Сохранение и чтение файлов во внутреннем хранилище

Вы можете сохранять и читать файлы во внутреннем хранилище в виде приватных файлов, которые могут видеть только ваше приложение, кроме других приложений с привилегиями root.  
Абсолютным путем этой папки является data \ data \ [ваш пакет приложения] \ файлы.   
В классе [Context](https://developer.android.com/reference/android/content/Context.html) содержатся некоторые методы, которые помогут вам выполнить такие операции, как:

* [openFileInput(String name)](https://developer.android.com/reference/android/content/Context.html#openFileInput(java.lang.String)) Откройте частный файл, связанный с этим пакетом приложений Context для чтения
* [openFileOutput(String name, int mode)](https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String,%20int)) Откройте частный файл, связанный с этим пакетом приложений Context для написания
* [getFilesDir()](https://developer.android.com/reference/android/content/Context.html#getFilesDir()) Получает абсолютный путь к каталогу файловой системы, в котором сохраняются ваши внутренние файлы
* [getDir()](https://developer.android.com/reference/android/content/Context.html#getDir(java.lang.String,%20int)) Создает (или открывает существующий) каталог в вашем внутреннем хранилище
* [deleteFile()](https://developer.android.com/reference/android/content/Context.html#deleteFile(java.lang.String)) Удаляет файл, сохраненный во внутреннем хранилище
* [fileList()](https://developer.android.com/reference/android/content/Context.html#fileList()) Возвращает массив файлов, сохраненных в настоящее время вашим приложением

и вам не нужно иметь специальные разрешения для использования этих методов.

Например, следующий код, взятый из « [Использование внутреннего хранилища»,](https://developer.android.com/guide/topics/data/data-storage.html#filesInternal) создает файл, содержащий строку «hello world!».

|  |
| --- |
| try {  String FILENAME = "hello\_file";  String string = "hello world!";    FileOutputStream fos = openFileOutput(FILENAME, Context.MODE\_PRIVATE);    fos.write(string.getBytes());  fos.close();    } catch (IOException e) {  Log.e("ERROR", e.toString());  } |

где в строке 5 вместо MODE\_PRIVATE вы можете использовать MODE\_APPEND, другие константы, такие как MODE\_WORLD\_READABLE и MODE\_WORLD\_WRITEABLE, устарели.

Следующий код читает тот же файл и показывает сообщение с текстом «hello world!».

|  |
| --- |
| try {  String FILENAME = "hello\_file";  byte[] bytes = new byte[1024];    FileInputStream fis = openFileInput(FILENAME);    fis.read(bytes);  fis.close();    String string = new String(bytes);  Toast.makeText(getApplicationContext(), string, Toast.LENGTH\_SHORT).show();    } catch (IOException e) {  Log.e("ERROR", e.toString());  } |

Но эти примеры имеют ограничение: их нельзя использовать в подпапках, другими словами, вы не можете создать древовидную структуру каталогов и файлов.   
Если вам нужно создавать файлы в подкаталогах, вы должны использовать стандартную технику java, например, следующий код создает файл внутри папки «sub»,

|  |
| --- |
| try {    String FILENAME = "hello\_file";  String FOLDERNAME = "sub";  String string = "hello world!";    Context context = getApplicationContext();  String folder = context.getFilesDir().getAbsolutePath() + File.separator + FOLDERNAME;    File subFolder = new File(folder);    if (!subFolder.exists()) {  subFolder.mkdirs();  }    FileOutputStream outputStream = new FileOutputStream(new File(subFolder, FILENAME));    outputStream.write(string.getBytes());  outputStream.close();    } catch (FileNotFoundException e) {  Log.e("ERROR", e.toString());  } catch (IOException e) {  Log.e("ERROR", e.toString());  } |

и следующий код читает один и тот же файл

Java

|  |
| --- |
| try {    String FILENAME = "hello\_file";  String FOLDERNAME = "sub";  byte[] bytes = new byte[1024];    Context context = getApplicationContext();  String folder = context.getFilesDir().getAbsolutePath() + File.separator + FOLDERNAME;    File subFolder = new File(folder);    FileInputStream outputStream = new FileInputStream(new File(subFolder, FILENAME));    outputStream.read(bytes);  outputStream.close();    String string = new String(bytes);  Toast.makeText(getApplicationContext(), string, Toast.LENGTH\_SHORT).show();    } catch (FileNotFoundException e) {  Log.e("ERROR", e.toString());  } catch (IOException e) {  Log.e(TAG, e.toString());  } |