**Guia 22: Permisos en android**

1. Crear un proyecto base con la restricción minSdkVersion 19.
2. Modificar el archivo AndroidManifest.xml

*<?***xml version="1.0" encoding="utf-8"***?>*

<**manifest xmlns:android="http://schemas.android.com/apk/res/android"**

**package="mobile.calyr.com.permisos"**>

**<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />**

<**application**

**android:allowBackup="true"**

**android:icon="@mipmap/ic\_launcher"**

**android:label="@string/app\_name"**

**android:roundIcon="@mipmap/ic\_launcher\_round"**

**android:supportsRtl="true"**

**android:theme="@style/AppTheme"**>

<**activity android:name=".MainActivity"**>

<**intent-filter**>

<**action android:name="android.intent.action.MAIN"** />

<**category android:name="android.intent.category.LAUNCHER"** />

</**intent-filter**>

</**activity**>

</**application**>

</**manifest**>

3. Editar el archivo activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*

<**LinearLayout**

**xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:app="http://schemas.android.com/apk/res-auto"**

**xmlns:tools="http://schemas.android.com/tools"**

**android:fitsSystemWindows="true"**

**android:orientation="vertical"**

**android:id="@+id/content\_main"**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"**

**tools:context="mobile.calyr.com.permisos.MainActivity"**

**tools:showIn="@layout/activity\_main"**>

<**Button**

**android:id="@+id/btn\_validar"**

**android:layout\_marginTop="20dp"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:onClick="validarPermisos"**

**android:text="Verificar Permiso"**/>

</**LinearLayout**>

4. Editar la clase MainActivity.java

**package** mobile.calyr.com.permisos;

**import** android.Manifest;

**import** android.content.DialogInterface;

**import** android.content.Intent;

**import** android.content.SharedPreferences;

**import** android.content.pm.PackageManager;

**import** android.net.Uri;

**import** android.os.Bundle;

**import** android.provider.Settings;

**import** android.support.v4.app.ActivityCompat;

**import** android.support.v7.app.AlertDialog;

**import** android.support.v7.app.AppCompatActivity;

**import** android.view.View;

**import** android.widget.Toast;

**public class** MainActivity **extends** AppCompatActivity {

**private static final int *EXTERNAL\_STORAGE\_PERMISSION\_CONSTANT*** = 100;

**private static final int *REQUEST\_PERMISSION\_SETTING*** = 101;

**private boolean sentToSettings** = **false**;

**private** SharedPreferences **permissionStatus**;

@Override

**protected void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main***);

**permissionStatus** = getSharedPreferences(**"permissionStatus"**,***MODE\_PRIVATE***);

}

**public void** validarPermisos(View view) {

**if** (ActivityCompat.*checkSelfPermission*(MainActivity.**this**, Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***) != PackageManager.***PERMISSION\_GRANTED***) {

**if** (ActivityCompat.*shouldShowRequestPermissionRationale*(MainActivity.**this**, Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***)) {

*//Show Information about why you need the permission*

AlertDialog.Builder builder = **new** AlertDialog.Builder(MainActivity.**this**);

builder.setTitle(**"Necesitamos permisos de almacenamiento"**);

builder.setMessage(**"Esta aplicación necesita permiso de almacenamiento."**);

builder.setPositiveButton(**"Grant"**, **new** DialogInterface.OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) {

dialog.cancel();

ActivityCompat.*requestPermissions*(MainActivity.**this**, **new** String[]{Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***}, ***EXTERNAL\_STORAGE\_PERMISSION\_CONSTANT***);

}

});

builder.setNegativeButton(**"Cancel"**, **new** DialogInterface.OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) {

dialog.cancel();

}

});

builder.show();

} **else if** (**permissionStatus**.getBoolean(Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***,**false**)) {

*//Previously Permission Request was cancelled with 'Dont Ask Again',*

*// Redirect to Settings after showing Information about why you need the permission*

AlertDialog.Builder builder = **new** AlertDialog.Builder(MainActivity.**this**);

builder.setTitle(**"Esta aplicación necesita permiso de almacenamiento."**);

builder.setMessage(**"Esta aplicación necesita permiso de almacenamiento."**);

builder.setPositiveButton(**"Grant"**, **new** DialogInterface.OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) {

dialog.cancel();

**sentToSettings** = **true**;

Intent intent = **new** Intent(Settings.***ACTION\_APPLICATION\_DETAILS\_SETTINGS***);

Uri uri = Uri.*fromParts*(**"package"**, getPackageName(), **null**);

intent.setData(uri);

startActivityForResult(intent, ***REQUEST\_PERMISSION\_SETTING***);

Toast.*makeText*(getBaseContext(), **"Ir a la configuración de permisos."**, Toast.***LENGTH\_LONG***).show();

}

});

builder.setNegativeButton(**"Cancel"**, **new** DialogInterface.OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) {

dialog.cancel();

}

});

builder.show();

} **else** {

*//just request the permission*

ActivityCompat.*requestPermissions*(MainActivity.**this**, **new** String[]{Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***}, ***EXTERNAL\_STORAGE\_PERMISSION\_CONSTANT***);

}

SharedPreferences.Editor editor = **permissionStatus**.edit();

editor.putBoolean(Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***,**true**);

editor.commit();

} **else** {

*//You already have the permission, just go ahead.*

proceedAfterPermission();

}

}

**private void** proceedAfterPermission() {

*//We've got the permission, now we can proceed further*

Toast.*makeText*(getBaseContext(), **"Tenemos el permiso de Almacenamiento"**, Toast.***LENGTH\_LONG***).show();

}

}

5. Ejecutar la aplicación en una versión anterior y una posterior a la api 23.