

ЛЕКЦИЯ I © 2021 Нет Ит

# Android: Saving Data #1

Теодор Костадинов



SOFTWARE  
ACADEMY



# Съдържание

---

1. Storage
2. Shared Prefs
3. Permissions
4. Internal Storage
5. Media Storage



# Types of Storages

За записване на постоянни данни, в Андроид имаме различни варианти.

- App-specific Storage
  - App folder - всеки апп може да записва файлове във вътрешната директория на аппа. Тези файлове са лични и не могат да се достъпват от други аппове.
  - Shared Preferences - всеки апп може да създаде прости файлове от тип речник. Всяка данна се записва като ключ и стойност.
- Shared Storage - всеки апп може да създава файлове, които иска да сподели с други аппове.
- Databases - всеки апп може да създава бази данни. Те се съхраняват като файлове в личната папка на апп-а.



# Shared Preferences



# Shared Preferences

Стандартен начин да запазим някакви настройки, които потребителя е задал.

Не се ползва за съхранение на обемни данни.

```
SharedPreferences sharedPref = context.getSharedPreferences(  
    "com.my.app.PREFERENCE_FILE_KEY", Context.MODE_PRIVATE);
```

# Shared Preferences

```
SharedPreferences.Editor editor = sharedPref.edit();  
editor.putInt(getString(R.string.saved_high_score_key), newHighScore);  
editor.apply();
```

```
int highScore =  
sharedPref.getInt(getString(R.string.saved_high_score_key), defaultValue);
```





# Permissions

# Permissions

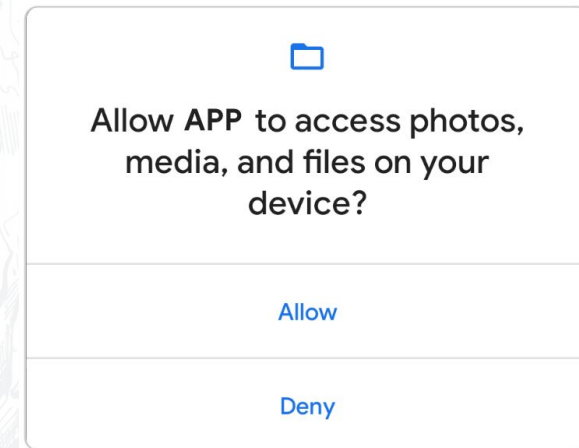
В Андроид не можем да правим каквото си искаме.

Потребителят трябва да е наясно с всяка опасна/тежка/публична операция, която вашето приложение прави.

За повечето такива операции, трябва да се поиска изрично позволение от потребителя под формата на permission. Има два вида:

- Implicit
- Explicit

И двата вида се дефинират в Манифеста на апп-а. Благодарение на това, те са видими още преди потребителя да си го инсталира - в Google Play.





# Implicit

Този тип е достатъчно само да бъде дефиниран в Манифеста.

Такива позволения са например това за Интернет.


Показва се в Google Play, но не и в самият апп.

## Messenger – Text and Video Chat for Free

Facebook

Показани са разрешенията за всички версии на това приложение

Това приложение има достъп до:

 Данни за идентификационния номер на устройството и за обажданията


- read phone status and identity

 SMS

- read your text messages (SMS or MMS)
- receive text messages (MMS)
- send SMS messages
- edit your text messages (SMS or MMS)

 Календар

- read calendar events plus confidential information

 Местоположение

- precise location (GPS and network-based)
- approximate location (network-based)

 Телефон

- read phone status and identity
- directly call phone numbers

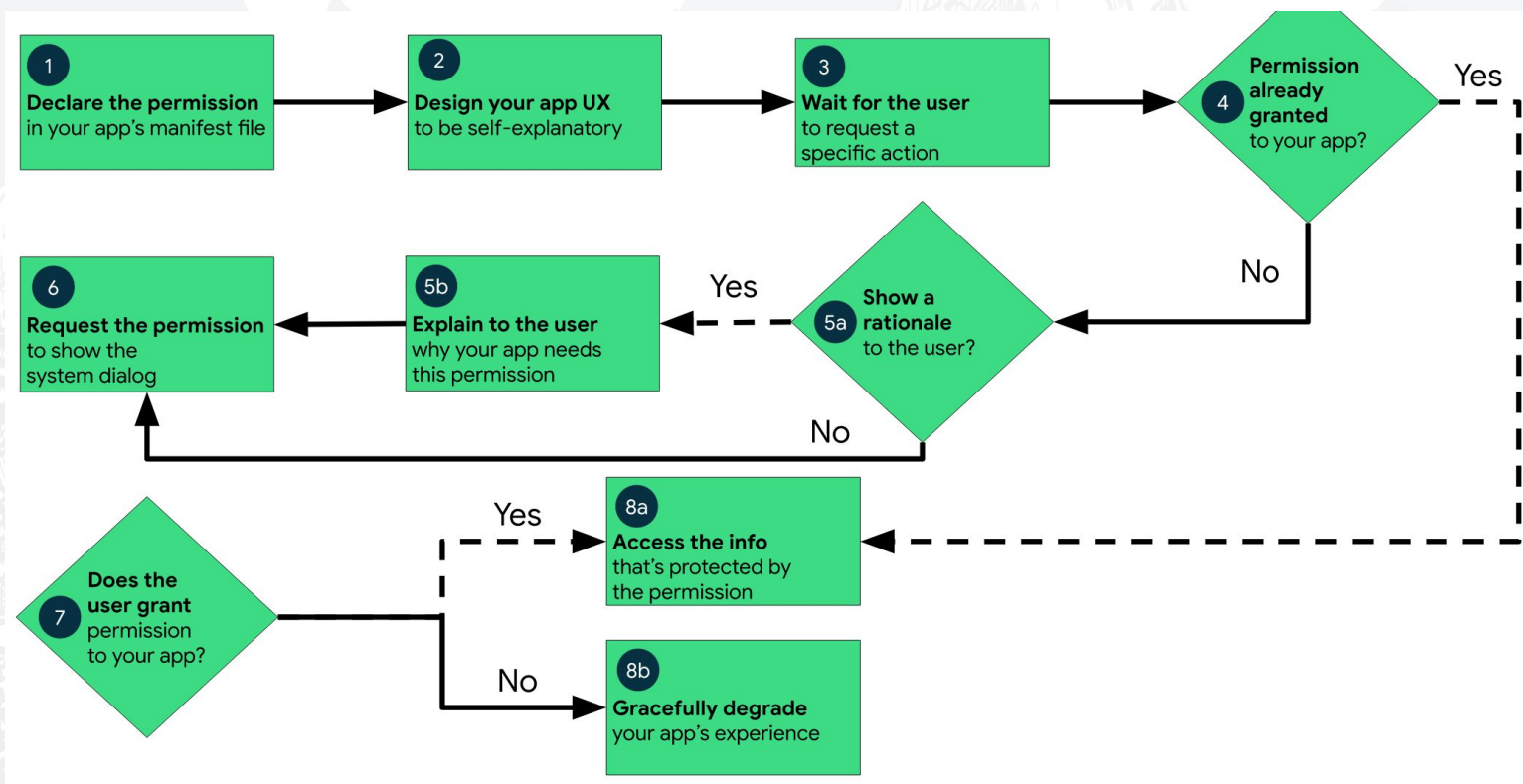
# Explicit

Този тип трябва да бъде дефиниран в Манифеста НО и да се поиска допълнително - по време на изпълнение на апа.

Такива позволения са например това за записване на файлове на телефона, достъп до контакти, блутуут, локация и тн.

Показва се в Google Play, но и в самият апа.

Потребителят може да избере да го откаже и трябва да можем да се справим с този случай.





# Permissions in Code

```
ContextCompat.checkSelfPermission(  
    CONTEXT, Manifest.permission.REQUESTED_PERMISSION) ==  
    PackageManager.PERMISSION_GRANTED
```

Проверяваме дали вече  
нямаме това  
разрешение. Това  
върща TRUE/FALSE.

```
requestPermissions(  
    new String[] { Manifest.permission.REQUESTED_PERMISSION },  
    REQUEST_CODE);
```

Искаме това  
разрешение.



**Storage**



# Internal Storage

Internal Storage е мястото където записваме данните на апликацията, които няма да се споделят с други апликации.

Той е ограничен, не искаме да записваме големи или много файлове там.

Винаги го има. Няма вариант при който устройството да няма internal storage.

Директорията се достъпва чрез:

- `getFilesDir()`
- `getCacheDir()`

# Internal Storage in Code

```
File file = new File(context.getFilesDir(),  
filename);
```

**Създаваме файл във  
вътрешната  
директория.**

```
String filename = "myfile";  
String fileContents = "Hello world!";  
try (FileOutputStream fos = context.openFileOutput(filename, Context.MODE_PRIVATE)) {  
    fos.write(fileContents.toByteArray());  
}
```

**Създаваме стрийм във  
вътрешна директория.**



# Internal Storage in Code

```
FileInputStream fis = context.openFileInput(filename);
InputStreamReader inputStreamReader =
    new InputStreamReader(fis, StandardCharsets.UTF_8);
StringBuilder stringBuilder = new StringBuilder();
try (BufferedReader reader = new BufferedReader(inputStreamReader)) {
    String line = reader.readLine();
    while (line != null) {
        stringBuilder.append(line).append('\n');
        line = reader.readLine();
    }
} catch (IOException e) {
    // Error occurred when opening raw file for reading.
} finally {
    String contents = stringBuilder.toString();
}
```

**Четене на текстови  
файл от вътрешна  
директория.**

# External Storage

External Storage е цялото останало пространство на телефона, което не се използва директно от апове. То се състои от различни папки за:

- Снимки
- Документи
- Книги
- Видео

Идеята е потребителя да намери всичките си снимки, видеа и тн на едно място, а не всеки апов да създава собствени си папки.

За Андроид 10 и надолу е възможно да се поиска позволение да се достъпват всички директории, но това ще бъде ограничено в бъдещи версии.



# Internal Storage in Code

```
// Request code for creating a PDF document.  
private static final int CREATE_FILE = 1;  
  
private void createFile(Uri pickerInitialUri) {  
    Intent intent = new Intent(Intent.ACTION_CREATE_DOCUMENT);  
    intent.addCategory(Intent.CATEGORY_OPENABLE);  
    intent.setType("application/pdf");  
    intent.putExtra(Intent.EXTRA_TITLE, "invoice.pdf");  
    intent.putExtra(DocumentsContract.EXTRA_INITIAL_URI, pickerInitialUri);  
  
    startActivityForResult(intent, CREATE_FILE);  
}
```

Res: <https://developer.android.com/training/data-storage/shared/media>

**Създаваме файл във  
вътрешната  
директория.**

ВЪПРОСИ?





© 2020 Нет Ит

# БЛАГОДАРЯ ЗА ВНИМАНИЕТО!



SOFTWARE  
ACADEMY

