

# Module 3

Data base Connectivity

# Contents

- Persistence data using the file system (external, internal, SD card)
- working with shared preferences
- Working with content providers
- CRUD operation using SQLite database connection.
- Self-Learning Topics: Interface of Database

# Data storage in Android

## Shared Preferences

Store primitive data in key-value pairs.

## Internal Storage

Store private data on the device memory.

## External Storage

Store public data on the shared external storage.(Eg SD card )

## SQLite Databases

Store structured data in a private database.(Particular mobile)

## Network Connection

Store data on the web with your own network server.

## Using the file system

- Android's file system is based on Linux and supports mode-based permissions.
- You can access this file system in several ways.
- You can create and read files from within applications, you can access raw resource files, and you can work with specially compiled custom XML files.

# Using the file system

## Storage areas

The **internal storage** is also referred to as permanent storage.

Files saved in internal storage are accessible only to your app by default.

**External storage** can be removed at any time and files saved here can be accessible to everyone.

Applications can allow themselves to be installed on the external storage by specifying the ***android:installLocation*** attribute.

The default is internal storage.

# Using the File System

To grant everyone permission to your files, you need to declare the following in your manifest file.

```
<manifest ...>  
    <uses-permission android_name=  
        "android.permission.READ_EXTERNAL_STORAGE" />  
    ...  
</manifest>
```

If we declare **READ\_EXTERNAL\_STORAGE**, global read permissions are granted implicitly.

To save files on internal storage, we need to acquire a directory as a file by calling either the **getFilesDir()** API, or by calling **getCacheDir()**, which represents your app's temp cache files.

A new file can be created by specifying the file and the name of the directory where the file needs to be created by using the File constructor as follows.

```
//filename="example.txt"  
File file = new File(context.getFilesDir(), "example.txt");
```

We also can create a file by calling **openFileOutput()** to get a FileOutputStream.

To create temporary files meant for caching purposes, use the **createTempFile()** API.

	Type of content	Access method	Permissions needed	Can other apps access?	Files removed on app uninstall?
App-specific files	Files meant for your app's use only	From internal storage, <code>getFilesDir()</code> or <code>getCacheDir()</code>	Never needed for internal storage	No	Yes
		From external storage, <code>getExternalFilesDir()</code> or <code>getExternalCacheDir()</code>	Not needed for external storage when your app is used on devices that run Android 4.4 (API level 19) or higher		



## Media

Shareable  
media files  
(images,  
audio files,  
videos)

MediaStore API

READ\_  
EXTERNAL\_  
STORAGE  
when  
accessing  
other apps'  
files on  
Android 11  
(API level 30)  
or higher

Yes, though  
the other  
app needs  
the READ\_  
EXTERNAL\_  
STORAGE  
permission

No

READ\_  
EXTERNAL\_  
STORAGE or  
WRITE\_  
EXTERNAL\_  
STORAGE  
when  
accessing  
other apps'  
files on  
Android 10  
(API level 29)

Permissions  
are required  
for **all** files on  
Android 9 (API  
level 28) or  
lower

Documents and other files	Other types of shareable content, including downloaded files	Storage Access Framework	None	Yes, through the system file picker	No
App preferences	Key-value pairs	Jetpack Preferences library	None	No	Yes
Database	Structured data	Room persistence library	None	No	Yes

# INTERNAL STORAGE

- Internal storage is a **closed non-shareable space** which is always available on your device.
- It holds dedicated directories for each application which can be used by the applications **to store sensitive and relevant data**.
- Thus, these dedicated directories are **only accessible to their owner applications and are completely inaccessible to users or foreign applications**.
- Also, because each of these directories are associated with an application, **if the user uninstalls an application, it's dedicated directory along with all it's contents are removed from the device**.

# INTERNAL STORAGE

- [getFilesDir\(\)](#) and [getCacheDir\(\)](#)
- These methods return the absolute paths to your application's dedicated directories which you can then use to open an OutputStream or InputStream.
  - For writing/reading to/from Internal storage, no permissions are required.
  - **However, keep in mind that the files you save to Internal storage though secure and inaccessible to other applications, are removed as soon as the user uninstalls your application.**

## getFilesDir() vs getCacheDir()

**getFilesDir()** returns the absolute path to the directory on the **filesystem** where files created with **openFileOutput(String, int)** are stored. It is used for **permanent storage directory**.

On the other hand, **getCacheDir()** returns the absolute path to the **application specific cache directory** on the file system. The system could **automatically delete files in this directory** in some cases such as **when memory is almost full**. Thus, it is **temporary storage directory**.

Two things are important to note when working with internal storage:

- A user cannot access these files through the file manager(u need permission)
- Files in this folder will be deleted when an app is uninstalled

**Create an android application for file handling in the internal 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.**

# 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="10dp"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/write"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"></TextView>
    <!--      view to get and display file data      -->
    <EditText
        android:id="@+id/edit_text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:hint="Enter text"
        android:lineHeight="20sp"
        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.internalstorage;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
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.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;

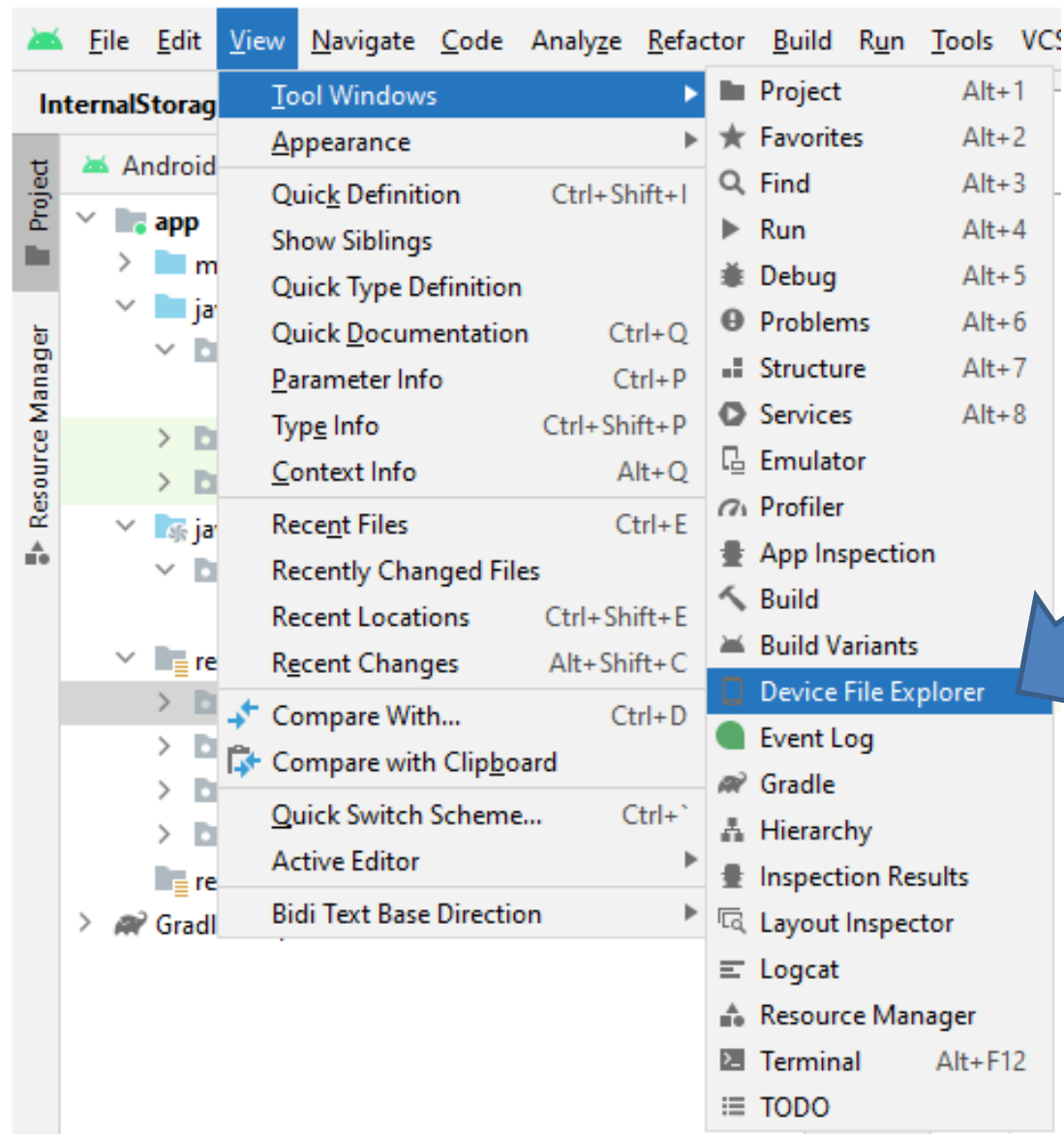
public class MainActivity extends AppCompatActivity {
    private EditText editText;
    private Button btnWrite, btnLoad;
    TextView write;
    private String FILENAME = "test.txt";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editText = findViewById(R.id.edit_text);
        btnWrite = findViewById(R.id.btn_write);
        btnLoad = findViewById(R.id.btn_load);
        write=findViewById(R.id.write);
    }
}
```

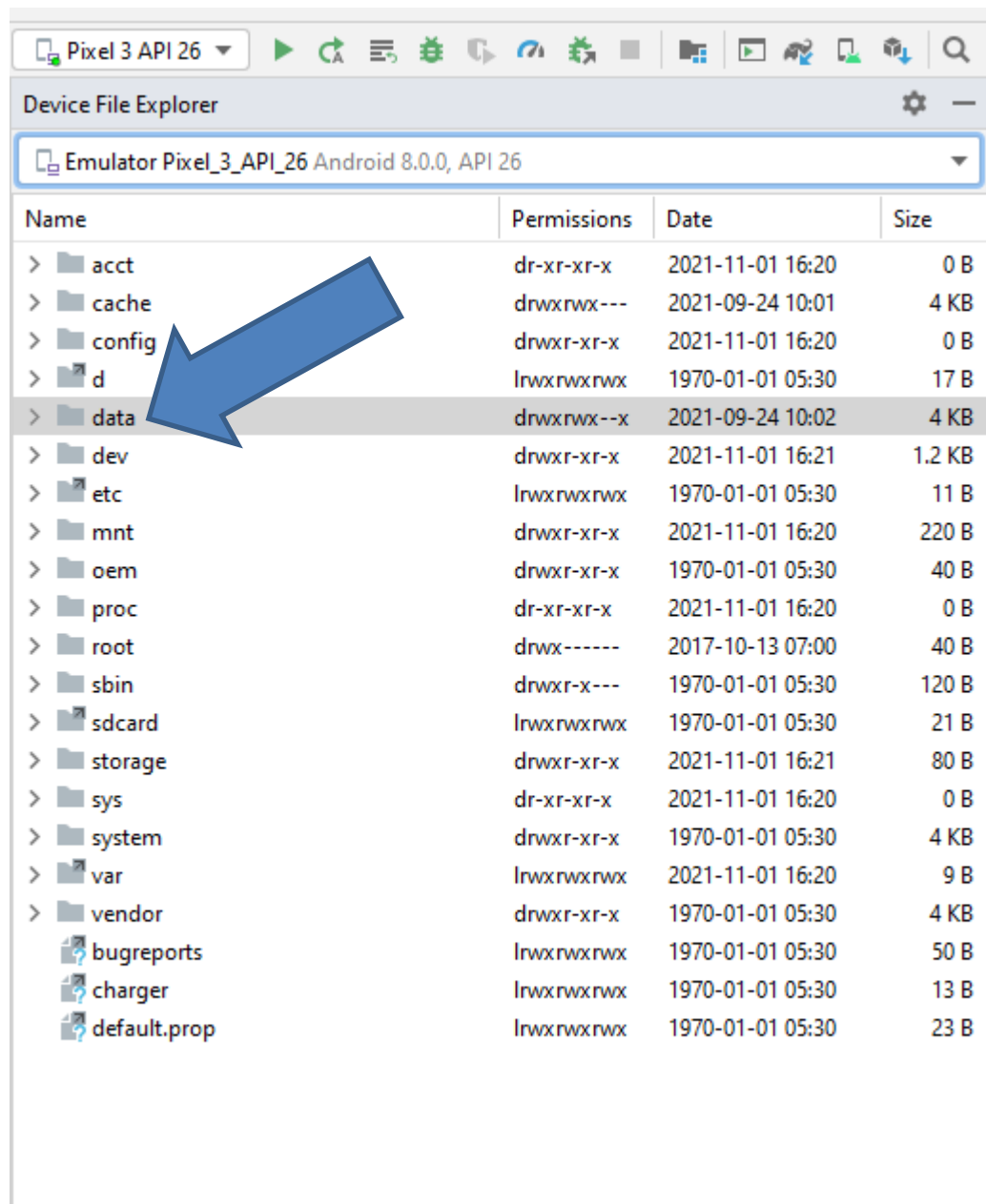
```
btnWrite.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        String data = editText.getText().toString();  
        try {  
            FileOutputStream fos =  
openFileOutput(FILENAME, MODE_PRIVATE);  
            fos.write(data.getBytes());  
            Toast.makeText(getApplicationContext(),  
"Data written successfully...", Toast.LENGTH_SHORT).show();  
            editText.getText().clear();  
            fos.close();  
        }  
        catch (IOException e) {  
            Toast.makeText(getApplicationContext(),  
e.getMessage(), Toast.LENGTH_SHORT).show();  
        }  
    }  
});
```

```
btnLoad.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        try {
            FileInputStream fis = openFileInput(FILENAME);
            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);
            }
            data.deleteCharAt(0);
            write.setText(data);
            Toast.makeText(getApplicationContext(), "Data
loaded successfully...", Toast.LENGTH_SHORT).show();
            fis.close();
        }
        catch (IOException e) {
            Toast.makeText(getApplicationContext(),
e.getMessage(), Toast.LENGTH_SHORT).show();
        }
    }
});
```

```
getDir();
}
private void getDir()
{
    StringBuilder builder=new StringBuilder();
    builder.append("Cache Directories:
").append(getCacheDir().getAbsolutePath()).append("\n")
        .append("File
Directories:").append(getFilesDir().getAbsolutePath());
    write.setText(builder.toString());
}
}
```

## View file in Device File Explorer





Device File Explorer <span>⚙️</span> <span>—</span>			
Emulator Pixel_3_API_26 Android 8.0.0, API 26			
Name	Permissions	Date	Size
>  acct	dr-xr-xr-x	2021-11-01 16:20	0 B
>  cache	drwxrwx---	2021-09-24 10:01	4 KB
>  config	drwxr-xr-x	2021-11-01 16:20	0 B
>  d	lrwxrwxrwx	1970-01-01 05:30	17 B
▼  data	drwxrwx--x	2021-09-24 10:02	4 KB
>  app	drwxrwx--x	2021-09-24 10:02	4 KB
▼  data	drwxrwx--x	2021-09-24 10:02	4 KB
>  android	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.backupconfirm	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.bips	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.bookmarkprovider	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.calculator2	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.calllogbackup	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.camera2	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.captiveportallogin	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.carrierconfig	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.cellbroadcastreceiv	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.certinstaller	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.chrome	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.companiondevicer	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.contacts	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.cts.ctsshim	drwxrwx--x	2021-09-24 10:02	4 KB
>  com.android.cts.priv.ctsshim	drwxrwx--x	2021-09-24 10:02	4 KB

Emulator Pixel\_3\_API\_26 Android 8.0.0, API 26

Name	Permissions	Date	Size
> com.example.capturevideo	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.contextmenu	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.drawingrectangle	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.fragment	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.frameanimation	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.getlocationaddress	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.googlemap	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.graphics	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.graphics1	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.graphicsoncanvas	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.implicitintent	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.intent	drwxrwx--x	2021-09-24 10:02	4 KB
▼ com.example.internalstorage	drwxrwx--x	2021-09-24 10:02	4 KB
cache	drwxrws--x	2021-11-01 16:22	4 KB
code_cache	drwxrws--x	2021-11-01 16:22	4 KB
▼ files	drwxrwx--x	2021-11-01 16:23	4 KB
test.txt	-rw-rw----	2021-11-02 11:07	20 B
> com.example.location	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.map	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.menudemo	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.myanimation	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.notificationapp1	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.popupmenu	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.progressbar	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.radiobutton	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.ratingbar	drwxrwx--x	2021-09-24 10:02	4 KB
> com.example.seekbar	drwxrwx--x	2021-09-24 10:02	4 KB