

Introduction

Dans ce TP, on voudrait améliorer l'application développée à la séquence 1 en récupérant les listes et les items des utilisateurs depuis une API Rest.

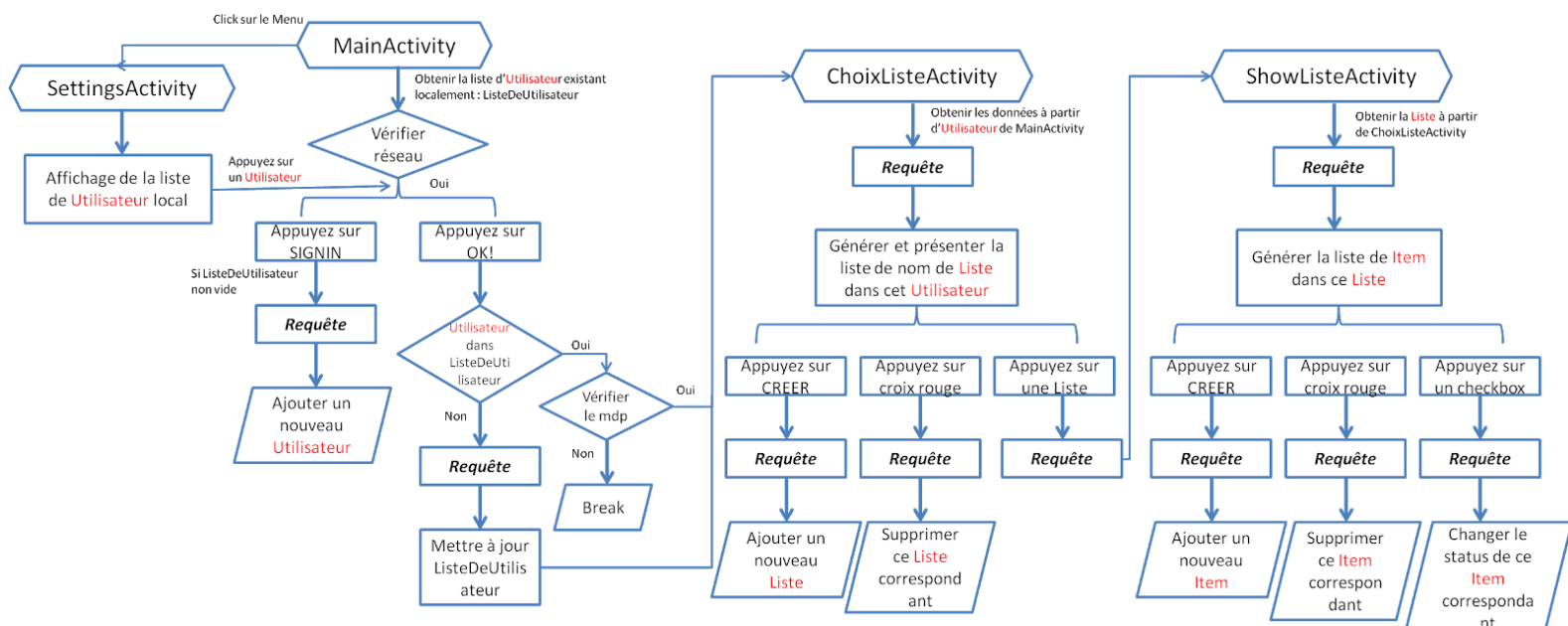
Pour réaliser les requêtes, on choisit d'utiliser la librairie Retrofit. D'après les règles de l'API todo-api, on encapsule tous les requêtes POST, GET, DELETE et PUT. On utilise les intents implicites à réaliser la transmission de données entre les activités.

Remarque : le programme de séquence 1 est dans la branche master, le programme de séquence 2 est dans la branche WEI.

Analyse

En bref, le programme que l'on réalise dans la séquence 2 réalise la même fonction que ceci de séquence 1, on change la transmission de données. Donc, la structure de programme ne change pas beaucoup.

On construit encore 4 class d'activity. Mais on ajoute un java class requestServiceFactory qui nous aide à créer l'objet de Retrofit, et un java interface requestService où on utilise des annotations pour décrire les requêtes. Voici la conception de notre programme.



Tout d'abord, on explique le fonctionnement de requête par retrofit : dans le java class requestServiceFactory on crée l'objet de Retrofit, et dans le java interface requestService où on utilise des annotations pour décrire les paramètres de requête réseau et configurer les paramètres de requête réseau. Pendant l'utilisation, on crée une instance de requestService,

et puis on encapsule la requête d'après les règles de Interface requestService, enfin on l'envoie et collecte le résultat. D'après le résultat on fait l'action prochaine.

Quand l'utilisateur lance l'application, il est dans l'interface de MainActivity. On construit une fonction `getProfiles()` pour créer une `listeDeUtilisateur` qui stock les utilisateurs locaux. Comme on transmet les données par l'Internet, il faut vérifier le réseau avant d'action de l'utilisateur. Pour créer un nouveau utilisateur en appuyant sur SIGNIN, il faut un hash code, donc il faut `listeDeUtilisateur.getUtilisateurs().isEmpty()`. Pour la connexion, On guide l'utilisateur à entrer le pseudo et le mot de passe. Le dernier pseudo saisi est automatiquement renseigné dans le champ de saisie. Si le mot de passe est correct, l'utilisateur entre dans l'interface de ChoixListeActivity en appuyant sur le bouton OK!.

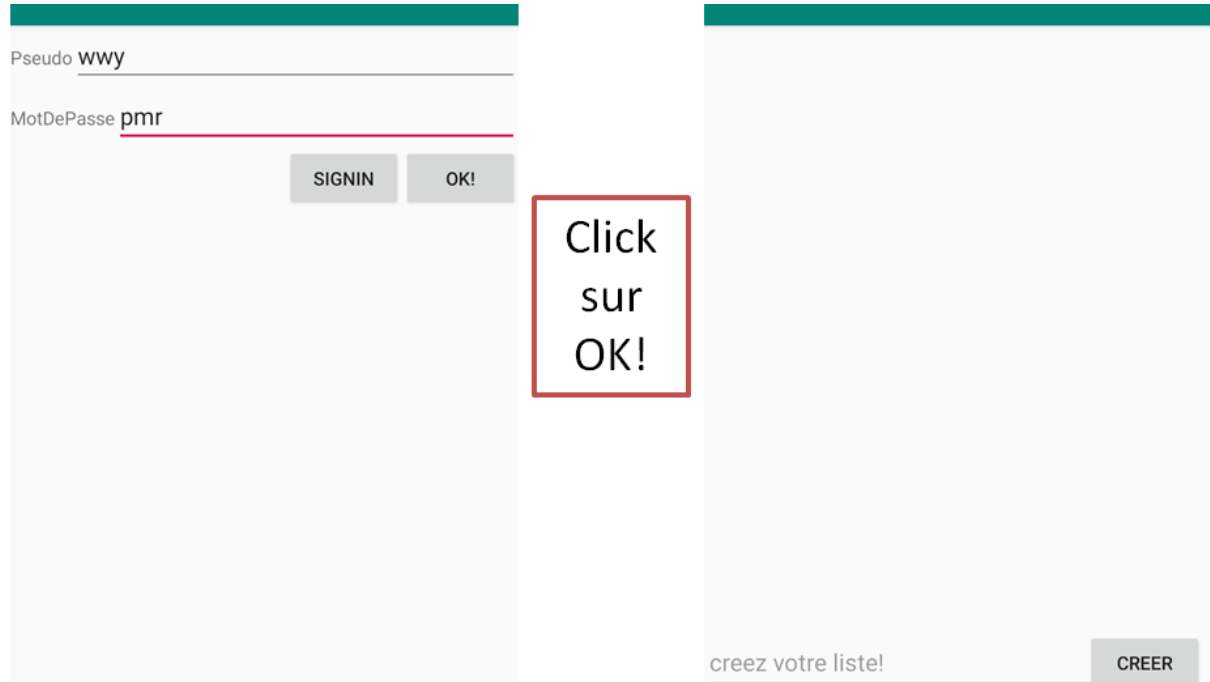
Dans la construction de fonction `ConvertToList()`, on utilise des `Intent.putExtra()` pour la transmission de donnés. L'application affiche la liste de Liste de ce utilisateur. Dans cet interface, l'utilisateur peut ajouter un nouveau Liste ou supprimer le Liste cliqué.

En cliquant sur un Liste, l'utilisateur peut entrer dans la liste d'Items de ce Liste. L'application affiche la liste de Items de ce Liste. Dans cet interface, l'utilisateur peut ajouter un nouveau Item, supprimer un Item ou changer le status d'un certain item.

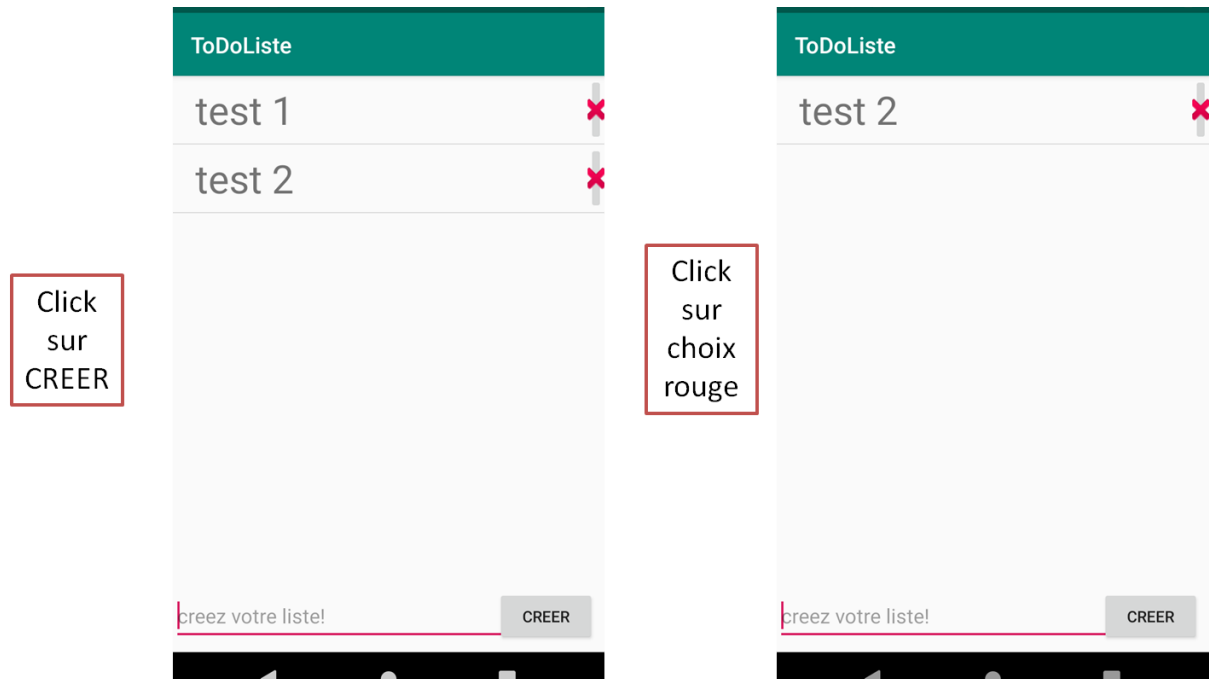
En cliquant sur le menu mis sur l'interface de MainActivity, l'utilisateur obtient tous les utilisateurs de préférence. Dans cet interface, l'utilisateur peut cliquer un profil pour revenir à l'interface de MainActivity avec ce profil automatiquement renseigné dans le champ de saisie.

Démonstration

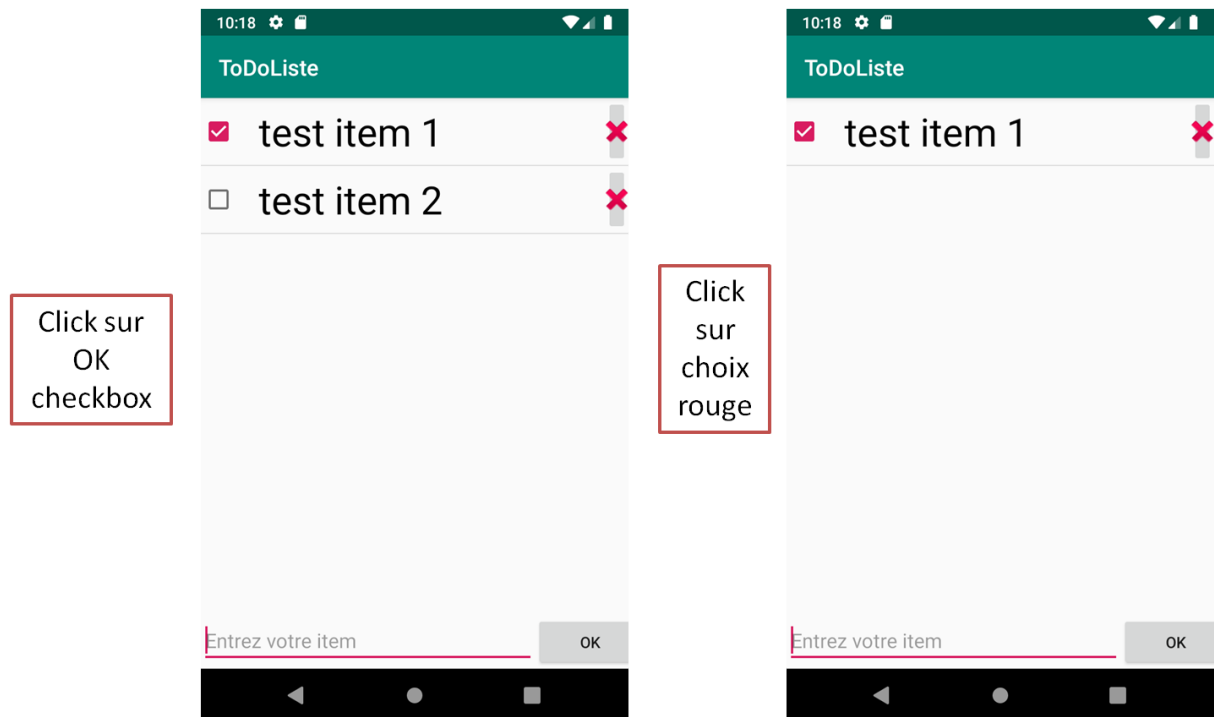
MainActivity compte existant nécessaire



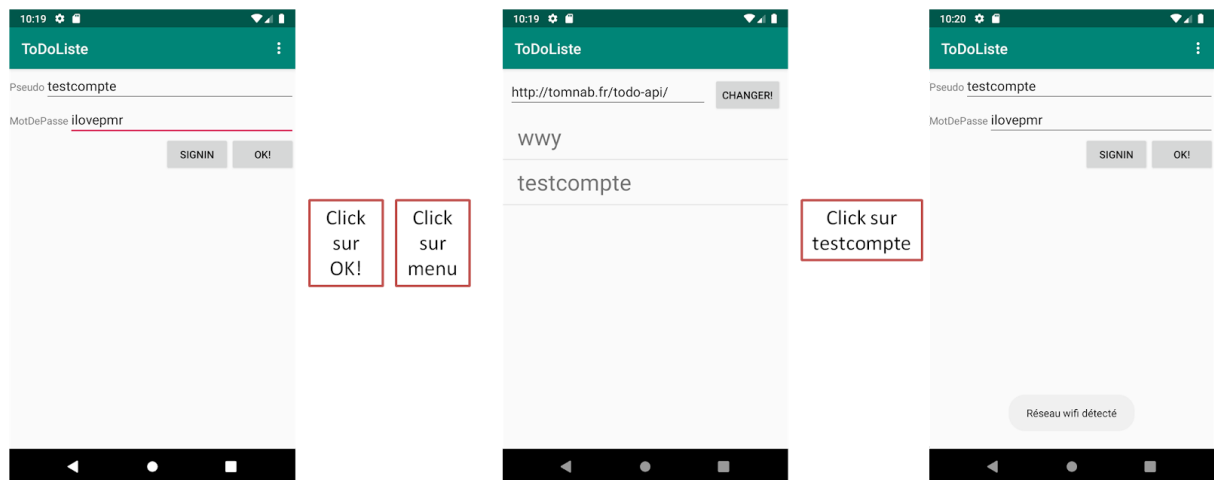
ChoixListeActivity



ShowListeActivity



Créer un nouveau utilisateur, SettingsActivity vs MainActivity



Conclusion

Pendant cette séance, basé sur l'application réalisée dans la séquence 1, on réussi à récupérer les listes et les items des utilisateurs depuis une API Rest. On réalise les requête par la librairie Retrofit, qui est une nouvelle méthode de faire des requêtes. On étudie bien le fonctionnement de Retrofit et l'utilise bien dans l'application.

Transformer une application locale en une application connectée avec l'Internet est un grand progrès. A part, on réalise aussi les autres perspectives proposées à la fin de séquence 1 comme l'ajoute de fonction de suppression. On trouve que la réalisation d'une application est un processus pas à pas. Nous sommes impatients d'apprendre la prochaine fonction.

Bibliographie

<https://cloud.tencent.com/developer/article/1179344>

<https://documenter.getpostman.com/view/375205/S1TYVGTa?version=latest>

https://blog.csdn.net/carson_ho/article/details/73732076

https://blog.csdn.net/Picasso_L/article/details/50523790