

# Les autorisations en android

---

Chaque application Android s'exécute dans un sandbox à accès limité. Si une application doit utiliser des ressources ou des informations en dehors de son propre sandbox, elle doit demander l'autorisation appropriée . Vous déclarez que votre application a besoin d'une autorisation en répertoriant l'autorisation dans le manifeste de l'application , puis en demandant à l'utilisateur d'approuver chaque autorisation au moment de l'exécution (sur Android 6.0 et versions ultérieures).

## Ajouter des autorisations au manifeste

Sur toutes les versions d'Android, pour déclarer que votre application a besoin d'une autorisation, insérez un élément dans le manifeste de votre application, en tant qu'enfant de l'élément de niveau supérieur . Par exemple, une application ayant besoin d'accéder à Internet aurait cette ligne dans le manifeste:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.snazzyapp">

    <uses-permission android:name="android.permission.INTERNET"/>
    <!-- other permissions go here -->

    <application ...>
        ...
    </application>
</manifest>
```

Si votre application nécessite une autorisation dangereuse, vous devez vérifier si vous disposez de cette autorisation à chaque fois que vous effectuez une opération nécessitant cette autorisation. Depuis Android 6.0 (API niveau 23), les utilisateurs peuvent révoquer les autorisations de n'importe quelle application à tout moment.

## Vérifier si vous avez une permission

```
if (ContextCompat.checkSelfPermission(thisActivity,
Manifest.permission.WRITE_CALENDAR)
    == PackageManager.PERMISSION_GRANTED) {

    // Permission granted
}

if (ContextCompat.checkSelfPermission(thisActivity,
Manifest.permission.WRITE_CALENDAR)
    == PackageManager.PERMISSION_DENIED) {

    // Permission denied
}
```

## Demander des autorisations

Lorsque votre application reçoit la notification **PERMISSION\_DENIED** de *checkSelfPermission()*, vous devez demander à l'utilisateur cette autorisation. Android fournit plusieurs méthodes que vous pouvez utiliser pour demander une autorisation, telles que *requestPermissions()*, comme indiqué dans l'extrait de code ci-dessous. L'appel de ces méthodes ouvre une boîte de dialogue Android standard, que vous ne pouvez pas personnaliser.

*Une approche que vous pouvez utiliser consiste à fournir une explication uniquement si l'utilisateur a déjà refusé cette demande d'autorisation. Android fournit une méthode utilitaire *shouldShowRequestPermissionRationale()*, qui retourne **true** si l'utilisateur a précédemment refusé la demande, et **false** si un utilisateur a refusé une autorisation et sélectionné l'option Ne plus demander dans la boîte de dialogue de ne plus demander d'autorisation.*

```
// Here, thisActivity is the current activity
if (ContextCompat.checkSelfPermission(thisActivity,
    Manifest.permission.READ_CONTACTS)
    != PackageManager.PERMISSION_GRANTED) {

    // Permission is not granted
    ActivityCompat.requestPermissions(thisActivity,
        arrayOf(Manifest.permission.READ_CONTACTS),
        MY_PERMISSIONS_REQUEST_READ_CONTACTS)

    // MY_PERMISSIONS_REQUEST_READ_CONTACTS is an
    // app-defined int constant. The callback method gets the
    // result of the request.

} else {
    // Permission has already been granted
}
```

## Gérer la réponse à la demande d'autorisation

Lorsque l'utilisateur répond à la demande d'autorisation de votre application, le système appelle la *onRequestPermissionsResult()* méthode de votre application et lui transmet la réponse de l'utilisateur. Votre application doit remplacer cette méthode pour savoir si l'autorisation a été accordée.

```
override fun onRequestPermissionsResult(requestCode: Int,
    permissions: Array<String>, grantResults: IntArray) {

    // Si la demande est annulé , le tableau grantResults sera vide

    if ((grantResults.isNotEmpty() &&
grantResults[0]==PackageManager.PERMISSION_GRANTED))
    {

        // permission accordé
    }
}
```

```
        go()
    } else {
        // permission denied, boo! Disable the
        // functionality that depends on this permission.
    }
    return
}
}
```