



Présentation

Les permissions à l'exécution

Par Soha LAGNEB



Pourquoi une Permission

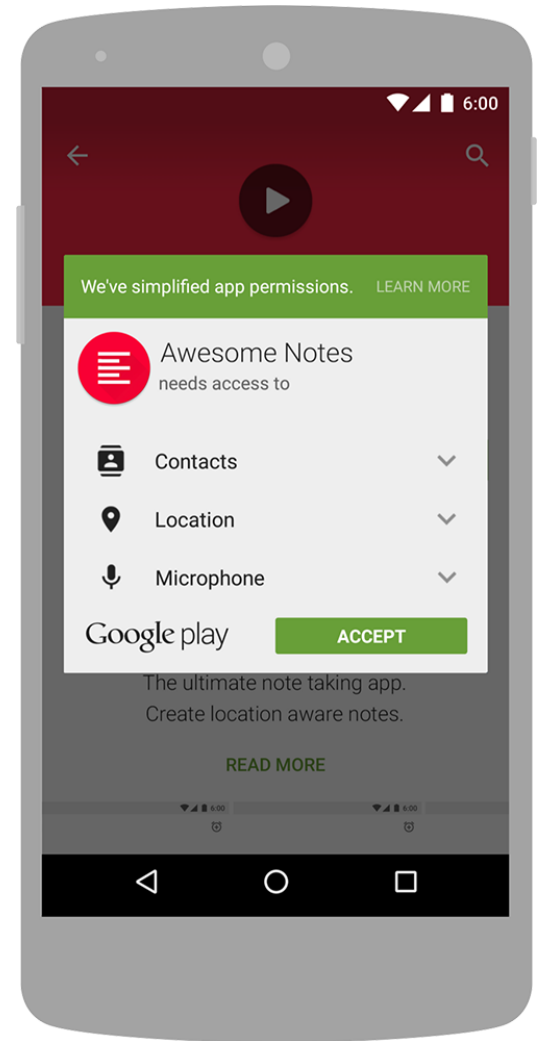
- Protection de la vie privée des utilisateurs
- L'utilisateur est celui qui décide d'autoriser ou non l'accès un des données personnelles ou à des fonctionnalités précises.

Trois types de permissions

- Permission normale: `BLUETOOTH, INTERNET..`
- Permission de signatures: `READ_VOICEMAIL..`
- Permissions dangereuses: `READ_CALENDAR, CAMERA, WRITE_CONTACTS, ACCESS_FINE_LOCATION, RECORD_AUDIO..`

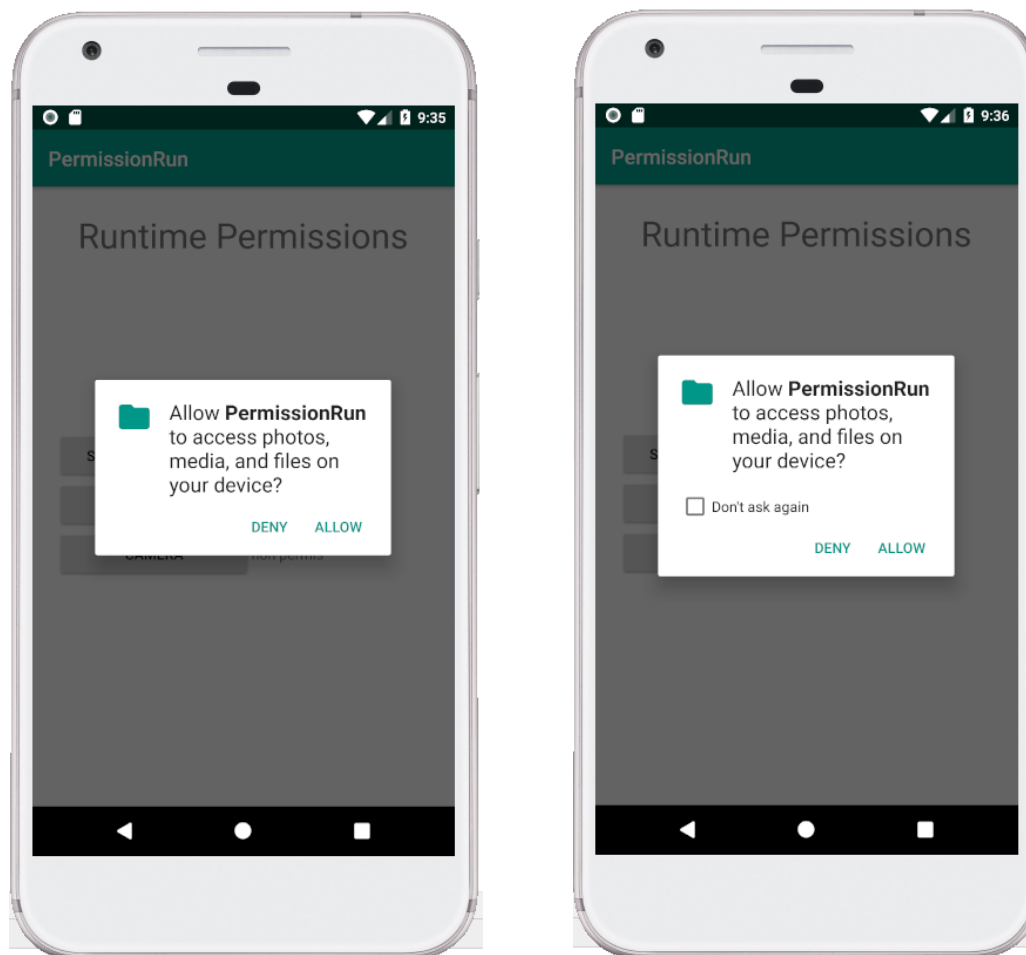
Permission à l'installation

- Android 5.1.1 (API 22) et moins
ou `targetSdkVersion 22` et moins

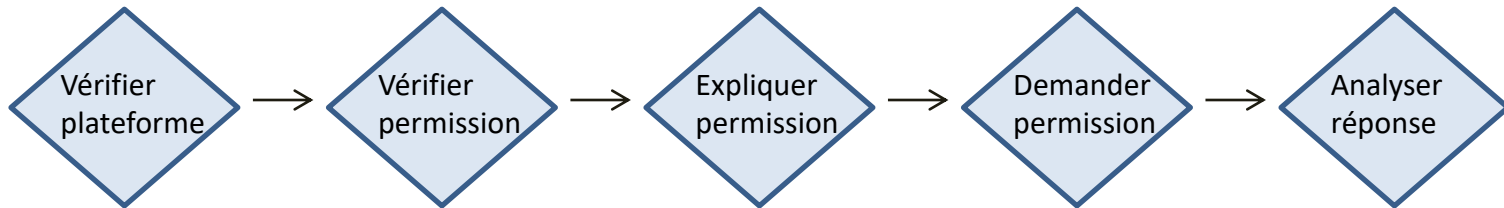


Permissions à l'exécution

- Android 6.0 (API 23) et plus



Comment ça marche?



Déclarer les permissions requises

```
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.CAMERA" />
```

Vérification de la permission

```
if (ContextCompat.checkSelfPermission(context: MainActivity.this,
    Manifest.permission.READ_CONTACTS) == PackageManager.PERMISSION_GRANTED) {
    Toast.makeText(context: MainActivity.this, text: "Vous avez déjà autorisé cette fonctionnalité",
        Toast.LENGTH_SHORT).show();
} else {
    requestContactsPermission();
}
```

Expliquer et demander la permission

```
private void requestCameraPermission() {  
    if (ActivityCompat.shouldShowRequestPermissionRationale activity: this,  
        Manifest.permission.CAMERA) {  
  
        new AlertDialog.Builder( context: this)  
            .setTitle("Permission requise")  
            .setMessage("L'application a besoin d'accéder à votre caméra")  
            .setPositiveButton( text: "ok", (dialog, which) → {  
                ActivityCompat.requestPermissions( activity: MainActivity.this,  
                    new String[] {Manifest.permission.CAMERA}, CAMERA_PERMISSION_CODE);  
            })  
            .setNegativeButton( text: "cancel", (dialog, which) → {  
                dialog.dismiss();  
            })  
            .create().show();  
    } else {  
        ActivityCompat.requestPermissions( activity: this,  
            new String[] {Manifest.permission.CAMERA}, CAMERA_PERMISSION_CODE);  
    }  
}
```

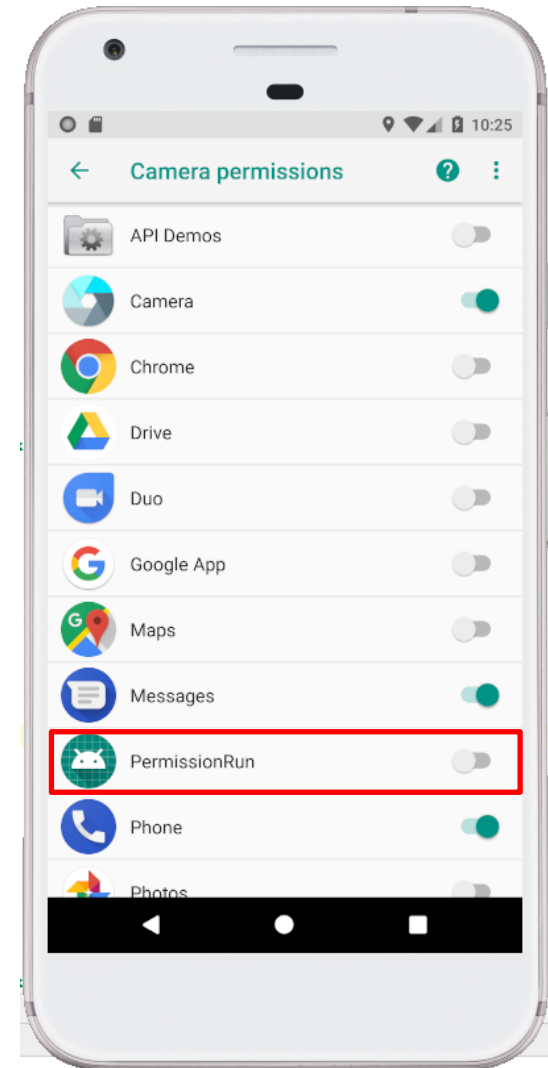

Analyser le résultat

```
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    if (requestCode == STORAGE_PERMISSION_CODE) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(context: this, text: "Permission acquise", Toast.LENGTH_SHORT).show();
            etat_stockage.setText("permis");
        } else {
            Toast.makeText(context: this, text: "Permission refusée", Toast.LENGTH_SHORT).show();
            etat_stockage.setText("non permis");
        }
    }
    if (requestCode == CONTACTS_PERMISSION_CODE) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(context: this, text: "Permission acquise", Toast.LENGTH_SHORT).show();
            etat_contacts.setText("permis");
        } else {
            Toast.makeText(context: this, text: "Permission refusée", Toast.LENGTH_SHORT).show();
            etat_contacts.setText("non permis");
        }
    }
}
```

Démonstration

Résiliation des permissions

- Soit manuellement par l'utilisateur



Résiliation des permissions

- Soit manuellement par l'utilisateur
- Soit implicitement via l'application

```
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    if (requestCode == STORAGE_PERMISSION_CODE) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(context: this, text: "Permission acquise", Toast.LENGTH_SHORT).show();
            etat_stockage.setText("permis");
        } else {
            Toast.makeText(context: this, text: "Permission refusée", Toast.LENGTH_SHORT).show();
            etat_stockage.setText("non permis");
            String permission = permissions[0];
            boolean showRationale = shouldShowRequestPermissionRationale( permission );
            if (! showRationale) {
                startActivity(new Intent(android.provider.Settings.ACTION_APPLICATION_DETAILS_SETTINGS, Uri.parse("package:" + BuildConfig.APPLICATION_ID)));
            }
        }
    }
}
```

Webographie

- [Permissions normales/dangereuses](#) (developer.android)
- [Demander une permission](#) (developer.android)
- [Tutoriel demander une permission dangereuse](#) (YouTube)
- [Analyse des résultats](#) (developer.android)
- [Vérifier si "don't ask again" coché](#) (stackoverflow)



Présentation

Les permissions à l'exécution

Par Soha LAGNEB

Merci

