# Android NotificationListenerService

## Android NotificatiotenerService

Avendo passato ore a capire come implementare (e soprattutto far funzionare) un Servizio di tipo Notification Listener in Android, ho deciso di condividere con voi la soluzione.

Innanzitutto, a cosa ci serve?
Un NotificationListenerService, come suggerisce il nome, serve a “catturare” tutte le notifiche (push e non) in arrivo sul nostro device Android.
La questione non banale, è come farlo funzionare correttamente.
Il trucco, che troverete alla fine dell’articolo, è richiedere all’utente l’autorizzazione esplicita per la nostra app di accedere alle notifiche.
Cosa che normalmente andrebbe fatta da qualche recondita schermata negli abissi delle impostazioni di Android, ma che possiamo far comparire all’utente tramite tre semplicissime righe di codice.

Ma procediamo con ordine.

La prima cosa da fare è dichiarare il servizio e le nostre intenzioni all’interno del file AndoidManifest.xml nel TAG <application>.
(In questo esempio il nome del servizio è NotificationListener, ma potete chiamarlo come volete, purché eriditi da NotificationListenerService)

```
<service android:name=".NotificationListener" android:exported="false" android:label="@string/app_name" android:permission="android.permission.BIND_NOTIFICATION_LISTENER_SERVICE"> <intent-filter> <action
```

```
android:name="android.service.notification.NotificationListene
rService" /> </intent-filter>
</service>
```

Implementazione del servizo:

```
public class NotificationListener extends
NotificationListenerService { public NotificationListener() {
super(); } @Override public void onCreate() {
super.onCreate(); } @Override public void onNotificationPosted
(StatusBarNotification sbn) { if(sbn != null &&
sbn.getNotification() != null) { Log.i("notifcation",
sbn.getNotification().toString()); String title =
sbn.getNotification().extras.getCharSequence(Notification.EXTR
A_TITLE).toString(); String text =
sbn.getNotification().extras.getCharSequence(Notification.EXTR
A_TEXT).toString(); Log.i("notifcation", "title: " + title + "
text: " + text); // Qui il vostro codice } } @Override public
void onNotificationRemoved (StatusBarNotification sbn) {
Log.i("notifcation", "onNotificationRemoved"); }
}
```

Ora, da qualche parte nella nostra applicazione ci farà sicuramente comodo controllare se la nostra applicazione ha i permessi per la lettura delle notifiche in arrivo, e il seguente metodo fa al caso nostro:

```
private boolean checkNotificationListenerPermission() {
ComponentName cn = new ComponentName(this,
NotificationListener.class); String flat =
Settings.Secure.getString(this.getContentResolver(),
"enabled_notification_listeners"); final boolean enabled =
flat != null && flat.contains(cn.flattenToString()); return
enabled;
}
```

Ora, come promesso, non resta altro che presentare all'utente la schermata per autorizzare la nostra app, e questo può essere fatto semplicemente richiamando l'intent di sistema.

```
Intent intent = new Intent();
intent.setAction(Settings.ACTION_NOTIFICATION_LISTENER_SETTING
```

```
S);
startActivity(intent);
```

https://italiancoders.it/android-notificationlistenerservice/