

Android

Almacenamiento externo

Almacenamiento en fichero de recursos

Android permite almacenar ficheros como recursos en la carpeta **res/raw**

Estos ficheros no van a ser comprimidos.

Se eliminan al desinstalar la aplicación.

Son únicamente de lectura

Se accede por el nombre del fichero sin su extensión

Por ejemplo a **res/raw/ayuda.txt** accederemos con **R.raw.ayuda**

Usaremos **getResources()** de la Activity para devolver un objeto **Resources**.

Usaremos el método **openRawResource()** para abrir el fichero de recursos

InputStreamReader

- Convierte un stream de bytes en caracteres
- Necesita un InputStream para leer de él
- Se puede leer de él carácter a carácter con **read()**
- O también un conjunto de caracteres con **read(char[] buffer, int offset, int count)**

BufferedReader

- Encapsula la lectura de un Reader (un InputStreamReader) proporcionando un buffer de lectura de 8Kb lo que acelera operaciones de lectura.
- Nos permite, a parte de los read, leer línea a línea con **readLine()**

```
InputStream instream = this.getResources().openRawResource(R.raw.ayuda);
InputStreamReader inputreader = new InputStreamReader(instream);
BufferedReader buffreader = new BufferedReader(inputreader);
String str = null;
try{
    while( (str=br.readLine())!= null){
        //Hacer algo con la línea leída
    }
    instream.close();
    buffreader.close();
}
catch (IOException e){
    e.printStackTrace();
}
```

Almacenamiento Memoria Externa

La memoria externa tiene más capacidad que la interna, usarla para ficheros grandes. Utilizaremos la memoria externa normalmente si queremos que se tenga acceso público a los ficheros, aunque también pueden ser privados a la aplicación.

- Memoria externa extraíble -> Tarjetas SD
- Memoria externa no extraíble -> Accedemos por USB
- Algunos dispositivos tienen ambas.

La memoria externa se monta en la carpeta `/sdcard/...`

Para obtener la ruta de montaje utilizar: `Environment.getExternalStorageDirectory()`

Necesitamos permisos de acceso (a partir de la versión 1.6) en el Manifest

```
<manifest ...>
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
                  android:maxSdkVersion="18" />
</manifest>
```

A partir de Android 4.4, estos permisos no son necesarios si deseas leer o escribir archivos que solamente son privados para tu app

`WRITE_EXTERNAL_STORAGE` (incluye acceso en lectura)

Hay que validar el estado de la memoria externa, si está disponible o no y en que modo, solo lectura o lectura/escritura.

Utilizaremos: `Environment.getExternalStorageState()`

Posibles valores:

`MEDIA_MOUNTED` (RW), `MEDIA_MOUNTED_READ_ONLY` (R-)

`MEDIA_SHARED`, `MEDIA_REMOVED`, `MEDIA_CHECKING`, `MEDIA_BAD_REMOVAL` ...

...en estos estados no se puede ni leer ni escribir.

Almacenamiento Memoria Externa

A partir del API 8 (version 2.2) podemos utilizar ficheros externos específicos y **privados** de nuestra aplicación en la ruta:

`/Android/data/<nombre_del_package>/files/`

Cuando la aplicación se desinstala se borran automáticamente estos ficheros.

Para saber la ruta:

`getExternalFilesDir(null)`

En lugar de null podemos utilizar otros valores para determinar el tipo de fichero de forma que se categoricen adecuadamente para el escaner de Medios de Android.

`getExternalFilesDir(String tipo)`

En función de tipo nos va a dar la ruta hasta la carpeta para el tipo de fichero que nos interesa. Si la carpeta no existe la crea a partir de la ruta indicada.

Un dispositivo puede particionar la memoria interna y ofrecer parte de ésta como memoria externa.

Además también puede ofrecer un slot para tarjetas SD.

Con Android 4.3 (API 18) y anteriores `getExternalFilesDir()` sólo da la ruta de la partición interna montada como externa.

A partir de Android 4.4 (API 19) devuelve un File array con cada una de las localizaciones.

La primera de ellas es la interna montada como externa, mientras no esté llena deberíamos usar esa.

Constante	carpeta	descripción
<code>DIRECTORY_MUSIC</code>	<code>Music</code>	ficheros de música
<code>DIRECTORY_PODCASTS</code>	<code>Podcasts</code>	descargas desde podcast
<code>DIRECTORY_RINGTONES</code>	<code>Ringtones</code>	tono de llamada de teléfono
<code>DIRECTORY_ALARMS</code>	<code>Alarms</code>	sonidos de alarma
<code>DIRECTORY_NOTIFICATIONS</code>	<code>Notifications</code>	sonidos para notificaciones
<code>DIRECTORY_PICTURES</code>	<code>Pictures</code>	ficheros con fotografías
<code>DIRECTORY_DOWNLOADS</code>	<code>Download</code>	descargas de cualquier tipo
<code>DIRECTORY_DCIM</code>	<code>DCIM</code>	carpeta que tradicionalmente crean las cámaras

Almacenamiento Memoria Externa

Lo normal es usarla para compartir ficheros con otras aplicaciones.

No será borrado al desinstalar la aplicación

En este caso hay que crearlo en cualquier directorio fuera de estas rutas de aplicación.

Lo ideal es utilizar los directorios **públicos** de la tabla anterior, que se crean a partir de la raíz de la memoria externa.

Con `getExternalStoragePublicDirectory(String tipo)` nos da la carpeta correspondiente al tipo pasado.

Con versiones anteriores a la 2.2 no funciona este método.

En cualquier caso es bueno usar las mismas carpetas.

Usaremos `getExternalStoragePublicDirectory() + "/Music"` para ficheros públicos

Y `getExternalStoragePublicDirectory() +`

`"/Android/data/<nombre_del_package>/files/ Music"` para privados