

Updated 06/28/2024

Sign up

# Kotlin Multiplatform mobile (KMM) training

3 days (21 hours)

## Presentation

Our Kotlin Multiplatform Mobile training course will teach you how to create robust, modern mobile applications on multiple platforms using [code reuse](#). Whether on Android or iOS, you'll be able to reduce the amount of code you need to duplicate manually, and drastically cut your development time and costs.

Our program will teach you how to create a cross-platform application using KMM, set up your development environment, create your first KMM project and share code seamlessly between Android and iOS.

Our training course will also teach you how to use the essential tools and plugins that make KMM a powerful tool for developers. By mastering it, you'll be able to create modern, high-performance user interfaces.

Like all our training courses, it will be run on the latest version of the tool: [KMM 0.8](#).

## Objectives

- Structuring shared code using MVVM and MVP architectures
- Creating a KMM project with Compose Multiplatform
- Master the IntelliJ IDEA IDE for KMM development
- Carry out exhaustive tests to ensure the quality and reliability of KMM applications
- Managing data persistence with SQL Delight

## Target audience

- **Mobile developers**
- Software developers

## Prerequisites

- Basic knowledge of Kotlin
- Experience in mobile/software development
- [Test My Knowledge](#)

## KMM training program

### INTRODUCTION TO KOTLIN MULTIPLATFORM MOBILE (KMM)

- What is KMM?
- Environment configuration
- Creating a KMM project
- Understanding code sharing between Android and iOS
- Essential KMM tools and plugins
- Specific code with expect/actual

### USING THE INTELLIJ IDE

- IntelliJ IDEA interface
- Project navigation
- IDE configuration
- Gradle dependency management in IntelliJ
- Debugging KMM applications with IntelliJ

### DATA MODELING AND PERSISTENCE

- Shared data models
- Introduction to SQLDelight
- Defining database schemas
- Writing SQL queries
- Data models and user interfaces
- Best practices for data integrity
- Database schema migrations

### ARCHITECTURE AND DESIGN PATTERNS

- Software architectures in KMM
- MVVM and MVP for structuring shared code
- Dependency injection with Koin
- Setting up an architecture

### DEVELOPMENT OF NATIVE USER INTERFACES

- Introduction to Compose Multiplatform
  - Desktop support
- Navigation and application status
- Customizing UI components
- Best practices for a consistent, high-performance user experience

## COMPETITION AND ASYNCHRONOUS PROGRAMMING

- Kotlin coroutines and asynchronous operations
- Competition management
- State sharing between threads
- Scope Functions
- Coroutine integration
- Avoid competition and blocking problems

## NETWORK AND API INTEGRATION

- Ktor for API calls
- Responses and network errors
- Kotlinx.serialization
- Secure network communications and authentication management
- Building a network data flow

## TESTING AND QUALITY ASSURANCE

- Writing unit tests
- Instrumented tests for Android and iOS
- Use of frameworks (MockK and Kotest)
- Tests for continuous development

## APPLICATION DEPLOYMENT AND MONITORING

- Deployment for KMM applications
- Version management
- Settings for App Stores
- Performance monitoring
- Monitoring and crash reporting tools

## Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

## Positioning on entry to training

Positioning on entry to training complies with Qualiopi quality criteria. As soon as

On final registration, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, and his or her expectations and personal objectives for the forthcoming training course, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

## Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

## Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

## Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

## Sanction

A certificate will be issued to each trainee who completes the course.