

.Net MAUI

Cross-Platform Mobile Application

Development

Length: 5 Days

Developing a mobile application for just one platform is becoming a thing of the past. Companies expect their apps to be supported on iOS, Android and Windows Phone, while leveraging the best native features on all three platforms. .Net MAUI's tools help ease this problem by giving developers a single toolset to target all three platforms. The main goal of this course is to equip you with knowledge to successfully analyze, develop, and manage .Net MAUI cross-platform projects using the most efficient, robust, and scalable implementation patterns.

After completing this course, students will be able to:

- Learn to deliver high-performance native apps that leverage platform specific acceleration, complied for native performance
- Learn development techniques that will allow you to use and create custom layouts for cross-platform UI
- Gain the knowledge needed to become more efficient in testing, deploying, and monitoring your applications
- Implement application life cycle management concepts to manage cross-platform projects

What You Will Learn

- Share C# code across platforms and call native Objective-C or Java libraries from C#
- Submit your app to the Apple App Store and Google Play
- Use the out-of-the-box services to support third-party libraries
- Find out how to get feedback while your application is used by your users
- Create shared data access using a local SQLite database and a REST service
- Test and monitor your applications
- Gain memory management skills to avoid memory leaks and premature code cycles while decreasing the memory print of your applications
- Integrate network resources with cross-platform applications
- Design and implement eye-catching and reusable UI components without compromising on nativity in mobile applications

Audience Profile

Mobile application developers wanting to develop skills required to steer cross-platform applications using .Net MAUI.

.Net MAUI

Cross-Platform Mobile Application Development

Length: 5 Days

.Net MAUI : Cross-Platform Mobile Application

Development

Table of Contents

Xamarin: Cross-Platform Mobile Application

Development

Credits

Preface

What this learning path covers

What you need for this learning path

- **Module 1:**
- **Module 2:**
- **Module 3:**

Who this learning path is for

Reader feedback

Customer support

Downloading the example code

Errata

Piracy

Questions

Module 1

1. Setting Up .Net MAUI

- The Xamarin tools
- Installing Xcode
- Installing .Net MAUI
- Choosing a .Net MAUI license
- Setting up the Android emulator
- Enrolling in the iOS Developer Program
- Registering as a Google Play developer
- Summary

2. Hello, Platforms!

- Building your first iOS application
- Understanding Apple's MVC pattern
- Using the iOS designer
- Building your first Android application
- Android activities
- .Net MAUI's Android designer
- Summary

3. Code Sharing between iOS and Android

- Learning the MVVM design pattern
 - Implementing MVVM in an example
- Comparing project organization strategies
 - Setting up a cross-platform solution
- Working with Portable Class Libraries
 - Using PCLs in .Net MAUI
- Using preprocessor statements
- Simplifying dependency injection
- Implementing Inversion of Control
- Summary

4. XamChat – a Cross-platform App

- Starting our sample application concept
- Developing our Model layer
- Writing a mock web service
- Writing the ViewModel layer
 - Implementing our LoginViewModel class
 - Implementing our RegisterViewModel class
 - Implementing our FriendViewModel class
 - Implementing our MessageViewModel class
- Writing unit tests
 - Setting up a new project for unit tests
 - Writing assertions
- Summary

5. XamChat for iOS

- Understanding the basics of an iOS app
- Using UINavigationController
 - Methods in Navigation Controllers
 - Setting up a Navigation Controller
- Implementing the login screen
 - Creating a LoginController class
 - Modifying the controller's layout
 - Registering and subscribing view models and services
- Using segues and UITableView
- Adding a friends list screen
- Adding a list of messages
- Composing messages
- Summary

.Net MAUI

Cross-Platform Mobile Application Development

Length: 5 Days

- Summary

6. XamChat for Android

- Introducing Android Manifest
 - Setting up the Manifest
 - ✓ Common manifest permissions
 - Creating and implementing the application class
- Adding a login screen
 - Layouts and ViewGroups in Android
 - Implementing the login functionality
- Using ListView and BaseAdapter
 - Implementing the conversations screen
 - Setting up the adapter
- Implementing the friends list
- Composing messages
- Summary

7. Deploying and Testing on Devices

- iOS provisioning
 - Prerequisites for deploying to iOS
 - Creating a provisioning profile
- Android device settings
- Understanding the linker
- Understanding AOT compilation
- Avoiding common memory pitfalls
 - Garbage collector
 - Memory leaks
 - Accessing objects disposed by GC
- Summary

8. Web Services with Push Notifications

- Learning Windows Azure
- Setting up your Azure account
- Exploring Azure Mobile Services
- Creating tables and scripts
- Adding a backend to XamChat
 - Adding the Azure Mobile Services NuGet package
- Using the Apple Push Notification service
 - Setting up proper provisioning
 - Setting up your provisioning profile
 - Setting up a certificate signing request
 - Making client-side changes for push notifications
- Implementing Google Cloud Messaging

9. Third-party Libraries

- The .Net MAUI Component Store
 - The most common components
- Porting existing C# libraries
- Objective-C bindings
 - Working with Objective Sharpie
- Java bindings
- Summary

10. Contacts, Camera, and Location

- Introducing .Net MAUI.Mobile
- Accessing contacts
 - Retrieving contacts on Android
- Looking up GPS location
 - Implementing GPS location on Android
- Accessing the photo library and camera
 - Accessing photos on Android
- Summary

11. .Net MAUI Forms

- Creating Hello World in .Net MAUI.Forms
- Understanding the architecture behind .Net MAUI.Forms
- Using XAML in .Net MAUI.Forms
- Using data binding and MVVM
- Summary

12. App Store Submission

- Following the iOS App Store Review Guidelines
 - General rules
 - Incorrect and incomplete information
 - Content present in the application
 - Apple's 70/30 revenue share
 - General Tips
- Submitting an app to the iOS App Store
 - Creating a distribution provisioning profile
 - Adding your app to iTunes Connect
 - Making an iOS binary for the App Store
- Signing your Android applications
- Submitting the app to Google Play
 - Google Play Developer Program Policies
- Tips to make a successful mobile app
- Summary

.Net MAUI

Cross-Platform Mobile Application Development

Length: 5 Days

Module 2

1. One Ring to Rule Them All

- Introduction
- Creating a cross-platform solution
 - Getting ready
 - How to do it...
 - How it works...
 - See also
- Creating a cross-platform login screen
 - How to do it...
 - How it works...
 - There's more...
- Using common platform features
 - How to do it...
 - How it works...
 - See also
- Authenticating with Facebook and Google providers
 - How to do it...
 - ✓ For the Android project
 - ✓ For the iOS project
- How it works...
- See also

2. Declare Once, Visualize Everywhere

- Introduction
- Creating a tabbed-page cross-platform application
 - How to do it...
 - How it works...
 - See also
- Adding UI behaviors and triggers
 - How to do it...
 - How it works...
 - See also
- Configuring XAML with platform-specific values
 - How to do it...
 - How it works...
 - See also
- Using custom renderers to change the look and feel of views
 - How to do it...
 - How it works...
 - See also

3. Native Platform-Specific Views and Behavior

- Introduction
- Showing native pages with renderers
 - How to do it...
 - How it works...
 - See also
- Attaching platform-specific gestures
 - How to do it...
 - How it works...
 - See also
- Taking an in-app photo with the native camera page
 - How to do it...
 - How it works...
 - See also

4. Different Cars, Same Engine

- Introduction
- Sharing code between different platforms
 - How to do it...
 - How it works...
 - See also
- Using the dependency locator
 - How to do it...
 - How it works...
 - See also
- Adding a third-party Dependency Injection Container
 - How to do it...
 - How it works...
 - See also
- Architecture design with Model-View-ViewModel (MVVM) pattern
 - How to do it...
 - How it works...
 - There's more...
 - See also
- Using the event messenger
 - How to do it...
 - How it works...
 - See also
- Adding localization
 - How to do it...

.Net MAUI

Cross-Platform Mobile Application Development

Length: 5 Days

- How it works...
- There's more...

5. Dude, Where's my Data?

- Introduction
- Creating a shared SQLite data access
 - How to do it...
 - How it works...
 - See also
- Performing CRUD operations in SQLite
 - How to do it...
 - How it works...
 - See also
- Consuming REST web services
 - How to do it...
 - How it works...
- Leveraging native REST libraries and making efficient network calls
 - How to do it...
 - How it works...
 - There's more...

6. One for All and All for One

- Introduction
- Creating cross-platform plugins
 - How to do it...
 - How it works...
 - There's more...
 - See also
- Taking or choosing photos
 - How to do it...
 - How it works...
 - There's more...
 - See also
- Getting the GPS location
 - How to do it...
 - How it works...
 - See also
- Show and schedule local notifications
 - How to do it...
 - How it works...
 - There's more...

7. Bind to the Data

- Introduction
- Binding data in code
 - How to do it...
 - How it works...

- Binding data in XAML
 - How to do it...
 - How it works...
- Configuring two-way data binding
 - How to do it...
 - How it works...
 - See also
- Using value converters
 - How to do it...
 - How it works...
 - See also

8. A List to View

- Introduction
- Displaying a collection and selecting a row
 - How to do it...
 - How it works...
 - See also
- Adding, removing, and refreshing items
 - How to do it...
 - How it works...
 - There's more...
 - See also
- Customizing the row template
 - How to do it...
 - How it works...
 - There's more...
 - See also
- Adding grouping and a jump index list
 - How to do it...
 - How it works...
 - There's more...
 - See also

9. Gestures and Animations

- Introduction
- Adding gesture recognizers in XAML
 - How to do it...
 - How it works...
 - There's more...
 - See also
- Handling gestures with native platform renderers
 - How to do it...
 - How it works...
- Adding cross-platform animations
 - How to do it...
 - How it works...
 - There's more...

.Net MAUI

Cross-Platform Mobile Application Development

Length: 5 Days

10. Test Your Applications, You Must

- Introduction
- Creating unit tests
 - How to do it...
 - How it works...
- Creating acceptance tests with .Net MAUI. UITest
 - How to do it...
 - How it works...
 - There's more...
 - See also
- Using the .Net MAUI .UITest REPL runtime shell to test the UI
 - How to do it...
 - How it works...
 - See also
- Uploading and running tests in .Net MAUI Test Cloud
 - How to do it...
 - How it works...
 - There's more...
 - See also

11. Three, Two, One – Launch and Monitor

- Introduction
- Using .Net MAUI Insights
 - How to do it...
 - How it works...
 - There's more...
- Publishing iOS applications
 - How to do it...
 - How it works...
- Publishing Android applications
 - How to do it...
 - How it works...
 - There's more...
 - See also
- Publishing Windows Phone applications
 - How to do it...
 - How it works...
 - There's more...

Module 3

1. Developing with .Net MAUI

- Cross-platform projects with .Net MAUI
 - .Net MAUI as a platform
 - .Net MAUI as a product
- Target platforms
 - .Net MAUI on Android
 - .Net MAUI on iOS
 - Windows Runtime apps
- Setting up the development environment
 - Choosing the right development OS
 - ✓ .Net MAUI Studio setup and configuration
 - ✓ Visual Studio setup and configuration
- Emulator options
 - Emulators for Android
 - iOS emulation
- A typical .Net MAUI solution structure
 - Portable class libraries
 - Shared projects
 - .Net MAUI.Forms
 - NuGet packages
 - Components
- Quality in cross-development
 - Reusability
 - Abstraction
 - Loose-coupling
 - Nativity
- Summary

2. Memory Management

- Application Component lifecycle
 - Activity lifecycle (Android)
 - ✓ Active/Running
 - ✓ Paused
 - ✓ Backgrounded
 - ✓ Stopped
 - ✓ Restarted
 - Application lifecycle (iOS)
- Garbage collection
 - GC on .Net MAUI projects
 - ✓ SGen garbage collector
 - ✓ Boehm garbage collector (iOS only)

.Net MAUI

Cross-Platform Mobile Application Development

Length: 5 Days

- Platform-specific concepts
 - Object reference types
 - Automatic Reference Counting (ARC)

- Troubleshooting and diagnosis
 - .Net MAUI Profiler
 - ✓ Allocations instrument
 - ✓ Time Profiler
 - Device Monitor (Android only)
 - Instruments (iOS only)
 - Monotouch Profiler (iOS only)
- Patterns and best practices
 - Disposable objects
 - The lapsed listener problem
 - Weak references
 - Cross-domain objects
 - Cyclic references (cycles)
- Summary

- 3. Asynchronous Programming**
- Multithreading on .Net MAUI
 - Single thread model
 - Task-based Asynchronous Pattern
 - Concurrency model on iOS
- Asynchronous methods
 - Continuation
 - Cancellation
 - Progress
 - Task batches
- Parallel execution
- Patterns and best practices
 - Async pattern conversions
 - Multi-threading with tasks
 - Exception handling
 - Initialization pattern
 - Semaphores
- Background tasks
 - Background tasks on iOS
 - Services (Android only)
- Summary

- 4. Local Data Management**
- Data in mobile applications
 - State
 - App data
 - Local files
 - External data
- Application data
 - Installation directory
 - ✓ Android
 - ✓ iOS

- Local storage
 - ✓ Android
 - ✓ iOS
- Temporary storage
- Local filesystem
- SQLite
- Patterns and best practices
 - Application preferences
 - File picker
- Backup/Roaming
 - Android and Backup API
 - iOS and ubiquitous storage
- Summary

- 5. Networking**
- Connected apps
- Web services
 - Transport
 - Messaging
 - SOAP/XML services
 - RESTful services
 - OData and OAuth
 - ✓ OData
 - ✓ OAuth
- SignalR
- Patterns and best practices
 - Async conversions
 - Data model abstraction
 - Service cache
- Platform-specific concepts
 - Permissions
 - NSURLConnection/NSURLSession (iOS Only)
 - Background downloads
 - Push notifications
- Cloud integration
 - Azure Mobile Services
 - Azure offline data
 - Azure authentication
- Summary

- 6. Platform Extras**
- Content sharing
 - File pickers and contracts (Windows Store apps)

.Net MAUI

Cross-Platform Mobile Application Development

Length: 5 Days

- Document Provider extensions (iOS)
 - ContentProvider and ContentResolver (Android)
- Peripherals
 - Bluetooth
 - Wi-Fi Direct
 - Near Field Communication
 - Location data
 - Android location and Google Play services
 - Location services on iOS
 - Location data on Windows Runtime
 - Geofencing
 - Native libraries
 - Managed callable wrappers (Android)
 - Linking versus binding (iOS)
 - Summary
- #### 7. View Elements
- Design philosophy
 - User expectations
 - Platform imperatives
 - Hardware dependency
 - ✓ Design metrics on Android
 - ✚ Design metrics on iOS
 - ✚ Design metrics on Windows Runtime
 - Design elements
 - The basic layout
 - Navigation
 - ✓ Horizontal navigation
 - ✓ Vertical navigation
 - ✓ Jump navigation
 - Content elements
 - ✓ Collection views
 - ✚ UITableView (iOS)
 - ✚ UICollectionView (iOS)
 - ✚ ListView (Android)
 - ✚ GridView (Android)
 - ✚ CardView (Android)
 - ✚ ListView and Listbox (Windows Phone)
 - ✚ GridView (Windows Phone)
 - ✚ Virtualizing panels (Windows Phone)
 - ✓ Modal views
 - ✚ Popover and alerts (iOS)
- ✚ Flyout, popups, and menus (Windows Phone)
- ✚ Dialogs (Android)
- ✓ Text views
 - ✓ Web views
- Feedback
 - ✓ Indeterminate progress
 - ✓ Determinate progress
- User interaction
 - Interactive controls
 - ✓ Text input
 - ✓ Dropdown selection
 - ✓ Option selection
 - Gestures
- Summary
- #### 8. .Net MAUI.Forms
- Under the hood
 - Anatomy of .Net MAUI.Forms
 - Project structure
 - Components
 - Pages
 - ✓ Tabbed page
 - ✓ The MasterDetail page
 - ✓ NavigationPage
 - ✓ CarouselPage
 - ✓ ContentPage
 - Layouts
 - Views
 - Extending forms
 - Styles
 - Triggers and behaviors
 - Custom renderers
 - Patterns and best practices
 - Messaging infrastructure
 - Dependency injection
 - Shared project versus portable project
 - Platform-specific fine-tuning
 - Summary
- #### 9. Reusable UI Patterns
- Visual assets
 - Text resources
 - ✓ .Net MAUI.Android
 - ✓ .Net MAUI.iOS
 - ✓ Windows Phone
 - Image resources

.Net MAUI

Cross-Platform Mobile Application Development

Length: 5 Days

- Adaptive visual assets
 - Reusable assets
 - Localization
 - Locale and culture
 - Windows Phone
 - .Net MAUI.iOS
 - .Net MAUI.Android
 - .Net MAUI.Forms
 - Architectural patterns
 - MVC
 - ✓ iOS app architecture
 - MVVM
 - ✓ Windows Runtime
 - ✓ MVVM on .Net MAUI.iOS and .Net MAUI.Android
 - ✓ MVVM with .Net MAUI.Forms
 - Summary
 - Live telemetry
 - .Net MAUI Insights
 - Application Insights
 - Summary
- #### 10. ALM – Developers and QA
- Development pipeline
 - Troubleshooting and diagnostics
 - Unit testing
 - Platform-agnostic unit tests
 - Platform-specific unit tests
 - UI testing
 - .Net MAUI.UITests and .Net MAUI Test Cloud
 - .Net MAUI Test Recorder
 - Coded UI tests (Windows Phone)
 - Calabash
 - Summary
- #### 11. ALM – Project and Release Management
- Source control
 - TFVC
 - Git
 - TFS/Git scenarios
 - ✓ Git bridge
 - ✓ NuGet packages
 - Subversion (SVN)
 - Continuous integration
 - Visual Studio Team Services
 - TeamCity
 - Other
 - Automated testing
 - Beta deployment
 - HockeyApp
- #### 12. ALM – App Stores and Publishing
- Release packages
 - .Net MAUI.Android app package (.apk)
 - ✓ Disabling debugging
 - ✓ Linking
 - ✓ Packing options
 - ✓ Packaging
 - .Net MAUI.iOS app bundle (.ipa)
 - ✓ Build options
 - ✓ Linking
 - ✓ Provisioning profile
 - Windows Phone app package (.appx)
 - Distribution options
 - App store(s)
 - Ad-hoc
 - Line of Business apps
 - Private channel distribution (Android)
 - Apple Developer Enterprise Program
 - Windows Phone private distribution
 - Summary