External storage

Android gives various options for storing apps data which uses a file system similar to the disk-based system on computer platforms

- App-Specific storage
  - Store data files within internal volume directories or external.
  - These data files are meant only for the app's use.
  - It uses internal storage directories to save sensitive information such as a username and password that other app should not access.
- Shared Storage
  - Store data files such as images, audio, video, documents, etc. that the app may need to share with other apps.
- Shared Preferences
  - Store primitive data type such as integer, float, Boolean, string, long in key-value pairs.
- Databases
  - Store structured data such as user-information(name, age, phone, email, address, etc. into private databases.

External storage

- If internal storage doesn't provide enough space to store app-specific files, consider using external storage instead.
- The system provides directories within external storage where an app can organize files that provide value to the user only within your app.

- Developers are advised to use the options available to store data depending upon the space required, reliable data access, and privacy of data.
- The data files saved over external storage devices are publicly accessible on shared external storage using USB mass storage transfer.
- Data files stored over external storage using a FileOutputStream object and can be read using a FileInputStream object.

Verify that storage is available

- Because external storage resides on a physical volume that the user might be able to remove, verify that the volume is accessible before trying to read app-specific data from, or write app-specific data to, external storage.
- You can query the volume's state by calling Environment.getExternalStorageState().
- If the returned state is MEDIA_MOUNTED, then you can read and write app-specific files within external storage.
- If it's MEDIA_MOUNTED_READ_ONLY, you can only read these files.

Select a physical storage location

- Sometimes, a device that allocates a partition of its internal memory as external storage also provides an SD card slot.
- This means that the device has multiple physical volumes that could contain external storage, so you need to select which one to use for your app-specific storage.
- To access the different locations, call ContextCompat.getExternalFilesDirs().

Methods to Store data in External Storage

- getExternalStoragePublicDirectory()
  - This is the present recommended method to keep files public and these files are not deleted even when the app is uninstalled from the system.
  - For eg: Images clicked by the camera are still available even after we uninstall the camera.
- getExternalFilesDir(String type)
  - This method is used to store private data that are specific to the app only.
  - And data are removed as we uninstall the app.
- getExternalStorageDirectory()
  - This method is not recommended.
  - It is now absolute and it is used to access external storage in older versions, API Level less than 7.

Saving files that are private to application(to store app specific data to external storage)

Create an android application for file handling in the external storage and do the following operations when you click the respective buttons: Write the contents to the file from the edit text, Read the contents of the file.(use getExternalFilesDir(String type) method **)**

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.externalstorageApp">
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-
permission>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.ExternalStorage">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">

    <!--    view to get and display file data    -->
    <TextView
        android:id="@+id/dir"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </TextView>

    <EditText
        android:id="@+id/input_text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:hint="Enter text"
        android:lineHeight="25sp"
        android:textColor="@color/black" />

    <!--    button to write data to file    -->
    <Button
        android:id="@+id/btn_write"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Write" />

    <!--    button to read data from file    -->
    <Button
        android:id="@+id/btn_load"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Load" />

</LinearLayout>
```

MainActivity.java

```
package com.example.externalstorageApp;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity {
    private EditText inputText;
    private Button btnWrite, btnLoad;
private TextView dir;
    private String filename = "hello.txt";
    private String filepath = "MyFileStorage";
    private File extFile;
    private String data = "";
```

```
@Override
protected void onCreate(Bundle
savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    inputText = findViewById(R.id.input_text);
    btnWrite = findViewById(R.id.btn_write);
    btnLoad = findViewById(R.id.btn_load);
    dir=findViewById(R.id.dir);
    if (!isExternalStorageAvailable() ||
isExternalStorageReadOnly()) {
        btnWrite.setEnabled(false);
    }
    else {
        extFile = new
File(getExternalFilesDir(filepath), filename);
    }
```

```
 getDir();
    btnWrite.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            data = inputText.getText().toString();
            try {
                FileOutputStream fos = new
FileOutputStream(extFile);
                fos.write(data.getBytes());
               // fos.write("Hello".getBytes());
                inputText.getText().clear();
                Toast.makeText(getApplicationContext(), filename +
" saved to external storage...", Toast.LENGTH_SHORT).show();
                fos.close();
            }
            catch (IOException ex) {
                ex.printStackTrace();
            }
        }
    });
```

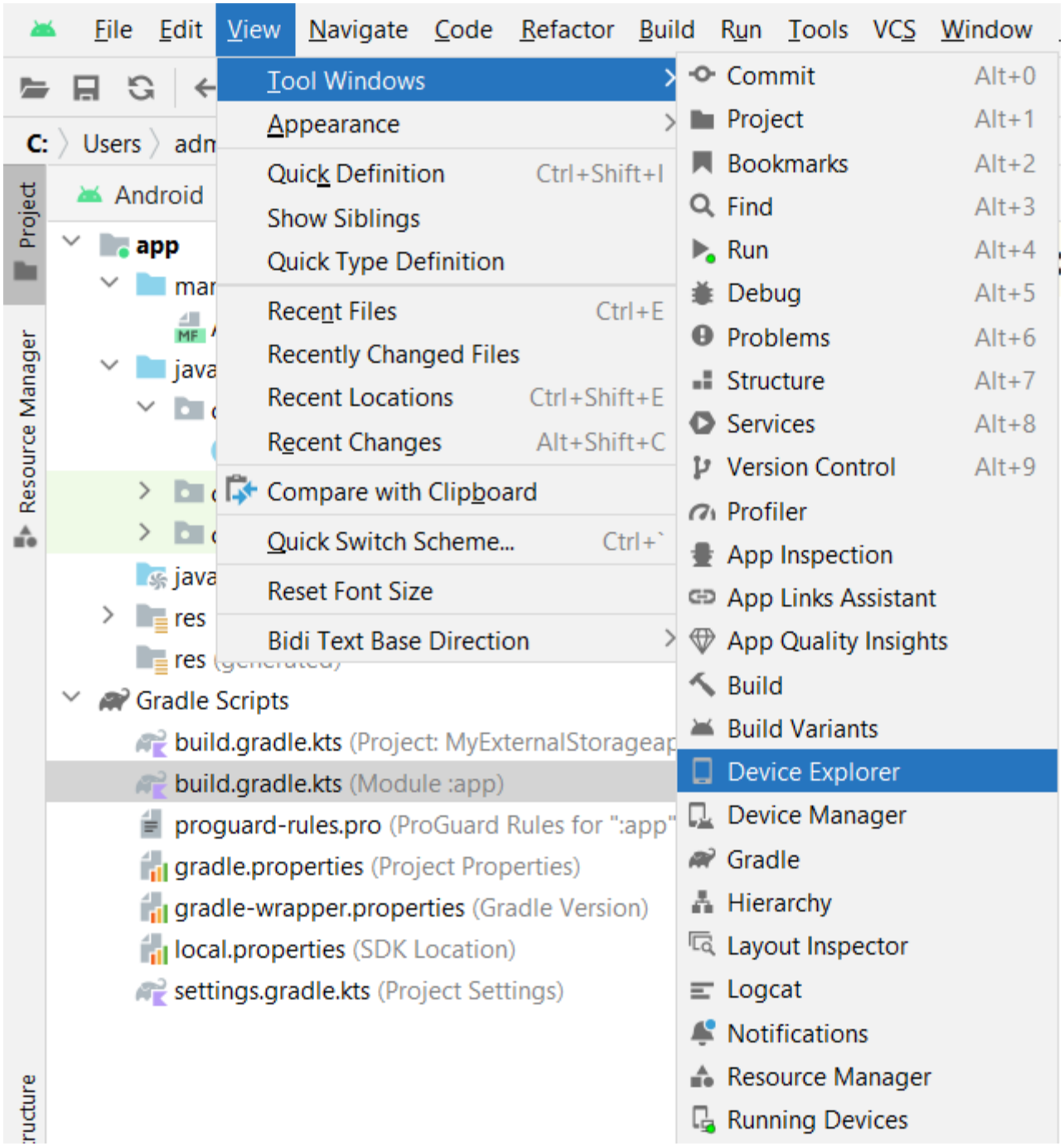
```
btnLoad.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            try {
                FileInputStream fis = new FileInputStream(extFile);
                InputStreamReader isr = new InputStreamReader(fis);
                BufferedReader br = new BufferedReader(isr);
                StringBuilder data = new StringBuilder();
                String line;
                while ((line = br.readLine()) != null) {
                    data.append("\n").append(line);
                }
                inputText.setText(data);
                Toast.makeText(getApplicationContext(), "Data
Retrieved from External File Successfully...",
Toast.LENGTH_SHORT).show();
                fis.close();
            }
            catch (IOException ex) {
                ex.printStackTrace();
            }
        }
    });
}
```

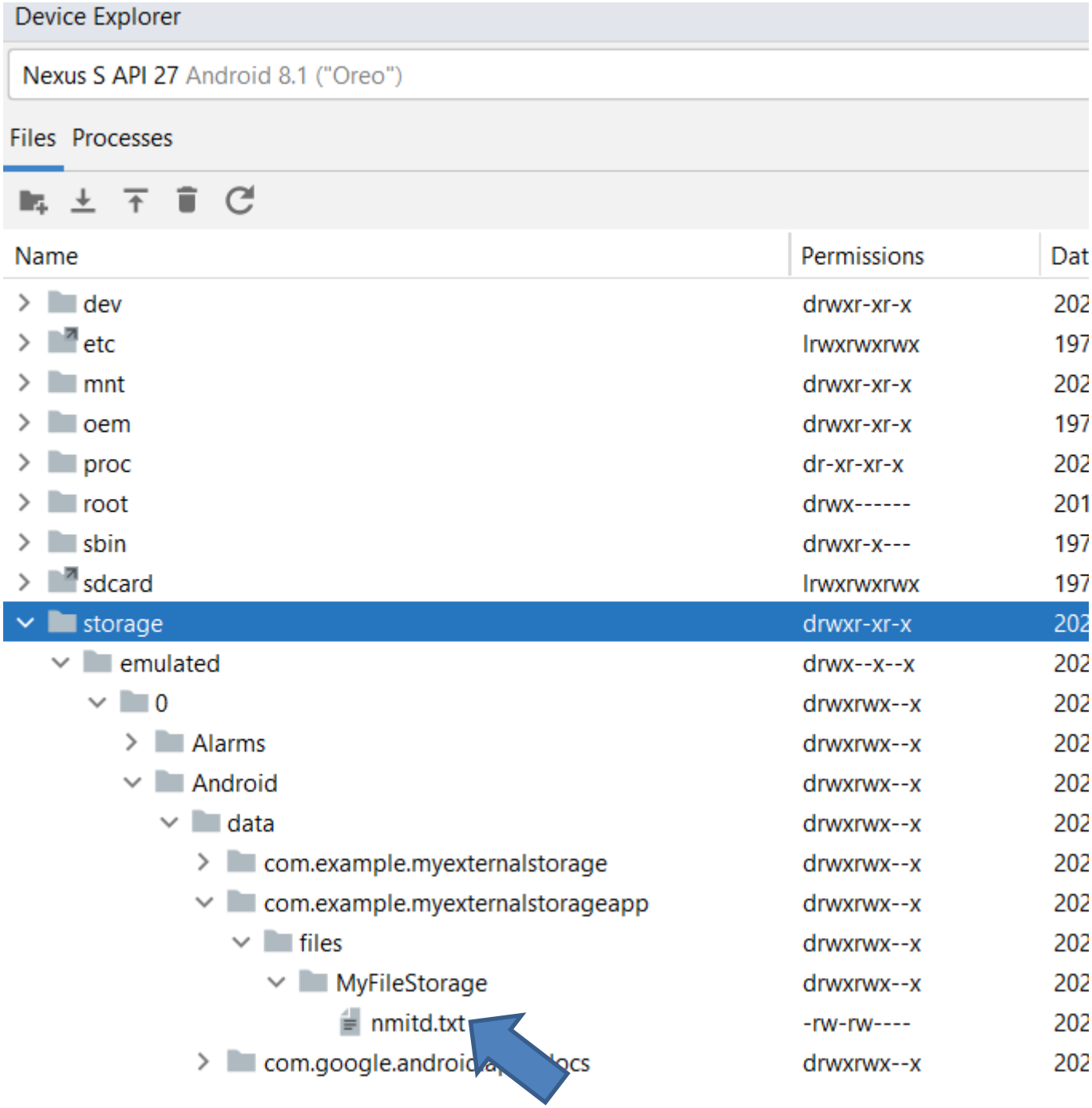
```
private static boolean isExternalStorageAvailable() {
        String extStorageState =
Environment.getExternalStorageState();
        return
Environment.MEDIA_MOUNTED.equals(extStorageState);
    }

    private static boolean isExternalStorageReadOnly() {
        String extStorageState =
Environment.getExternalStorageState();
        return
Environment.MEDIA_MOUNTED_READ_ONLY.equals(extStorageState);
    }
    private void getDir()
    {
        StringBuilder builder=new StringBuilder();
        builder.append("External File Directories:
").append(getExternalFilesDir(filepath).getAbsolutePath()).appe
nd("\n");
        dir.setText(builder.toString());
    }
}
```

In Emulator, to view this file click on Settings>Storage>Internal shared storage>Files>Android>data>com.example.exernal storageApp>files>MyFileStorage

OR
To view this file in file explorer
View>Tool Windows>Device Explorer
In Device Explorer, select storage>emulated>0>Android>data>com.example. externalstorageApp>files>MyFileStorage


File
Edit
View
Navigate
Code
Refactor
Build
Run
Tools
VCS
Window
Tool Windows
Appearance
Quick Definition Ctrl+Shift+I
Show Siblings
Quick Type Definition
Recent Files Ctrl+E
Recently Changed Files
Recent Locations Ctrl+Shift+E
Recent Changes Alt+Shift+C
Compare with Clipboard
Quick Switch Scheme... Ctrl+`
Reset Font Size
Bidi Text Base Direction
Commit Alt+0
Project Alt+1
Bookmarks Alt+2
Find Alt+3
Run Alt+4
Debug Alt+5
Problems Alt+6
Structure Alt+7
Services Alt+8
Version Control Alt+9
Profiler
App Inspection
App Links Assistant
App Quality Insights
Build
Build Variants
Device Explorer
Device Manager
Gradle
Hierarchy
Layout Inspector
Logcat
Notifications
Resource Manager
Running Devices
C: Users adm
Android
Project
Resource Manager
app
Gradle Scripts
build.gradle.kts (Project: MyExternalStorageap
build.gradle.kts (Module :app)
proguard-rules.pro (ProGuard Rules for ":app"
gradle.properties (Project Properties)
gradle-wrapper.properties (Gradle Version)
local.properties (SDK Location)
settings.gradle.kts (Project Settings)


Device Explorer
Nexus S API 27 Android 8.1 ("Oreo")
Files Processes
Name
Permissions
Dat
dev
drwxr-xr-x
202
etc
lrwxrwxrwx
197
mnt
drwxr-xr-x
202
oem
drwxr-xr-x
197
proc
dr-xr-xr-x
202
root
drwx------
201
sbin
drwxr-x---
197
sdcard
lrwxrwxrwx
197
storage
drwxr-xr-x
202
emulated
drwx--x--x
202
0
drwxrwx--x
202
Alarms
drwxrwx--x
202
Android
drwxrwx--x
202
data
drwxrwx--x
202
com.example.myexternalstorage
drwxrwx--x
202
com.example.myexternalstorageapp
drwxrwx--x
202
files
drwxrwx--x
202
MyFileStorage
drwxrwx--x
202
nmitd.txt
-rw-rw----
202
drwxrwx--x
202

Saving files that can be shared with other applications

- **Create an android application for file handling in the external storage and do the following operations when you click the respective buttons: Write the contents to the file from the edit text, Read the contents of the file.(use** getExternalStoragePublicDirectory() method **OR** getExternalStorageDirectory() **)**

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.publicexternalstorage">
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" ></uses-
permission>
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"></uses-
permission>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.PublicExternalStorage">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

<EditText
    android:id="@+id/data"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    android:hint="Enter text"
    android:textColor="@color/black"></EditText>
   <LinearLayout
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:orientation="horizontal">
    <Button
        android:id="@+id/write"
        android:layout_marginLeft="16dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Save File In External"
        tools:layout_editor_absoluteX="132dp"
        tools:layout_editor_absoluteY="135dp" />
    <Button
        android:id="@+id/read"
        android:layout_marginLeft="30dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Read File"
        tools:layout_editor_absoluteX="132dp"
        tools:layout_editor_absoluteY="135dp" />
   </LinearLayout>

</LinearLayout>
```

MainActivity.java

```
package com.example.publicexternalstorage;

import android.Manifest;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity {
   Button write,read;
EditText text;
    private int EXTERNAL_STORAGE_PERMISSION_CODE = 23;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
write=findViewById(R.id.write);
read=findViewById(R.id.read);
text=findViewById(R.id.data);
```

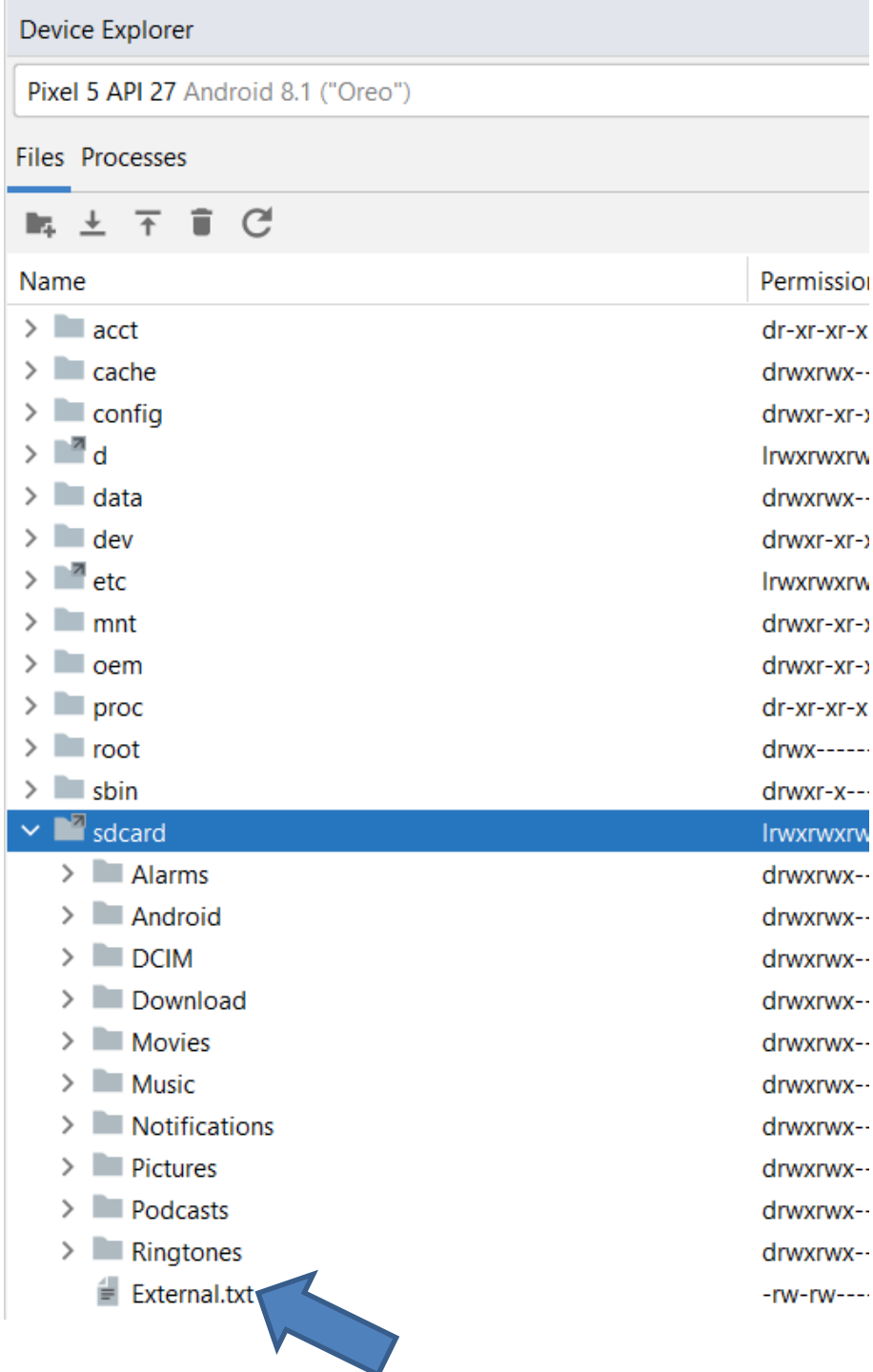
```
File folder=Environment.getExternalStorageDirectory();
        File file = new File(folder, "External.txt");
write.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Requesting Permission to access External Storage
        ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE,
Manifest.permission.WRITE_EXTERNAL_STORAGE},
                EXTERNAL_STORAGE_PERMISSION_CODE );
        String editTextData = text.getText().toString();  // Content
        writeTextData(file, editTextData);
    }
});
```

```
 read.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                try {
                    FileInputStream fis = new FileInputStream(file);
                    InputStreamReader isr = new InputStreamReader(fis);
                    BufferedReader br = new BufferedReader(isr);
                    StringBuilder data = new StringBuilder();
                    String line;
                    while ((line = br.readLine()) != null) {
                        data.append("\n").append(line);
                    }
                    text.setText(data);
                    Toast.makeText(getApplicationContext(), "Data Retrieved from
External File Successfully...", Toast.LENGTH_SHORT).show();
                    fis.close();
                }
                catch (IOException ex) {
                    ex.printStackTrace();
                }
            }
        });
    }
```

```
private void writeTextData(File file, String data) {
        FileOutputStream fileOutputStream = null;
        try {
            fileOutputStream = new FileOutputStream(file);
            fileOutputStream.write(data.getBytes());
            text.getText().clear();
            Toast.makeText(this, "Written data successfully to" +
file.getAbsolutePath(), Toast.LENGTH_SHORT).show();

        }
        catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (fileOutputStream != null) {
                try {
                    fileOutputStream.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

To view this file
In Device Explorer, select sdcard


Device Explorer
Pixel 5 API 27 Android 8.1 ("Oreo")
Files Processes
Name
Permissio
acct
dr-xr-xr-x
cache
drwxrwx--
config
drwxr-xr-x
d
lrwxrwxrw
data
drwxrwx--
dev
drwxr-xr-x
etc
lrwxrwxrw
mnt
drwxr-xr-x
oem
drwxr-xr-x
proc
dr-xr-xr-x
root
drwx------
sbin
drwxr-x---
sdcard
lrwxrwxrw
Alarms
drwxrwx--
Android
drwxrwx--
DCIM
drwxrwx--
Download
drwxrwx--
Movies
drwxrwx--
Music
drwxrwx--
Notifications
drwxrwx--
Pictures
drwxrwx--
Podcasts
drwxrwx--
Ringtones
drwxrwx--
External.txt
-rw-rw----