

Xamarin Mobile Developer Certification Study Guide

V2.0.0 2018-03-06

ListViews and Adapters in Android [AND110]

Populate a ListView using an ArrayAdapter

- Add a ListView to an Activity
- Use ArrayAdapter to populate a ListView
- Explore the limitations of ArrayAdapter

Handle list-item click events

- Respond to taps on a ListView
- Determine which list item was clicked

Implement a custom adapter

- Inflate a layout defined in axml using LayoutInflater
- Create a custom Adapter

Use layout recycling and the view-holder pattern

- Reuse inflated layouts to reduce memory usage
- Cache view references to increase performance

Enable fast scrolling and code a section indexer

- Enable ListView fast scrolling
- Display a section indexer on a ListView

Fundamentals of TableViews [IOS110]

Explore Table Views

- Add a Table View to your UI
- Display data in a Table View

Utilize built-in cell styles

- Specify a built-in cell style to be used in a table view
- Display your data using the built-in cell styles

Add selection behavior

- Add an accessory view to a cell
- Respond to taps on the accessory view

Implement cell reuse

- Enable cell reuse in the designer
- Enable cell reuse programmatically

Customizing TableViews [IOS115]

Customize table view cells in code

- View the anatomy of a cell
- Customize the default cell styles
- Create a custom cell

Customize table view cells in the designer

- Choose static or dynamic cells based on your app's data
- Design a custom cell using the designer

Group data in the table view

- Changing the grouping visualization for a table view
- Display an index
- Add headers and footers
- Customize headers and footers

Toolbar and App Bar [AND180]

Add a Toolbar to an Activity

- Add a Toolbar to your UI
- Set text and logo

Populate Toolbar Actions from an XML menu file

- Define an XML menu file
- Inflate the XML menu file into a Toolbar's actions menu
- Create hierarchical menus
- Create checkable menu items

Use a Toolbar as an Activity's app bar

- Set a Toolbar as your Activity's app bar
- Inflate the Toolbar menu items
- Respond to item click

Set a navigation icon on an app bar

- Enable the app bar navigation button
- Set the navigation icon

- Respond when the navigation button is clicked

Android Navigation [AND205]

Use Fragment and FrameLayout to swap views

- Add a Fragment to a FrameLayout dynamically

Implement tab navigation with TabLayout

- Include a TabLayout in your UI
- Add tabs using code-behind
- Add tabs using XML
- Respond when a tab is tapped

Implement gestural navigation with ViewPager

- Add a ViewPager to your UI
- Code an adapter to supply the ViewPager with Fragments

Combine gestural and tab navigation

- Use a ViewPager to populate a TabLayout with tabs

Implement drawer navigation with DrawerLayout

- Use DrawerLayout to display a drawer menu
- Use an app bar navigation button to open the drawer
- Code an XML file for your menu
- Swap content when your menu is clicked

Navigation Patterns [IOS205]

Progress through pages of data with stack-based navigation

- Create a Navigation Controller programmatically
- Utilize the designer to create a Navigation Controller
- Customize the Navigation Controller

Show different views of related data with tab navigation

- Create a Tab Bar Controller
- Populate a Tab Bar Controller
- Customize the Tab Bar Controller

Display hierarchical relationships with master/detail navigation

- Create a Split View Controller
- Use a Split View Controller programmatically

- Use the iOS Designer to define a Split View Controller

Mobile Application Architecture [XAM301]

Apply Model-View-Controller to a Xamarin.iOS app

- Architect an iOS app using MVC

Apply Model-View-Presenter to a Xamarin.Android app

- Architect an Android application using MVP

Apply Model-View-ViewModel to a Xamarin.Forms app

- Architect a Xamarin.Forms app using MVVM

Patterns for Cross Platform Mobile Development [XAM250]

Locate dependencies using the Factory Pattern

- Define a Factory
- Assign a dependency to a Factory
- Access the Factory from shared code

Use a Service Locator to register and retrieve dependencies

- Define a Service Locator
- Register dependencies with a Service Locator
- Resolve dependencies from a Service Locator

Use an IoC container to automatically inject dependencies

- Register dependencies with an IoC container
- Inject dependencies
- Automate dependence injection

Data Binding in Xamarin.Forms [XAM270]

Use data binding to decouple code-behind from UI

- Map data to visuals
- Create bindings in code
- Create bindings in XAML
- Work with binding context
- Change binding modes
- Implement property change notifications

Use value converters to bind incompatible types

- Perform textual conversions in XAML
- Create a value converter
- Apply a value converter in XAML

Using ListView in Xamarin.Forms [XAM280]

Display a collection with ListView

- Provide data to a ListView
- Manage selection in a ListView

Add and remove items dynamically

- Add, remove and update data in the ListView
- Make UI-safe collection changes
- Modify collections in the background

Customize ListView rows

- Alter the row visuals
- Use Data Templates
- Change the ListView separator
- Use built-in cell templates

Xamarin.Forms Effects [XAM330]

Customize control appearance

- Change element properties
- Use platform themes to update native control appearance

Apply effects to controls

- Resolve an effect
- Apply an effect programmatically
- Wrap an effect with RoutingEffect
- Apply an effect in XAML

Create an effect

- Create an effect
- Update platform specific UI
- Respond to UI changes and notifications
- Export an effect

Add configurable properties

- Add attached properties to an effect
- Use and update attached properties

Diagnosing Memory Management Issues [XAM370]

Identify and fix memory leaks in your code

- Find memory leaks in your code
- Fix common leaks in managed code

Recognize and fix Xamarin.iOS specific memory problems

- Identify strong reference cycles
- Dispose native resources
- Manage event handler lifecycles

Recognize and fix Xamarin.Android specific memory problems

- Improve garbage collector performance
- Free native resources

© 2014 - 2018 Xamarin Inc., Microsoft. All rights reserved.