

Guide d'installation et de paramétrage d'Android Studio

1 Option: l'installer sur votre propre ordinateur

Si vous avez une machine sous Linux, Windows, MacOS ou ChromeOS, avec une bonne connexion Internet et, au moins 8Go de mémoire si vous utilisez un émulateur, tout est très simple: téléchargez et installez Android Studio pour votre OS sur <https://developer.android.com/studio/>. Ne le mettez pas à jour jusqu'à la fin de cet enseignement.

2 Utiliser Android Studio sur les machines dans les salles 2031 et 2032

2.1 Avertissements

Les instructions de cette section concernent uniquement l'installation spécifique d'Android Studio sur les serveur et les PC de l'UFR d'info. Ne les appliquez pas sur vos propres machines!

Ne pas utiliser les machines de nom "she-said-evil" en 2031 et "mekanik" et "pissing" en 2032 (première rangée). Il est très recommandé d'utiliser toujours le même PC, sinon le paramétrage de la section 2.3 devra impérativement être refait.

2.2 Nettoyage

Dans un terminal lancez `ls -l .*` pour voir vos fichiers cachés. Si vous avez les répertoires `.android`, `.AndroidStudioXX`, `.gradle` (les redoublants les ont certainement), effacez-les tous. Une façon rapide pour effacer un répertoire est `rm -rf nom_repertoire`. Les versions récentes mettent la configuration dans le répertoire `.config/Google/AndroidStudioXX`, à supprimer aussi.

Si vous avez un répertoire `AndroidStudioProjects` avec vos projets de l'an dernier gardez-le.

2.3 Deux répertoires locaux et liens symboliques — à refaire si vous changez de poste

Notre installation d'Android Studio, ainsi que vos fichiers, se trouvent sur un disque réseau (NFS), assez lent. Pour accélérer le travail, on va créer et utiliser deux répertoires locaux sur le disque dur du PC (pour le moteur de production `gradle` et pour l'émulateur de téléphone, qui sollicitent beaucoup le disque). Un étudiant de login `dupont` le fera comme ceci.

```
mkdir -p /local2/tmp/dupont-gradle
ln -s /local2/tmp/dupont-gradle ~/.gradle

mkdir -p ~/.android
rm -rf ~/.android/avd

mkdir -p /local2/tmp/dupont-android
ln -s /local2/tmp/dupont-android ~/.android/avd
```

2.4 Premier lancement et paramétrage d'Android Studio

Définissez la variable d'environnement (dans un fichier de configuration `.bashrc` ou autre, ou directement en ligne de commande) `export ANDROID_SDK_ROOT=/opt/android`

Depuis la ligne de commande, lancez la commande `android-studio`. Suivez à la lettre les indications ci-dessous:

- Au premier écran, choisissez “Do not import settings”.
- On tombe ensuite sur un message “unable to access Android SDK add-on list” cliquez sur “Setup Proxy”, puis “Manual proxy configuration”, mettez “cache” comme hostname et 3128 comme port, puis OK.
- Vous arrivez alors à un écran de “Setup Wizard”. **Important: *annuler* ce wizard en cliquant sur Cancel**, sinon il télécharge et met dans le compte utilisateur toute une installation d'Android :-(. Confirmez cette annulation dans l'écran suivant en choisissant **"Do not re-run the setup wizard"** puis “OK”.
- L'écran suivant a pour titre “Welcome to Android Studio”.
- Vous pouvez maintenant créer un nouveau projet comme décrit dans la section suivante (s'il y a tout de suite une erreur SDK appelez le secours). **Important:** Vous aurez durant la création un écran sur le réglage du proxy, mettez de nouveau “cache” et “3128”, et choisissez aussi “Enable HTTPS Proxy” (et “Do not show this dialog in the future”). Puis “Ok”.
- Il faudra aussi à un moment opportun se rendre dans **Project Structure** et choisir **Compile SDK version 29**.

3 Pour tester l'installation: Hello World!

Commencez un projet: Choisissez “Start a new Android Studio project”. À l'écran suivant, choisissez “Empty Activity”, et appuyez “Next”. Mettez le nom d'application “TP0”, un nom de package sympa (par exemple `fr.uparis.dupont`), le langage Kotlin, appuyez sur “Finish” et patientez. Le démarrage du projet est un peu lent la première fois, mais si vous avez l'impression qu'il bloque vraiment, appelez votre chargé de TD. L'écran de travail d'Android Studio doit s'ouvrir durant ce processus.

Configurez un émulateur – A faire une seule fois Dans le menu choisissez Tools→AVD Manager. Dans la fenêtre qui s'ouvre cliquez sur “create virtual device”, Sélectionnez Nexus 6.

Dans la fenêtre suivante choisissez l'onglet “x86 images”, puis une des images disponibles, par exemple Pie 9, puis “Next” et enfin “Finish”. Il y a un peu d'attente avant d'avoir une fenêtre “Your virtual devices”, fermez cette fenêtre (vous pouvez lancer l'émulateur).

Lancer l'application (et si nécessaire l'émulateur): Cliquez sur la flèche verte. Au bout d'un temps un peu long, un téléphone apparaît, puis votre application (Hello World) se lance, testez-la.

L'émulateur met un certain temps à charger, ne l'éteignez surtout pas jusqu'à la fin du TP, vous pourrez lancer vos applications dessus rapidement.

4 Optionnel: pour connecter votre téléphone

C'est parfois plus rapide et plus amusant d'exécuter vos programmes sur un vrai téléphone (ou tablette) sous Android. C'est très facile

<https://developer.android.com/studio/debug/dev-options>

et un peu magique:

- Sur votre téléphone allez à **Paramètres** → **A propos** → **Numéro de version**, que vous taperez 7 fois. Le message "vous êtes sous mode développeur" apparaîtra.
- Allez maintenant dans la nouvelle rubrique **Paramètres** → **Options pour développeurs** et activez le Débogage USB.
- Connectez le téléphone à la machine par câble USB.
- A partir d'Android Studio, vous pouvez maintenant lancer une application directement sur le téléphone, sans émulateur.