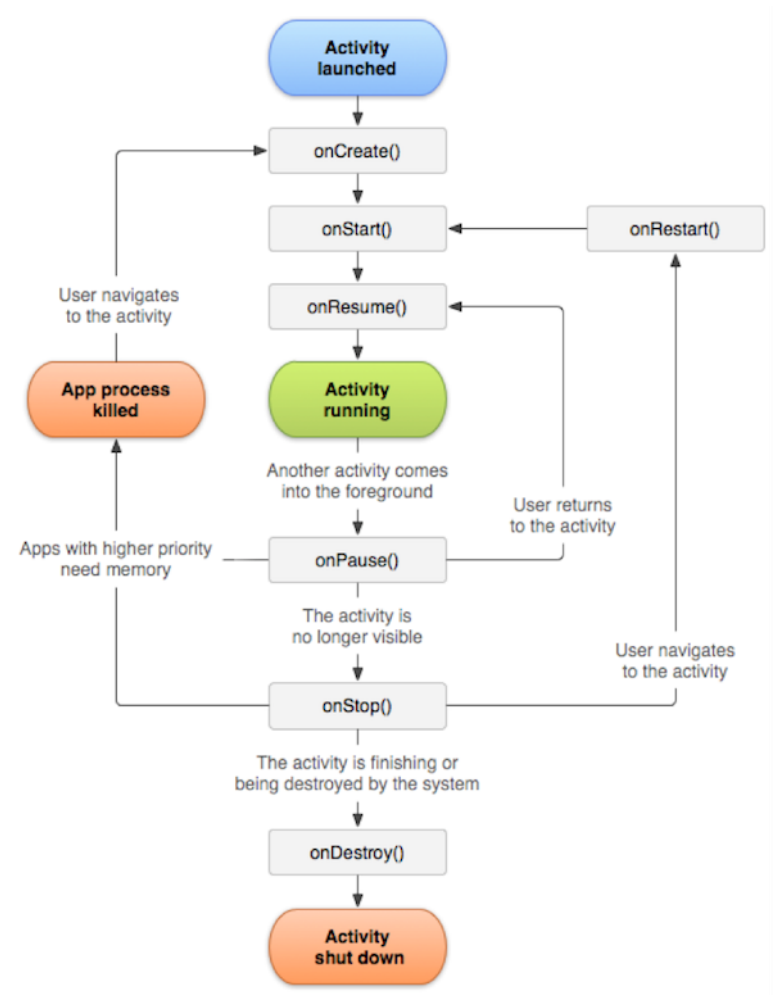
Cycle de vie des activités

Le cycle de vie d'une activité correspond aux différents états d'une activité lors de sa gestion par le système Android. Il est très important car il va vous permettre de suivre l'état de votre activité au fur et mesure de son existence dans le système Android.



Lorsque l'activité est lancée, le système Android appelle les méthodes onCreate, onStart et onResume.

Lorsque l'activité s'arrête, le système Android appelle les méthodes onPause, onStop et onDestroy.

Lorsque l'activité n'est plus au premier plan, mais qu'elle est tout de même affichée, onPause est appelée. Lorsque l'activité retourne au premier plan, onResume est appelée.

Lorsque l'activité n'est plus affichée, onResume et onStop sont appelées. Lorsque l'activité est de nouveau affichée, onRestart, onStart et onResume sont appelées.

Lorsque le système Android décide de tuer votre application, il appelle la méthode onSaveInstanceState qui vous permet de sauvegarder l'état de votre activité. Cet état sera passé en paramètre à la méthode onCreate, afin que vous puissiez restaurer l'état de l'activité.

Le système Android s'occupe lui-même de sauvegarder et restaurer l'état des vues des layouts, à condition que ces vues aient chacune un ID unique dans le layout.

Le changement de configuration comme le changement d'orientation de l'écran provoque un redémarrage de l'activité, vous devez donc sauvegarder et restaurer l'état dans ce cas

Toast

```
Toast.makeText(context, “Votre message..”, Toast.LENTGH_LONG).show;
```

Ou bien

```
Context context = getApplicationContext();
CharSequence text = "Hello toast!";
int duration = Toast.LENGTH_SHORT;

Toast toast = Toast.makeText(context, text, duration);
toast.show();
// on peut ajouter
toast.setGravity(Gravity.TOP|Gravity.LEFT, 0, 0);
```

AlertDialog

```
// 1. Instantiate an AlertDialog.Builder with its constructor
AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());

buildder.setMessage(“message”)
       .setTitle(“title”);
builder.setPositiveButton(“button ok”,new
DialogInterface.OnClickListener() {
           public void onClick(DialogInterface dialog, int id) {
               // User clicked OK button
           }
       });
builder.setNegativeButton(“button NO”,new
DialogInterface.OnClickListener() {
           public void onClick(DialogInterface dialog, int id) {
               // User clicked NO button
           }
       });
builder.setNeutralButton(“Cancel”,new DialogInterface.OnClickListener() {
           public void onClick(DialogInterface dialog, int id) {
               // User clicked Cancel button
           }
       });
// 3. Get the AlertDialog from create()
AlertDialog dialog = builder.create();
dialog.show();
```

Notification

Basic notification

```
NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(
Nom_Activite.this,NotificationChannel.DEFAULT_CHANNEL_ID)
        .setSmallIcon(R.drawable.notification_icon)
        .setContentTitle(textTitle)
        .setContentText(textContent)
        .setPriority(NotificationCompat.PRIORITY_DEFAULT);
```

Notification avec action(intent)

```
// Create an explicit intent for an Activity in your app
Intent intent = new Intent(this, AlertDetails.class);
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
PendingIntent pendingIntent = PendingIntent.getActivity(this,0,intent,0);

NotificationCompat.Builder mBuilder = new
NotificationCompat.Builder(this, NotificationChannel.DEFAULT_CHANNEL_ID)
        .setSmallIcon(R.drawable.notification_icon)
        .setContentTitle("My notification")
        .setContentText("Hello World!")
        .setPriority(NotificationCompat.PRIORITY_DEFAULT)
    // Set the intent that will fire when the user taps the notification
        .setContentIntent(pendingIntent)
        .setAutoCancel(true);

        //Vibration
         mBuilder.setVibrate(new long[] { 1000, 2000,1000,1000 });

        //son
          Uri alarmSound =
 RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        mBuilder.setSound(alarmSound);
```

Afficher la notification

```
NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(this);

// notificationId is a unique int for each notification that you must
```

```
define
notificationManager.notify(notificationId, mBuilder.build());
```

Ajout d'un écouteur de messagerie

Etape1 :

Ajouter les permissions suivantes au manifest :

```
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

Etape2 :

Créer une BroadCast Receiver avec un clic droite sur le package puis New et puis Other. Cela implique l'ajout de la balise **<receiver.../>** et la classe MyReceiver.

- Ajouter le filtre 'intent-filter' dans la balise receiver comme la suivante :

```
<receiver
    android:name=".MySMSReceiver"
    android:enabled="true"
    android:exported="true"
    >
    <intent-filter >
        <action android:name="android.provider.Telephony.SMS_RECEIVED"/>
    </intent-filter>
</receiver>
```

- Modifier le code de la méthode onReceive dans la classe MyReceiver

```
@Override
public void onReceive(Context context, Intent intent) {


// TODO: This method is called when the BroadcastReceiver is receiving an
// Intent broadcast.
String messageBody,phoneNumber;
if(intent.getAction().equals("android.provider.Telephony.SMS_RECEIVED"))
{
    Bundle bundle =intent.getExtras();
    if (bundle != null) {
            Object[] pdus = (Object[]) bundle.get("pdus");
            final SmsMessage[] messages = new SmsMessage[pdus.length];
            for (int i = 0; i < pdus.length; i++) {
                messages[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
            }
            if (messages.length > -1) {
                 messageBody = messages[0].getMessageBody();
                 phoneNumber = messages[0].getDisplayOriginatingAddress();

         Toast.makeText(context,
          "Message : "+messageBody+"Reçu de la part de;"+ phoneNumber,
           Toast.LENGTH_LONG )
         .show();

            }
        }
    }
   }
```