

# Mise en place de l'environnement de développement

Emmanuel BRUNO  
Université de Toulon, LIS UMR CNRS 7020  
emmanuel.bruno@univ-tln.fr

2026-10-14

## Table des matières

|                                                  |   |
|--------------------------------------------------|---|
| 1 L'écosystème Android .....                     | 1 |
| 2 Composants clés .....                          | 1 |
| 2.1 Android SDK (Software Development Kit) ..... | 1 |
| 2.2 Android Studio (IDE officiel) .....          | 1 |
| 2.3 Gradle (système de build) .....              | 1 |
| 2.4 AVD (Android Virtual Device) .....           | 1 |
| 2.5 ADB (Android Debug Bridge) .....             | 1 |
| 3 Installation des outils .....                  | 2 |
| 3.1 Configuration appareil physique ....         | 2 |
| 3.2 Configuration émulateur .....                | 2 |
| 4 Création d'un projet .....                     | 2 |
| 4.1 Structure type .....                         | 2 |
| 4.2 Déployer l'application .....                 | 2 |
| 4.3 Modifier le code .....                       | 2 |
| 5 Interaction avec un appareil .....             | 3 |
| 6 Ressources et liens utiles .....               | 3 |
| 7 Conclusion .....                               | 3 |

## 1 L'écosystème Android

- Système d'exploitation mobile open-source
- Basé sur le noyau Linux
- Développé par Google
- Large adoption mondiale (>70% part de marché)

## 2 Composants clés

- Android SDK (Software Development Kit)
- Android Studio (IDE officiel)
- Gradle (système de build)

- AVD (Android Virtual Device)
- ADB (Android Debug Bridge)

### 2.1 Android SDK (Software Development Kit)

- Collection d'outils de développement
- API Android et bibliothèques
- Outils de débogage et de test
- Versions SDK par niveau d'API

### 2.2 Android Studio (IDE officiel)

- Basé sur IntelliJ IDEA
- Intégration complète Kotlin
- Éditeur de layout visuel
- Outils de profiling et débogage

### 2.3 Gradle (système de build)

- Build system flexible et performant
- Support du Kotlin DSL
- Gestion des dépendances

### 2.4 AVD (Android Virtual Device)

- Émulateurs de périphériques Android
- Multiple versions d'Android
- Différentes configurations matérielles

### 2.5 ADB (Android Debug Bridge)

- Interface en ligne de commande
- Communication avec émulateur/appareil
- Installation et débogage d'apps
- Capture de logs
- Accès à la console système
- USB et Wi-Fi

### 3 Installation des outils

- Android Studio
- Configuration initiale
  - **SDK Manager** :
    - Android SDK Platform-Tools
    - Au moins une version d'Android SDK
    - Android Emulator
    - Android SDK Build-Tools

#### 3.1 Configuration appareil physique

1. **Activer mode développeur**
  - Paramètres > À propos du téléphone
  - Taper 7 fois sur « Numéro de build »
2. **Activer débogage USB**
  - Paramètres > Options développeur
  - Activer « Débogage USB »
  - Autoriser RSA fingerprint
3. **Possible de connecter via Wi-Fi**
  - Paramètres > Options développeur
  - Activer « Débogage USB via Wi-Fi »
4. **Vérifier connexion**

```
adb devices
# Devrait afficher votre appareil
```

#### 3.2 Configuration émulateur

1. **AVD Manager** (Tools > AVD Manager)
  - Create Virtual Device
  - Sélectionner un profil d'appareil
  - Choisir une image système
    - API Level recommandé : 24+
    - Google APIs recommandé
2. **Configuration matérielle**
  - RAM : 2GB minimum
  - Stockage interne : 2GB minimum

### 4 Création d'un projet

- Dans Android Studio :
  1. File → New → New Project
  2. Sélectionner « Empty Activity »
  3. Configurer le projet :

- Name : Nom de l'application
- Package name : com.example.myapplication
- Language : Kotlin
- Minimum SDK : API 24 (Android 7.0)

#### 4.1 Structure type

La structure d'un projet Android suit une organisation standardisée :

```
MonProjet/                                     #
Racine du projet
├─ app/                                         #
Module principal
│   └─ src/                                     #
Sources de l'application
│   │   └─ main/                               #
Code principal
│   │   │   └─ java/                           # Code
Kotlin ou Java
│   │   │   │   └─ com/example/app/
│   │   │   │   └─ res/                         #
Ressources (layouts, images...)
│   │   │   └─ AndroidManifest.xml
│   │   └─ test/                               #
Tests unitaires
│   │   └─ androidTest/                         # Tests
d'instrumentation
│   └─ build.gradle.kts                         #
Configuration du module
└─ settings.gradle.kts                         #
Configuration du projet
```

#### 4.2 Déployer l'application

- **Appareil physique** :
  - Connecter l'appareil via USB
  - Activer le mode débogage
  - Exécuter l'application
- **Émulateur** :
  - Lancer l'émulateur via AVD Manager
  - Exécuter l'application

#### 4.3 Modifier le code

- **MainActivity.kt** : Point d'entrée de l'application

- Modifier le texte de la fonction `Greetings()`
- Exécuter l'application

## 5 Interaction avec un appareil

```
# Commandes ADB essentielles
adb devices          # Liste
des appareils
adb install app.apk  #
Installation d'app
adb logcat           # Logs
en temps réel
adb shell            # Shell
sur l'appareil
```

## 6 Ressources et liens utiles

- Documentation Android
- Documentation Kotlin
- Android Developers Blog
- CodeLabs Android

## 7 Conclusion

À partir de maintenant, vous devriez être capable de :

- 1. Installer et configurer l'environnement**
  - Installer Android Studio
  - Configurer le SDK Android
- 2. Créer et structurer un projet**
  - Initier un nouveau projet Android
  - Comprendre l'arborescence des fichiers
  - Modifier le code source de base
- 3. Configurer les environnements de test**
  - Créer et paramétrer un émulateur
  - Gérer les images système Android
- 4. Préparer un appareil physique**
  - Activer les options développeur
  - Configurer le débogage USB/Wi-Fi
  - Établir la connexion ADB
- 5. Déployer et tester**
  - Compiler l'application
  - Déployer sur les différents supports