

SAMSUNG



Kotlin

Базовый Курс

5.1. Хранение файлов приложения



Обзор способов хранения данных и файлов

SAMSUNG



На сегодняшний день в системе есть несколько вариантов хранения данных приложения:

- 1. Специфичные для приложения файлы***
- 2. Общее (разделяемое) хранилище***
- 3. Настройки***
- 4. Базы данных***

Android Application
Data Storage





Разрешения и доступ к внешнему хранилищу

SAMSUNG



```
<manifest ...  
    android:installLocation="preferExternal"> ...  
</manifest>
```

Android определяет следующие разрешения, связанные с хранением файлов во внешней памяти:

- READ_EXTERNAL_STORAGE
- WRITE_EXTERNAL_STORAGE
- MANAGE_EXTERNAL_STORAGE

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>  
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```





Запись чтение в файл во внешнем хранилище

SAMSUNG



```
var myExternalFile:File = File(getExternalFilesDir(filepath),fileName)
try {
    val fileOutputStream = FileOutputStream(myExternalFile)
    fileOutputStream.write(fileContent.toByteArray())
    fileOutputStream.close()
} catch (e: IOException) { e.printStackTrace() }
```

```
var myExternalFile:File = File(getExternalFilesDir(filepath), fileName)
val filename = fileName.text.toString()
myExternalFile = File(getExternalFilesDir(filepath),filename)
var fileInputStream =FileInputStream(myExternalFile)
var inputStreamReader: InputStreamReader = InputStreamReader(fileInputStream)
val bufferedReader: BufferedReader = BufferedReader(inputStreamReader)
val stringBuilder: StringBuilder = StringBuilder()
var text: String? = null
while ({ text = bufferedReader.readLine(); text }()) != null) {
    stringBuilder.append(text)
}
fileInputStream.close()
```





Доступ к файлам приложения

SAMSUNG



Система Android предоставляет следующие возможности для хранения файлов приложения:

- **Внутренние каталоги**
- **Внешние каталоги**

В Android есть несколько методов, с помощью которых можно получить доступ к файлам и сохранить их.

1. File API

```
val file = File(context.filesDir, filename)
```

2. В качестве альтернативы использованию файлового API можно задействовать метод `openFileOutput()`.

```
val filename = "myfile"  
val fileContents = "My IT Academy!"  
context.openFileOutput(filename, Context.MODE_PRIVATE).use {  
    it.write(fileContents.toByteArray())  
}
```





Доступ к файлам приложения

SAMSUNG



Для чтения файла в потоковом режиме необходимо использовать метод `openFileInput()`. Сделать это можно следующим образом:

```
context.openFileInput(filename).bufferedReader().useLines { lines ->
    lines.fold("") {
        my, message -> "$my\n$message "
    }
}
```

Еще одной опцией является получение массива из имен каталогов `filesDir`. Для этого целесообразно вызвать метод `fileList()`, например следующим образом:

```
var files: Array<String> = context.fileList()
```

Также существует возможность создания вложенных каталогов или открытия внутреннего через вызов метода `getDir()`, например следующим образом:

```
context.getDir(dir, Context.MODE_PRIVATE)
```





Доступ к файлам приложения

SAMSUNG



Для определения доступного места в каталоге кэша можно вызвать метод `getCacheQuotaBytes()`, а для создания кэшированного файла можно использовать следующий код:

```
File.createTempFile(filename, null, context.cacheDir)
```

Для удаления ненужных файлов можно использовать следующий код:

```
cacheFile.delete()
```

Метод, представленный ниже проверяет размер памяти внешнего хранилища для чтения и записи:

```
fun isExternalStorageWritable(): Boolean {  
    return Environment.getExternalStorageState() == Environment.MEDIA_MOUNTED  
}
```





Доступ к файлам приложения

SAMSUNG



Для доступа к файлам внешнего хранилища, требуется вызов метода `getExternalFilesDir()`:



```
val appSpecificExternalDir = File(context.getExternalFilesDir(null), filename)
```

Для добавления файла в кэш внешнего хранилища, можно написать следующий код:

```
val externalCacheFile = File(context.externalCacheDir, filename)
```

Для удаления файла из каталога внешнего кэша, задействуйте метод `delete()` для объекта `File`:

```
externalCacheFile.delete()
```



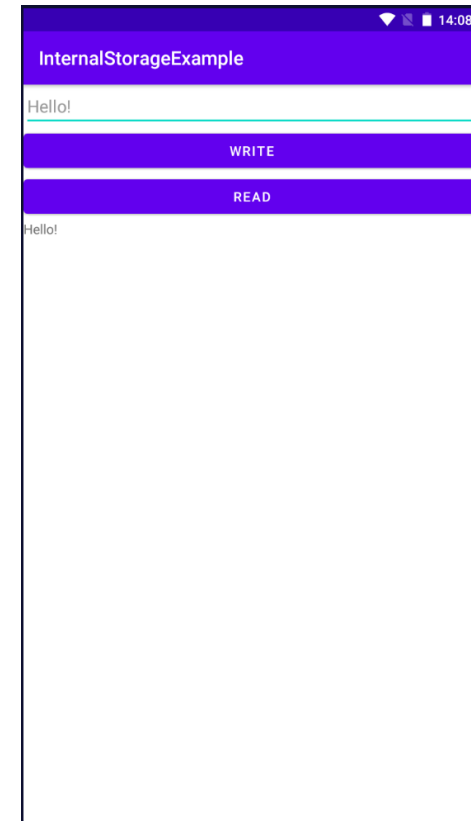
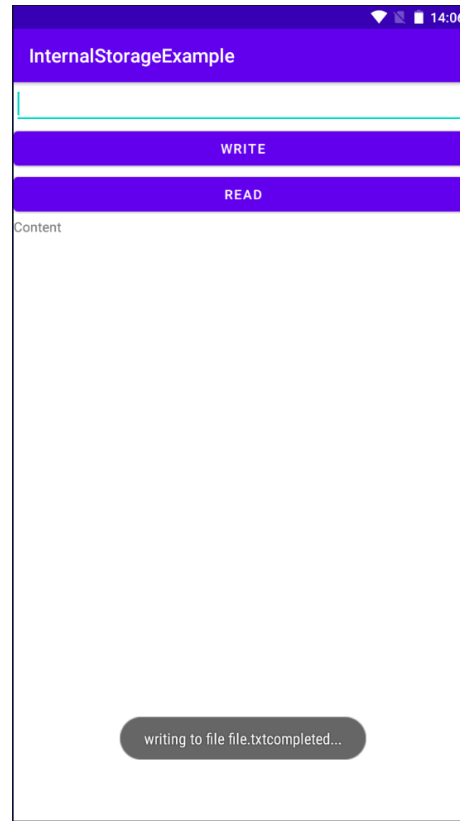
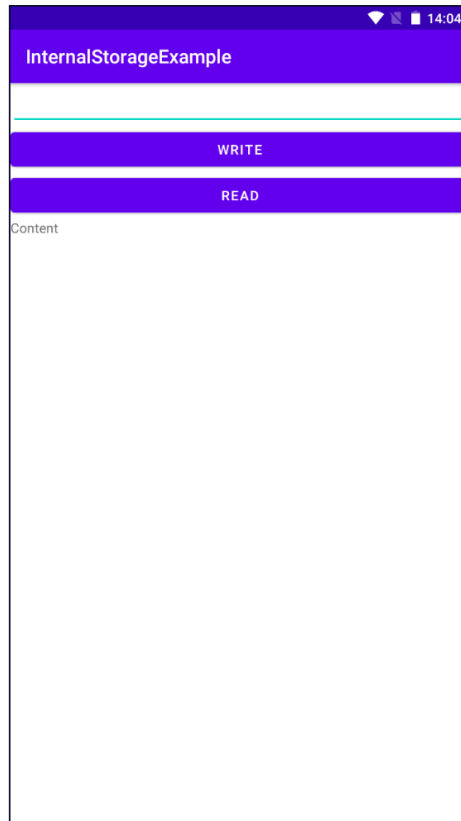


Упражнение

SAMSUNG



Создадим проект, в котором будем записывать и считывать данные из файла из внутреннего хранилища устройства.





SAMSUNG



Спасибо

