**Android External Storage пример работы с внешней памятью**

В прошлой статье мы говорили о внутренней памяти Android устройства и рассмотрели пример работы с ней. Сегодня мы продолжим знакомиться с хранением данных в Android на примере External Storage (внешней памятью Android устройства).

**Краткий обзор Android External Storage**

Внешнее запоминающее устройство, например SD-карта, может хранить данные приложений.  
Всего существует два типа внешних накопителей:

1. **Primary External Storage**. Встроенное общее хранилище будет доступно пользователю, если он подключит USB кабель и вмонтирует его в виде диска на компьютере.
2. **Secondary External Storage**. Съемное хранилище данных, например SD- карта

Все приложения могут считывать и записывать данные, размещенные на внешнем запоминающем устройстве и пользователь может удалить эти данные. Чтобы работать с внешним запоминающим устройством, всего нужно проверять доступность SD-карты.

**Пример работы с Android External Storage**

Сегодня мы напишем простое приложение, которое будет работать с внешней памятью Android устройства. Структура проекта ничем не отличается простого приложения [Hello World](http://prologistic.com.ua/pishem-hello-world-na-android-rabotaem-v-android-studio.html" \t "_blank), которое мы писали в первых уроках по Android. Создайте простой проект в [Android Studio](http://prologistic.com.ua/struktura-proekta-v-android-studio.html" \t "_blank) и приступим к программированию.

Прежде всего мы должны получить разрешение на чтение и запись данных, расположенных на внешней памяти. Для этого в AndroidManifest.xml добавьте следующие разрешения:

|  |  |
| --- | --- |
| 1  2 | <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>  <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/> |

Ниже представлен макет файла activity\_main.xml:

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:orientation="vertical">        <EditText          android:id="@+id/editTxt"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:hint="@string/edit\_txt"          android:inputType="text" />        <LinearLayout          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:orientation="horizontal">            <Button              android:id="@+id/saveBtn"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:text="@string/save\_txt" />            <Button              android:id="@+id/readBtn"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:text="@string/read\_txt" />      </LinearLayout>    </LinearLayout> |

Как видно из макета, мы вынесли текст на кнопках и в виджете EditText в файл strings.xml(всегда так делайте):

|  |
| --- |
| <resources>      <string name="app\_name">ExternalStorageApp</string>      <string name="save\_txt">Сохранить данные</string>      <string name="read\_txt">Считать данные</string>      <string name="edit\_txt">Введите текст...</string>  </resources> |

Листинг класса MainActivity представлен ниже:

|  |
| --- |
| package ua.com.prologistic.externalstorageapp;    import android.os.Bundle;  import android.os.Environment;  import android.support.v7.app.AppCompatActivity;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;  import android.widget.Toast;    import java.io.BufferedReader;  import java.io.DataInputStream;  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 implements View.OnClickListener {      private EditText editText;      private Button saveBtn, readBtn;        private String fileName = "mFile.txt";      private String filePath = "MyFileStorage";        private File mFile;      private String mData = "";    *@Override*      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);            // инициализирум виджеты          editText = (EditText) findViewById(R.id.editTxt);          saveBtn = (Button) findViewById(R.id.saveBtn);          readBtn = (Button) findViewById(R.id.readBtn);            saveBtn.setOnClickListener(this);          readBtn.setOnClickListener(this);            // проверям внешнюю память на доступность          // и возможность записи          if (!isAvailable() || isReadOnly()) {              // если доступа нет, то делаем кнопки для сохранения              // и считывания с внешней памяти неактивными              saveBtn.setEnabled(false);              readBtn.setEnabled(false);          } else {              // если доступ есть, то создаем файл в ExternalStorage              mFile = new File(getExternalFilesDir(filePath), fileName);          }        }        // является ли внешнее хранилище только для чтения      private static boolean isReadOnly() {          String storageState = Environment.getExternalStorageState();          return Environment.MEDIA\_MOUNTED\_READ\_ONLY.equals(storageState);      }        // проверяем есть ли доступ к внешнему хранилищу      private static boolean isAvailable() {          String storageState = Environment.getExternalStorageState();          return Environment.MEDIA\_MOUNTED.equals(storageState);      }        // простой метод для создания всплывающих окон      public void showToast(String message) {          Toast.makeText(this, message, Toast.LENGTH\_SHORT).show();      }    *@Override*      public void onClick(View v) {          int id = v.getId();          switch (id) {              case R.id.readBtn:                  try {                      FileInputStream fis = new FileInputStream(mFile);                      DataInputStream in = new DataInputStream(fis);           BufferedReader br = new BufferedReader(new InputStreamReader(in));                      String strLine;                      // считываем данные с файла в mData                      while ((strLine = br.readLine()) != null) {                          mData += strLine;                      }                      in.close();                  } catch (IOException e) {                      e.printStackTrace();                  }                  // очищаем поле для ввода от старой информации                  editText.setText("");                  // вставляем считанное из файла                  editText.setText(mData);                  showToast("Данные получены из внешней памяти!");                  break;                case R.id.saveBtn:                  // обнуляєм данные поля mData                  mData = "";                  try {                      FileOutputStream fos = new FileOutputStream(mFile);                      fos.write(editText.getText().toString().getBytes());                      fos.close();                  } catch (IOException e) {                      e.printStackTrace();                  }                  showToast("Данные сохранены на внешней памяти!");                  break;          }      }  } |

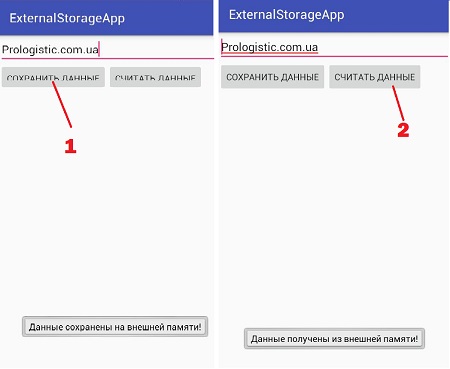
Обратите внимание:

Метод Environment.getExternalStorageState() возвращает путь к внутреннему хранилищу по адресу /mnt/sdcard.

Метод getExternalFilesDir() возвращает путь к файлам папкиandroid/data/data/application\_package/ на SD-карте. Он используется для хранения необходимых вашему приложения файлов, например, скачанных из интернета изображений или кэш приложения). Следует отметить, что с удалением приложения все файлы, хранящиеся в этой папке, также будут удалены.

Ниже представлен результат работы с External Storage на реальном устройстве Android.

Сначала вводим в поле какие-то данные, потом нажимаем «Сохранить данные», далее очищаем поле и нажимаем «Считать данные»:

[](http://prologistic.com.ua/wp-content/uploads/2016/03/res.jpg)