



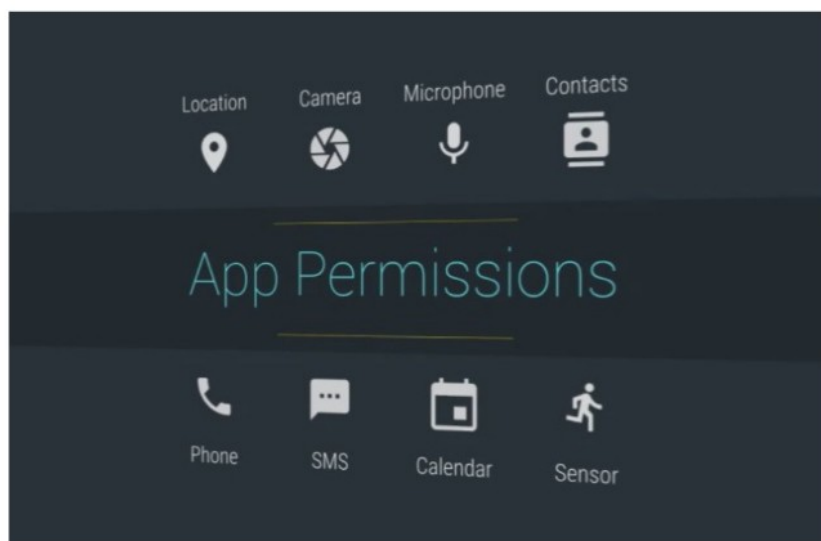
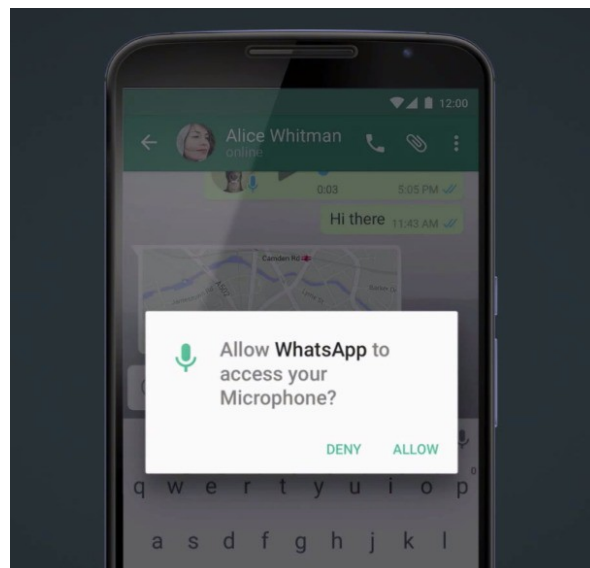
Android -TP

Les permissions à la volée

Source : doc officielle developers.android.com

Quésako ?

A partir d'Android 6 (SDK 23+), l'acceptation des permissions par l'utilisateur ne se fait plus à l'installation mais **pendant l'exécution**.



Création du projet (avec Android 6 sinon l'appli va partir en cacahuète)

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☒ Phone and Tablet

Minimum SDK: API 23: Android 6.0 (Marshmallow)

Lower API levels target more devices, but have fewer features available.
By targeting API 23 and later, your app will run on approximately 4.7% of the devices that are active on the Google Play Store.
[Help me choose](#)

☐ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Android Auto

☐ Glass

Minimum SDK: Glass Development Kit Preview (API 19)

Previous Next Cancel Finish

Déclarer une permission

Dans le manifest de l'application, utiliser la balise `<uses-permission />`

```
AndroidManifest.xml x
manifest
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3    package="com.example.aristide.myapplication">
4
5    <uses-permission android:name="android.permission.SEND_SMS"/>
6
7    <application
8        android:allowBackup="true"
9        android:icon="@mipmap/ic_launcher"
10       android:label="My Application"
11       android:supportsRtl="true"
12       android:theme="@style/AppTheme">
13        <activity android:name=".MainActivity">
14            <intent-filter>
15                <action android:name="android.intent.action.MAIN" />
16
17                <category android:name="android.intent.category.LAUNCHER" />
18            </intent-filter>
19        </activity>
20    </application>
21
22 </manifest>
```

Demander une permission

Code : <http://pastebin.com/ejqKgvyE>

1. La fonction **checkSelfPermission(...)** permet de vérifier si une permission est accordée.

```
int checkSelfPermission (Context context,  
                        String permission)
```

Determine whether you have been granted a particular permission.

Parameters	
context	Context
permission	String: The name of the permission being checked.

Returns	
int	PERMISSION_GRANTED if you have the permission, or PERMISSION_DENIED if not.

- .

Exemple d'implémentation :

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.READ_CONTACTS) != PackageManager.PERMISSION_GRANTED) {
```

2. La fonction **shouldShowRequestPermissionRationale(...)** permet de savoir si la permission a déjà été demandée.

```
boolean shouldShowRequestPermissionRationale (Activity activity,  
                                             String permission)
```

Parameters	
activity	Activity: The target activity.
permission	String: A permission your app wants to request.
Returns	
boolean	Whether you can show permission rationale UI.

Exemple d'implémentation :

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.READ_CONTACTS) != PackageManager.PERMISSION_GRANTED) {  
        if (ActivityCompat.shouldShowRequestPermissionRationale(this, Manifest.permission.READ_CONTACTS)) {  
            // Sacrebleu ! La permission a déjà été demandée mais ce clavier d'utilisateur a refusé  
        } else {  
            // Demander la permission  
        }  
    }  
}
```

3. La fonction **requestPermission(...)** permet de demander une permission.

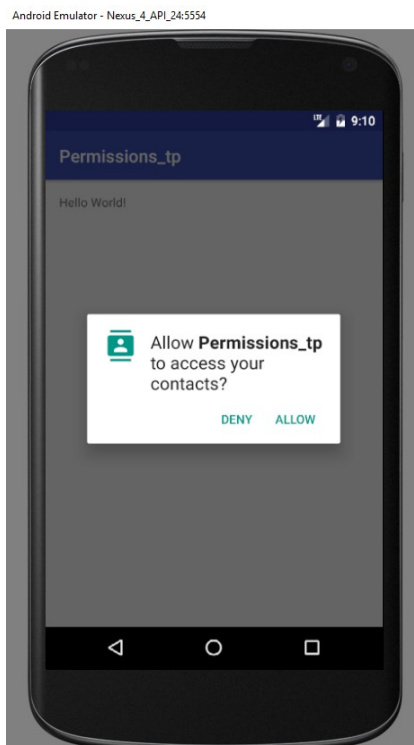
```
void requestPermissions (Activity activity,  
                        String[] permissions,  
                        int requestCode)
```

Parameters	
activity	Activity: The target activity.
permissions	String: The requested permissions. Must be non-null and not empty.
requestCode	int: Application specific request code to match with a result reported to <code>onRequestPermissionsResult(int, String[], int[])</code> . Should be ≥ 0 .

Exemple d'implémentation :

```
private static final int MY_PERMISSIONS_REQUEST_READ_CONTACTS = 1;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.READ_CONTACTS) != PackageManager.PERMISSION_GRANTED) {  
        if (ActivityCompat.shouldShowRequestPermissionRationale(this, Manifest.permission.READ_CONTACTS)) {  
            // Sacrebleu ! La permission a déjà été demandée mais ce clavier d'utilisateur a refusé  
        } else {  
            // Demander la permission  
            ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.READ_CONTACTS}, MY_PERMISSIONS_REQUEST_READ_CONTACTS);  
        }  
    }  
}
```

Tada !



Callback

Le système renvoie un callback à l'app pour lui notifier de la réponse de l'utilisateur.

```
void onRequestPermissionsResult (int requestCode,  
    String[] permissions,  
    int[] grantResults)
```

Parameters	
<code>requestCode</code>	<code>int</code> : The request code passed in <code>requestPermissions(android.app.Activity, String[], int)</code>
<code>permissions</code>	<code>String</code> : The requested permissions. Never null.
<code>grantResults</code>	<code>int</code> : The grant results for the corresponding permissions which is either <code>PERMISSION_GRANTED</code> or <code>PERMISSION_DENIED</code> . Never null.

Exemple d'implémentation :

```
@Override  
public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {  
    switch (requestCode) {  
        case MY_PERMISSIONS_REQUEST_READ_CONTACTS: {  
            if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {  
                // Permission accordée hourra  
            } else {  
                // Permission refusée  
            }  
            return;  
        }  
    }  
}
```

FIN

