Faysal Aziz

# Kotlin in the New Era

## From Mobile Code to Distributed Systems, Building Scalable, Intelligent, and Secure Kotlin Applications - 3 days

FA-0278 · 3 days

Software Development · DevOps, Cloud & Infrastructure

## Introduction

Modern Kotlin development no longer lives solely inside the boundaries of a mobile device. Applications today are fragments of larger ecosystems, distributed across cloud services, powered by streaming data, and increasingly infused with AI. The shift from monolithic, device-centric design to microservices and event-driven architectures has fundamentally redefined what it means to be an Android or Kotlin developer. This course embraces that shift head-on, equipping learners with the architectural thinking, performance discipline, and system-level awareness demanded by today's industry.

Led by an instructor with over 30 years of industry experience, this course focuses on real-world, industry-demanded practices rather than academic abstractions. Every topic reflects how modern production systems are actually built, scaled, and maintained.

## Learning outcomes

By the end of this course, participants will be able to:

- Design scalable Kotlin applications using modern architectural patterns
- Implement and extend MVVM with production-grade practices
- Apply ViewModel strategies that support lifecycle resilience and testability
- Secure applications through authentication, authorization, and code protection techniques
- Optimize performance across memory, threading, and app size dimensions
- Navigate platform changes and migrate Android apps from version 12 to 16
- Integrate AI capabilities both on-device and via external APIs

- Build and consume backend microservices using Kotlin and Java
- Understand and apply event-driven architecture using streaming platforms such as Kafka
- Transition from monolithic thinking to cloud-native, distributed system design

# Prerequisites

- Basic proficiency in Kotlin programming
- Familiarity with Android development fundamentals
- Understanding of object-oriented programming concepts
- Basic knowledge of REST APIs and networking
- Exposure to Git and version control workflows
- Optional but helpful: prior experience with Java or backend systems

---

# Training outline

## 1. Evolution of Modern Development Paradigms

1. Transition from monolithic to microservices architecture
2. Limitations of on-device centric application design
3. Rise of cloud-native applications and distributed systems

   Role of event-driven systems and streaming platforms

   1. Fundamentals of event streaming
   2. Introduction to Apache Kafka concepts
   3. Producer-consumer patterns and real-time data pipelines
5. Impact of AI and intelligent systems on application design

## 2. Kotlin Project Structure & Code Architecture*

   Modularization strategies for large-scale applications

   1. Feature-based modules
   2. Layered architecture separation

   Clean architecture principles in Kotlin

   1. Domain, data, and presentation layers
   2. Dependency inversion and abstraction boundaries
3. Code organization for scalability and maintainability
4. Dependency injection patterns and frameworks

5. Multi-module Gradle configurations

## 3. MVVM Pattern Deep Dive

1. Core principles of MVVM in modern Android
2. Separation of concerns and data flow management

   State management strategies

      1. Immutable state handling
      2. UI state vs business state

   Reactive programming integration

      1. LiveData vs Flow vs StateFlow

5. Handling complex UI interactions

   Testing MVVM architecture

      1. Unit testing ViewModels
      2. Mocking repositories and data sources

## 4. ViewModel Best Practices

1. Lifecycle awareness and configuration changes
2. Avoiding memory leaks and improper references

   State persistence strategies

      1. SavedStateHandle usage

4. Coroutine scope management within ViewModels
5. Error handling and retry mechanisms
6. Structuring ViewModels for scalability
7. Sharing data between ViewModels

## 5. Security Fundamentals*

   Core principles of application security

      1. Confidentiality, integrity, availability

2. Secure coding practices in Kotlin

   Code protection techniques

      1. Obfuscation and shrinking
      2. Reverse engineering prevention

   Authentication mechanisms

      1. Token-based authentication

2. OAuth flows

Authorization strategies

1. Role-based access control
2. Claims and scopes

Secure data storage

1. Encryption techniques
2. Key management systems

Network security

1. HTTPS enforcement
2. Certificate pinning

## 6. Performance Tuning and Optimization

Memory management in Android

1. Heap usage and garbage collection
2. Detecting and fixing memory leaks

Profiling tools and techniques

1. CPU profiler
2. Memory profiler
3. Network profiler

Application size reduction strategies

1. Resource shrinking
2. Code minification

Threading and concurrency

1. Kotlin coroutines deep dive
2. Structured concurrency
3. Dispatchers and thread pools

5. Avoiding UI thread blocking
6. Performance monitoring in production

## 7. Migrating from Android 12 to Android 16

1. Overview of platform changes across versions
2. Behavioral changes impacting existing apps
3. Permissions and privacy updates
4. Background execution limitations

Compatibility strategies

1. Feature flags
2. Backward compatibility handling

6. Refactoring legacy code for modern APIs
7. Testing across multiple Android versions

## 8. Integrating AI into Kotlin Applications*

1. Overview of AI in modern applications

On-device AI integration

1. ML model deployment
2. Performance considerations

API-based AI integration

1. Consuming AI services via REST
2. Handling latency and reliability

Use cases for AI in mobile apps

1. Recommendations
2. Natural language processing

5. Data handling and preprocessing
6. Ethical considerations and data privacy

## 9. Backend Services and Microservices Architecture*

Fundamentals of microservices design

1. Service decomposition
2. API gateway patterns

Communication between services

1. REST vs gRPC
2. Event-driven communication

3. Service discovery and orchestration

Fault tolerance and resilience

1. Circuit breakers
2. Retry policies

Cloud deployment models

1. Containerization concepts
2. Scaling strategies

## 10. Backend Development with Kotlin and Java*

Kotlin for backend development

1. Frameworks and ecosystem overview

2. Interoperability with Java

Building RESTful APIs

1. Routing and controllers
2. Request/response handling

Database integration

1. ORM and query handling

Asynchronous processing

1. Coroutines in backend systems

6. Logging and monitoring
7. Testing backend services

## 11. Event Streaming and Real-Time Systems*

1. Introduction to streaming architectures

Kafka ecosystem fundamentals

1. Topics and partitions
2. Producers and consumers

3. Designing event-driven systems
4. Handling data consistency and ordering
5. Integrating mobile apps with streaming backends
6. Real-time analytics and processing pipelines

## 12. Bridging Mobile and Cloud Systems*

1. Designing mobile apps as clients of distributed systems
2. API consumption patterns
3. Handling offline-first scenarios
4. Synchronization strategies

Observability across systems

1. Logging
2. Metrics
3. Tracing

---

* This course positions Kotlin developers to operate beyond the confines of mobile development and into the broader world of distributed, intelligent, and scalable systems, where modern software engineering is rapidly heading. These items require bypassing the firewall during training.

---

© 2026 Faysal Aziz · Kuala Lumpur