

## CR TP Séquence 2

### 1. Introduction

J'ai repris mon projet Android du TP Séquence 1 et j'ai utilisé pour la première fois la librairie Retrofit pour effectuer les requêtes sur l'API REST *todo-api* que j'ai mis en œuvre sur ma machine à partir du logiciel MAMP.

Il m'a fallu beaucoup de temps pour réussir à mettre en œuvre l'API REST *todo-api* sur ma machine puisqu'elle était sous Macintosh et que seules les mises en œuvre sur des machines Linux et Windows avaient été explicitées.

Concernant l'utilisation de la librairie Retrofit, cela ne m'a pas posé problème avec les cours de M.Boukadir.

### 2. Analyse

D'une manière générale l'application fonctionne correctement et répond aux fonctionnalités exigées dans l'énoncé du TP voir aussi quelques fonctionnalités facultatives (modification du label d'un item ou d'une liste, suppression d'un item ou d'une liste, suppression de tous les items d'une liste ou de toutes les listes, amélioration de la visualisation des items d'une liste par le tri entre les items non cochés et cochés par une disposition des items non cochés au-dessus des items cochés).

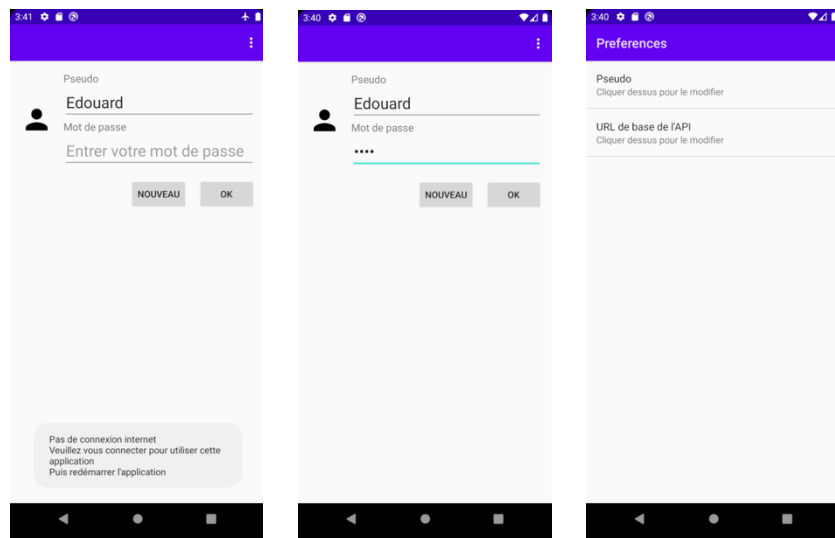
En résumé, j'ai modifié le code des 4 activités suivantes : *MainActivity*, *ChoixListActivity* et *ShowListActivity* qui hérite de la classe *AppCompatActivity*, qui ont chacune leur layout et leur menu respectifs, et qui utilise la même toolbar ; et *SettingsActivity* qui hérite de *PreferenceActivity* qui a son propre fichier xml.

J'ai également modifié les 3 fichiers kotlin suivant : les classes *ListeAdapter* et *ItemAdapter* qui hérite de *RecyclerView.Adapter* et qui implémentent les méthodes nécessaires à la gestion des adaptateurs des recyclerviews des activités *ChoixListActivity* et *ShowListActivity* pour l'affichage des listes et des items dans les listes, et *Data* qui contient 3 classes *User*, *Liste* et *Item* nécessaires au stockage des données de la base de données *todo-api.sql* ainsi que 4 nouvelles classes *LoginResponse*, *UsersResponse*, *ListesResponse* et *ItemsResponse* nécessaire à la désérialisation des fichiers Json transmis par l'API *todo-api* en réponse aux requêtes effectuées.

J'ai dû créer un nouveau fichier kotlin *ServiceApi* qui regroupe l'ensemble des requêtes qui sont faites auprès de l'API *todo-api*.

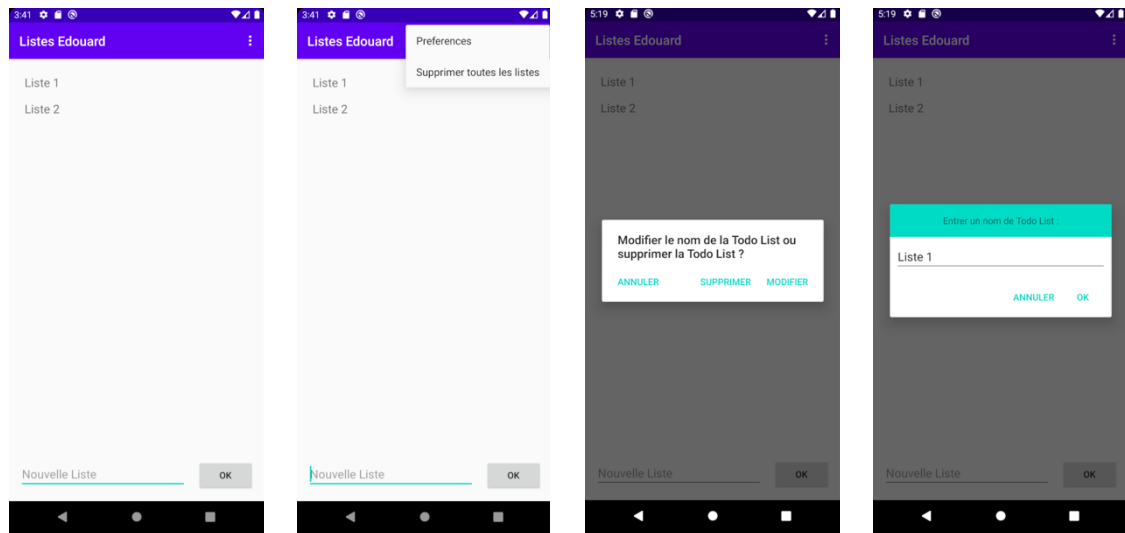
Par ailleurs je n'ai pas eu à modifier les 3 layouts suivant : *liste* et *item* qui sont utilisés au sein des ViewHolders des recyclerviews des activités *ChoixListActivity* et *ShowListActivity*, *dialog\_new* qui est utilisé lors de l'affichage d'AlertDialog pour modifier le label d'un item ou d'une liste.

Ci-dessous des captures d'écran de l'application en fonctionnement :

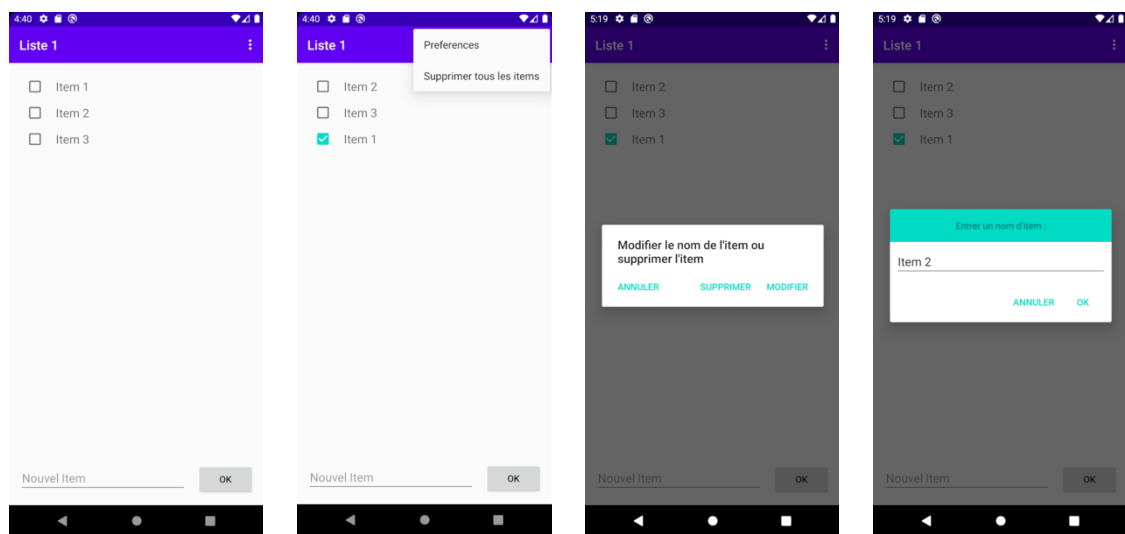


*MainActivity*

*SettingsActivity*



*ChoixListActivity*



*ShowListActivity*

### 3. Conclusion

A l'issue de la fin de la séquence 2, je suis satisfait du rendu de l'application et de son fonctionnement. Et la programmation de cette application m'a permis de comprendre l'utilisation des APIs REST pour récupérer et stocker des données ainsi que la programmation Android avec la librairie Retrofit pour utiliser une API REST.

J'ai également pu améliorer et mettre plus au propre le code Android effectué pour la séquence 1.

### 4. Perspective

Concernant l'activité Settings, je pourrais ajouter une préférence sur la taille de la police et une préférence sur la langue de l'application afin de proposer l'application dans diverses langues.

Concernant la disposition des listes et la disposition des items de ces listes, je pourrais ajouter une fonctionnalité permettant de les déplacer afin de pouvoir choisir l'ordre de priorité des listes ainsi que des items au sein d'une liste.

Concernant l'activité *ShowListActivity*, je pourrais permettre l'association d'un lien à un item et la possibilité d'afficher ce lien dans un navigateur web.

### 5. Bibliographie

Developer Guides <https://developer.android.com/guide/>

Material Design <https://material.io/design>

Les cours et screencasts des séquences 1, 2 & 3 de l'électif PMR.