

Listes et fragments

On souhaite créer une petite app de gestion de notes personnelles. Chaque note a un titre et un contenu (sous forme de texte).

En mode écriture, on se contentera d'afficher une zone de texte éditable et un bouton "Envoyer".

En mode consultation, l'app sera décomposée en deux parties : la liste des notes enregistrées (affichant uniquement leur titre), et le détail d'une note (affichant titre et contenu de la note sélectionnée).

Dans un premier temps, on ne se préoccupera pas du stockage à long terme de ces notes (elles seront détruites à la fermeture de l'app).

1. Récupérer le squelette d'app dans l'archive *RecyclerView.tar.gz*. Il contient un ensemble de classes à compléter dans le cadre de cet exercice.
2. La classe Note est destinée à représenter une note (titre + contenu, tous deux de type String). Elle doit comporter des méthodes publiques *getTitre()* et *getContenu()* permettant d'accéder à son contenu. Implémenter cette classe dans le fichier existant (les méthodes à implémenter sont déjà déclarées, mais vides).
3. Compléter ensuite la classe ListeNotes permettant de gérer une liste de notes. Cette classe disposera des méthodes suivantes :
 - *ajouteNote(String titre, String contenu)* créant une nouvelle note et l'ajoutant en fin de liste ;
 - *get(int)* renvoyant une note à partir de sa position ;
 - *deleteNote(int)* permettant d'effacer la note dont l'index est passé en paramètre ;
 - *deleteNote(String)* permettant d'effacer la note dont le titre est passé en paramètre, et retournant l'index de la position qu'occupait cette note ;
 - *size()* retournant le nombre de notes stockées.NB : le constructeur de cette classe contient déjà quelques lignes permettant d'ajouter automatiquement quelques exemples de notes dans tout ensemble que l'on crée. Ne pas retirer ces lignes pour l'instant (elles serviront de test).
4. exécuter le programme. Les notes de tests devraient s'afficher dans la partie haute de l'écran (leur contenu étant cependant tronqué dans cette liste)

NB : le squelette d'app fourni comprend deux fragments : celui affiché en haut s'occupe de l'affichage des notes, tandis que celui du bas aura pour fonction de gérer l'affichage complet d'une note sélectionnée. Il sera câblé plus tard.

Modification

La modification d'un élément d'une liste affichée via un RecyclerView se fait en deux temps : d'abord, on procède à la modification proprement dite, ensuite, on notifie au RecyclerView.Adapter le changement via la méthode la plus appropriée.

La liste des méthodes permettant d'effectuer ces notifications est disponible dans la documentation de référence¹.

1. Dans NoteFragment, créer une méthode *addNote* permettant d'ajouter une note à la liste et notifiant cet ajout à l'adapter.
2. Créer une méthode *deleteNote* permettant de supprimer une note à la liste à partir de son titre (et notifiant aussi cette suppression à l'adapter).
3. Premier test : Ajouter un bouton Créer ayant pour effet d'ajouter une nouvelle note (au titre et contenu aléatoires dans un premier temps) via la nouvelle méthode *addNote*.

1. <https://developer.android.com/reference/android/support/v7/widget/RecyclerView.Adapter.html>

4. Second test : Ajouter un bouton Supprimer ayant pour effet de supprimer la première note de la liste via la nouvelle méthode `deleteNote`

Communication Activity/Fragment et affichage complet d'une note.

L'appui sur un élément de la liste ne provoque pour le moment qu'un affichage complet de la note dans le Logcat. On veut pouvoir afficher la note à l'écran dans le fragment correspondant. Cela s'effectue par le truchement de la méthode *onListFragmentInteraction* définie dans l'interface *NotesFragment.OnListFragmentListener* que l'activité principale implémente.

1. Récupérer l'instance du *NoteDetailFragment* que l'activité principale a instancié à sa création. Pour ce faire, on peut utiliser le *FragmentManager* de l'activité (accessible via *getSupportFragmentManager*), possédant une méthode *findFragmentById*.
2. Compléter la définition de la méthode *onListFragmentInteraction* de manière à appeler la méthode *Update* du fragment récupéré dans la question précédente.

Finitions

1. Créer une activité retournant un résultat (elle sera appelée via *startActivityForResult*) permettant à l'utilisateur de saisir une nouvelle note. Cette activité comprendra deux zones *EditText* (une pour le titre, l'autre pour le contenu) et un bouton *Valider*.
Faire en sorte que l'appui sur le bouton *Creer* défini dans *NotesFragment* appelle désormais cette activité et crée la note en fonction de ce que l'utilisateur a entré.
2. faire en sorte que le bouton *Supprimer* défini dans *NotesFragment* supprime désormais la note dont les détails sont actuellement affichés (indice : on voudra garder un identifiant de cette note lorsque l'on affiche les détails).
3. faire en sorte que l'affichage des deux fragments se fasse côte à côte (et non plus l'un en dessous de l'autre) lorsque l'appareil est tourné en mode paysage. (c'est possible en ne touchant qu'au répertoire *res*)