

## Various approaches to storage management on Android

When approaching storage management, there are various factors that need to be considered. First and foremost, it is vital to identify what kind of data needs to be stored, as well as the nature of the data (size, frequency of use, access). Needless to say, the type of data will influence the storage management techniques. Android has some best practices when it comes to storage management

Android provides two physical storage locations: internal and external

- App specific files can be stored in internal storage. The problem with this is that if you have a large amount of data that needs to be stored, internal storage will not be sufficient; other types of storage will need to be used for substantial amounts of data. External storage directories can then be leveraged for both persistent files as well as cache data.
- Similarly, any data that needs to be present for basic functionality such as startup should be stored in internal storage or a database.
- Data that is only meaningful for the specific app should be stored in app-specific storage.
- Sensitive data should be stored in internal storage directories as other apps are barred from accessing this data, and these locations are encrypted.
- Shared storage is for data that is fine to be accessed by other apps and is saved on app uninstall. The main types of shareable data types that are stored include media content, pdf and epub books, as well as datasets.
- A small collection of key-value pairs can be saved in either SharedPreferences (outdated) or DataStore.

Pros and Cons for my project (Food Travel App)

- DataStore can be used for user preferences considering that we can represent the user's culinary preferences in a small set of key value pairs. The con of this would be if we began to ask for more preferences (which might happen as the app scales in order to improve the algorithm that solves the cold start problem), we might need a larger storage. Alternatively, since we want to save the user's preferences on app uninstall (or tie it to an account) shared storage could be a potential solution.
- While internal storage may be convenient for my app, external storage needs to be used if the app needs access to large amounts of data, which it likely will in order to give restaurant recommendations based on location.
- Some form of shared storage will likely be used in order to gain access to google maps API.
- Room can be used to persist data locally and cache relevant pieces of data - when the user cannot access google maps or briefly lose internet access, they should be able to browse content while offline.

App specific storage is cleared when users uninstall the app. SD card appears in the file system as part of external storage.