

Module :

Développement mobile native :



Design Pattern
&
Déploiement d'une application mobile sous Android

Pr. EL UAHHABI Zakia

Design Pattern

Les "**design patterns**" sont des solutions reconnues pour des problèmes de conception récurrents dans le développement logiciel.

En Android, plusieurs "**design patterns**" sont couramment utilisés pour organiser le code de manière efficace et maintenable.

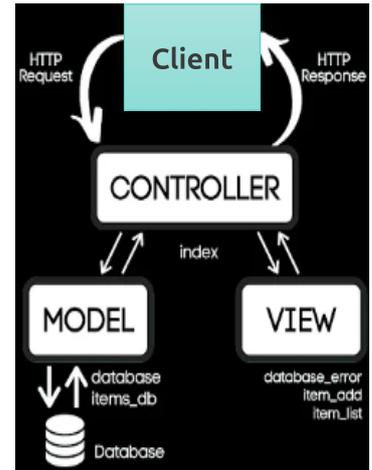
Voici quelques-uns des "**design patterns**" les plus courants en Android, accompagnés d'exemples d'organisation du code :



Design Pattern

Modèle-Vue-Contrôleur (MVC) :

- **Explication** : Divise l'application en trois composants principaux : le modèle (contient les données), la vue (interface utilisateur) et le contrôleur (gère les interactions entre le modèle et la vue).
- **Exemple d'organisation du code** :
 - Modèle : Classes de données ou d'accès aux données (ex : Base de données SQLite, classes de modèle de données).
 - Vue : Activités (Activity) ou fragments qui représentent l'interface utilisateur.
 - Contrôleur : Contrôleurs ou présentateurs (ex : classes qui réagissent aux événements utilisateur et mettent à jour le modèle ou la vue en conséquence).



3

Design Pattern

```
app/  
├─ src/  
│   └─ main/  
│       └─ java/  
│           └─ com.example.myapp/  
│               └─ model/  
│                   └─ Task.java  
│                   └─ TaskRepository.java  
│               └─ view/  
│                   └─ TaskListActivity.java  
│                   └─ controller/  
│                       └─ TaskListController.java  
│           └─ res/  
│               └─ layout/  
│                   └─ activity_task_list.xml
```



4

Design Pattern

Patron Singleton :

- **Explication** : Garantit qu'une classe n'a qu'une seule instance et fournit un point d'accès global à cette instance.
- **Exemple d'organisation du code** :
 1. Création d'une classe Singleton pour la gestion d'une connexion réseau ou d'une base de données.
 2. Déclarez un constructeur privé pour la classe.
 3. Créer une variable statique privée pour stocker l'instance unique de la classe à l'intérieur de celle-ci.
 4. Déclarez une méthode publique et statique pour récupérer l'instance du singleton.

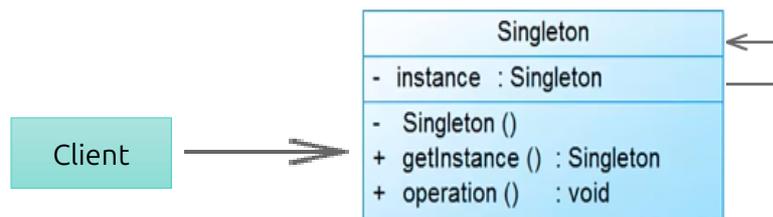


5

Design Pattern

Patron Singleton

-Structure :



- La classe `Singleton` déclare la méthode statique `getInstance` qui retourne la même instance de sa propre classe.
- Le code client ne doit pas avoir de visibilité sur le constructeur du singleton. Seule la méthode `getInstance` doit permettre l'accès à l'objet du singleton.



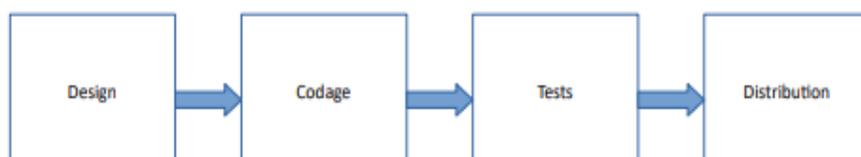
6

Déploiement d'une application mobile sous Android



Déploiement d'une application mobile sous Android

- La distribution de l'application est la dernière étape d'une série de processus :



Déploiement d'une application mobile sous Android

▢ Mettez de l'ordre dans votre application.

- Tests, nettoyage, version, nom, icône, etc.
- Par exemple, Désactivez le débogage : soit la retirer du fichier « AndroidManifest.xml » ou la mettre à « false ». `android:debuggable="false"`

▢ Signez votre application.

- Indispensable pour pouvoir la publier.
- Android exige d'une application qu'elle soit préalablement signée avant d'être installée ou mise à jour sur un appareil.

▢ Testez votre application comme un utilisateur final.

Se mettre à la place de l'utilisateur final

▢ Publiez votre application.

Déployez votre application en ligne.



Déploiement d'une application mobile sous Android

Signer l'application :

- ▢ Les applications dans « Google Play » sont représentées par un fichier au format « **apk** » ou « appbundle ».
- ▢ On commence par exporter le projet de l'interface de développement sous la forme désirée (apk ou appbundle).
- ▢ Cette procédure fait en sorte que le projet est exporté en mode « **release** » et non pas « **debug** ».
- ▢ Durant ce processus, nous sommes ramenés à signer l'application.
- ▢ Android exige d'une application qu'elle soit préalablement signée avant d'être installée ou mise à jour sur un appareil.

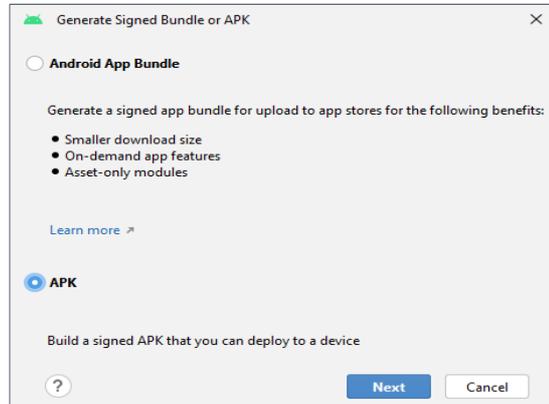


Déploiement d'une application mobile sous Android

Signer l'application :

▣ Signez l'application en utilisant Android Studio :

Dans le menu, cliquez sur « **build** » puis « **Generate signed Bundle/APK...** », cochez « **APK** », « **Next** »



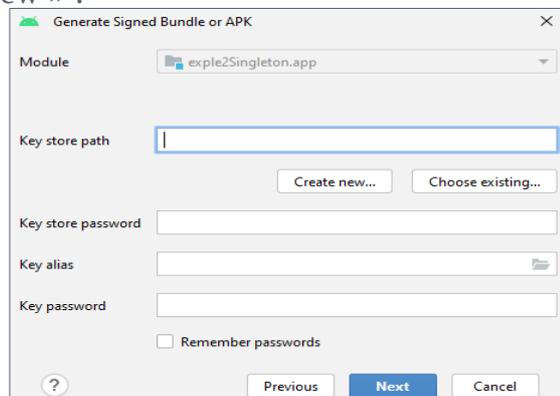
11

Déploiement d'une application mobile sous Android

Signer l'application :

▣ Signez l'application en utilisant Android Studio :

On doit préciser une clé. Pour cela, on va commencer par générer un dépôt (« Key store ») pour les clés. Cliquez sur « **Create new** » :



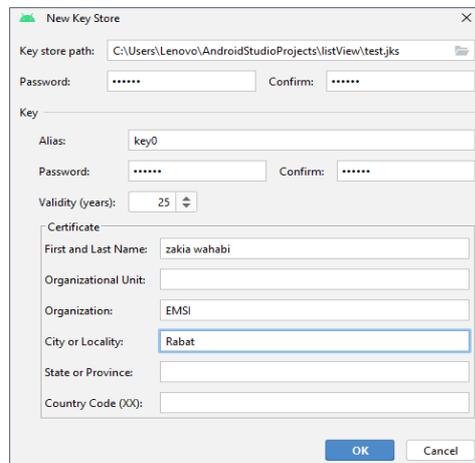
12

Déploiement d'une application mobile sous Android

Signer l'application

▣ Signez l'application en utilisant Android Studio :

Compléter les différents champs et cliquez sur « OK »

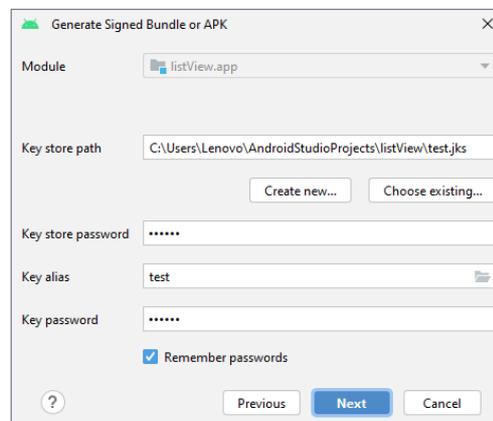


Déploiement d'une application mobile sous Android

Signer l'application

▣ Signez l'application en utilisant Android Studio :

On va obtenir cette fenêtre complétée, cliquez sur « Next » :

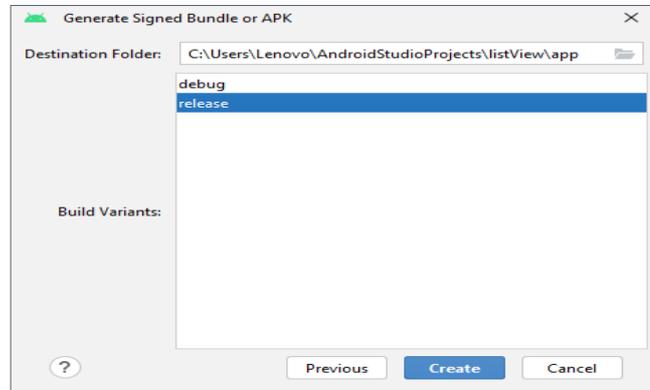


Déploiement d'une application mobile sous Android

Signer l'application

- ☐ **Signez l'application en utilisant Android Studio :**

Choisir la destination et la variante, puis cliquez sur « Finish ».



Déploiement d'une application mobile sous Android

Signer l'application

:

- ☐ **Signez l'application en utilisant Android Studio :**

- Si l'opération a réussi, vous allez obtenir ce message :

Generate Signed APK

APK(s) generated successfully for 1 module:

Module 'app': locate or analyze the APK.

- Noter le chemin où le paquetage « **APK** » sera sauvegardé.



Déploiement d'une application mobile sous Android

Signer l'application

:

▢ **Signez l'application en utilisant Android Studio - Signez un « bundle »:**

- Avant la version « 3.2 » d'Android Studio, le développeur devait générer une version du fichier « APK » pour les différentes variantes de son application (taille de l'écran, type de l'appareil, etc) .



- ✓ La taille de fichier APK très large.
- ✓ La création de plusieurs variantes nécessite d'inclure des heures de travail supplémentaires pour réaliser cette tâche.
- ✓ La taille de l'application a une influence directe sur la décision de l'utilisateur de l'installer ou pas



17

Déploiement d'une application mobile sous Android

Signer l'application

:

▢ **Signez l'application en utilisant Android Studio - Signez un « bundle »:**

- Pour pallier tous ces problèmes, Google a proposé une livraison dynamique. Le développeur doit inclure les différentes images pour les différents appareils dans un fichier compressé « bundle ». Au moment de l'installation, Google va générer de manière dynamique à partir de ce « bundle », le fichier « APK » qui correspond aux caractéristiques de l'appareil.

C'est l'approche recommandée par Google.



18

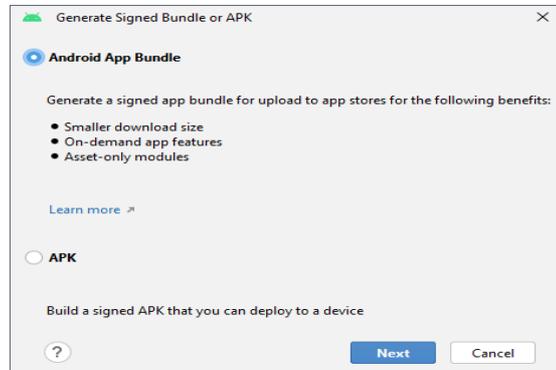
Déploiement d'une application mobile sous Android

Signer l'application



Signez l'application en utilisant Android Studio - Signez un « bundle »:

- Dans le menu, cliquez sur « build » puis « Generate signed Bundle/APK... » , cochez « Android App Bundle », « Next » :



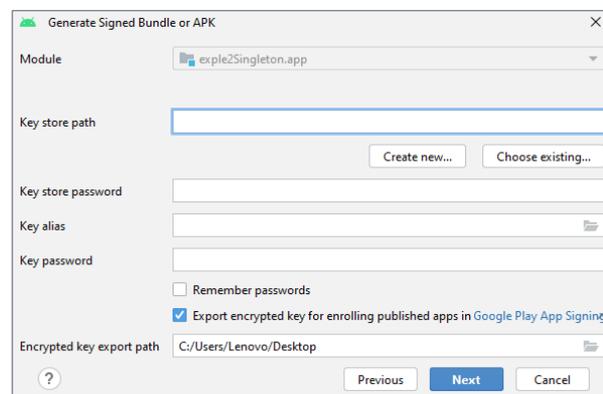
Déploiement d'une application mobile sous Android

Signer l'application



Signez l'application en utilisant Android Studio - Signez un « bundle »:

- Complétez les différents champs, puis cliquez sur « Next »:



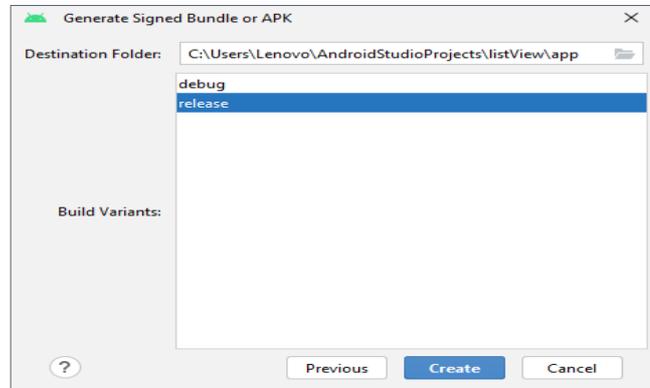
Déploiement d'une application mobile sous Android

Signer l'application

⋮

▢ **Signez l'application en utilisant Android Studio - Signez un « bundle »:**

- Choisissez la destination et la variante, puis cliquez sur « Finish ».



Déploiement d'une application mobile sous Android

Signer l'application

:

▢ **Signez l'application en utilisant Android Studio - Signez un « bundle »:**

- Si l'opération est un succès, vous allez obtenir cette série de messages :

Generate Signed Bundle

App bundle(s) generated successfully:

Module 'app': locate or analyze the app bundle.

Locate exported key file.



Déploiement d'une application mobile sous Android

Distribuer l'application :

- ▢ Assurez-vous d'abord d'avoir suivi les recommandations pour déployer l'application.
- ▢ Le déploiement peut se faire de plusieurs manières :
 - **Manuelle** : via le web, espace local ou sur le réseau.
 - **Boutique en ligne** : une boutique qui peut héberger des applications gratuites ou payantes, dédiées entre autres à des appareils Android. C'est l'option idéale si l'on veut ratisser large.



Déploiement d'une application mobile sous Android

Distribuer l'application :

▢ **Déploiement manuel**

- ▢ Vous pouvez donner accès à l'application via le web par exemple.
- ▢ Vous n'avez qu'à fournir le lien « URL » vers la page web en question.

`Telecharger App`

- ▢ Vous devez activer dans votre appareil l'option « **installation à partir d'une source inconnue** ».
- ▢ Le serveur qui héberge la page doit ajouter ce « **MIME** » (**Multipurpose Internet Mail Extensions type**):



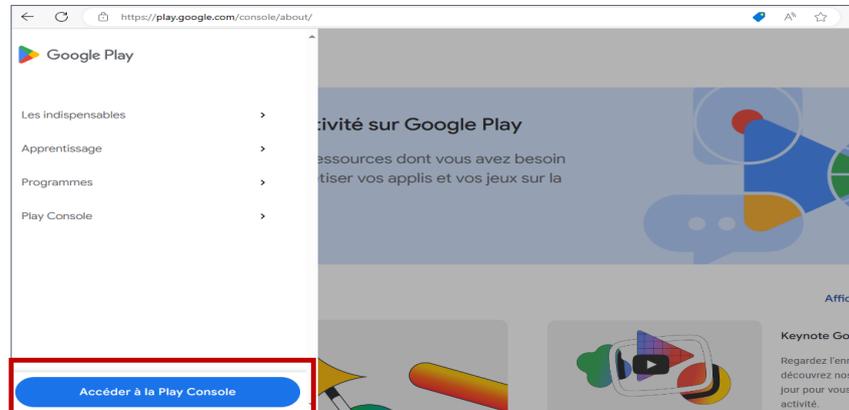
« application/vnd.android.package-archive »

Déploiement d'une application mobile sous Android

Distribuer l'application :

▮ **Déploiement à travers « Google Play »**

- ▮ « **Google Play Store** » est une place de marché qui a été créée et exploitée par Google qui permet aux Développeurs enregistrés dans certains pays de distribuer des Produits directement aux utilisateurs d'Appareils. Vous accédez au site suivant: <https://play.google.com/console/about/>



25

Déploiement d'une application mobile sous Android

Distribuer l'application :

▮ **Déploiement à travers « Google Play »**

- ▮ La boutique est utilisable par l'intermédiaire d'un compte Google, c'est-à-dire « Gmail ».

- ▮ Ce compte peut être associé à une personne physique ou morale.



- ▮ La personne gère les accès à la boutique pour tous les appareils dont elle est propriétaire, sans restriction de nombre.

26

Déploiement d'une application mobile sous Android

Distribuer l'application :

▢ Déploiement à travers « Google Play »

- ▢ Ainsi elle peut déployer une application payante ou gratuite acquise sur la boutique pour l'ensemble des appareils dont elle est propriétaire, en ne payant qu'une seule fois l'application (payante).



27

Déploiement d'une application mobile sous Android

Distribuer l'application :

▢ Déploiement à travers « Google Play »

- ▢ Pour rendre disponible une application via la boutique, il faut avoir un compte développeur.
- ▢ Un compte développeur est un compte de publication attribué aux développeurs qui permet la distribution de Produits via **le Play Store**.

Créer un compte de développeur

Ce nouveau compte de développeur sera associé au compte Google sélectionné. Si vous essayez de rejoindre un compte de développeur existant, veuillez contacter votre administrateur pour faire une demande d'invitation.

Si vous représentez une organisation, nous vous conseillons de ne pas utiliser un compte personnel pour configurer des comptes de développeur. Vous pouvez configurer des comptes Google avec une adresse e-mail existante. [En savoir plus](#)

ⓘ Pour créer un compte, vous devez vous acquitter des frais d'enregistrement de 25 \$ (payables une seule fois). Pour finaliser l'enregistrement de votre compte, vous devez peut-être valider votre identité à l'aide d'une pièce d'identité en cours de validité. Si votre identité ne peut pas être validée, les frais d'enregistrement ne vous seront pas remboursés.

Nom de développeur public *

Les utilisateurs de Google Play verront ce nom de développeur 0/50



28

Déploiement d'une application mobile sous Android

Distribuer l'application :

▢ **Déploiement à travers « Google Play »**

- ▢ Acceptez les conditions générales associées à la distribution sur « Google Play » pour les développeurs, puis cliquez « **Continue to payment** ».
- ▢ Réglez les frais d'inscription de 25USD.
- ▢ L'étape finale consiste à fournir des informations relatives à votre compte (nom du développeur, adresse courriel et numéro de téléphone).



Déploiement d'une application mobile sous Android

Distribuer l'application :

▢ **Déploiement à travers « Google Play »**

- ▢ Le compte développeur requiert des frais d'inscription uniques de 25\$.
- ▢ Pour une application payante : 30% des revenus sont pour Google, 70% pour le développeur.

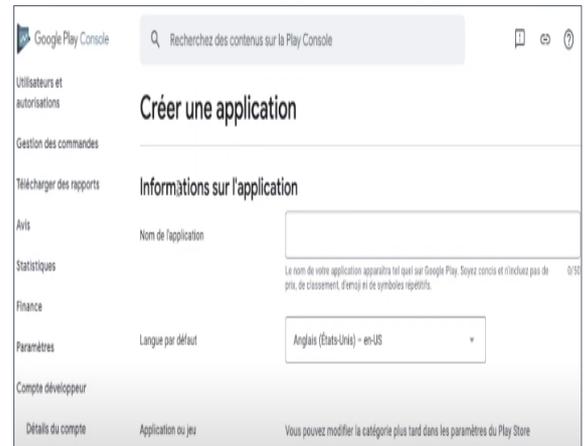


Déploiement d'une application mobile sous Android

Distribuer l'application :

▢ Déploiement à travers « Google Play »

- ▢ Google vous redirige par la suite sur cette page.
- ▢ Cette interface va vous permettre d'importer des applications, de définir les prix, d'ajouter des utilisateurs de comptes et de gérer les autorisations et finalement, de consulter des rapports, des statistiques et des informations relatives à vos applications.
- ▢ La publication d'une application nécessite l'envoi du fichier « apk » et de fournir des captures d'écran et des métadonnées relatives à votre application.



31

Déploiement d'une application mobile sous Android

Distribuer l'application :

▢ Déploiement à travers « Google Play »

- ▢ Une application peut-être :
 - **Payante** : elle est facturée avant qu'elle ne soit téléchargée.
 - **Gratuite** : mais vraiment gratuite sans les « extras » !
 - **Mixte** : offrir une version gratuite basic et une version améliorée payante.
 - **Financée par la Pub** : elle est gratuite, mais vous allez y inclure de la pub.
 - **Produits intégrés** : elle est gratuite, mais vous allez y inclure un contenu qui peut être acheté au fur et à mesure
 - **Abonnements** : le contenu nécessite le paiement d'un abonnement périodique.



32