

Jetpack DataStore

Jetpack DataStore is a modern data storage solution from **Android Jetpack**, designed to replace the older **SharedPreferences API**. It provides a more robust, scalable, and asynchronous way to store small amounts of data such as user preferences or app settings.

There are two types of DataStore:

1. Preferences DataStore:

- Stores **key-value pairs**, similar to SharedPreferences.
- Does **not require a predefined schema**.
- Use case: Simple user preferences like dark mode, notification toggles, etc.



```
val Context.dataStore: DataStore<Preferences> by preferencesDataStore(name = "settings")
```



2. Proto DataStore:

- Stores **typed objects** using **Protocol Buffers**.
- Requires a **schema definition** (.proto files).
- Use case: Complex or structured data (like user profiles).

```
val Context.userDataStore: DataStore<UserPreferences> by datastore(  
    fileName = "user_prefs.pb",  
    serializer = UserPreferencesSerializer  
)
```

Key Benefits:

-  Built on Kotlin Coroutines → **asynchronous and safe**.
-  **Transactional consistency** using Flow and suspend functions.

-  Handles data migration from SharedPreferences.
-  Works seamlessly with **Jetpack Compose** and **modern Android architecture**.