

UTILISATION DE LA MÉMOIRE DE STOCKAGE SUR ANDROID

@.fR Frédéric RALLO



AUTORISATIONS DANS LE MANIFEST

- ❑ **Une application qui doit accéder à la mémoire de stockage doit avoir une autorisation qui ressemble à cette ligne dans le Manifeste :**

```
<manifest xmlns:android = "http://schemas.android.com/apk/res/android"
    package = « edu.demo.ihm" >

    <uses-permission android: name = "android.permission.WRITE_EXTERNAL_STORAGE" />
    <! - d'autres autorisations vont ici ->

    <application ... >
        ...
    </application>
</manifest>
```

- ❑ **Une autorisation de lire ou d'écrire dans la mémoire est considérée comme une « autorisation dangereuse »**

DIFFÉRENTES MÉMOIRES

- ❑ **Avant Android 6.0 (API 23)**, il était possible mettre de la musique, des vidéos et des images sur une carte SD, mais vous ne pouviez pas installer une application sur la carte.



Après Android 6.0 (API 23), Google supprime cette restriction des données via une fonctionnalité Adoptable Storage.

Le problème avec cette méthode, c'est qu'elle formate la carte SD pour ne fonctionner que sur le mobile en question, et rend difficile sa réutilisation sur un autre appareil.

DIFFÉRENTES MÉMOIRES





TERMINOLOGIE DE GOOGLE

Le stockage interne. Il s'agit des fichiers de votre application.

- ❑ Ils sont privés : les autres applications ne peuvent pas y accéder, ni l'utilisateur à moins de rooter le téléphone Android.
 - ❑ Espace partagé :
 - *AUCUNE POSSIBILITE*
 - ❑ Espace privé :
 - `getDir()`
 - `getFilesDir()`
 - `getCacheDir()`
- **Emplacement** : `/data/data/your.package.appname/someDirectory`



TERMINOLOGIE DE GOOGLE

Le stockage externe.

Ce type de stockage peut être partagé par toutes les applications et par l'utilisateur depuis un PC en le connectant au téléphone via un câble USB.

❑ Espace partagé :

- `getExternalStorageDirectory()` //déprecié pour API > 29
- `getExternalStoragePublicDirectory()` //répertoire spécifique

Emplacement :

`/storage/emulated/0/` ou `/storage/sdcard1/`

❑ Espace privé :

- `getExternalFilesDir()`
- `getExternalCacheDir()`
- `getExternalMediaDirs()`

Emplacement :

`/storage/emulated/0/Android/data/your.application.package.appname/files`
`/storage/sdcard1/Android/data/your.application.package.appname/files`



TERMINOLOGIE DE GOOGLE

Le stockage externe.

On distingue deux types de stockage externes :

- **Stockage externe primaire** : Pas de permission à demander si API > 19
 - De même nature que la mémoire interne du téléphone. Accessible depuis un PC en sélectionnant *Camera (PTP)* ainsi que *Transfert de fichiers* dans la notification des options USB
 - Emplacement :
`/storage/Android/data/your.package.appname/someDirectory`
- **Stockage externe secondaire** : Permission indispensable
 - Il s'agit d'un stockage amovible. Accessible depuis un PC en sélectionnant *Transfert de fichiers* dans la notification des options USB
 - Emplacement :
`/sdcard/Android/data/your.package.appname/someDirectory`

NOTES IMPORTANTES

- ❑ **Si l'appareil n'a pas de support de stockage externe secondaire, le lien vers la carte SD est automatiquement considéré comme stockage externe principal sur la plupart des appareils (c'est le cas des émulateurs Android studio API <29)**
- ❑ **Sur les versions antérieures à API 29, la méthode `Environment.getExternalStorageDirectory ()` renverra l'un de ces deux chemins de support de stockage**
- ❑ **La gestion de la mémoire externe peut être difficile si on veut passer de la primary à secondary. En effet, lorsqu'on a visité un dossier, celui-ci est mémorisé par le système où qu'il soit.**
- ❑ **→ Si on bascule primary/secondary, le système recrée alors tous les dossiers visités (mémorisés) en mémoire externe en mélangeant alors la provenance primary et secondary. !!!**



STOCKAGE ET AUTORISATIONS ?

méthode	Demander une autorisation ?	Qui a accès aux fichiers ?	Fichiers supprimés après désinstallation ?
getFilesDir() getCacheDir()	Tacite pour la mémoire interne	Accès privé pour la mémoire interne	oui
getExternalFilesDir() getExternalCacheDir() <i>getExternalMediaDirs()</i> <i>getExternalStorageDirectory()</i> <i>getExternalStoragePublicDirectory()</i>	Pas indispensable pour la mémoire externe si on utilise une API 19 ou plus	Accès public pour la mémoire externe	Oui lorsqu'ils sont rattachés à un dossier nominatif Non s'ils sont dans un espace partagé
MediaStore API via ContentResolver	READ_EXTERNAL_STORAGE WRITE_EXTERNAL_STORAGE Indispensable depuis API 28	Accès public mais soumis à approbation de l'utilisateur	non

OÙ SE TROUVENT LES FICHIERS DEPUIS « DEVICE FILE EXPLORER »



Internal
storage

Device File Explorer

Emulator Pixel_2_API_27 Android 8.1.0, API 27

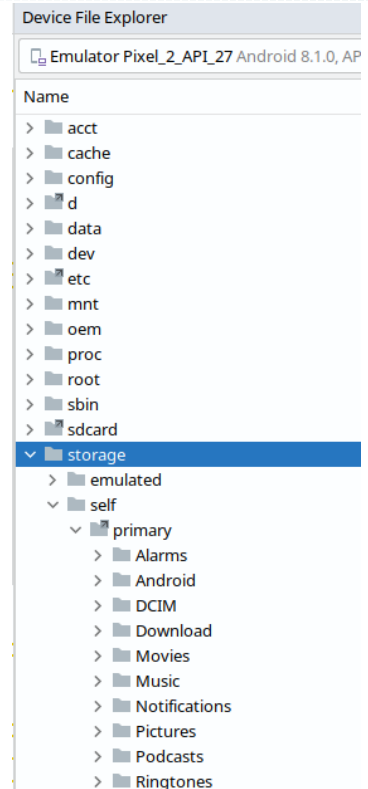
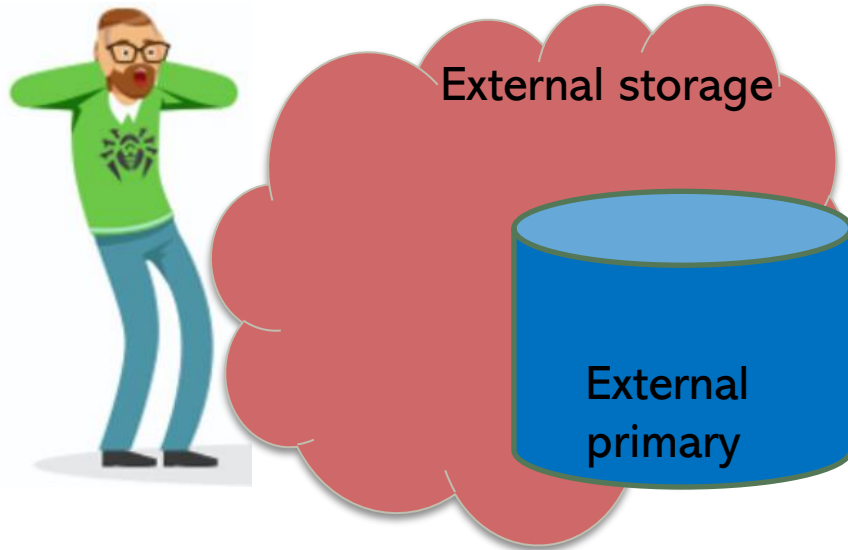
Name

- > acct
- > cache
- > config
- > d
- ▼ data
 - > app
 - ▼ data
 - > android
 - > com.android.backupconfirm
 - > com.android.bips
 - > com.android.bookmarkprovider
 - > com.android.calculator2
 - > com.android.calllogbackup
 - > com.android.camera2
 - > com.android.captiveportallogin

```
File internal_m1 = getDir("custom", 0);  
/data/data/your.application.package.appname/app_custom
```

```
File internal_m2 = getFilesDir();  
/data/data/your.application.package.appname/files
```

OÙ SE TROUVENT LES FICHIERS DEPUIS « DEVICE FILE EXPLORER »



```
File external_m1 = Environment.getExternalStorageDirectory();  
/storage/emulated/0
```

```
File external_m2 = getExternalFilesDir(null);  
/storage/emulated/0/Android/data/your.application.package.appname/files
```

```
File external_m2_Args = getExternalFilesDir(Environment.DIRECTORY_PICTURES);  
/storage/emulated/0/Android/data/your.application.package.appname/files/Pictures
```

```
File external_m3 = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);  
/storage/emulated/0/Pictures
```

OÙ SE TROUVENT LES FICHIERS DEPUIS « DEVICE FILE EXPLORER »

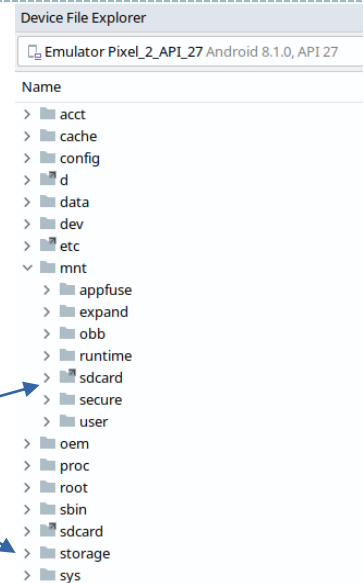


External
storage

Lorsqu'aucune carte SD n'est présente,
l'émulateur crée un lien sdcard vers
external primary

SD
External
secondary

Oui...
mais NON



```
File[] external_AND_removable_storage_m1 = getExternalFilesDirs(null);  
(first path) /storage/emulated/0/Android/data/your.application.package.appname/files  
(second path) /storage/sdcard1/Android/data/your.application.package.appname/files
```

```
File[] external_AND_removable_storage_m1_Args = getExternalFilesDirs(DEFAULT_FOLDER);  
(first path) /storage/emulated/0/Android/data/your.application.package.appname/files/Pictures  
(second path) /storage/sdcard1/Android/data/your.application.package.appname/files/Pictures
```

EN RÉSUMÉ

`File internal_m1 = getDir("custom", 0);`

→ `/data/data/your.application.package.appname/app_custom`

`File internal_m2 = getFilesDir();`

→ `/data/data/your.application.package.appname/files`

`File external_m1 = Environment.getExternalStorageDirectory();`

→ `/storage/emulated/0`

`File external_m2 = getExternalFilesDir(null);`

→ `/storage/emulated/0/Android/data/your.application.package.appname/files`

`File external_m2_Args = getExternalFilesDir(Environment.DIRECTORY_PICTURES);`

→ `/storage/emulated/0/Android/data/your.application.package.appname/files/Pictures`

`File external_m3 = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);`

→ `/storage/emulated/0/Pictures`

`File[] external_AND_removable_storage_m1 = getExternalFilesDirs(null);`

→ (first path) `/storage/emulated/0/Android/data/your.application.package.appname/files`

→ (second path) `/storage/sdcard1/Android/data/your.application.package.appname/files`

`File[] external_AND_removable_storage_m1_Args = getExternalFilesDirs(DEFAULT_FOLDER);`

→ (first path) `/storage/emulated/0/Android/data/your.application.package.appname/files/Pictures`

→ (second path) `/storage/sdcard1/Android/data/your.application.package.appname/files/Pictures`

EXEMPLE D'UTILISATION

❑ Trouver le chemin (mémoire externe)

```
public File getPath() {  
    File[] external_AND_removable_storage_m1_Args = getExternalFilesDirs(DEFAULT_FOLDER + currentFolder);  
    path = external_AND_removable_storage_m1_Args[isSdcard ? 1 : 0];  
    return path;  
}
```

❑ Explorer le contenu d'un répertoire .

```
private void loadToExternalStorage() {  
    path = getPath();  
    listOfFile = new ArrayList<>();  
    if (!currentFolder.equals("")) folders.add("..");  
    for (String element : path.list()) {  
        listOfFile.add(element);  
    }  
    FileAdapter adapter = new FileAdapter(this, listOfFile);  
    listView.setAdapter(adapter);  
}
```



EXEMPLE D'UTILISATION

❑ Enregistrer un fichier

```
private void saveToExternalStorage(String[] lines, String fileName) {
    if (Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED_READ_ONLY)) return;
    path = getPath();
    File myfile = new File(path, fileName);
    try {
        myfile.createNewFile();
        try {
            FileOutputStream fout = new FileOutputStream(myfile);
            PrintWriter pw = new PrintWriter(fout);
            for (String line : lines) {
                pw.println(line + "\r\n");
            }
            pw.flush();
            pw.close();
            fout.close();
            loadToExternalStorage();
            Log.d(TAG, "File Written successfully" );
        } catch (Exception e) {
            Log.d(TAG, "ERROR: " + e.toString());
        }
    } catch (IOException e1) {
        Log.d(TAG, "I/O ERROR: " + e1.toString());
    }
}
```



EXEMPLE D'UTILISATION

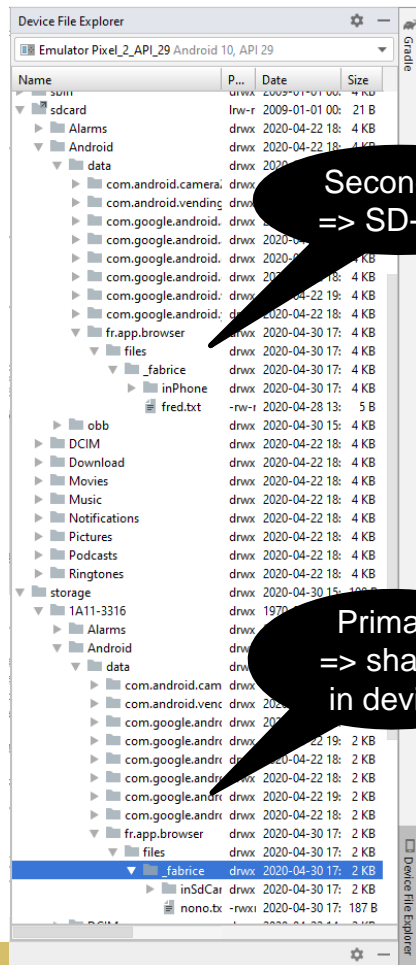
❑ Enregistrer une image API 29+ (avec gestion des permission)

```
public void saveToInternalStorage(Bitmap picture) {
    if (ContextCompat.checkSelfPermission(getContext(), Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_DENIED) {
        ActivityCompat.requestPermissions( getActivity(),
            new String[] {Manifest.permission.READ_EXTERNAL_STORAGE},
            IStorageActivity.REQUEST_MEDIA_READ);
    } else { //permission is still granted
        ContentValues contentValues = new ContentValues();
        contentValues.put(MediaStore.MediaColumns.DISPLAY_NAME, pictureName);
        contentValues.put(MediaStore.MediaColumns.MIME_TYPE, "image/*");
        contentValues.put(MediaStore.MediaColumns.RELATIVE_PATH, directoryName);
        contentValues.put(MediaStore.Images.Media.MIME_TYPE, "image/jpeg");
        getActivity().getContentResolver().insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, contentValues);
        File = new File(directoryName, pictureName);
        FileOutputStream fos = null;
        try {
            fos = new FileOutputStream(file);
            picture.compress(Bitmap.CompressFormat.PNG, 90, fos);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    } //end if
}
```



INFORMATIONS COMPLÉMENTAIRES

❑ Pour mettre en application, voici une application que je vous propose de réaliser



Indique si on écrit dans la PRIMARY ou la SECONDARY

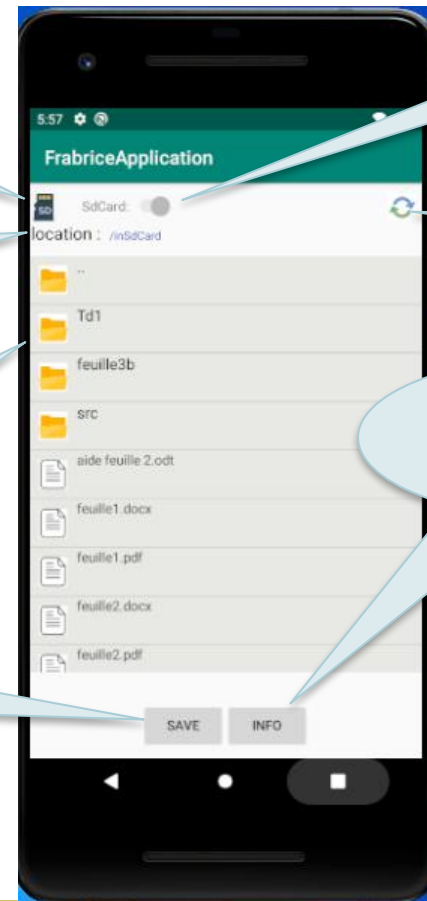
Secondary
=> SD-card

Chemin du répertoire courant

Contenu du répertoire

Primary
=> shared in device

Crée un fichier texte appelé nono.txt



Choisir PRIMARY ou SECONDARY

actualiser

Affiche un AlertDialog pour expliquer le fonctionnement

