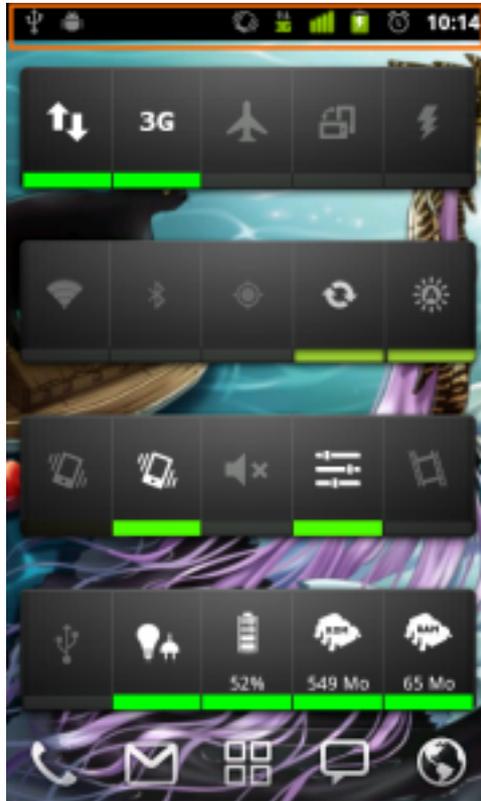


## Les notifications sous Android



par Nazim Benbourahla

Date de publication :

Dernière mise à jour :

Le but de ce tutoriel est d'apprendre à utiliser les notifications sous Android et de prévenir les utilisateurs de vos applications d'un événement particulier ou interagir avec eux.

1 - Qu'est qu'une notification ?.....	3
2 - Création de la vue principale.....	3
3 - Rajouter le Listener sur les boutons.....	4
4 - Création d'une notification.....	5
5 - Suppression d'une notification.....	6
6 - Conclusion.....	7
6 - Remerciements.....	7
7 - Liens.....	7

## 1 - Qu'est qu'une notification ?

Une notification est une indication qui s'affiche sur la barre qui se situe en haut d'un téléphone Android. Cette notification sert à prévenir un utilisateur de certains événements, comme la réception d'un message par exemple.

Nous allons créer un projet qui comprendra deux boutons :

- un pour créer une notification ;
- un autre pour en supprimer.

Commençons par créer un projet avec les données suivantes :

- **Nom du projet** : tuto\_notification ;
- **SDK** : 2.1 ;
- **Nom de l'application** : Tuto Notification ;
- **Nom du package** : com.tutos.android.notification ;
- **Activité** : TutoNotificationHomeActivity ;

## 2 - Création de la vue principale

Nous allons modifier le fichier "**main.xml**" afin d'avoir deux boutons :

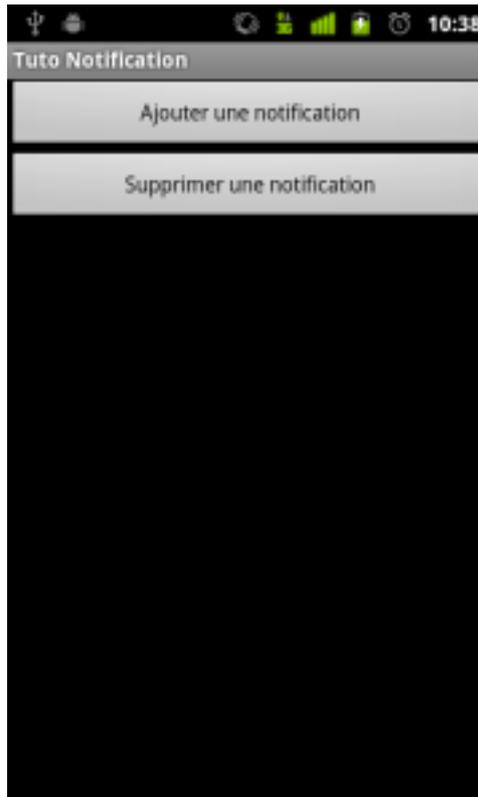
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/add_notification"
        android:id="@+id/add_notification"
        />

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/delete_notification"
        android:id="@+id/delete_notification"
        />
</LinearLayout>
```

Le fichier "**strings.xml**" pour rajouter les différentes strings pour le texte des boutons :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="add_notification">Ajouter une notification</string>
    <string name="delete_notification">Supprimer une notification</string>
    <string name="app_name">Tuto Notification</string>
</resources>
```

Voici le résultat que vous allez obtenir :



### 3 - Rajouter le Listener sur les boutons

Nous allons rajouter des "Listeners" sur les boutons pour gérer le clic. Ce qui donnera sur la classe "TutoNotificationHomeActivity" :

```

package com.tutos.android.notification;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class TutoNotificationHomeActivity extends Activity {
    private Button addNotificationBtn;
    private Button deleteNotificationBtn;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        addNotificationBtn = (Button) findViewById(R.id.add_notification);
        addNotificationBtn.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {
                Toast.makeText(getApplicationContext(), "Ajout d'une notification", Toast.LENGTH_SHORT).show();
            }
        });

        deleteNotificationBtn = (Button) findViewById(R.id.delete_notification);
        deleteNotificationBtn.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {

```

```

        Toast.makeText(getApplicationContext(), "Suppression d'une notification", Toast.LENGTH_SHORT).show();
    }
});
}
}

```

## 4 - Création d'une notification

Nous allons créer une méthode "**createNotification**" qu'on appellera au clic sur le bouton "**Ajouter une notification**". Voici à quoi ressemble cette méthode :

```

private final void createNotification() {
    //Récupération du notification Manager
    final NotificationManager notificationManager =
        (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);

    //Création de la notification avec spécification de l'icône de la notification et le texte qui apparaît à la cr
    final Notification notification = new Notification(R.drawable.notification, notificationTitle,
        System.currentTimeMillis());

    //Définition de la redirection au moment du clic sur la notification. Dans notre cas la notification redirige v
    final PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, new Intent(this,
        TutoNotificationHomeActivity.class), 0);

    //Récupération du titre et description de la notification
    final String notificationTitle = getResources().getString(R.string.notification_title);
    final String notificationDesc = getResources().getString(R.string.notification_desc);

    //Notification & Vibration
    notification.setLatestEventInfo(this, notificationTitle, notificationDesc, pendingIntent);
    notification.vibrate = new long[] {0,200,100,200,100,200};

    notificationManager.notify(NOTIFICATION_ID, notification);
}

```

Je vous mets l'icône que j'ai utilisée pour la notification [ici](#).

Il suffit maintenant d'appeler la méthode dans votre code, ce qui donnera :

```

addNotificationBtn = (Button) findViewById(R.id.add_notification);
addNotificationBtn.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View arg0) {
        createNotification();
    }
});

```

Il faut rajouter la permission "**Vibration**" dans l' **AndroidManifest.xml** :

```

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

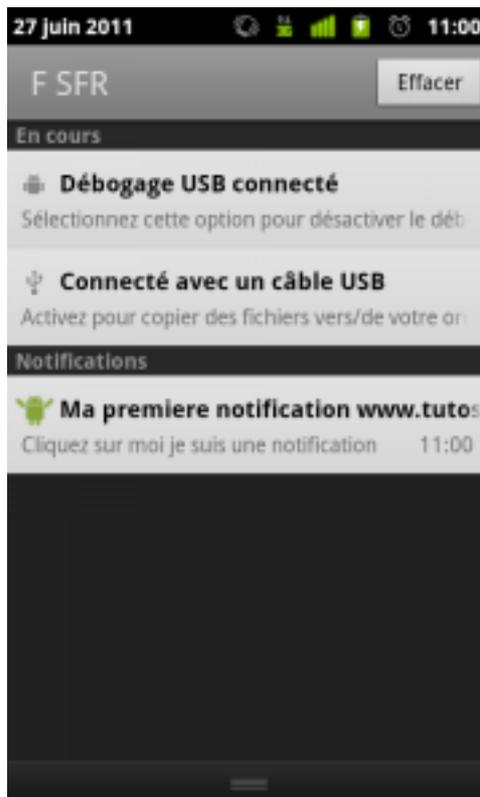
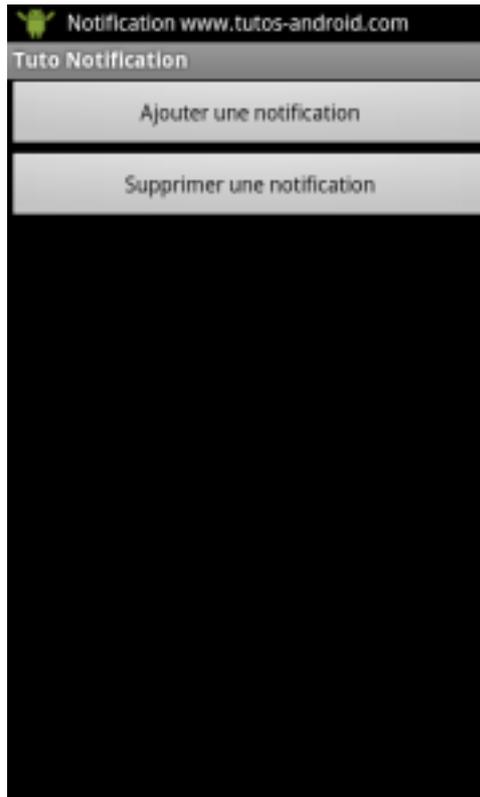
```

et les nouveaux textes dans le fichier "Strings.xml" :

```

<string name="notification">Notification www.tutos-android.com</string>
<string name="notification_title">Ma première notification www.tutos-android.com</string>
<string name="notification_desc">Cliquez sur moi je suis une notification</string>

```



## 5 - Suppression d'une notification

Nous allons créer une méthode " **deleteNotification** ", pour supprimer notre notification depuis l'application :

```
private void deleteNotification() {
```

```
final NotificationManager notificationManager =  
(NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);  
//la suppression de la notification se fait grâce à son ID  
notificationManager.cancel(NOTIFICATION_ID);  
}
```

Puis l'appeler dans le code Java au moment du clic :

```
deleteNotificationBtn = (Button) findViewById(R.id.delete_notification);  
deleteNotificationBtn.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View arg0) {  
        deleteNotification();  
    }  
});
```

Vous pouvez trouver [ici](#) le projet réalisé durant ce tuto.

## 6 - Conclusion

En espérant que ce tutoriel vous aura aidé à comprendre comment fonctionnent les notifications sous Android.

## 6 - Remerciements

Je tiens à remercier tout particulièrement [plegat](#) qui a mis ce tutoriel au format Developpez.com.  
Merci également à [ClaudeLELOUP](#) d'avoir pris le temps de le relire et de le corriger.

## 7 - Liens

**Tutoriel origine sur Tutos-Android**