Android Internal Storage Example

We are able to save or read data from the device internal memory. FileInputStream and FileOutputStream classes are used to read and write data into the file.

Here, we are going to read and write data to the internal storage of the device.

Example of reading and writing data to the android internal storage

activity_main.xml

Drag the 2 edittexts, 2 textviews and 2 buttons from the pallete, now the activity_main.xml file will like this:

File: activity_main.xml

```
<RelativeLayout 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"
    tools:context=".MainActivity" >

    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginRight="20dp"
        android:layout_marginTop="24dp"
        android:ems="10" >

        <requestFocus />
    </EditText>

    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/editText1"
        android:layout_below="@+id/editText1"
        android:layout_marginTop="24dp"
        android:ems="10" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/editText1"
        android:layout_alignBottom="@+id/editText1"
        android:layout_alignParentLeft="true"
        android:text="File Name:" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/editText2"
        android:layout_alignBottom="@+id/editText2"
        android:layout_alignParentLeft="true"
        android:text="Data:" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/editText2"
        android:layout_below="@+id/editText2"
        android:layout_marginLeft="70dp"
        android:layout_marginTop="16dp"
        android:text="save" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/button1"
        android:layout_alignBottom="@+id/button1"
        android:layout_toRightOf="@+id/button1"
        android:text="read" />

</RelativeLayout>
```

Activity class

Let's write the code to write and read data from the internal storage.

File: MainActivity.java

```
package example.javatpoint.com.internalstorage;

import android.content.Context;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity {
    EditText editTextFileName,editTextData;
    Button saveButton,readButton;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editTextFileName=findViewById(R.id.editText1);
        editTextData=findViewById(R.id.editText2);
        saveButton=findViewById(R.id.button1);
        readButton=findViewById(R.id.button2);

        //Performing Action on Read Button
        saveButton.setOnClickListener(new View.OnClickListener(){

            @Override
            public void onClick(View arg0) {
                String filename=editTextFileName.getText().toString();
                String data=editTextData.getText().toString();

                FileOutputStream fos;
                try {
                    fos = openFileOutput(filename, Context.MODE_PRIVATE);
                    //default mode is PRIVATE, can be APPEND etc.
                    fos.write(data.getBytes());
                    fos.close();

                    Toast.makeText(getApplicationContext(),filename + " saved",
                            Toast.LENGTH_LONG).show();


                } catch (FileNotFoundException e) {e.printStackTrace();}
                catch (IOException e) {e.printStackTrace();}

            }

        });

        //Performing Action on Read Button
        readButton.setOnClickListener(new View.OnClickListener(){

            @Override
            public void onClick(View arg0) {
                String filename=editTextFileName.getText().toString();
                StringBuffer stringBuffer = new StringBuffer();
                try {
                    //Attaching BufferedReader to the FileInputStream by the help of InputStreamReader
                    BufferedReader inputReader = new BufferedReader(new InputStreamReader(
                            openFileInput(filename)));
                    String inputString;
                    //Reading data line by line and storing it into the stringbuffer
                    while ((inputString = inputReader.readLine()) != null) {
                        stringBuffer.append(inputString + "\n");
                    }

                } catch (IOException e) {
                    e.printStackTrace();
                }
                //Displaying data on the toast
                Toast.makeText(getApplicationContext(),stringBuffer.toString(),Toast.LENGTH_LONG).show();

            }
        });
    }
}
```

Output: