**Session 19**

**Folder metrics and File reading/writing**

1. **Compare Internal and External storage**

|  |  |  |
| --- | --- | --- |
|  | **Internal Storage** | **External Storage** |
| **Availability** | Always | Varies |
| **Accessibility** | Your app only | World readable |
| **App Uninstallation** | Delete all files | Delete all files (if saved in the directory given by *[getExternalFilesDir()](https://developer.android.com/reference/android/content/Context.html" \l "getExternalFilesDir(java.lang.String))* |
| **Permission** | No permission needed | Require either  *WRITE\_EXTERNAL\_STORAGE*  Or  *READ\_EXTERNAL\_STORAGE* |
| **Best for** | Files that must not be accessed by either the user or other apps. | Files that don’t need access restriction, need to be shared with other apps, or that the user can access. |
| **Access directory** | [*getFilesDir()*](https://developer.android.com/reference/android/content/Context.html#getFilesDir()) | [*getExternalFilesDir()*](https://developer.android.com/reference/android/content/Context.html#getExternalFilesDir(java.lang.String))  *or*  [*getExternalStoragePublicDirectory()*](https://developer.android.com/reference/android/os/Environment.html#getExternalStoragePublicDirectory(java.lang.String)) |

1. **Note on External storage**

External storage is **NOT** always stored on a removable storage device (E.g. SD-card), even if it can be.

In fact, nowadays, in most cases, Android devices split the device permanent storage into Internal and External storage.

“*Many devices now divide the permanent storage space into separate ‘internal’ and ‘external’ partitions.”*

Source : <https://developer.android.com/training/data-storage/files>

Before accessing External storage, you should verify that it is available and that where you want to read/write exists.

You can verify the availability with [*getExternalStorageState()*](https://developer.android.com/reference/android/os/Environment.html#getExternalStorageState())

1. **Read and Write files**

There are two major ways of writing files in Android.

[***FileOutputStream***](https://developer.android.com/reference/java/io/FileOutputStream.html)

Code example:

val filename = "outputstream.txt"  
val fileContents = "Hello world! outputstream"  
  
*applicationContext*.openFileOutput(filename, *MODE\_PRIVATE*).*use* **{  
 it**.write(fileContents.*toByteArray*())  
**}**

[***PrintWriter***](https://developer.android.com/reference/java/io/PrintWriter.html#PrintWriter(java.lang.String))

Code example:

val filename = "printwriter.txt"  
val fileContents = "Hello world! printwriter"  
File(*applicationContext*.*filesDir*, filename).*printWriter*().*use* **{  
 it**.write(fileContents)  
**}**

There is one major way of reading files in Android.

[***FileInputStream***](https://developer.android.com/reference/java/io/FileInputStream.html)

Code example:

var filename = "outputstream.txt"  
var fileContents = "Hello world! outputstream"  
*applicationContext*.openFileInput(filename).*bufferedReader*().*use* **{** lblStorageDir.*text* = **it**.readLine()  
**}**

Access this file:

https://github.com/vincent-picot/Android-apps