

[ESCRIBIR EL NOMBRE DE LA COMPAÑÍA]

SharedPreferences en Android

Juan Antonio Japón de la Torre

Juan

09/09/2016

Contenido

SharedPreferences	3
¿Qué es un SharedPreference?	3
Actualizar o insertar nuevas preferencias.....	3
Recuperando los datos almacenados en SharedPreferences	4
Borrando o limpiando el contenido de nuestro SharedPreferences	4

SharedPreferences

¿Qué es un SharedPreferences?

Las Preferencias o datos de sesión son un componente disponible en Android, que nos permite almacenar información de una aplicación, como por ejemplo información personal, opciones de configuración, los datos de inicio de sesión de un formulario de login, etc.

Android proporciona un método alternativo al almacenamiento en bases de datos o ficheros, diseñado específicamente para administrar este tipo de datos: las preferencias compartidas o **SharedPreferences**. Cada preferencia se almacenará en forma de **clave-valor**, es decir, cada una de ellas estará compuesta por un identificador único ("**email**") y un valor asociado a dicho identificador ("**hola@ejemplo.com**"). Los datos no se guardan en ficheros XML

Toda la gestión se centraliza en la clase `SharedPreferences`, que representará a una colección de preferencias. Una aplicación Android puede gestionar varias colecciones de preferencias, que se diferenciarán mediante un identificador único. Para obtener una referencia a una colección determinada utilizaremos el método `getSharedPreferences()` al que pasaremos el identificador de la colección y un modo de acceso. El modo de acceso indicará qué aplicaciones tendrán acceso a la colección de preferencias y qué operaciones tendrán permitido realizar sobre ellas. Así, tendremos tres posibilidades principales:

- **MODE_PRIVATE**. Sólo nuestra aplicación tiene acceso a estas preferencias.
- **MODE_WORLD_READABLE**, **MODE_WORLD_WRITEABLE**. Estas dos opciones se han declarado como obsoletas en la API 17.

Actualizar o insertar nuevas preferencias

Este proceso por ejemplo se realizaría cuando hubiéramos realizado el login de usuario y queramos guardar los datos de usuario logeado para que en una posterior apertura de la aplicación usara dicha sesión guardada. Para ello realizamos:

```
//Creamos nuestro objeto SharedPreferences
SharedPreferences prefs =
getSharedPreferences("MisPreferencias",getApplicationContext().MODE_PRIVATE);
```

```
//Abrimos el editor para editar en el archivo donde guardaremos datos
SharedPreferences.Editor editor = prefs.edit();
```

```
//Añadimos datos (clave / valor) para recuperarlos al iniciar la app
posteriormente
editor.putString("logeado", "si");
editor.putString("usuariologeado", usu.getText().toString());
//editor.putBoolean("claveBooleana", false);
//editor.putFloat("claveFloat", (float) 2.4);
//editor.putInt("claveEntera", 5);
```

```
//Guardamos los cambios
editor.commit();
```

Recuperando los datos almacenados en SharedPreferences

Para recuperar los datos que guardamos a la hora de realizar el login o cualquier acción y queramos evitar que el usuario se tenga que volver a logear posteriormente realizamos lo siguiente para recuperar dichos datos:

```
//Recuperamos el sharedPreferences
SharedPreferences prefs =
getSharedPreferences("MisPreferencias",getApplicationContext().MODE_PRIVATE);

//Una vez recuperado usamos el metodo getString pasandole la clave que
pusimos y un valor por defecto en caso de no devolver nada para que no
devuelva nulo
String logeado = prefs.getString("logeado", "no");
String usuario = prefs.getString("usuariologeado", "ejemplo");
```

Borrando o limpiando el contenido de nuestro SharedPreferences

Este caso se utiliza cuando queremos cerrar la sesión del usuario que se logeo en nuestra aplicación por ejemplo. Para ello tenemos que realizar lo siguiente:

```
//Creamos nuestro objeto SharedPreferences
SharedPreferences prefs =
getSharedPreferences("MisPreferencias",getApplicationContext().MODE_PRIVATE);

//Abrimos el editor para editar en el archivo donde guardaremos
datos o borrarremos el contenido
SharedPreferences.Editor editor = prefs.edit();

//borramos el contenido
editor.clear();

//comiteamos o guardamos cambios
editor.commit();
```