

Formation Développeur Android

Cette formation Développeur Android s'adresse aux développeurs qui souhaitent se spécialiser en mobilité, ainsi qu'aux chefs de projet ou architectes responsables d'applications mobiles. Au cours d'une formation jalonnée d'ateliers pratiques, vous apprendrez à configurer votre environnement de développement, concevoir des interfaces réactives en XML et Jetpack Compose, gérer la persistance des données et consommer des services REST/JSON avec Retrofit et les coroutines. Vous aborderez également les tests unitaires et UI (JUnit/Espresso) et le déploiement d'applications signées sur terminaux et le Play Store. À l'issue de ce parcours, vous disposerez de toutes les compétences nécessaires pour concevoir, développer et publier des applications Android performantes et professionnelles.

Durée

5 jours

Objectifs pédagogiques

- ◆ Configurer l'environnement Android Studio et créer une application de base
- ◆ Concevoir et implémenter des interfaces utilisateur réactives en XML ou Jetpack Compose en respectant les principes du Material Design
- ◆ Implémenter la persistance locale des données
- ◆ Intégrer et consommer des services REST/JSON et gérer l'asynchronisme via les coroutines
- ◆ Tester l'application avec des tests unitaires et UI (JUnit/Espresso) et déployer une version signée sur un terminal ou le Play Store

Public

Développeurs, chefs de projet, architectes

Prérequis

Bonnes connaissances d'un langage orienté objet (Java ou Kotlin), bases en programmation XML et notions de design d'interface, compréhension des concepts de programmation asynchrone et des patterns MVVM.

Programme de formation

Phase d'inclusion

Module 1 : Installation et prise en main de l'environnement

Installation d'Android Studio, configuration du SDK et des AVD
Présentation de la structure d'un projet (manifest, Gradle, ressources)
Introduction aux bases du langage Kotlin et rappels Java
Cycle de vie des activités et tests sur émulateur et appareil
Bonnes pratiques de configuration, versionnement et gestion de dépendances

Exemples d'activités pratiques :

Installer l'environnement et créer un projet Hello World
Configurer un émulateur et déployer l'application sur un smartphone
Explorer et modifier le manifest et les ressources de base

Module 2 : Conception d'interfaces utilisateur

Layouts classiques (ConstraintLayout, RecyclerView) et création de vues
Principes et syntaxe de Jetpack Compose et comparaison avec XML
Composants Material Design et personnalisation de thèmes
Navigation component, gestion de la pile d'activités et fragments
Accessibilité et internationalisation des écrans

Exemples d'activités pratiques :

Créer un écran d'accueil réactif en Compose et XML
Développer une liste de données avec RecyclerView ou LazyColumn
Mettre en place une navigation entre écrans avec données passées en arguments

Module 3 : Architecture et persistance des données

Différences entre SharedPreferences, fichiers et bases SQLite/Room
Création et utilisation d'une base de données Room avec DAO et entités
Patterns MVVM : ViewModel, LiveData et Repository
DataBinding et gestion des états dans Compose
Aspects sécurité : protection des données locales et chiffrement

Exemples d'activités pratiques :

Développer une application de prise de notes utilisant Room et ViewModel
Sauvegarder des préférences utilisateur et les afficher dans l'UI
Synchroniser les données entre activité et fragment via LiveData

Module 4 : Réseaux, services et asynchronisme

Principes HTTP/REST, JSON et sécurité des échanges réseau
Utilisation de Retrofit et OkHttp pour consommer des API
Coroutines et Flow pour l'asynchronisme et la gestion des threads
Gestion des erreurs réseau, du cache et de la connectivité avec WorkManager
Introduction à la dépendance Hilt/Dagger pour l'injection et les services

Exemples d'activités pratiques :

Appeler une API publique et afficher les données dans une liste
Implémenter des appels en coroutines avec indicateurs de chargement et d'erreur
Ajouter un cache local et gérer l'actualisation hors ligne

Module 5 : Tests, optimisation et déploiement

Tests unitaires avec JUnit et tests d'interface avec Espresso et Compose Test
Configuration de Gradle pour les différents variants et signatures
Profilage de l'application : mémoire, CPU, réseau et énergie
Création d'un bundle ou APK signé et bonnes pratiques de versioning
Présentation du Play Console et étapes de mise en production

Exemples d'activités pratiques :

Écrire des tests unitaires pour les ViewModel et Repository
Automatiser un test UI de formulaire de connexion avec Espresso
Générer un fichier AAB, le signer et l'installer sur un terminal de test

Moyens et méthodes pédagogiques

- ♦ La formation alterne entre présentations des concepts théoriques et mises en application à travers d'ateliers et exercices pratiques (hors formation de type séminaire).
- ♦ Les participants bénéficient des retours d'expérience terrains du formateur ou de la formatrice
- ♦ Un support de cours numérique est fourni aux stagiaires

Modalités d'évaluation

- ♦ **En amont de la session de formation**, un questionnaire d'auto-positionnement est remis aux participants, afin qu'ils situent leurs connaissances et compétences déjà acquises par rapport au thème de la formation.
- ♦ **En cours de formation**, l'évaluation se fait sous forme d'ateliers, exercices et travaux pratiques de validation, de retour d'observation et/ou de partage d'expérience, en cohérence avec les objectifs pédagogiques visés.
- ♦ **En fin de session**, le formateur évalue les compétences et connaissances acquises par les apprenants grâce à un questionnaire reprenant les mêmes éléments que l'auto-positionnement, permettant ainsi une analyse détaillée de leur progression.