

Listes et fragments - Un peu plus de fun

Recherche de messages dans le ViewModel L'un des avantages du modèle MVVM est la capacité du ViewModel à traiter et préparer les données spécifiquement pour l'UI. Dans cette section, nous allons ajouter une fonctionnalité de recherche de messages directement dans le ViewModel, permettant ainsi à l'utilisateur de filtrer et d'afficher des messages pertinents.

1. Recherche basique :

- Dans le *Message ViewModel*, ajoutez un *LiveData* qui stockera la chaîne de recherche. (on pourra ainsi éventuellement l'afficher ailleurs sans se soucier de la mise à jour.)
- Ajoutez également un *LiveData* pour la liste filtrée des messages. Cette liste sera mise à jour en fonction de la valeur de *searchQuery*.
- Implémentez une méthode *searchMessages(query: String)* qui mettra à jour *searchQuery* et filtrera la liste des messages en fonction de cette requête.
- Dans l'UI, ajoutez une barre de recherche (vous pouvez utiliser un *EditText* accompagné d'un bouton pour valider, ou directement un *SearchView*). Lors de la soumission d'une requête, elle devrait appeler *searchMessages(query: String)* de votre ViewModel.

2. Affichage conditionnel :

- Modifiez l'interface pour afficher soit la liste complète des messages, soit la liste filtrée, selon la présence ou non d'une requête de recherche.
- Implémentez une fonctionnalité pour afficher un message "Aucun résultat trouvé" si la requête ne renvoie aucun message correspondant.

3. Effacement de la recherche :

- Ajoutez une fonctionnalité permettant d'effacer la requête de recherche. Cela pourrait être un bouton "Effacer" à côté de la barre de recherche ou une icône "x" intégrée dans celle-ci. (NB : si vous avez utilisé la *SearchView*, normalement, c'est déjà fait ☺)
- En cliquant sur ce bouton ou cette icône, l'affichage devrait revenir à la liste complète des messages.

Affichage des derniers messages Il pourrait être utile pour les utilisateurs de ne voir que les messages les plus récents, surtout si la liste devient très longue.

1. Exposition des 10 derniers messages :

- Dans le *MessageViewModel*, ajoutez une méthode *displayLastMessages(count: Int)*. Cette méthode filtrera et exposera les *count* derniers messages de la liste.
- Dans l'UI, ajoutez un bouton ou une option pour "Afficher les 10 derniers messages" qui, lorsqu'il est activé, appelle *displayLastMessages(10)*.

2. Retour à l'affichage complet :

- Ajoutez une option ou un bouton "Afficher tous les messages" qui, lorsqu'il est activé, réinitialise l'affichage pour montrer tous les messages, sans se limiter aux 10 derniers.

Bonus: Thème sombre Les applications modernes offrent souvent une option de thème sombre pour améliorer la lisibilité dans des conditions de faible luminosité.

1. Ajoutez un bouton ou une option dans les paramètres de l'application pour permettre à l'utilisateur de basculer entre le thème clair et sombre.
2. Utilisez les ressources Android et les styles pour implémenter un thème sombre pour l'application.