

Android : Services de téléphonie

Frédéric Lemoine

Conservatoire National des Arts et Métiers

Services de téléphonie

```
TelephonyManager telephonyManager=  
(TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);
```

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

Information	Commentaire
int getPhoneType()	Type de téléphone (voir plus loin)
String getDeviceId()	L'identifiant unique du téléphone (IMEI)
String getDeviceSoftwareVersion()	Le numéro de la version du logiciel utilisé
String getLine1Number()	Le numéro de téléphone de l'abonné
boolean hasIccCard()	Retourne vrai si l'appareil contient une carte SIM (Integrated Circuit Card)

Services de téléphonie

String getNetworkOperatorName()	Nom de l'opérateur de téléphonie
String getNetworkOperator()	Identifiant de l'opérateur de téléphonie
String getNetworkCountryIso()	Code ISO du pays de l'opérateur (fr, de ...)
int getNetworkType()	Type de la technologie radio employée (GPRS, EDGE, UMTS (3G)...).
String getSimCountryIso()	Code ISO du pays du fabricant de la carte SIM (fr, de ...)
String getSimOperatorName()	Le nom de l'opérateur de l'abonné (carte SIM)
String getSimOperator()	Identifiant de l'opérateur de l'abonné (carte SIM)
String getSimSerialNumber()	Numéro de série de la carte SIM.
int getSimState()	Etat de la carte SIM (voir plus loin)

Services de téléphonie

String getSubscriberId()	Identifiant de l'abonné
String getVoiceMailAlphaTag()	Nom donné pour la messagerie. Habituellement: "Messagerie vocale".
String getVoiceMailNumber()	Numéro de la messagerie vocale.
boolean isNetworkRoaming()	Vrai si l'appareil est en mode d'itinérance sur le réseau.
int getDataActivity()	Type de la connexion pour les données (voir plus loin)
int getDataState()	Etat de la connexion pour les données (voir plus loin)
int getCallState()	Etat de l'appel (voir plus loin)

Services de téléphonie

```
int phoneType=telephonyManager.getPhoneType();
```

Type	Commentaire
TelephonyManager.PHONE_TYPE_NONE	Pas d'appels vocaux
TelephonyManager.PHONE_TYPE_GSM	Norme la plus connue pour la téléphonie mobile (Global System for Mobile Communications).
TelephonyManager.PHONE_TYPE_CDMA	Système de codage des transmissions, utilisant la technique d'étalement de spectre (Code Division Multiple Access)
TelephonyManager.PHONE_TYPE_SIP	Session Initiation Protocol: Téléphonie par internet (Voice over IP: VoIP)

Services de téléphonie

```
int simState=telephonyManager.getSimState();
```

Etat de la carte SIM	Commentaire
TelephonyManager.SIM_STATE_UNKNOWN	Inconnu ou en cours de basculement entre 2 états.
TelephonyManager.SIM_STATE_ABSENT	Aucune carte SIM n'est présente dans l'appareil.
TelephonyManager.SIM_STATE_PIN_REQUIRED	La carte SIM est verrouillée et réclame le code personnel de l'abonné (Personal Identification Number).
TelephonyManager.SIM_STATE_PUK_REQUIRED	La carte SIM est bloquée et réclame le code de déblocage (PIN Unlock Key).
TelephonyManager.SIM_STATE_NETWORK_LOCKED	La carte SIM est verrouillée et réclame le code de déverrouillage.
TelephonyManager.SIM_STATE_READY	La carte SIM est prête et fonctionne.

Services de téléphonie

```
int dataActivity=telephonyManager.getDataActivity();
```

Type de connexion	Commentaire
TelephonyManager.DATA_ACTIVITY_NONE	Aucun trafic
TelephonyManager.DATA_ACTIVITY_IN	Données en réception
TelephonyManager.DATA_ACTIVITY_OUT	Données en émission
TelephonyManager.DATA_ACTIVITY_INOUT	Données à la fois en réception et en émission
TelephonyManager.DATA_ACTIVITY_DORMANT	Connexion active mais le lien a été rompu.

Services de téléphonie

```
int dataState=telephonyManager.getDataState();
```

Etat de la connexion	Commentaire
TelephonyManager.DATA_DISCONNECTED	Déconnecté
TelephonyManager.DATA_CONNECTING	En cours de connexion
TelephonyManager.DATA_CONNECTED	Connecté
TelephonyManager.DATA_SUSPENDED	Connexion active mais pas de transmission. La transmission est suspendue du fait de la réception d'un appel par exemple.

Services de téléphonie

```
int callState=telephonyManager.getCallState();
```

Etat de l'appel	Commentaire
TelephonyManager.CALL_STATE_IDLE	Aucun appel.
TelephonyManager.CALL_STATE_OFFHOOK	Un appel est en cours de traitement. L'utilisateur a décroché.
TelephonyManager.CALL_STATE_RINGING	Un appel arrive. Le téléphone sonne.

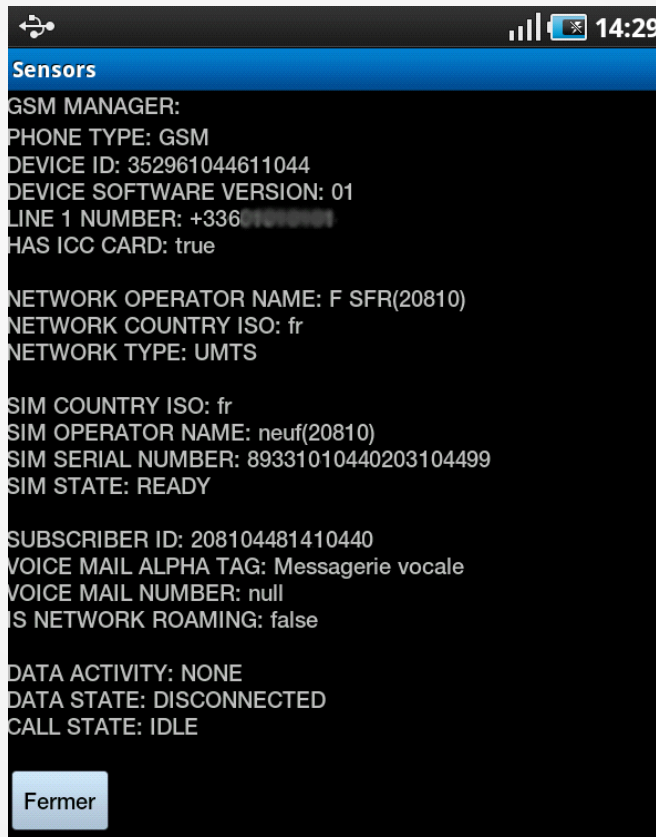
Services de téléphonie

Changement d'état du téléphone

Changement d'état du téléphone	Commentaire
PhoneStateListener. LISTEN_CALL_FORWARDING_INDICATOR	L'indicateur de renvoi d'appel a changé d'état.
PhoneStateListener.LISTEN_CALL_STATE	Etat de l'appel (voir précédemment)
PhoneStateListener.LISTEN_CELL_LOCATION	Changement de cellule
PhoneStateListener.LISTEN_DATA_ACTIVITY	Changement dans la direction du trafic de données (voir précédemment)
PhoneStateListener. LISTEN_DATA_CONNECTION_STATE	Changement d'état de connexion (voir précédemment)
PhoneStateListener. LISTEN_MESSAGE_WAITING_INDICATOR	L'indicateur de dépôt d'un message vocal sur la messagerie a changé d'état.
PhoneStateListener. LISTEN_SIGNAL_STRENGTHS	Force du signal reçu.

Services de téléphonie

Affichage des informations sur les services de téléphonie



The screenshot shows the 'Sensors' app interface on an Android phone. The status bar at the top displays signal strength, battery level, and the time 14:29. The app title 'Sensors' is at the top of the screen. Below it, the following telephony-related information is displayed:

```
GSM MANAGER:  
PHONE TYPE: GSM  
DEVICE ID: 352961044611044  
DEVICE SOFTWARE VERSION: 01  
LINE 1 NUMBER: +336 Fermer  
HAS ICC CARD: true  
  
NETWORK OPERATOR NAME: F SFR(20810)  
NETWORK COUNTRY ISO: fr  
NETWORK TYPE: UMTS  
  
SIM COUNTRY ISO: fr  
SIM OPERATOR NAME: neuf(20810)  
SIM SERIAL NUMBER: 89331010440203104499  
SIM STATE: READY  
  
SUBSCRIBER ID: 208104481410440  
VOICE MAIL ALPHA TAG: Messagerie vocale  
VOICE MAIL NUMBER: null  
IS NETWORK ROAMING: false  
  
DATA ACTIVITY: NONE  
DATA STATE: DISCONNECTED  
CALL STATE: IDLE
```

At the bottom of the screen, there is a button labeled 'Fermer'.

Services de téléphonie

Changement d'état du téléphone

```
PhoneStateListener psl=new PhoneStateListener() {
    @Override
    public void onSignalStrengthsChanged(SignalStrength signalStrength) {
        super.onSignalStrengthsChanged(signalStrength);
        // Modification du niveau de signal
        int strength=signalStrength.getGsmSignalStrength();
    }
};
```

Services de téléphonie

Changement d'état du téléphone

```
public void listen (PhoneStateListener listener, int events)
```

```
@Override
```

```
protected void onResume() {  
    telephonyManager.listen(psl,  
PhoneStateListener.LISTEN_SIGNAL_STRENGTHS);  
}
```

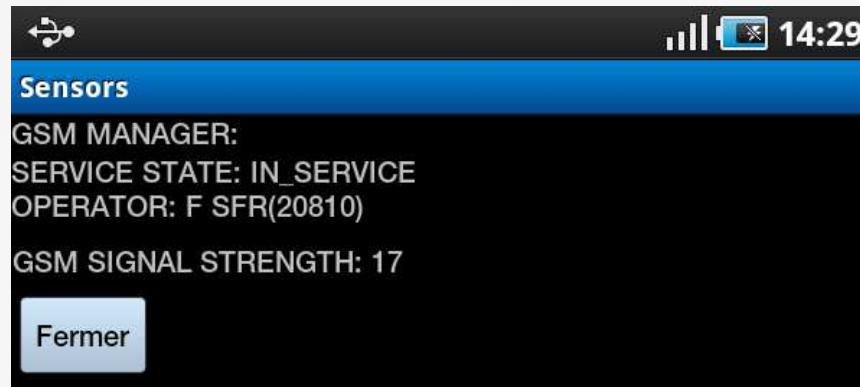
```
@Override
```

```
protected void onPause() {  
    super.onPause();  
    telephonyManager.listen(psl, PhoneStateListener.LISTEN_NONE);  
}
```

Services de téléphonie

Changement d'état du téléphone

Application affichant le niveau du signal reçu:



Gestionnaire de connexion

```
ConnectivityManager connectivityManager=  
(ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
```

Gestionnaire de connexion

Méthodes de la classe ConnectivityManager

Méthode	Commentaire
<code>NetworkInfo</code> <code>getActiveNetworkInfo()</code>	Donne les caractéristiques de l'interface réseau active.
<code>NetworkInfo[]</code> <code>getAllNetworkInfo()</code>	Donne la liste de toutes les interfaces réseau et leurs caractéristiques.
<code>NetworkInfo</code> <code>getNetworkInfo(int networkType)</code>	Donne les caractéristiques de l'interface réseau du type donné.
<code>static boolean</code> <code>isNetworkTypeValid(int networkType)</code>	Retourne vrai si le type d'interface réseau spécifié est valide.
<code>int</code> <code>getNetworkPreference()</code>	Donne le type d'interface réseau préféré.
<code>void</code> <code>setNetworkPreference(int preference)</code>	Modifie le type d'interface réseau préféré.
<code>boolean</code> <code>requestRouteToHost(int networkType, int hostAddress)</code>	S'assure qu'une route existe vers l'hôte ayant l'adresse spécifiée en empruntant l'interface réseau donnée.

Gestionnaire de connexion

Les types d'interface (`int networkType`)

Interface	Commentaire
<code>ConnectivityManager.TYPE_BLUETOOTH</code>	Bluetooth
<code>ConnectivityManager.TYPE_DUMMY</code>	Interface factice. Ne doit pas être utilisée.
<code>ConnectivityManager.TYPE_ETHERNET</code>	Ethernet
<code>ConnectivityManager.TYPE_MOBILE</code>	Mobile
<code>ConnectivityManager.TYPE_MOBILE_DUN</code>	Spécifiques à certaines connexions (Multimedia Messaging Service, haute priorité...). Normalement identique à Mobile mais peut être différent.
<code>ConnectivityManager.TYPE_MOBILE_HIPRI</code>	
<code>ConnectivityManager.TYPE_MOBILE_MMS</code>	
<code>ConnectivityManager.TYPE_MOBILE_SUPL</code>	
<code>ConnectivityManager.TYPE_WIFI</code>	Wifi
<code>ConnectivityManager.TYPE_WIMAX</code>	Wimax (Worldwide Interoperability for Microwave Access)

Gestionnaire de connexion

Informations obtenues par la classe [NetworkInfo](#)

Méthode	Commentaire
<code>NetworkInfo.State</code> getState()	Donne l'état de l'interface réseau
<code>NetworkInfo.DetailedState</code> getDetailedState()	Donne l'état détaillé de l'interface réseau
int getType()	Donne le type d'interface (wifi, mobile..)
int getSubtype()	Donne le sous-type de l'interface.
String getTypeName()	Donne le type d'interface (wifi, mobile..) sous la forme d'une chaîne de caractères.
String getSubtypeName()	Donne le sous-type de l'interface sous la forme d'une chaîne de caractères.
boolean isAvailable()	La connexion est-elle possible ? Est-on par exemple dans la zone de couverture
boolean isConnected()	Est-on connecté ?
boolean isConnectedOrConnecting()	Est-on connecté ou en cours de connexion ?

Gestionnaire de connexion

Informations obtenues par la classe [NetworkInfo](#)

boolean isFailover()	Indique que l'on tente de se connecter sur ce réseau car la connexion sur le précédent a échoué.
boolean isRoaming()	Indique si l'itinérance est activée. (changement de cellule ou de réseau tout en restant en communication).
String getExtraInfo()	Donne une information supplémentaire sur le réseau si disponible.
String getReason()	Si disponible, indique pourquoi la tentative de connexion précédente a échoué.

Gestionnaire de connexion

Etat que peut prendre l'interface réseau

Etat standard (<code>NetworkInfo.State</code>)	Etat détaillé (<code>NetworkInfo.DetailedState</code>)	Commentaire
CONNECTED	CONNECTED	Connecté
CONNECTING	CONNECTING	En cours de connexion
DISCONNECTED	DISCONNECTED	Déconnecté
DISCONNECTING	DISCONNECTING	En cours de déconnexion
SUSPENDED	SUSPENDED	Le trafic IP est suspendu.
	AUTHENTICATING	En cours d'authentification
	BLOCKED	L'accès au réseau est bloqué
	FAILED	La tentative de connexion a échoué
	IDLE	Prêt. En attente de connexion.
	OBTAINING_IPADDR	Attend d'un serveur DHCP que celui-ci fournisse une adresse IP.
	SCANNING	En cours de recherche de points d'accès.
UNKNOWN		Etat inconnu.

Gestionnaire de connexion

```
// Réseau préféré
int networkPreference= connectivityManager.getNetworkPreference();
// Réseau actif
NetworkInfo
activeNetWorkInfo=connectivityManager.getActiveNetworkInfo();
String activeNetworkName=activeNetWorkInfo.getTypeName();

// Caractéristiques de l'ensemble des interfaces réseau disponibles
NetworkInfo[] netWorkInfo=connectivityManager.getAllNetworkInfo();
for(NetworkInfo netwi:netWorkInfo) {
    String networkName=netwi.getTypeName();
    NetworkInfo.DetailedState nstate=netwi.getDetailedState();
    boolean isRoaming=netwi.isRoaming();
    ...
}
```

Gestionnaire de connexion

```
↳ WiFi 15:27
Sensors
WIFI MANAGER:
Network preference: WIFI
Active Network: WIFI
All Networks:
mobile
SUBTYPE NAME: UNKNOWN
DETAILED STATE: IDLE
AVAILABLE: FALSE
CONNECTED: FALSE
CONNECTED OR CONNECTING: FALSE
FAILOVER: FALSE
ROAMING: FALSE

WIFI
SUBTYPE NAME:
DETAILED STATE: CONNECTED
AVAILABLE: TRUE
CONNECTED: TRUE
CONNECTED OR CONNECTING: TRUE
FAILOVER: FALSE
ROAMING: FALSE

mobile_mms
SUBTYPE NAME: UNKNOWN
DETAILED STATE: IDLE
AVAILABLE: FALSE
CONNECTED: FALSE
CONNECTED OR CONNECTING: FALSE
FAILOVER: FALSE
ROAMING: FALSE

mobile_supl
SUBTYPE NAME: UNKNOWN
DETAILED STATE: IDLE
AVAILABLE: FALSE
CONNECTED: FALSE
CONNECTED OR CONNECTING: FALSE
FAILOVER: FALSE
ROAMING: FALSE
```

Gestionnaire de connexion

Changement d'état

ConnectivityManager.[CONNECTIVITY_ACTION](#)

```
BroadcastReceiver connectionReceiver = new BroadcastReceiver() {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        if(intent.getAction().equals(ConnectivityManager.CONNECTIVITY\_ACTION)  
        {  
            // L'état a changé  
        }  
    }  
};
```

Gestionnaire de connexion

Changement d'état

```
IntentFilter filterConnectivity = new  
IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION);
```

```
@Override  
protected void onResume() {  
    super.onResume();  
    registerReceiver(connectionReceiver, filterConnectivity);  
}
```

```
@Override  
protected void onPause() {  
    super.onPause();  
    unregisterReceiver(connectionReceiver);  
}
```