DAA Rapport Lab06

Auteurs: Edwin Häffner, Guillaume Dunant, Arthur Junod

Frontend avec Composable

Nous avons décider d'utiliser la version du laboratoire utilisant composable. L'affichage des composables se fait avec une variable contenue dans le ViewModel `applicationStatus`. Selon son état, l'utilisateur verra la liste des contacts, l'écran de création d'un nouveau contact ou bien l'écran d'édition/suppression de contact. Cet état est mis à jour lorsque l'utilisateur appuie sur certains boutons spécifique (comme appuyer sur un contact, ou bien en appuyant sur le bouton "+" pour ajouter un nouveau contact ,etc.).

Dans l'affichage de la liste des contacts, si un contact n'est pas encore synchronisé avec la database sur le net, il sera affiché avec un font rouge. Du moment ou le contact est synchronisé, son affichage repasse à la normale.

Enrollment

Pour l'enrollment, nous avons mis en place une *MutableLiveData* dans le **ContactRepository** qui prend la valeur de l'UUID retourné par l'endpoint *"/enroll"*, si nous avons réussi à l'atteindre. L'UUID est exposé par le **ContactViewModel** pour le récupérer dans l'Activité afin de le sauvegarder dans les préférences. Au démarrage de l'application, si on arrive à récupérer un UUID depuis les préférences, on va le passer au Repository en passant par le ViewModel et on essaye de synchroniser la DB locale et en ligne. Si, à l'inverse, on ne trouve pas d'UUID dans les préférences, alors on va en demander automatiquement un nouveau en appelant la méthode `enroll()` du ViewModel. Cette méthode appelle simplement l'`enroll()` du Repository dans une coroutine qui va récupérer l'UUID depuis l'endpoint et le poster dans sa LiveData.

Synchronisation

Le remote ID de chaque contact permettra de trouver sur quel contact de la DB en ligne il faudra répercuter les changements faits sur la DB locale (sauf pour la création où on n'en aura aucun, car il sera créé quand on insérera ce nouveau contact dans la DB en ligne). Les contacts modifiés (MOD), supprimés (DEL) ou créés (NEW) ont tous un statut différent de celui de base (OK) quand ces actions sont seulement effectuées dans la DB locale.

Prenons la modification de contact :

- Quand on change les informations d'un contact, on modifie la DB locale ; celui-ci aura alors son statut changé de OK à MOD tant que la modification ne sera pas répercutée sur la DB en ligne.
- Si le changement a pu être envoyé à la DB en ligne, alors le statut de notre contact sera remis à OK.
- Sinon, il sera laissé sur MOD.
- Ce contact garde son statut, et quand nous appelons la fonction `refresh()` depuis le bouton de l'UI, il sera détecté comme "dirty" et on essaiera de répercuter le changement décrit par son statut sur la DB en ligne.

Il y a donc différentes étapes pour chaque changement fait à la DB :

1. Application du changement sur la DB locale avec un statut qui reflète ce changement.

2. Essai de répercussion de ce changement sur la DB en ligne.
3. Si succès, alors on remet le statut du contact à OK.
4. Si échec, alors on laisse le statut du contact comme il est afin qu'il soit géré plus tard par l'appel à la synchronisation à travers la fonction `refresh()` qui réessayera les points 2 et 3 ou 4 suivant la réussite ou l'échec.

Le refresh se fait en parcourant la DB locale. Pour chaque contact :

- On l'ignore si son statut est OK.
- On essaie de faire le changement décrit par le statut sur la DB en ligne pour tout autre.