Advanced UI Architecture using MVVM and XAML

Course 1392 – 40 Hours

Overview

This is an advanced programing and architecture course that deals with the question: "How do I construct a full working XAML application from scratch". In this course we will go over the MVVM separation pattern as the base for a XAML app architecture. We will learn the concepts of MVVM and how to apply them in real life. We will go over specific patterns, each deals with different aspect of the application, such as navigation, presenting collections, presenting multi state data, editing data, working with async services and validation.

On Completion, Delegates will be able to

Design an architecture for a complex XAML application. You will be familiar with the different parts of the application structure and will know how to compose them in a way that will allow you to maintain your code, use design tools effectively, and test your code. We will complete the course with a full example from scratch.

Who Should Attend

- This course is intended for experienced UI Developers and Architects, wishing to learn the best practices of UI development in MVVM and XAML.

Prerequisites

- At least 3 years' experience developing in C#
- At least 1-year experience developing XAML based applications

Course Contents

Module 1: Introduction
- About MVVM
- Separation of Concerns
- Evolution of MVVM
- MVVM Application in today's platforms

Module 2: MVVM Responsibilities
- The Role of the Model
- The Role of client services
- The Role of the Views
- The Role of the View Models

- Data Model Wrappers

Module 3: First Taste of MVVM

- Sample application – No MVVM
- Sample application – Basic Binding and MVVM
- Summary and Open Questions

Module 4: Wiring Up Views and View Models

- View First – In XAML
- View First – Code Behind
- View First – View Model Locator
- Demo: Writing a View Model Locator
- View Model First – Explicit
- View Model First – With Data Templates
- Demo: Data Templates Implicit and Explicit Selection
- Demo: Resolving Views using Data Template Selector
- No One First – Using Bootstrapper

Module 5: Data Binding in MVVM

- Overview
- Sources and Targets
- Data Context and Scopes
- Demo: Binding to Data Context
- Data Sources – Binding to Entities and Collections
- Demo: Implementing INotifyPropertyChanged
- Observable Collections
- Demo: Observable Collections
- Property Paths
- Demo: Property Paths
- Binding Mode
- Demo: Binding Mode

Module 6: Advanced Binding

- Relative Sources
- Demo: Relative Sources
- Update Source Triggers
- Demo: Update Source Trigger
- Converters
- Demo: Converters and String Formats
- Async Binding
- Fallback Values
- Demo: Fallback and Target Null
- Multi Binding
- Demo: Multi Binding

Module 7: Communication in MVVM

- Commands
- Demo: Relay Commands
- Attached Properties
- Behaviors
- Demo: Calling method on load using Attached Properties and Using Behaviors
- Custom Behaviors
- Demo: Content Control Notifications using Behaviors

Module 8: Navigation and Hierarchy

- Naming Guidance
- Hierarchy and Navigation overview
- Navigation using Content Control and Data Templates
- Demo: 3 Page application
- Navigation with Parameters
- Demo: Details Page
- Messaging between view models
- Demo: Event Aggregator navigation

Module 9: Multi State Views

- Supporting Multi – State in View Model
- Visual State Manager in the view
- Switching States using Data Triggers
- Switching States using Behaviors
- Demo: Multi State Data Template

Module 10: Editable Views

- Wrapper View Model Pattern
- Read Model / Context method
- Write / Wrap Model method
- Demo: Action Item editor
- Validation using INotifyDataErrorInfo
- Demo: Creating ValidatableObservableObject
- Templating Validation

Module 11: Dialogs and View Services

- Overview
- Creating the View and View Model
- Wrapping method
- Async wrap method
- Demo: Confirmation dialog box

Module 12: Design Time Data and Blendability

- Overview – Approaches to Design Time Data
- Design Time data as Resource
- Demo: Main view model
- Demo: Data model wrapper
- Using in Blend
- Demo: Debugging Design Time data

Module 13: Dependency Injection

- Overview and Motivation
- IoC Containers
- Injection Methods
- Demo: Unity
- Demo: Injection Methods
- Demo: IoC Helper

Module 14: Working with Async Services

- Overview of Services
- Async – Await explained
- Demo: Building data service

Module 15: Full Demo

- Introduction: The Todo List app
- Building the Data Model
- Building the Data Service
- Building the Bootstrapper and containers
- The Main View models
- The Lists page
- The Item Editor page