Réaliser une application Android « TODO »

Contents

[VIDEO #1 3](#_Toc84376036)

[I Préparation du code : 3](#_Toc84376037)

[1. Création d’un nouveau projet 3](#_Toc84376038)

[2. Ajout des bibliothèques : 4](#_Toc84376039)

[II Début du codage 5](#_Toc84376040)

[1. Code de création de la base de données : 5](#_Toc84376041)

[2. Code model 6](#_Toc84376042)

[3. Ajout du contrôleur de base de données 9](#_Toc84376043)

[4. Ajout des vecteurs (Drawables) 12](#_Toc84376044)

[5. Ajout des couleurs et du thème de l’application 13](#_Toc84376045)

[6. Création du splash screen 13](#_Toc84376046)

[7. Changement du Manifest 15](#_Toc84376047)

[8. Supprimer la barre d’action de l’application 15](#_Toc84376048)

[VIDEO #2 15](#_Toc84376049)

[III Codage de l’interface principale 15](#_Toc84376050)

[IV Création des adaptateurs 17](#_Toc84376051)

[1. Java ListAdapter 17](#_Toc84376052)

[V listAdapter Layouts 19](#_Toc84376053)

[2. Suite code Java de l’adaptateur 22](#_Toc84376054)

[VIDEO #3 24](#_Toc84376055)

[3. Ajout des fonctions d’interaction avec la liste 24](#_Toc84376056)

[4. Adaptateur Java pour la liste vide 25](#_Toc84376057)

[VI Codage de la classe principale 26](#_Toc84376058)

[1. Initialisation des variables 26](#_Toc84376059)

[2. Appel à la base de données 27](#_Toc84376060)

[VII Modal de création et modification d’une TODO 28](#_Toc84376061)

[1. Création de l’interface 28](#_Toc84376062)

[2. Création du code Java pour l’interface 30](#_Toc84376063)

[VIII Finalisation de la classe principale 32](#_Toc84376064)

[1. Remplissage de l’instance listener 32](#_Toc84376065)

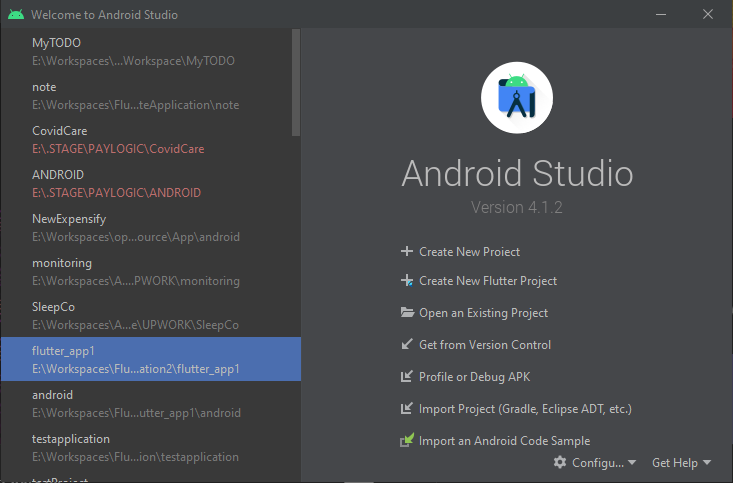
[2. Ajout des listeners des boutons de la page principale 33](#_Toc84376066)

# VIDEO #1

## Préparation du code :

### Création d’un nouveau projet

Pour créer un nouveau projet, ouvrez votre IDE Android Studio. Si une petite fenêtre comme celle-là apparait :

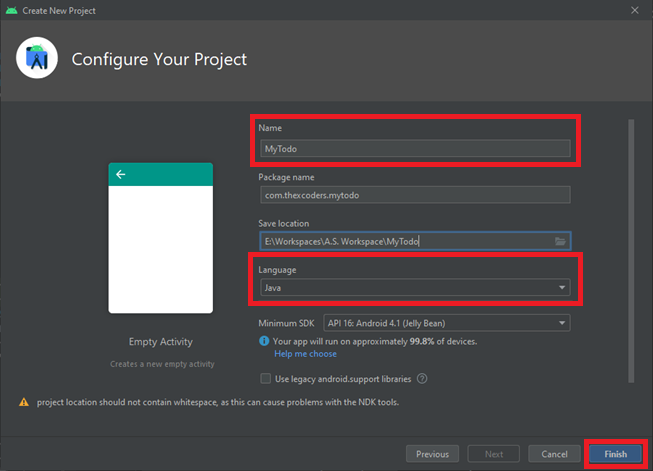


Cliquez sur ***Create new Project***

Sinon, si vous avez déjà ouvert un projet cliquer sur ***File > New > New Project…***

Sélectionnez ***Empty Activity*** *(sélectionné par défaut)* puis **Next.**

Pour le nom de l’application mettez « TODO » vérifier la sélection de Java puis appuyez sur ***Finish***

**

### Ajout des bibliothèques :

Une fois le projet généré, sur le menu à droite cliquez sur ***Gradle Script*** puis sur le premier fichier ***build.gradle (****Project :MyTODO).*

Allez à l’intérieur des accolades de ***repositories*** qui se trouve à l’intérieur de ***buildscript*** et assurez-vous qu’il y a deux lignes :



Faites la même chose pour le ***repositories*** qui se trouve à l’intérieur des accolades de ***allprojects***.

Si vous avez fait une modification, un bouton ***sync now*** apparaitra en haut à droite de votre IDE.

Ouvrez maintenant le fichier ***build.gradle*** *(Module :MyTODO.app)* et ajoutez les lignes suivantes dans ***dependencies***



## Début du codage

### Code de création de la base de données :

Sur le menu à gauche, cliquer sur le dossier **app**, puis **java** et faites un clic droit sur le package que vous avez choisis lors de la création du projet (dans mon cas c’est **com.thexcoders.mytodo**) ensuite cliquez sur ***New > java class****. Dans la fenêtre qui apparait tapez :*

**

Une classe sous le nom de ***DatabaseInitializer*** se crée. Après le nom de la classe et avant l’accolade ouvrante, tapez :

**

Cliquez sur ***implement methods*** et remplir les méthodes ou copiez le code suivant et mettez-le à l’intérieur des accolades :

**

### Code model

Le code model est comme un « wrapper » qui permet d’englober toutes les caractéristiques de la tache (TODO) comme le contenu, la date, l’id, et l’état.   
Créez la classe ***Todo*** dans le package ***model*** de la même manière que vous avez fait pour la création de la classe précédente.

(Clic droit sur le package du projet, puis ***new> Java class*** puis tapez : **model.Todo** )

On essaie de remplir la classe en commençant par les variables de classe qu’on met à l’intérieur des accolades

**

Ensuite on crée des constructeurs ainsi qu’une méthode statique qui permet de créer une instance de la classe ***Todo***.

**

Juste après, on ajoute des getters et des setters pour obtenir et modifier (respectivement) les données de toute instance de la classe ***Todo***



Après les getters et les setters on ajoute des fonctions et méthodes de reformatage des données de la ***Todo***



### Ajout du contrôleur de base de données

Créez une classe ***DbController*** à l’interieur du package ***Database*** en faisant un clic droit puis  ***new > Java class*** puis tapez uniquement le nom de la classe qui est : ***DbController.***

Le contrôleur nous permet de d’interconnecter la classe ***Todo*** et ses instances avec la base de données.

A l’intérieur de l’accolade, ajoutez ceci :

**

Puis ajoutez les fonctions :



Une fois terminé, ajoutez de la même façon une classe dans le même package (***database***) sous le nom de ***SQLExecution.*** Cette classe nous permet d’éxecueter le code d’accès à la base de données dans des threads ( des processus qui s’exécutent en parallèle pour ne pas bloquer l’application)

Ajoutez le contenu suivant à l’intérieur de la classe :



### Ajout des vecteurs (Drawables)

Ici on créer les icones qu’on utilise pour notre application. Les icones sont sous format XML (SVG)

Pour ajouter les icones, cliquez sur le dossier ***res > drawable*** puis faites un clic droit ***New > Drawable Ressource File*** puis collez le contenu du fichier à l’intérieur ou uniquement ***copier*** tous les fichiers allez sur le dossier ***drawable*** et faites un clic droit puis ***coller***

[LIEN DE TELECHARGEMENT DES DRAWABLES](https://drive.google.com/file/d/1wQJDw0W9gy3JzjCaPr6tyw8CZuQcWBwz/view?usp=sharing)

Si vous avez des problèmes pour la copie des fichiers vous pouvez faire un clic droit sur un des fichiers à l’intérieur du dossier ***Drawable***puis ***show in explorer*** ceci ouvre un dossier où vous allez devoir coller les ***drawables*** téléchargés

### Ajout des couleurs et du thème de l’application

Dans le dossier ***res*** cliquez sur ***values > colors.xml*** ajoutez les lignes suivantes :



Une fois terminé ouvrez le fichier ***themes > themes.xml*** et changez les ***item*** dont le nom est ***colorPrimary***  et ***colorPrimaryVariant*** vers les valeurs suivantes :



Puis ajoutez la ligne suivante n’importe où à l’intérieur des tags ***style***



### Création du splash screen

Pour créer une nouvelle page allez au menu et cliquez sur :

***File > New > Activity > Empty Activity***

Pour ne nom entrez ***SplashScreen*** *et* cliquez sur ***Finish***

Ouvrez la fenêtre ***activity\_splash\_screen.xml*** qui se trouve dans le dossier ***res > layout*** et remplissez TOUT le contenu avec ce code :



De retour vers le code java associé dans la classe ***SplashScreen*** ajoutez les lignes suivantes à l’intérieur des accolades de la méthode ***onCreate()***



Ce code attend 2 seconds (2 000 millisecondes) puis ouvre la classe ***MainActivity*** à partir de la classe où on est qui est ***SplashScreen.***

### Changement du Manifest

Afin de lancer la classe ***SplashScreen*** en premier au lien de ***MainActivity*** ouvrez le dossier ***manifests*** puis le fichier ***AndroidManifest.xml***

Cherchez la ligne où il y a



Et permutez la propriété ***name*** entre les deux valeurs : .***SplashScreen*** *et* **.MainActivity**

### Supprimer la barre d’action de l’application

Allez sur le fichier **res > values > themes > theme.xml**et changez la dernière valeur après le dernier point de l’attribut ***parent*** du tag ***style*** de la valeur ***DarkActionBar*** vers ***NoActionBar.***

# VIDEO #2

## Codage de l’interface principale

De retour à l’interface principale : ***activity\_main.xml*** qui se trouve dans l’emplacement ***res > layout***

Ajoutez l’attribut background au tag principal pour changer la couleur d’arrière-plan de l’application en ajoutant la ligne suivante :



Supprimez le tag ***<TextView*** avec tous ses attributs etcopiez le contenu suivant :



Et juste après, ajoutez le code suivant :



Ouvrez maintenant la classe ***MainActivity*** qui se trouve sur ***java > [vote package]*** et ajoutez les variables de classe :



## Création des adaptateurs

### Java ListAdapter

Afin de créer notre adaptateur faites un clic droit sur votre package puis ***New > java class*** et tapez :

***adapters.ListAdapters***

Ajoutez le code suivant avant l’accolade ouvrante :



Puis ajoutez la classe interne ***ViewHolder*** en copiant ce code à l’intérieur des accolades :



Implémentez et remplissez les fonctions ou copiez le code suivant à l’intérieur des accolades de la classe ***ListAdapter***:



Puis ajoutez ce code à l’intérieur des accolades de la classe ***ViewHolder***



Ajoutez l’interface ainsi que les variables de classes et le constructeur pour la classe ***ListAdapter***



## listAdapter Layouts

Maintenant on crée les layouts de notre Adapter, on commence par l’adaptateur vide :

Allez sur ***res > layouts*** et faites un click droit puis ***New > Layout Ressource File***, tappez le nom :

***todo\_empty\_item***

Puis collez le code suivant :



Refaites la même chose pour le fichier ***todocard*** avec le code suivant :



### Suite code Java de l’adaptateur

Revenez vers le code Java qui se trouve sur ***app > java > [votre package] > adapters > ListAdapter***

Remplissez la méthode ***onCreateViewHolder()*** par le code suivant :



Changez le contenu de la classe interne ***ViewHolder*** avec le contenu suivant :



Dans la méthode ***OnBindViewHolder()*** ajoutez le code suivant :



Ajoutez les méthodes complémentaires quelque part avant la classe interne ***ViewHolder*** comme suit :



Changez le ***0*** de la méthode ***getItemCount()*** vers

***todoList.size() ;***

# VIDEO #3

### Ajout des fonctions d’interaction avec la liste

Ajoutez les méthodes suivantes après la définition de l’interface (après l’accolade fermante)



### Adaptateur Java pour la liste vide

Dans le package ***adapters*** créez une classe java sous le nom de ***EmptyTodoListAdapter****,* avant l’accolade ouvrante de la classe ajoutez le code suivant :



Puis collez à l’intérieur de la classe le code suivant :



## Codage de la classe principale

### Initialisation des variables

Copiez les deux méthodes suivantes sur votre code :





Appelez les deux méthodes à l’intérieur de la méthode ***onCreate()***



***Vous pouvez faire un petit test***

### Appel à la base de données

Ajoutez la méthode suivante après ***clickListeners()***



***Vous pouvez faire un petit test***

## Modal de création et modification d’une TODO

### Création de l’interface

Faites un click droit sur ***layout*** puis ***New > New Ressource File*** et nomez le ***todo\_editor\_layout*** et remplacez la totalité du code par le code suivant :



### Création du code Java pour l’interface

Faites un click droit sur le package puis ***new > Java Class*** et tapez dans le nom

***modals.TodoModal***

Avant l’ouverture de l’accolade de la classe ajoutez le code suivant :

extends Dialog

Ensuite collez ce code à l’intérieur de la classe ( l’intérieur des accolades)



Ajoutez la méthode **onCreate**en faisant un override oubien en copiant le code suivant :



Ajoutez les méthodes ***setupViews()*** *et* ***setupClickListeners()***avec ce code :



## Finalisation de la classe principale

### Remplissage de l’instance listener

A la fin de la méthode ***InitVariables()*** dans la classe ***MainActivity***, ajoutez ce lignes :



Après la méthode ***clickListeners()*** ajoutez les trois méthodes suivantes



### Ajout des listeners des boutons de la page principale

Dans la fonction ***clickListeners()*** supprimez les toast et remplacez le premier code de « Edit » par :



Et remplacez le Toast de « Add New » par :



Et voilà!

Your android application is ready to be published!

You have just finished an **average** **difficulty** application

If you enjoyed this mini-tutorial and you have any questions my contact information will be at the end of this document as well as some useful links that I recommend.



Ce document a été rédigé par Anass AIT BEN EL ARBI, élève ingénieur en génie Informatique option IQL promotion 2022.

Si vous avez des requêtes ou des questions, vous pouvez me les communiquez par mail ou par LinkedIn

**Contact info:**

[annassabe@gmail.com](mailto:annassabe@gmail.com)

[anassaitbenelarbi@student.emi.ac.ma](mailto:anassaitbenelarbi@student.emi.ac.ma)

<https://www.linkedin.com/in/anass-abea/>

**Useful links:**

My Fav Mobile Development Channel:

<https://www.youtube.com/c/CodinginFlow>

Mobile development community

<https://discord.com/invite/TSnMvmc>

Android MVVM architecture

<https://www.journaldev.com/20292/android-mvvm-design-pattern>