Data Persistence
•••
Files (External Storage)

Android External Storage

- Data files are stored publically on the shared external storage using a FileOutputStream object.
- We can read the data files from the device using a FileInputStream object.
- The Data files are not deleted on uninstalling the app.
- External storage needs read/write permission.

Android External Storage

Grant permissions to External Storage

- To read or write files on the external storage, our app must acquire the WRITE_EXTERNAL_STORAGE and READ_EXTERNAL_STORAGE system permissions

Android External Storage

Grant permissions to External Storage

Add the following permissions in the android manifest file like as shown below:

```
 <manifest>
    ....
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    ....
</manifest>
```

Android External Storage

Checking External Storage Availability

- Before using external storage, we must check if the media is available by calling getExternalStorageState().
- The media may be read-only, mounted, missing, or in some other state.

Android External Storage (checking if media mounted read only)

```
private static boolean isExternalStorageReadOnly() {
            // on below line getting external storage and checking if it is
media mounted read only.
        String extStorageState = Environment.getExternalStorageState();
        if (Environment.MEDIA_MOUNTED_READ_ONLY.equals(extStorageState)) {
            return true;
        }
        return false;
    }
```

Android External Storage (checking if media is available or not)

```
private static boolean isExternalStorageAvailable() {
        // on below line checking external storage weather it is available
or not.
        String extStorageState = Environment.getExternalStorageState();
        if (Environment.MEDIA_MOUNTED.equals(extStorageState)) {
            return true;
        }
        return false;
    }
```

Android External Storage

Write a File to External Storage

- We can easily generate and write data to a file in the external storage by using the **android File & FileOutputStream** object **getExternalFilesDir()** method.

Note:

Use MediaStore and ContentValues instead of getExternalStoragePublicDirectory()

Android External Storage

Write a File to External Storage

```
String filename = "user_details";

String folder = "demo"

String name = "admin";

File externalFile = new File(getExternalFilesDir(folder), filename);

 FileOutputStream fstream = new FileOutputStream(externalFile);

fstream.write(name.getBytes());

 fstream.close();
```

Android External Storage

Read a File from External Storage

- We read data from a file in the internal storage by using the **android File & FileInputStream** object **getExternalFilesDir()** method.

Android External Storage

Read a File from External Storage

```
String filename = "user_details";
String folder = "demo"
File myFile = new File(getExternalFilesDir(folder), filename);
          FileInputStream fstream = new FileInputStream(myFile);
           StringBuffer sbuffer = new StringBuffer();
           int i;
           while ((i = fstream.read())!= -1){
               sbuffer.append((char)i);
           }
           fstream.close();
```

Internal Storage Example (Code)

Activity_main.xml

AndroidManifest.xml

activity_details.xml

details.java

MainActivity.java