

Explication des choix

Fonctionnement général de l'application :

- L'application a comme concept de lister des animes pour l'utilisateur en forme d'une liste ou d'un grid. Lorsque l'utilisateur clique sur un de ces animes l'application lui redirige vers une page affichant ses détails. En plus, l'application enregistre un historique des animes consultés. La page de l'historique peut être consultée en utilisant la barre de navigation en bas de l'application.

Architecture du projet :

- Pour avoir une application propre et bien structurée j'ai décidé d'utiliser un des patterns d'architecture des applications android. J'ai choisi MVVM car il est un des Design Pattern les plus utilisés.

Librairies et Concepts utilisés :

- RecyclerView : Passage facilement entre List et Grid View.
- Fragments : Permettre à avoir des affichages différents sans avoir besoin de créer plusieurs activités.
- Repository : Avoir une classe responsable de communiquer avec les API et avec la base de données locale.
- ViewModel : Responsable d'utiliser les repositories afin de récupérer les données et les envoyées à l'adaptateur du recyclerView pour l'affichage.
- Retrofit : Permettre d'établir les connexions, envoyer les appels et recevoir les réponses.
- OkHttp : Un client HTTP que j'utilise avec Retrofit pour la récupération des données des APIs publiques.
- GSON : permettre de convertir les réponses des API en des objets JAVA.
- SQLite : Permettre de stocker les données en local.

Les choix qui ont été fait :

- Initialiser la base de données dans la répositorie et initiliser ce dernier dans l'activité Main. Puisque c'est la répositorie qui est responsable de contacter avec la base de données locale.
- Utiliser un task async lors de stockage des données dans la base de données au lieu d'utiliser le type **Single** puisqu'on attend aucun retour.
- Garder un lien entre de l'adapteur du recyclerView dans le fragment afin de le notifier du changement des données lors de leurs réception du viewModel.
- Utiliser le type **Single** afin de recevoir les données soit de la base de données soit de l'API publique sans bloquer le Main Thread.
- Afficher le vue liste et le vue grid dans un seul fragment et passer de l'un à l'autre en tapant sur un simple bouton. Car je n'ai trouvé aucun raison pour faire un deuxième fragment juste pour un changment de Layout.