

CAI-IP01 SMB116

# Les Permissions





# Les Permissions / Autorisations <sup>(1)</sup>

- ▶ **But:** Assurer la sécurité du système et de ses utilisateurs
- ▶ **Comment:** Android impose aux applications de demander la permission pour:
  - ▶ Accéder à certaines informations
  - ▶ Utiliser certaines fonctionnalités
- ▶ Permissions classées en deux niveaux :
  - ▶ **Normales** : Les données et fonctionnalités exploitées par l'application ne sont pas considérées comme confidentielles ou à risque. Android donne automatiquement la permission aux applications qui la réclament. (*Exemple: régler l'heure*)  
<https://developer.android.com/guide/topics/permissions/normal-permissions.html>
  - ▶ **Dangereuses** : Les données qui vont être manipulées sont des informations privées de l'utilisateur (*liste des contacts*) . Les fonctionnalités à utiliser pourraient mettre en péril le fonctionnement d'autres applications. **Ces permissions doivent être données explicitement par l'utilisateur.**  
<https://developer.android.com/guide/topics/permissions/requesting.html#normal-dangerous>



# Les Permissions / Autorisations (2)

- **Groupe de permissions** : Les Permissions « Dangereuses » sont classées en groupes.

| Permission Group | Permissions   |
|------------------|---|
| CONTACTS         | <ul style="list-style-type: none"><li>• READ_CONTACTS</li><li>• WRITE_CONTACTS</li><li>• GET_ACCOUNTS</li></ul> |

**Accepter une Permission d'un groupe revient à accepter toutes les permissions de ce groupe**



# Définir les permissions pour une application

- ▶ Une application a besoin de Permissions pour **accéder aux données et fonctionnalités qui lui sont nécessaires**
- ▶ **Seul les accès directs nécessitent des Permissions:** Si l'application utilise un composant externe qui accède à des données soumises à Permission, **c'est au composant d'obtenir la Permission.**
- ▶ Les Permissions nécessaires à l'application sont déclarées dans le Manifest:

```
<manifest xmlns:android=http://schemas.android.com/apk/res/android package="edu.cai.skoup">  
    <uses-permission android:name="android.permission.READ_CONTACTS"/>  
    <application> . . . </application>  
</manifest>
```



# Contrôler les Permissions

- ▶ Quand une application réalise une action sans la Permission une exception est levée

`java.lang.SecurityException: Permission Denial`

- ▶ Il faut vérifier que la Permission a été obtenue pour l'action avant de la réaliser:

```
int permissionCheck = ContextCompat.checkSelfPermission(this, Manifest.permission.READ_CONTACTS);
```

La méthode **checkSelfPermission** retourne :

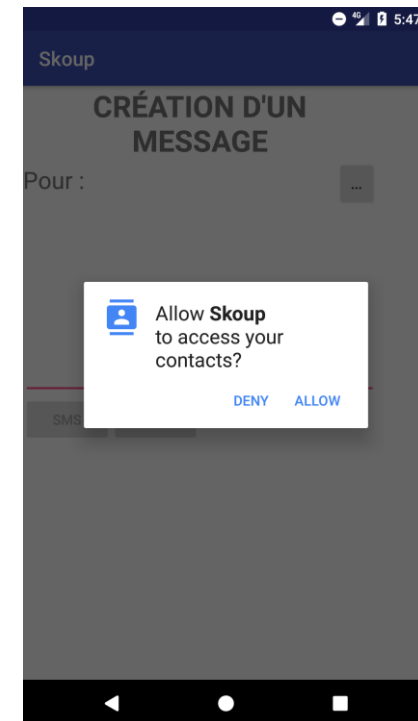
`PackageManager.PERMISSION_GRANTED` ou `PackageManager.PERMISSION_DENIED`

Lorsque la valeur de retour est **PERMISSION\_DENIED**  
l'action doit être abandonnée ou bien la permission doit être demandée



## Demander des Permissions (1)

- ▶ Avant de réaliser une action Si l'application **n'a pas obtenu la permission elle doit la demander**
- ▶ La demande d'une permission s'effectue avec la méthode: **requestPermission()**
- ▶ Le traitement de la demande s'effectue de manière **asynchrone**.
  - ▶ Une boîte de dialogue modale est affichée
  - ▶ Lorsque l'utilisateur répond: « Accepte » ou « Refuse »  
d'octroyer la permission la méthode **onRequestPermissionsResult** est invoquée.





## Demander des Permissions (2)

### ► Demande de la permission de lire les Contacts de l'utilisateur:

```
private static final int REQUEST_PERMISSION_READ_CONTACTS = 2;
/**
 * Selection du contact destinataire
 */
public void selectionDestinataire() {
    // Contrôle autorisation d'accéder aux contacts
    int permissionCheck = ContextCompat.checkSelfPermission(this, Manifest.permission.READ_CONTACTS);
    if(permissionCheck == PackageManager.PERMISSION_DENIED) {
        // Demande la permission de lire les contacts
        ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.READ_CONTACTS},
                                          REQUEST_PERMISSION_READ_CONTACTS
        );
    } else {
        // Sélection d'un destinataire
        Intent intent = new Intent(Intent.ACTION_PICK);
        intent.setType(ContactsContract.Contacts.CONTENT_TYPE);
        startActivityForResult(intent, REQUEST_SELECT_CONTACT);
    }
}
```



## Demander des Permissions (3)

### ► Traitement de la réponse de la demande de permission :

```
@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    switch (requestCode) {
        case REQUEST_PERMISSION_READ_CONTACTS:
            // Si une réponse est présente et qu'il s'agit d'une autorisation
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                // Lancement de l'action
                selectionDestinataire()
            }
        }
    }
    return;
}
```





# Une application avec ses propres Permissions

## ► Création de la Permission :

```
<permission android:name="edu.cai.logincomponent.LOGIN"
    android:label="Authentification par logincomponent"
    android:description="Authentification en utilisant le composant logincomponent"
    android:permissionGroup="android.permission-group.TOTO"
    android:protectionLevel="dangerous" />
```

## ► Création de la permission dans le Manifest de l'application

- **Label et Description** : textes descriptif de la permission
- **Groupe**: groupe de rattachement de la permission
- **Niveau**: niveau de la permission (Normale ou Dangereuse)



# Imposer une Permission pour lancer une Activité

## ► Pour contrôler l'accès à une Activité en imposant une permission:

- Ajout de l'attribut Permission dans la déclaration de l'activité :

```
<activity android:name=".LoginActivity" android:permission="edu.cai.logincomponent.LOGIN">
```

- Pour lancer l'activité la Permission est requise sinon une exception est levée

- L'application utilisatrice doit avoir déclaré la permission dans le Manifest par **uses-permission**

## ► Pour contrôler les autorisations de gestion d'un Service :

- Ajout de l'attribut Permission dans la déclaration du Service dans le Manifest.

- Cette permission est nécessaire aux processus pour :

- Démarrer le service [startService\(Intent\)](#)
  - Arrêter le service [stopService\(Intent\)](#)
  - Se connecter au service [bindService\(Intent, ServiceConnection, int\)](#)



# Broadcast / Receiver et Permissions

## ► Envoyer un Broadcast réservé aux Receiver ayant obtenu une Permission:

- Ajout de l'attribut Permission dans la déclaration de l'activité :

```
sendBroadcast(intentTick, "app.permission.RECEVOIR_BROADCAST_TEMPERATURE");
```

- Pour recevoir le Broadcast l'application du Receiver doit avoir obtenu la Permission
  - L'application du Receiver doit avoir déclaré la permission dans le Manifest par `uses-permission`

## ► Recevoir uniquement les Broadcast envoyés par des applications ayant une Permission

- Ajout de l'attribut Permission dans la déclaration du Receiver dans le Manifest.
- Ou bien enregistrement du receiver avec une Permission:

```
IntentFilter filter = new IntentFilter(Intent.ACTION_AIRPLANE_MODE_CHANGED);  
registerReceiver(receiver, filter, Manifest.permission.SEND_SMS, null );
```



# Permissions Android

- ▶ Elles permettent de limiter les actions que peut réaliser une application
- ▶ L'utilisateur propriétaire du périphérique doit donner explicitement les permissions nécessaire à l'application
- ▶ Il existe des Permissions livrées avec le système Android
- ▶ Il est possible de créer nos propres permissions
- ▶ Elles s'appliquent sur les Activités, les Services, Les BroadcastReceivers