Proto DataStore in Action

Mehdi Haghgoo

@IAmMehdiHaghgoo

in/MehdiHaghgoo

Section Overview

- Connect our Proto DataStore to ProtoViewModel
- Add ComicsProtoScreen that will enable us to see how the Proto DataStore works
- Run and verify the app

Code Challenge Review

- Add **BY_NAME** sort order in **user_pref.proto** file
- Add enableSortByName() function in **UserProtoPreferencesRepository**

ProtoViewModel

The ProtoViewModel will consume functions present in UserProtoPreferencesRepository

```
 fun filterComicsByCategory(
    comicsCategory: ComicCategory
 ) {}

fun sortComicsByRating(enabled: Boolean) {}

fun sortComicsByDateAdded(enabled: Boolean) {}

fun sortComicsByName(enabled: Boolean){}

fun disabledSorting() {}
```

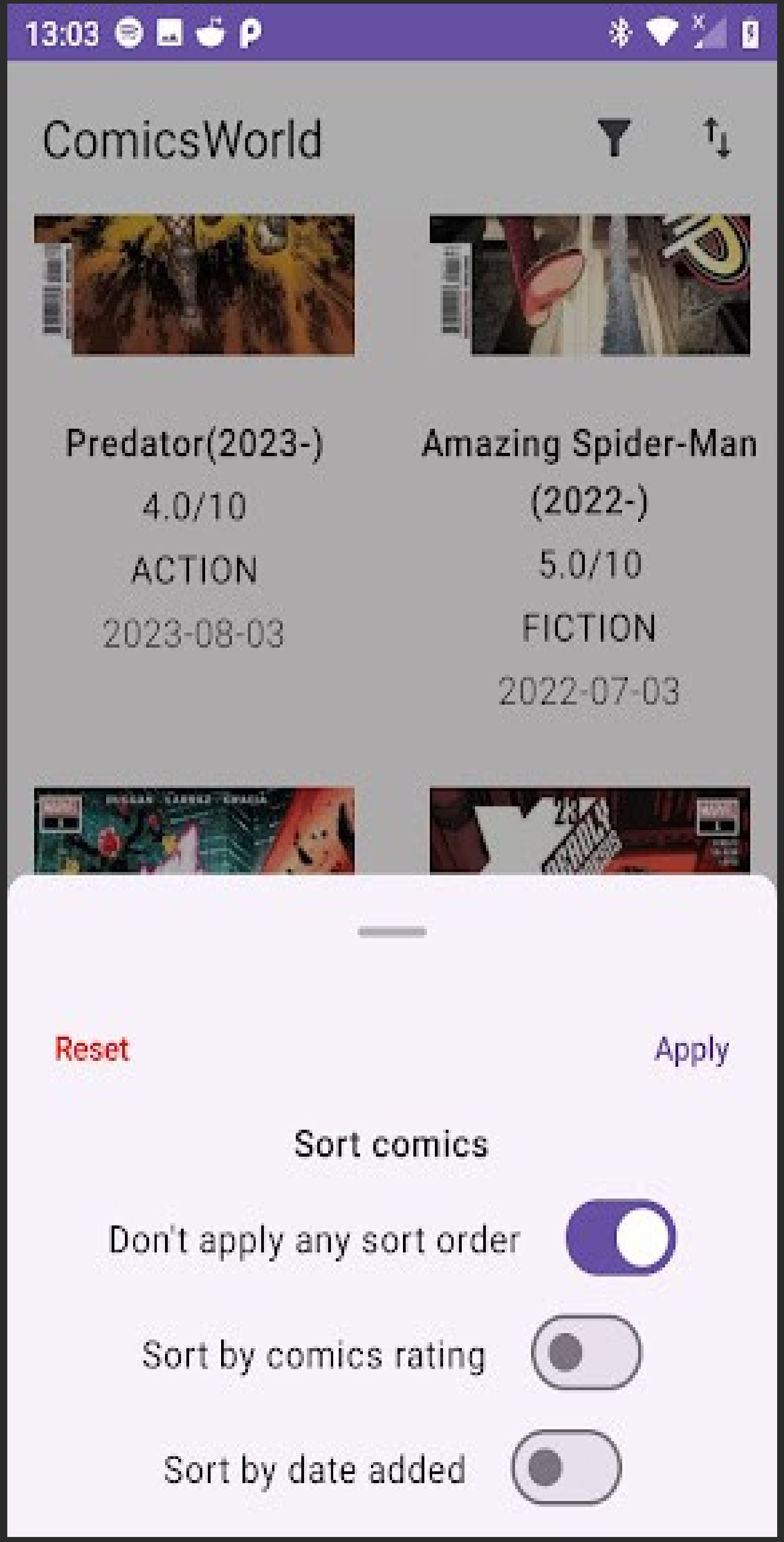
droidcon academy

ComicsProtoScreen

- UI Screen the user will interact with
- The user's sort options will be persisted inside our Proto DataStore
- The ui-screen will interact with the Proto DataStore via the ProtoViewModel

```kotlin
@Composable
fun ComicsProtoScreen(comicsProtoViewModel:
ProtoViewModel = viewModel ( )  )  {
    val comicsUiModel by
comicsProtoViewModel.comicsUiModel.collectAsState ( )
    ComicsScreenView(
        comics = comicsUiModel.comics,
        userPreference=comicsUiModel.userPreference,
    FilterComicsByCategory=
comicsProtoViewModel::filterComicsByCategory ,
        DisableSorting =
comicsProtoViewModel::disabledSorting)
}
```

Running the App



Predator(2023-)
4.0/10
ACTION
2023-08-03

Amazing Spider-Man (2022-)
5.0/10
FICTION
2022-07-03

Reset
Apply

Sort comics

Don't apply any sort order [✓]

Sort by comics rating

Sort by date added

Summary

- Create a ViewModel that acts as a bridge between the our Proto DataStore and the UI screen
- Create the ComicsProtoScreen
- Verify that the Proto DataStore is working

Testing Preferences and Proto DataStores

Up Next