

# Formation Kotlin Multiplatform, Développer des applications Android

## Présentation

Cette formation de trois jours (21 h) initie les développeurs aux fondamentaux de Kotlin Multiplatform (KMP) : création d'un projet commun Android-iOS-Web, organisation d'une architecture partagée et implémentation d'interfaces Jetpack Compose Multiplatform / SwiftUI. Renforcez la qualité avec tests multiplateformes, analyse statique, CI/CD GitHub Actions et bonnes pratiques de sécurité (chiffrement local, TLS, scan dépendances).

Clôturez par l'optimisation : profilage, cache réseau afin de livrer un socle industrialisable et pérenne.

Durée : 21,00 heures (3 jours)

Tarif INTRA : Nous consulter

## Objectifs de la formation

A l'issue de la formation, le stagiaire sera capable de créer des applications multiplateformes (Web, iOS, Android) avec Kotlin Multiplatform (KMP).

- Connaître les concepts de Kotlin Multiplatform
- Mettre en place un environnement de développement
- Concevoir l'architecture d'une application mobile multiplateforme
- Construire une interface utilisateur fluide et performante
- Réaliser des tests pour assurer la qualité et la fiabilité des applications

## Prérequis

Avoir des connaissances en langage de programmation (Java, C#, C++, JavaScript, Python, etc.)

## Public

Développeurs, architectes

## Programme de la formation



## **Jour 1 – Concepts, environnement et sécurité KMP**

- Comprendre les bases de Kotlin Multiplatform (KMP)
- Mettre en place un projet et configurer l'environnement
- Intégrer des bonnes pratiques de sécurité multiplateforme

### **Introduction à KMP**

1. Historique et avantages du multiplateforme vs Flutter, React Native
2. Architecture KMP : modules common, Android, iOS et autres targets
3. Support IDE
4. Cas d'usage et réutilisation de code

### **Kotlin pour le multiplateforme**

1. Fonctions, bases, objets, data
2. Coroutines, extensions

### **Configuration de l'environnement**

1. Création d'un projet KMP avec Gradle
2. Configuration des targets : JVM, Android, iOS (Xcode interop)
3. Plugins et dépendances courants (Kotlinx Serialization, Ktor)

### **Sécurité multiplateforme**

1. Gestion des secrets et des clés API dans common et plateformes
2. Chiffrement des données locales (SQLDelight encrypted DB)
3. Validation des entrées et protection contre l'injection de code

### **Atelier fil rouge - mise en place**

1. Initialiser un squelette de projet KMP avec un module common
2. Configurer les sources communes et spécifiques
3. Compiler et exécuter des tests unitaires common

## **Jour 2 – Architecture, UI, accès aux données et sécurité**

- Définir une architecture sécurisée pour KMP
- Construire une interface partagée et gérer la persistance
- Assurer la sécurité des communications et des données

### **Architecture d'application KMP**

1. Patterns de clean architecture adaptés (Domain, Data, UI)
2. Sécurisation des modules common et spécifiques
3. Communication sécurisée entre modules (TLS, mutual TLS)

### **UI partagée et plateformes natives**

1. Approches : Jetpack Compose Multiplatform, SwiftUI interop
2. Gestion sécurisée des ressources (cryptage, permissions)

### **Accès aux données**

1. Utiliser Ktor pour consommer un web service commun via HTTPS et authentification OAuth2
2. Interception des erreurs et affichage UI adapté
3. Persistance sécurisée avec SQLDelight (chiffrement, migrations)

### **Atelier fil rouge codé**

1. Implémenter un module réseau commun sécurisé pour récupérer des données
2. Afficher une liste simple dans l'UI Android et iOS
3. Ajouter un cache local chiffré

## **Jour 3 – Qualité, CI/CD et optimisation multiplateforme**

- Automatiser l'analyse de qualité, les tests et le déploiement d'un projet KMP
- Optimiser performances, sécurité et maintenance du code partagé

### **Analyse statique & couverture**

- Configuration Detekt, Ktlint et Kover dans le module common et sur chaque target pour détecter dettes et vulnérabilités
- Scan dépendance (OWASP Dependency check)

**Tests automatisés multiplateformes**

- Mise en place d'un jeu de tests commun (JUnit/JVM)
- Tests instrumentés Android & iOS

**Pipeline CI/CD**

- Workflow type
- Publication d'une lib partagée
- Distribution iOS et Android
- Notions de déploiement interne

**Optimisation performance**

- Profilage mémoire/CPU avec Android Studio & Xcode
- Stratégie de caching réseau et persistance chiffrée

**Atelier fil rouge pratique**

- Finaliser l'app
- Exécuter les lint/tests
- Mesurer la couverture avant de déclencher un build signé

## Organisation

**Formateur**

Les formateurs de Docaposte Institute sont des experts de leur domaine, disposant d'une expérience terrain qu'ils enrichissent continuellement. Leurs connaissances techniques et pédagogiques sont rigoureusement validées en amont par nos référents internes. Riches de leur expérience sur le sujet, ils sauront accompagner vos collaborateurs dans leur montée en compétences.

**Moyens pédagogiques et techniques**

- Apports des connaissances communes.
- Mises en situation sur le thème de la formation et des cas concrets.
- Méthodologie d'apprentissage attractive, interactive et participative.

- Équilibre théorie / pratique : 60 % / 40 %.
- Supports de cours fournis au format papier et/ou numérique.
- Ressources documentaires en ligne et références mises à disposition par le formateur.
- Pour les formations en présentiel dans les locaux mis à disposition, les apprenants sont accueillis dans une salle de cours équipée d'un réseau Wi-Fi, d'un tableau blanc ou paperboard. Un ordinateur avec les logiciels appropriés est mis à disposition (le cas échéant).

### **Dispositif de suivi de l'exécution et de l'évaluation des résultats de la formation**

En amont de la formation :

- Recueil des besoins des apprenants afin de disposer des informations essentielles au bon déroulé de la formation (profil, niveau, attentes particulières...).
- Auto-positionnement des apprenants afin de mesurer le niveau de départ.

Tout au long de la formation :

- Évaluation continue des acquis avec des questions orales, des exercices, des QCM, des cas pratiques ou mises en situation...

A la fin de la formation :

- Auto-positionnement des apprenants afin de mesurer l'acquisition des compétences.
- Evaluation par le formateur des compétences acquises par les apprenants.
- Questionnaire de satisfaction à chaud afin de recueillir la satisfaction des apprenants à l'issue de la formation.
- Questionnaire de satisfaction à froid afin d'évaluer les apports ancrés de la formation et leurs mises en application au quotidien.

### **Accessibilité**

Nos formations peuvent être adaptées à certaines conditions de handicap. Nous contacter pour toute information et demande spécifique.