Android avancé : architecture et testabilité

Appliquer et intégrer l'approche TDD dans la conception et le développement de ses applications Android

DESCRIPTION

Depuis son lancement en 2008, le système d'exploitation de Google a connu de nombreuses mutations et se positionne comme l'OS mobile le plus répandu au monde. Parallèlement, la plateforme Android est devenue un système techniquement mature et sur lequel s'appuient de très larges projets. Développer aujourd'hui pour Android requiert un savoir-faire et une méthodologie particulière.

Cette session prépare au développement d'applications Android en milieu professionnel. Une très grande attention sera portée sur la mise en place de tests unitaires sur les différentes couches de l'application. La présentation et la mise en pratique de concepts d'architecture fondamentaux permettra de voir comment une application Android peut être entièrement conçue et développée en Test Driven Development.

Il sera également question de méthodologie, tant dans l'organisation des équipes, que dans les stratégies d'intégration continue et de déploiement.

OBJECTIFS PEDAGOGIQUES

- Développer une application de manière professionnelle et testée
- Industrialiser la couche réseau et la persistance
- Concevoir les différentes couches en Test Driven Development
- Appliquer des concepts architecturaux avancés
- Maîtriser son environnement de développement
- Monter une usine d'intégration continue et piloter les indicateurs de qualité
- Partager les bonnes pratiques de travail en équipe et de déploiement

PUBLIC CIBLE

- Développeur Android confirmé
- Architecte
- Chef de projet technique

Stage pratique
Mobile

Code :
WMAND

Durée :
3 jour(s) (21,00 heures)

Exposés : **30.00 %**
Cas pratiques : **70.00 %**
Echanges d'expérience :
Variable non renseignée %

Inter-entreprises :
Prochaines sessions disponibles sur notre site web.
Tarif : 2 050,00 € HT / participant

Intra-entreprise :
Tarifs et dates sur demande.

PRE-REQUIS
Cette formation avancée ne rappellera pas les concepts de bases d'Android. Les participants doivent avoir une bonne maîtrise de la plateforme Android et des langages Kotlin ou Java.

METHODE PEDAGOGIQUE
Durant les 3 jours de la formation, le développement complet d'une application sera l'occasion d'alterner des apports théoriques et des travaux pratiques. Ceux-ci se dérouleront en binôme ou en coding dojo (tous les développeurs passent tour à tour au clavier pour résoudre un problème ensemble).

PROFIL DES INTERVENANTS
Toutes nos formations sont animées par des consultants-formateurs expérimentés et reconnus par leurs pairs.

MODALITÉS D'ÉVALUATION ET FORMALISATION À L'ISSUE DE LA FORMATION
L'évaluation des acquis se fait tout au long de la session au travers des ateliers et des mises en pratique. Une évaluation à chaud sur la satisfaction des stagiaires est réalisée systématiquement en fin de session et une attestation de formation est délivrée aux participants mentionnant les objectifs de la formation, la nature, le programme et la durée de l'action de formation ainsi que la formalisation des acquis.

PROGRAMME PEDAGOGIQUE DETAILLE

Jour 1

INTRODUCTION

COUCHE RÉSEAU

- État de l'art des principaux frameworks réseau
- Bonnes pratiques sur la gestion du cache
- Mise en pratique : "Faire des appels réseau avec Retrofit"

PERSISTANCE DES DONNÉES

- Architecture ORM
- Nouvelles bases : Room

- Mise en pratique : "Réaliser un CRUD avec Room"

TESTER UNE APPLICATION

- Tests manuels
- Tests unitaires
- Tests d'interface
- Développement d'une fonctionnalité à partir des tests
- Mise en pratique : “Calcul d'un score d'une partie de bowling en TDD”

Jour 2

ARCHITECTURE ANDROID MODERNE

- MVP, MVVM, Redux
- Principe des différentes couches
- Présentation du modèle de Clean Architecture

MISE EN PRATIQUE : DÉVELOPPEMENT DES RÈGLES FONCTIONNELLES SELON LE MODÈLE DE CLEAN ARCHITECTURE

- Création d'un module sans aucune dépendance au framework Android pour la gestion de ces règles Définir les exigences de notre application via des tests unitaires
- Présentation du concept des mocks pour définir ces exigences
- Implémenter le code de production nécessaire pour répondre à ces exigences
- Gestion des erreurs fonctionnelles et techniques par des exceptions
- Utiliser les fonctionnalités et les raccourcis d'Android Studio pour refactorer le code

MISE EN PRATIQUE : DÉVELOPPEMENT DE LA COUCHE PRÉSENTATION

- Compréhension de la frontière entre la View et le Presenter
- Transformation des données en données présentables (ViewModel)
- Présentation de Robolectric pour tester unitairement des composants Android

Jour 3

MISE EN PRATIQUE : INTÉGRATION DES DIFFÉRENTES COUCHES DÉVELOPPÉES DANS L'APPLICATION

- Utilisation du pattern Controller
- Implémentation d'un fournisseur de données bouchonné
- Injection des dépendances dans les différentes couches créées
- Intégration et branchement au design définitif de l'application

GESTION DE L'ASYNCHRONISME

- Présentation des composants JetPack & Kotlin : ViewModel, LiveData et Coroutine
- Mise en pratique : “Exécuter les méthodes du Controller en asynchrone avec des Coroutines et celles de la View sur le main thread via des LiveData”

EQUIPES MOBILES

- Retours d'expérience d'organisations multi-équipes
- Bonnes pratiques méthodologiques : feature flips, A/B testing, release train

AUTOMATISER LE BUILD

- Choisir les outils pour l'intégration continue
- Différents jobs : continu, nightly, release
- Mesurer la qualité
- Mise en pratique : "Automatiser un build dans le Cloud"

DISTRIBUER SUR LE PLAY STORE

- Préparer et signer un package
- Maîtriser le Play Store : builds alpha, déploiements progressifs

SYNTHÈSE ET RAPPEL DES POINTS CLÉS DE LA FORMATION

Accessibilité

L'inclusion est sujet important pour OCTO Academy.
Nos référent·es sont à votre disposition pour faciliter l'adaptation de votre formation à vos besoins spécifiques.
Pour les contacter : academy.accessibilite@octo.com