Please consult the following sources when deciding to make use of custom android permissions.

- https://developer.android.com/training/permissions/declaring.html

Beginning in Android 6.0 (API level 23), users grant permissions to apps while the app is running, not when they install the app. This approach streamlines the app install process, since the user does not need to grant permissions when they install or update the app. It also gives the user more control over the app's functionality; for example, a user could choose to give a camera app access to the camera but not to the device location. The user can revoke the permissions at any time, by going to the app's Settings screen.

System permissions are divided into two categories, *normal* and *dangerous:*

- **Normal permissions do not directly risk the user's privacy. If your app lists a normal permission in its manifest, the system grants the permission automatically.**
- **Dangerous permissions can give the app access to the user's confidential data. If your app lists a normal permission in its manifest, the system grants the permission automatically. If you list a dangerous permission, the user has to explicitly give approval to your app.**

For more information, see Normal and Dangerous Permissions.

On all versions of Android, your app needs to declare both the normal and the dangerous permissions it needs in its app manifest, as described in Declaring Permissions. However, the *effect* of that declaration is different depending on the system version and your app's target SDK level:

- **If the device is running Android 5.1 or lower, or your app's target SDK is 22 or lower: If you list a dangerous permission in your manifest, the user has to grant the permission when they install the app; if they do not grant the permission, the system does not install the app at all.**
- **If the device is running Android 6.0 or higher, and your app's target SDK is 23 or higher: The app has to list the permissions in the manifest, *and* it must request each dangerous permission it needs while the app is running. The user can grant or deny each permission, and the app can continue to run with limited capabilities even if the user denies a permission request.**

Note: Beginning with Android 6.0 (API level 23), users can revoke permissions from any app at any time, even if the app targets a lower API level. You should test your app to verify that it behaves properly when it's missing a needed permission, regardless of what API level your app targets.

Add Permissions to the Manifest

To declare that your app needs a permission, put a <uses-permission> element in your app manifest, as a child of the top-level <manifest> element. For example, an app that needs to send SMS messages would have this line in the manifest:

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

    <uses-permission android:name="android.permission.SEND_SMS"/>


    <application ...>
        ...
    </application>

</manifest>
```