

How to persist data?

Here, we are going to learn how to persist data after the application is closed.

Let's take an example.

In our Activity code, we have `String name = "Amit"`. The name is being shown in the UI which is currently "Amit" in the `TextView`. After the button click, we changed the name = "Amit Shekhar" and updated the `TextView` with "Amit Shekhar". Now "Amit Shekhar" is being visible in the UI.

Now, when you close and restart the application, you will see, "Amit" in the `TextView` not "Amit Shekhar" because the data was not persisted using the File System. Now, What is File System?

A file system is a process that manages how and where data on a storage disk, typically a hard disk drive (HDD), SDCard in Mobile, is stored, accessed and managed.

Now, we have two types of memory:

- RAM - Temporary
- File System - Permanent

Whatever we keep in RAM will not be persisted, like we created the variable name = "Amit" in our Java/Kotlin class.

Whatever we keep in the File System will be persisted until we delete it specifically.

Let's see how to use it to persist data.

In Android, we have the following ways to deal with the File System:

- Through SharedPreferences - Android has provided us a very easy way to handle this which is through the SharedPreferences API. It stores the value in Key-Value pairs.
- Through Databases
- Store File as image, PDF, and, etc directly in a folder.

Here, we will discuss only about the **SharedPreferences**:

SharedPreferences is like a register/notebook.

We can name the register anything like "notes-app".

It keeps the data as key-value.

We can add any value with a unique key and get the value with that key.

key	value
NAME	Amit Shekhar
USER_ID	67
USER_NAME	amitshekhar

If we would have stored and updated the name in the SharedPreferences, we would have retrieved the “Amit Shekhar” even after the application killed and restarted.

We will see how to use it in the code implementation part of this course.