

# Notification Detector

## Development

- Android Studio
- API 21 (V5.0 **minimum support**)

## Setting Notification Listener Service

### 1. Get the permission

- Write the follow in AndroidManifest.xml
- Change **android:name** and **android:label** to meet your project

```
<service
    android:name="[@subclass_name implements the service]
"
    android:label="[@service_name show for user]"
    android:permission="android.permission.BIND_NOTIFICATION_LISTENER_SERVICE">
    <intent-filter>
        <action android:name="android.service.notification.NotificationListenerService" />
    </intent-filter>
</service>
```

## 2. Extends NotificationListenerService class

- Create a class to extend NotificationListenerService
- Override two methods : **onNotificationPosted** ,  
**onNotificationRemoved**

```
public class NotificationListener extends NotificationListenerService {  
    @Override  
    public void onNotificationPosted(StatusBarNotification sbn) {  
        //called when get a notification  
    }  
    @Override  
    public void onNotificationRemoved(StatusBarNotification sbn) {  
        //called when remove a notification  
    }  
}
```

###Usage

### 1. Create a NotificationListenerService

See the above

### 2. Check the NotificationListener Access

1. Use the below code to check the access

```

public static String NOTIFICATION_ACCESS = "enabled_notification_listeners";

private boolean isNotificationAccessEnabled(){
    String access = Settings.Secure.getString(getContentResolver(), NOTIFICATION_ACCESS);
    String pkgName = getApplicationContext().getPackageName();
    return access.contains(pkgName);
}

```

2. If this method return false, open the access page

```

public static String NOTIFICATION_LISTENER_SETTINGS = "android.settings.ACTION_NOTIFICATION_LISTENER_SETTINGS";
Intent intent = new Intent(NOTIFICATION_LISTENER_SETTINGS);
startActivity(intent);

```

### 3. Handle the notification in onNotificationPosted

Notification will be handled at **onNotificationPosted** method

```

public void onNotificationPosted(StatusBarNotification sbn) {

    //get the notification info
    Notification mNotification = sbn.getNotification();
}

```

```
if(mNotification != null) {
```

```
    //create a bundle to save data
```

```
    Bundle data = new Bundle();
```

```
    data.putString("title", mNotification.extras.getString(Notification.EXTRA_TITLE));
```

```
    data.putCharSequence("text", mNotification.extras.getCharSequence(Notification.EXTRA_TEXT));
```

```
    data.putCharSequence("subtext", mNotification.extras.getCharSequence(Notification.EXTRA_SUB_TEXT));
```

```
    data.putString("pkgname", sbn.getPackageName());
```

```
    data.putString("key", sbn.getKey());
```

```
    data.putString("tag", sbn.getTag());
```

```
    //create a intent to pass data to activity
```

```
    //MainActivity.INTENT_ACTION_NOTIFICATION is static variable that use to filter the brocat, see next step
```

```
    Intent intent = new Intent(MainActivity.INTENT_ACTION_NOTIFICATION);
```

```
    intent.putExtras(data);
```

```
    //use BroadcastReceiver, see next step
```

```
    sendBroadcast(intent);
```

```
    //remove the notification at the top of screen
```

```
    this.cancelNotification(sbn.getKey());
```

```
}
```

```
}
```

## 4. Receive the data at activity

1. Create a subclass to extend BroadcastReceiver class in activity

```
private class mBroadcastReceiver extends BroadcastReceiver {  
    public Bundle notification;  
  
    @Override  
    public void onReceive(Context context, Intent intent) {  
  
        //handle the notification data  
        if (intent != null) {  
            notification = intent.getExtras();  
        }  
    }  
}
```

2. Register and unregister the BroadcastReceiver at **onResume** and **onPause**

```
private mBroadcastReceiver broadcastReceiver = new mBroadcastReceiver();  
  
@Override  
protected void onResume() {
```

```

super.onResume();

//register
if(broadcastReceiver == null)
    broadcastReceiver = new mBroadcastReceiver();

//IntentFilter can filter the broadcast
registerReceiver(broadcastReceiver, new IntentFilter(IN
TENT_ACTION_NOTIFICATION));
}

@Override
protected void onPause() {

//unregister
unregisterReceiver(broadcastReceiver);
super.onPause();
}

```

## Reference

- [Android Service](#)
- [NotificationListenerService](#)
- [Github: NotificationListener](#)