



04/2024

# Documentation Note App

Android Studio



Lucas LAGARDERE, Melvil NGUYEN, Edouard  
JANIN, Roméo LEHMANN  
ESEO E4A S8

## Table des matières

Documentation .....	2
A. NoteViewModel .....	2
B. NoteViewModelFactory .....	2
C. NoteAdapter .....	2
D. NoteActivity .....	4
E. AddActivity .....	5
F. UpdateActivity .....	5
G. NoteDatabase .....	5
H. NoteRepository.....	6
I. Note .....	6
J. NoteDao .....	7
Guide d'utilisation de l'application .....	8
1. Création d'une Nouvelle Note .....	8
2. Affichage des Notes .....	8
3. Mise à Jour d'une Note .....	8
4. Suppression d'une Note .....	8

# Documentation

## A. NoteViewModel

### Méthodes :

`insertNote(note: Note)`

*Insère une nouvelle note dans la base de données via le repository.*

`deleteNote(id: Int)`

*Supprime une note existante de la base de données via le repository.*

`updateNote(note: Note)`

*Met à jour une note existante dans la base de données via le repository.*

### Variables :

`notesLiveData : LiveData<List<Note>>`

*LiveData pour observer les changements dans la liste des notes.*

## B. NoteViewModelFactory

### Méthodes :

`create(modelClass: Class<T>): T`

*Crée une instance de ViewModel.*

`modelClass` : La classe du ViewModel à créer.

*Vérifie si la classe fournie est une instance de NoteViewModel.*

*Si oui, crée et retourne une nouvelle instance de NoteViewModel avec le repository fourni.*

*Si non, lance une exception IllegalArgumentException.*

## C. NoteAdapter

### Méthodes :

`onCreateViewHolder(parent: ViewGroup, viewType: Int): NoteViewHolder`

*Crée un ViewHolder en fonction du type de vue à afficher.*

viewType : Le type de vue à afficher.

onBindViewHolder(holder: NoteViewHolder, position: Int)

*Lie les données de la note à l'élément de vue dans le ViewHolder.*

holder : Le ViewHolder à mettre à jour.

getItemCount(): Int

*Retourne le nombre total d'éléments dans la liste de notes.*

loadNotes(notes: List<Note>?)

*Charge une liste de notes dans l'adaptateur.*

notes : La liste de notes à charger.

### **Variables :**

notes: MutableList<Note>

*Liste des notes à afficher dans l'adaptateur.*

Initialisation : Initialisée avec une liste mutable vide dans le constructeur.

### **Classes internes :**

NoteViewHolder

*ViewHolder pour chaque élément de la liste de notes.*

### Composants :

noteTitle: TextView : Titre de la note.

noteContent: TextView : Contenu de la note.

editNoteImage: ImageView : Icône d'édition de note.

deleteNoteImage: ImageView : Icône de suppression de note.

### Méthodes :

bindNote(note: Note): Lie les données de la note à l'élément de vue dans le ViewHolder.

## D. NoteActivity

### Méthodes :

onCreate(savedInstanceState: Bundle?)

*Méthode appelée lors de la création de l'activité.*

savedInstanceState : Les données de l'instance précédemment sauvegardées.

setNoteRecycler()

*Configuration du RecyclerView pour afficher la liste des notes.*

onActivityResult(requestCode: Int, resultCode: Int, intentData: Intent?)

*Méthode appelée lorsque l'activité d'ajout de note retourne un résultat.*

requestCode : Le code de requête utilisé pour lancer l'activité d'ajout de note.

resultCode : Le code de résultat indiquant le succès ou l'échec de l'activité d'ajout de note.

intentData : Les données retournées par l'activité d'ajout de note.

### Variables :

newNoteActivityResultCode : Int

*Code de requête utilisé pour lancer l'activité d'ajout de note.*

noteViewModel : NoteViewModel

*ViewModel associé à cette activité pour gérer les données de note.*

binding : ActivityNoteBinding

*Binding pour lier le layout de l'activité.*

noteAdapter : NoteAdapter

*Adapter pour afficher la liste des notes dans le RecyclerView.*

## E. AddActivity

### Méthodes :

onCreate(savedInstanceState: Bundle?)

*Méthode appelée lors de la création de l'activité.*

savedInstanceState : Les données de l'instance précédemment sauvegardées.

### Variables :

binding : ActivityAddBinding

*Binding pour lier le layout de l'activité.*

## F. UpdateActivity

### Méthodes :

onCreate(savedInstanceState: Bundle?)

*Méthode appelée lors de la création de l'activité.*

savedInstanceState : Les données de l'instance précédemment sauvegardées.

### Variables :

binding : ActivityUpdateBinding

*Binding pour lier le layout de l'activité.*

noteViewModel : NoteViewModel

*ViewModel associé à cette activité pour gérer les données de note.*

## G. NoteDatabase

### Méthodes :

getNoteDAO(): NoteDao

*Méthode abstraite pour obtenir l'interface DAO associée à la base de données.*

getNoteDatabase(context: Context): NoteDatabase

*Crée une nouvelle instance de la base de données si elle n'existe pas encore.*

## H. NoteRepository

### Méthodes :

insertNote(note: Note)

*Insère une nouvelle note dans la base de données.*

deleteNote(id: Int)

*Supprime une note de la base de données en fonction de son ID.*

updateNote(note: Note)

*Met à jour une note dans la base de données.*

### Variables :

mAllNotes : LiveData<List<Note>>

*LiveData contenant la liste de toutes les notes.*

### Constructeur :

NoteRepository(mNoteDao: NoteDao?)

*Initialise le repository avec le DAO des notes.*

Paramètres : mNoteDao : Le DAO des notes.

## I. Note

### Propriétés :

id : Int

*Identifiant de la note.*

Annotation : @PrimaryKey indiquant que cette propriété est la clé primaire de la table.

mNoteTitle : String

*Titre de la note.*

Annotation : @ColumnInfo pour spécifier le nom de la colonne dans la table de la base de données.

mNoteContent : String

*Contenu de la note.*

Annotation : @ColumnInfo pour spécifier le nom de la colonne dans la table de la base de données.

### **Héritage :**

La classe Note implémente Serializable pour pouvoir être sérialisée et désérialisée.

## **J. NoteDao**

### **Méthodes :**

insertNote(note: Note)

*Insère une nouvelle note dans la base de données.*

deleteNote(id: Int)

*Supprime une note de la base de données en fonction de son ID.*

deleteAll()

*Supprime toutes les notes de la base de données.*

getAllNotes(): LiveData<List<Note>>

*Récupère toutes les notes de la base de données sous forme de LiveData.*

updateNote(note: Note)

*Met à jour une note dans la base de données.*

## **K. NoteApplication**

La classe NoteApplication est une classe qui initialise la base de données et le repository de l'application.



# Guide d'utilisation de l'application

## 1. Création d'une Nouvelle Note

Pour créer une nouvelle note dans NoteApp, suivez ces étapes :

1. Lancez l'application NoteApp sur votre appareil Android.
2. Sur l'écran d'accueil, appuyez sur le bouton flottant "+" pour créer une nouvelle note.
3. Dans l'écran de création de note, saisissez un titre pour votre note dans le champ "Titre".
4. Ensuite, saisissez le contenu de votre note dans le champ "Contenu".
5. Une fois que vous avez saisi le titre et le contenu de votre note, appuyez sur le bouton "Ajouter cette note" pour l'enregistrer.

## 2. Affichage des Notes

Pour afficher toutes vos notes dans NoteApp, suivez ces étapes :

1. Lancez l'application NoteApp sur votre appareil Android.
2. Sur l'écran d'accueil, vous verrez la liste de toutes vos notes existantes.
3. Faites défiler la liste pour afficher toutes vos notes. Chaque note affiche son titre et son contenu.

## 3. Mise à Jour d'une Note

Pour mettre à jour une note existante dans NoteApp, suivez ces étapes :

1. Sur l'écran d'accueil de NoteApp, trouvez la note que vous souhaitez mettre à jour.
2. Appuyez sur la note pour ouvrir son contenu.
3. Sur l'écran de détail de la note, appuyez sur le bouton d'édition (icône de crayon) pour modifier la note.
4. Dans l'écran de mise à jour de la note, modifiez le titre et/ou le contenu de la note selon vos besoins.
5. Une fois que vous avez terminé les modifications, appuyez sur le bouton "Mettre à jour" pour enregistrer les modifications.

## 4. Suppression d'une Note

Pour supprimer une note dans NoteApp, suivez ces étapes :

1. Sur l'écran d'accueil de NoteApp, trouvez la note que vous souhaitez supprimer.
2. Appuyez longuement sur la note pour sélectionner la note.

3. Une fois la note sélectionnée, vous verrez une option de suppression apparaître en haut de l'écran.

4. Appuyez sur l'icône de suppression (corbeille) pour supprimer la note de manière permanente.