Android Essentials

Permisos

Permisos en Android

FOLDER IT

Todos los permisos de Android, comienzan con `android.permission` y son listados en la API de Android.
Igualmente las aplicaciones como las que desarrollamos nosotros, pueden tener sus propios permisos.

Algunos de los permisos más comunes son:

`INTERNET` : si la aplicación desea acceder a internet de cualquier manera

`WRITE_EXTERNAL_STORAGE` : se necesita para escribir datos en la SD.

`NFC` : permite realizar acciones de IO con el protocolo “near-field communication (NFC)”

`ACCESS_COARSE_LOCATION` y `ACCESS_FINE_LOCATION` : son permisos que podemos solicitar para determinar la ubicación del dispositivo.

`CALL_PHONE` permite a la aplicación realizar llamadas directamente

Permisos en Android

FOLDER IT

En modo desarrollador o cuando usamos `adb install` , no se preguntan los permisos

Si no se obtuvo el permiso al ejecutar una acción se obtendrá una `SecurityException`, en la que se informa el permiso faltante

Esto solo sucede cuando nos olvidamos de solicitar un permiso. Si el permiso fue declarado en el manifest, se aceptará al instalar la aplicación y ejecutar normalmente sin `SecurityException's`

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

Permisos normales

FOLDER IT

Son aquellos que no representan mayores riesgos para la privacidad del usuario o el funcionamiento del dispositivo

Si son declarados correspondientemente en el manifest, son concedidos automáticamente

Cubren áreas donde la aplicación necesita acceder a datos o recursos fuera de su sandbox, pero el riesgo de la privacidad del usuario o la operación de otras aplicaciones es bajo

Permisos normales

ACCESS_LOCATION_EXTRA_COMMANDS
ACCESS_NETWORK_STATE
ACCESS_NOTIFICATION_POLICY
ACCESS_WIFI_STATE
BLUETOOTH
BLUETOOTH_ADMIN
BROADCAST_STICKY
CHANGE_NETWORK_STATE
CHANGE_WIFI_MULTICAST_STATE
CHANGE_WIFI_STATE
DISABLE_KEYGUARD
EXPAND_STATUS_BAR
GET_PACKAGE_SIZE
INSTALL_SHORTCUT
INTERNET
KILL_BACKGROUND_PROCESSES
MODIFY_AUDIO_SETTINGS
NFC
READ_SYNC_SETTINGS
READ_SYNC_STATS
RECEIVE_BOOT_COMPLETED
REORDER_TASKS
REQUEST_IGNORE_BATTERY_OPTIMIZATIONS
REQUEST_INSTALL_PACKAGES
SET_ALARM
SET_TIME_ZONE
SET_WALLPAPER
SET_WALLPAPER_HINTS
TRANSMIT_IR
UNINSTALL_SHORTCUT
USE_FINGERPRINT
VIBRATE
WAKE_LOCK
WRITE_SYNC_SETTINGS

Permisos peligrosos

FOLDER IT

Son aquellos que pueden potencialmente afectar la privacidad o el funcionamiento del dispositivo
El sistema pide que el usuario conceda explícitamente estos permisos
Cubren áreas de la app donde se quiere acceder a datos o recursos donde hay información privada del usuario o que puede afectar la información almacenada o funcionamiento de otras apps

```
READ_CALENDAR
WRITE_CALENDAR
CAMERA
READ_CONTACTS
WRITE_CONTACTS
GET_ACCOUNTS
ACCESS_FINE_LOCATION
ACCESS_COARSE_LOCATION
RECORD_AUDIO
READ_PHONE_STATE
CALL_PHONE
READ_CALL_LOG
```

```
WRITE_CALL_LOG
ADD_VOICEMAIL
USE_SIP
PROCESS_OUTGOING_CALLS
BODY_SENSORS
SEND_SMS
RECEIVE_SMS
READ_SMS
RECEIVE_WAP_PUSH
RECEIVE_MMS
READ_EXTERNAL_STORAGE
WRITE_EXTERNAL_STORAGE
```

Permisos estáticos y dinámicos

FOLDER IT

Depende de la versión de Android los permisos se solicitarán de manera distinta

Si el dispositivo corre sobre Android 5 (API 22) o inferior y el targetSdkVersion es 22 o menor
Los permisos se solicitarán al momento de instalarla
Si se agregan permisos, se pedirán en el momento de actualizar la aplicación
La única forma de revocar los permisos es desinstalar la aplicación

Si el dispositivo corre sobre Android 6 (API 23) o superior y el targetSdkVersion es 23 o mayor
Los permisos se solicitarán en tiempo de ejecución

El usuario puede revocar los permisos en cualquier momento, por esto debemos verificarlos en todo momento.

Permisos dinámicos

FOLDER IT

Primero que todo debemos verificar en que version de Android estamos ejecutando

```
if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){
   // Tenemos que pedir los permisos en runtime
} else {
   // ya deberiamos tener los permisos
}
```

Luego, cada vez que necesitamos un permiso peligroso, tenemos que verificar si tenemos el permiso (**Recordemos:** El usuario puede revocar el permiso en cualquier momento)

```
if (ContextCompat.checkSelfPermission(getApplicationContext(),
        android.Manifest.permission.MY_PERMISSION)
        != PackageManager.PERMISSION_GRANTED) {
   // No tengo el permiso hay que pedirlos
} else {
   // ya tengo los permisos
}
```

Pedir permisos

FOLDER IT

En caso de no tener permisos debemos invocar al método `requestPermissions()`
Este método recibe un `String[]` con la lista de permisos a pedir, y un código para identificar nuestra petición luego

Puede que el usuario deniegue nuestra petición. Por lo tanto necesitamos explicarle porque necesitamos los permisos

Para esto necesitamos el método `shouldShowRequestPermissionRationale(context, perm)` retorna verdadero en caso de que el permiso haya sido solicitado antes y el usuario no lo haya concedido

`requestPermissions()`, **Este método es Asincrónico, no bloquea la UI.**

Entonces cómo sabemos desde qué `Activity` se lanzó el método?

Pedir permisos

FOLDER IT

Implementamos el Listener `onRequestPermissionsResult()`

```
@Override
public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
    switch (requestCode) {
        case MyActivity.MY_PERMISSION_REQUEST: {
            // si el request es cancelado el arreglo es vacio.
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                // tengo el permiso!!!.
            } else {
                // no tenemos permisos
            }
            return;
        }
    }
}
```

Pedir permisos

FOLDER IT

```
if (ActivityCompat.shouldShowRequestPermissionRationale(getApplicationContext(),
        android.Manifest.permission.MY_PERMISSION)) {

    AlertDialog.Builder builder = new AlertDialog.Builder(getApplicationContext());
    builder.setTitle("Me das permiso ?!?!");
    builder.setPositiveButton(android.R.string.ok, null);
    builder.setMessage("Me los das o no ?");
    builder.setOnDismissListener(new DialogInterface.OnDismissListener() {
        @Override
        public void onDismiss(DialogInterface dialog) {
            requestPermissions(new String[] {android.Manifest.permission.MY_PERMISSION}
                    , MY_PERMISSION_REQUEST_CODE);
        }
    });
    builder.show();

}
```