

# Rapport n°3

## I. Introduction

Le but de ce TEA 3 était de développer un mode hors ligne pour l'application développée lors des deux travaux précédents.

## II. Analyse

### 1. Accès aux données hors ligne

Pour concevoir le mode hors-ligne, on va utiliser une base de données en local qui stocke les données de l'utilisateur quand celui-ci se connecte. Afin d'y parvenir, on utilise la librairie open source Room basée sur le fonctionnement et la syntaxe de Retrofit (aussi open source). Ainsi, on procède de la même manière que lors du TEA précédent.

Tout d'abord, il faut créer des classes de données qui vont correspondre au format de la base de données, dont les tables sont décrites ci-dessous.

- ListEntity (table "list")
  - id (primary key)
  - label
  - userHash
- ItemEntity
  - id (primary key)
  - listId (foreign key)
  - label
  - checked
  - url

De la même manière qu'avec Retrofit, on écrit les requêtes SQL dans des interfaces DAO (Data access object). Ces requêtes ont les mêmes fonctionnalités que les requêtes à l'API.

Ensuite, on adapte le DataProvider de sorte à ce que lorsque la requête API échoue, ce soit la base de données locale qui soit sollicitée. Lorsque la requête est une demande d'ajout et de modification, on modifie la base de données locale.

### 2. Synchronisation des modifications

Pour assurer la synchronisation entre les données de l'API et les données locales de l'application, on utilise une troisième table dans notre base de données :

- PendingEntity
  - id (primary key)
  - type (*addList / addItem / deleteList / updateItemChecked / updateItemName*)
  - hash (de l'utilisateur)
  - listId (nullable)
  - itemId (nullable)
  - label (nullable)

- checked (nullable)
- url (nullable)

On stocke alors les différents changements effectués dans la base de données locale lorsque la connexion à l'API échoue.

On met alors en place une méthode dans le DataProvider qui vide cette table en y appliquant les modifications sur l'API. Cette méthode est appelée dans la MainActivity lorsque l'utilisateur démarre l'application et qu'il a accès à internet.

### **III. Conclusion**

L'application implémente désormais un mode hors-ligne qui permet d'utiliser l'application et d'interagir avec tout en permettant une synchronisation distante à posteriori.

### **IV. Perspectives**

Dans l'implémentation actuelle, le mode hors-ligne ou le monde en ligne est décidé au lancement de l'application. On pourrait mettre en place un basculement dynamique entre les deux modes lorsque l'état de la connexion change.