studio
Lowest Common Denominator
Xamarin Standard App Architecture

- Business logic, data models and common code shared between all platforms.
- All platform-specific UI and API calls are in C# as well.
- Retain fine-grain control app user interface
- Recommended for sophisticated UX requirements (complicated gestures, animations, design)
Native Performance

Xamarin.iOS does full Ahead Of Time (AOT) compilation to produce an ARM binary for Apple’s App Store.

Xamarin.Android takes advantage of Just In Time (JIT) compilation on the Android device.
Windows APIs
iOS APIs | 100% Coverage
Android APIs | 100% Coverage
Meet Xamarin.Forms

Build native UIs for iOS, Android and Windows Phone from a single, shared C# codebase.
Considerations

• Not for all types of apps
• Is your app very customised?
• Great for data driven apps
• Utility apps
• Also supports maps
Xamarin App Architectures

Option for OS – specific UI/UX elements
Recommended for: Sophisticated UX (complicated gestures, animations, design)

Option for 100% code share
Recommended for: forms-based, apps with a lot of data entry screens
Xamarin.Forms

Quickly and easily build native user interfaces using shared code

Xamarin.Forms elements map to native controls and behaviours

Mix-and-match Xamarin.Forms with native APIs
What’s Included

- 40+ Pages, Layouts, and Controls
  - Build from code behind or XAML
- Two-way Data Binding
- Navigation
- Animation API
- Dependency Service
- Messaging Center

Shared UI Code

Shared C# Backend
Pages

Content

MasterDetail

Navigation

Tabbed

Carousel
Layouts

Stack
Absolute
Relative
Grid
ContentView
ScrollView
Frame
# Controls

<table>
<thead>
<tr>
<th>ActivityIndicator</th>
<th>BoxView</th>
<th>Button</th>
<th>DatePicker</th>
<th>Editor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry</td>
<td>Image</td>
<td>Label</td>
<td>ListView</td>
<td>Map</td>
</tr>
<tr>
<td>OpenGLView</td>
<td>Picker</td>
<td>ProgressBar</td>
<td>SearchBar</td>
<td>Slider</td>
</tr>
<tr>
<td>Stepper</td>
<td>TableView</td>
<td>TimePicker</td>
<td>WebView</td>
<td>EntryCell</td>
</tr>
<tr>
<td>ImageCell</td>
<td>SwitchCell</td>
<td>TextCell</td>
<td>ViewCell</td>
<td></td>
</tr>
</tbody>
</table>
Demo
Pages

• Single screen of content
• ContentPage holds one visual element
Layout

- Layouts handle child elements
- Layouts come in two types: managed and unmanaged
Read the book

Charles Petzold
Xamarin

Creating Mobile Apps with Xamarin.Forms
Cross-platform C# programming for iOS, Android, and Windows Phone

CHARLES PETZOLD
Where to find some interesting samples

- https://github.com/MikeCodesDotNet/
- https://github.com/pierceboggan
Looking for a Forms weekend project?

beerdrink.in
Questions?
Thanks !