APPLI\_MOBILES\_Android\_TD1\_20161122

Présentation d’Android

1. Qu’est-ce qu’Android ? Quel est l’intérêt qu’il soit basé sur Kernel Linux ?

Android est un système d’exploitation pour smartphones et tablettes open source. Le fait qu’il soit basé sur Kernel Linux et qu’il possède une licence Apache, donne l’avantage qu’il sépare la couche matérielle de la couche logicielle, et par conséquent, n’importe quel téléphone Android peut faire fonctionner la même application.

1. Quelle est la version actuelle d’Android ?

La dernière version d’Android est Android 7, Nougat.

1. À quoi sert un émulateur ?

À pouvoir tester le rendu d’une application en simulant un smartphone précis ou une tablette Android (et donc avec ses propres caractéristiques d’affichage [résolution, taille d’écran, etc…]) avec une version d’Android choisie au préalable.

1. Pourquoi ne se sert-on pas d’Eclipse ?

Eclipse est certes un environnement de développement, mais Android Studio bénéficie de son propre SDK, qui embarque tous les outils nécessaires aux développeurs, et est plus simple d’installation et d’utilisation.

APK

1. Qu’est-ce qu’un APK ? Que contient-il ? De quels types ? Inconvénient de l’un des 2 types d’APK ?

Un APK (ou Android Package) est une archive compilée et packagée pour en faire une application Android. Il contient :

* Le code de l’application compilé en .dex
* Les ressources
* Les assets
* Les certificats
* Le fichier manifeste (xml)

Il existe deux types d’APK : signé et non signé. Le type non signé présente l’inconvénient de ne pas pouvoir être publié sur Google Play, même s’il peut être utilisé pour tester l’application.

1. Le Keystore, à quoi sert-il ?

Lorsque l’on crée un APK signé, notamment pour la distribution de l’application, la clé générée est stockée dans le Keystore : un fichier protégé qui contient toutes les clés de nos applications. Sans clé, pas de publication sur le store.

1. À quoi sert l’alias ?

L’alias correspond au nom de notre clé, demandé à chaque création d’un .apk.

Composantes d’Android

1. Qu’est-ce qu’une activité ?

Un écran représente une activité. C’est l’implémentation et les interactions des interfaces (mais pas des pop-up).

1. Qu’est-ce qu’un service ?

Le service ne possède pas d’interface. Il permet l’exécution d’un traitement en tâche de fond, comme lire de la musique en tâche de fond.

1. Qu’est-ce qu’un Content Provider ?

Le Content Provider, ou fournisseur de contenu, permet de partager les données d’une application. Elles peuvent être stockées en base SQLite sur le web ou dans des fichiers.

Cycle de vie d’une activité

1. 4 états à expliquer succinctement

* **En cours d’exécution :** Activité au 1er plan reçoit les interactions utilisateur. En cas de besoin de ressources, l’activité en bas de la Backstack est tuée.
* **En pause :** Activité partiellement cachée, l’utilisateur ne peut pas interagir.
* **Stopper :** Activité n’est plus visible mais toujours en cours d’exécution. Informations relatives à l’exécution gardées en mémoire.
* **Tuer :** L’activité est détruite, elle n’est plus en cours d’exécution et disparaît dans la BackStack.

1. Quelle est la méthode du cycle d’une activité que vous avez employée jusqu’à présent ?

La méthode onCreate().

Contexte d’une application

1. Que représente le contexte ?

L’état actuel d’une application et les informations sur son environnement. Il permet de récupérer des objets transmis par d’autres parties de l’application.

1. Que font les méthodes suivantes :
   1. getApplicationContext() : récupère le **contexte de l’application**
   2. getContext() : récupère le **contexte de la vue courante qui est exécutée**
   3. getBaseContext() : récupère le **contexte défini grâce à la méthode setBaseContext()**
   4. this : s’utilise **uniquement quand on se situe dans une classe héritant** (directement ou indirectement) **de la classe Context**

Le Manifeste

1. À quoi sert ce fichier XML ?

C’est une description complète de l’application : ses composantes, activités, services, permissions, fournisseurs de contenu, etc…

1. Où le trouve-t-on ?

À la racine de tout projet Android, dans le dossier manifests.

1. Que contient-il ?

Cela va varier en fonction de l’application, mais globalement :  
- Les permissions : droits d’accès à une fonctionnalité ou à une donnée ;  
- Les instrumentations : active à travers de tests la surveillance des interactions entre l’application et le système;  
- Uses-Configuration (ou données de configuration) : précise les spécificités de navigation supportées par votre application  
- Uses-Feature (spécificités matérielles) : précise les spécificités matérielles indispensables à l’utilisation de l’application (audio, bluetooth, caméra, gps, localisation, etc…) ;  
- Supports-screens ;  
- L’application.

1. Décrivez le contenu de la partie application (son contenu)

La partie application comprend toutes les spécificités et les composantes de l’application comme son nom, son icône, le thème, etc…

Architecture d’un projet

1. Que contient le répertoire java ?

Le répertoire java contient deux sous-dossiers :  
- Le code source de l’application ;  
- Le code source des tests.

1. Que contient le répertoire RES ? Décrivez le contenu de chaque sous répertoire.

Plusieurs sous-dossiers :  
- drawable : les ressources images et îcones utilisées  
- layout : les différentes interfaces (arborescence des vues)  
- menu (si il y en a un) : les menus contextuels ou d’application  
- mipmap : images et icônes utilisées  
- values : valeurs de configuration (langue entre autres), textes, etc…

1. Dans le répertoire Values, figure le Strings.xml, comment-vous en servir pour l’internationalisation de votre application ? Comment changez-vous la langue au final ?

Il sert de langue de base. Pour rajouter une langue, l’anglais par exemple, il faut créer un dossier values-en avec un autre fichier string.xml. Ce fichier contiendra les mêmes balises, mais le contenu entre les balises en anglais.