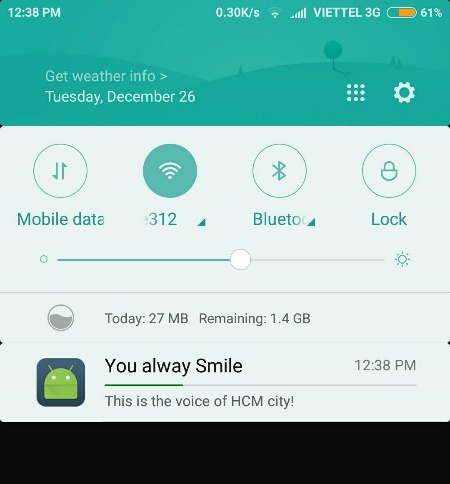
**Notification – Thông báo.**

****

public static final String EXTRA\_KEY\_NUMBER = "EXTRA KEY NUMBER";

public static final int REQUEST\_CODE\_INTENT = 1;

private static final int MY\_NOTIFY\_ID = 1; //Mã quản lý Notification

/\*\* Hiển thị Notification \*/

private void showNotification() {

//Quản lý Notification

final NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION\_SERVICE);

Notification.Builder mNotifiBuilder = new Notification.Builder(this);

//--- Tạo Nội dung Notification: Phải có Ảnh + Chữ ---

mNotifiBuilder.setContentTitle("Day la tieng noi")

.setContentText("Nội dung text bên trong tiêu đề")

.setSmallIcon(android.R.drawable.ic\_lock\_idle\_low\_battery);

//--- Tạo Intent gọi khi Click Notification ---

Intent intent = new Intent(this, Detail1Activity.class);

intent.putExtra(EXTRA\_KEY\_NUMBER, 12121991); //Gởi 1 số đến Activity hiện khi Notification click<

if (Build.VERSION.SDK\_INT < Build.VERSION\_CODES.JELLY\_BEAN) {

// api < 16

// Intent( ý định) Pending (chờ xử lý): Intent chạy suốt Ứng dụng

PendingIntent resultPendingIntent = PendingIntent

.getActivity(this, REQUEST\_CODE\_INTENT, intent, PendingIntent.FLAG\_UPDATE\_CURRENT);

mNotifiBuilder.setContentIntent(resultPendingIntent);

//--- Hiển thị Notification ---

manager.notify(MY\_NOTIFY\_ID, mNotifiBuilder.getNotification()); // hiển thị, cập nhật Notification

} else {

// api >= 16

// ---Tạo hàng chờ Activity trở về---

TaskStackBuilder stackBuilder = TaskStackBuilder.create(this);

stackBuilder.addParentStack(Detail1Activity.class); // thêm vào ngăn chứa

stackBuilder.addNextIntent(intent); // thêm Intent trên cùng ngăn chứa, Thêm được nhiều Intent

PendingIntent resultPendingIntent = stackBuilder

.getPendingIntent(0, PendingIntent.FLAG\_UPDATE\_CURRENT); //Tạo PendingIntent từ StackBuilder

// Đặt PendingIntent vào NotifiBuilder

mNotifiBuilder.setContentIntent(resultPendingIntent);

//--- Hiển thị Notification ----

manager.notify(MY\_NOTIFY\_ID, mNotifiBuilder.build()); // hiển thị, cập nhật Notification

}

}

// sử dụng: NotificationCompat, không cần chia phiên bản android

//Phải thêm thư viện: android.support.v4+

void runNotifyCompat(View v){

//---Cài đặt giao diện ---

NotificationCompat.Builder notifiBuilder = new NotificationCompat.Builder(getBaseContext())

.setContentTitle("You alway Smile") //phai co

.setContentText("This is the voice of HCM city!")

.setSmallIcon(R.mipmap.ic\_abc) //phai co

.setLargeIcon(BitmapFactory.decodeResource(getResources(), R.mipmap.ic\_abc))

.setAutoCancel(true) //Ẩn khi User click

.setOngoing(true) //Không cho User hủy

.setPriority(Notification.PRIORITY\_MAX) //Độ ưu tiên cao nhất

.setProgress(100, 25, false) //Hiện Progress với 25%

//.setProgress(0, 0, false) //Ẩn Progress

.setColor(0x008000);

//--- Thêm Intent khi click chuyển đến Activity ---

Intent intent = new Intent(this, Detail1Activity.class);

intent.putExtra(EXTRA\_KEY\_NUMBER, 12121991);

//Tạo hàng chờ Activity trở về

TaskStackBuilder stackBuilder = TaskStackBuilder.create(this);

stackBuilder.addParentStack(Detail1Activity.class); // thêm vào ngăn chứa

stackBuilder.addNextIntent(intent); // thêm Intent trên cùng ngăn chứa, Thêm được nhiều Intent

// Gets a PendingIntent containing the entire back stack

PendingIntent resultPendingIntent = stackBuilder

.getPendingIntent(0, PendingIntent.FLAG\_UPDATE\_CURRENT);

// ấn vào gọi Intent chuyển đến Activity

notifiBuilder.setContentIntent(resultPendingIntent);

//--- Hiển thị Notification ---

final NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION\_SERVICE);

manager.notify(MY\_NOTIFY\_ID, notifiBuilder.build());

// lấy Dữ liệu từ Intent của Notification từ Activity khác

//int value = getIntent().getIntExtra(MainActivity.EXTRA\_KEY\_NUMBER, 0);

}

//--- Close Notification ---

public void cancelNotifiCompat(View v){

final NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION\_SERVICE);

manager.cancel(MY\_NOTIFY\_ID);

}

//--- Chạy Service Hiện Notification ---

public void runServiceNotification(View v) {

Intent i = new Intent(getBaseContext(), ProgressService.class);

startService(i);

}

**Service giao tiếp Notification**

public class ProgressService extends Service

{

private static final int PROGRESS\_NOTIFY\_ID = 2;

public boolean running = false; //kiểm tra đã chạy chưa

private NotificationCompat.Builder mNotifiBuilder;

@Override

public void onCreate()

{

super.onCreate();

mNotifiBuilder = new NotificationCompat.Builder(getApplicationContext());

}

@Override

public int onStartCommand(Intent intent, int flags, int startId) {

super.onStartCommand(intent, flags, startId);

startServiceWithNotification();

return START\_STICKY; //Tự hồi sinh

}

void startServiceWithNotification() {

//Check Running

if (running) return;

running = true;

final int minValue = 0;

final int maxValue = 1000;

//--- Hiện giao diện khi Running ---

mNotifiBuilder.setSmallIcon(android.R.drawable.btn\_default)

.setContentTitle("Running...")

.setOngoing(true); // ko cho user huy

mNotifiBuilder.build().flags = Notification.FLAG\_NO\_CLEAR; // NO\_CLEAR makes the notification stay when the user performs a "delete all" command

startForeground(PROGRESS\_NOTIFY\_ID, mNotifiBuilder.build()); //Chay notifi duoi nen

final NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION\_SERVICE);

// Hiển thị process

new Thread(new Runnable() {

@Override

public void run() {

try {

//---Chay & Hien Progress dang chay ---

for (int i = minValue; i <= maxValue; i += 100) {

mNotifiBuilder.setProgress(maxValue, i, false);

manager.notify(PROGRESS\_NOTIFY\_ID, mNotifiBuilder.build());

Thread.sleep(2 \* 1000); // dừng lại 2s

}

//--- Hien thi giao dien Finish ---

stopForeground(true); //Hủy Notification, để đặt lại lệnh setOngoing

mNotifiBuilder.setContentTitle("Finish")

.setProgress(0,0,false) //Hide progress

.setOngoing(false); //Cho user hủy

manager.notify(PROGRESS\_NOTIFY\_ID, mNotifiBuilder.build()); //Hiện lại Notifi, user được hủy

} catch (Exception ex) {

ex.printStackTrace();

}

}

}).start();

}

@Override

public IBinder onBind(Intent p1)

{

return null;

}

@Override

public void onDestroy()

{

//khi ung dung bi close, crash

running = false;

Toast.makeText(getApplicationContext(),

"Stop Running! by Sys", Toast.LENGTH\_SHORT).show();

super.onDestroy();

}

//--- Dừng Service ---

public void stopService() {

stopForeground(true);

stopSelf();

running = false;

Toast.makeText(getApplicationContext(),

"Stop Running! by user", Toast.LENGTH\_SHORT).show();

}

}