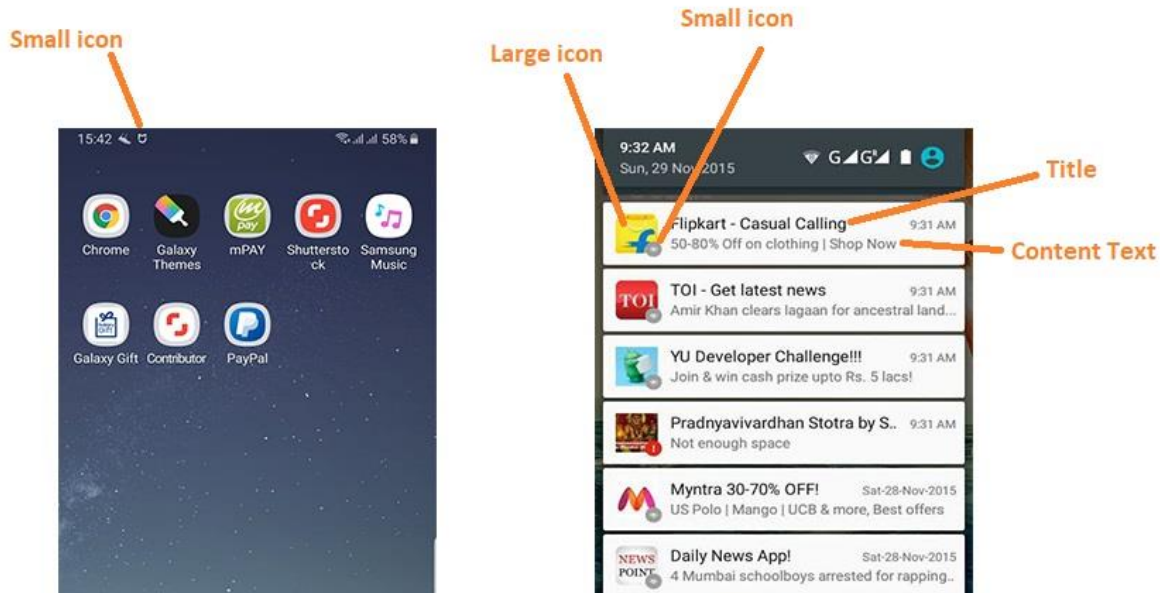


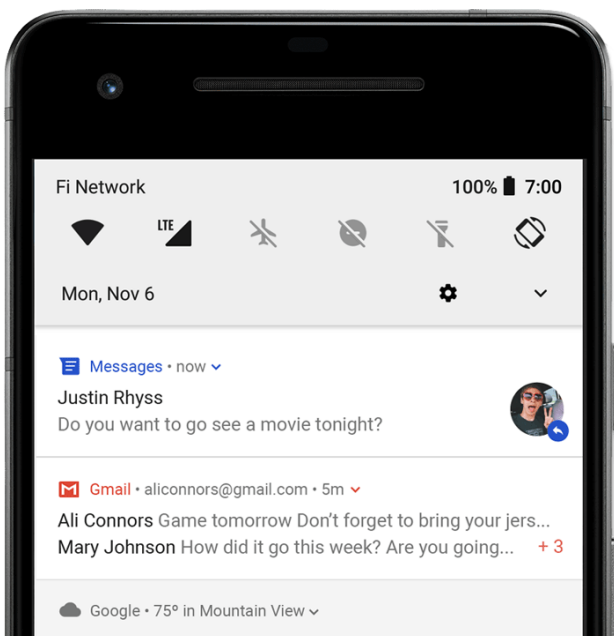


Notification การแจ้งเตือนแบบพื้นฐาน

มาดูหน้าตา Notification ก่อน หน้าตาของ notification ก็จะต่างไปตาม andriodเวอร์ชัน รูปข้างล่างคือของ api 21 – 23



พอเป็น api24 ก็จะเปลี่ยนไป สังเกตชัดๆเลย คือย้าย ไอคอนมาด้านขวา แต่โดยหลักแล้วก็คล้ายๆของเดิม

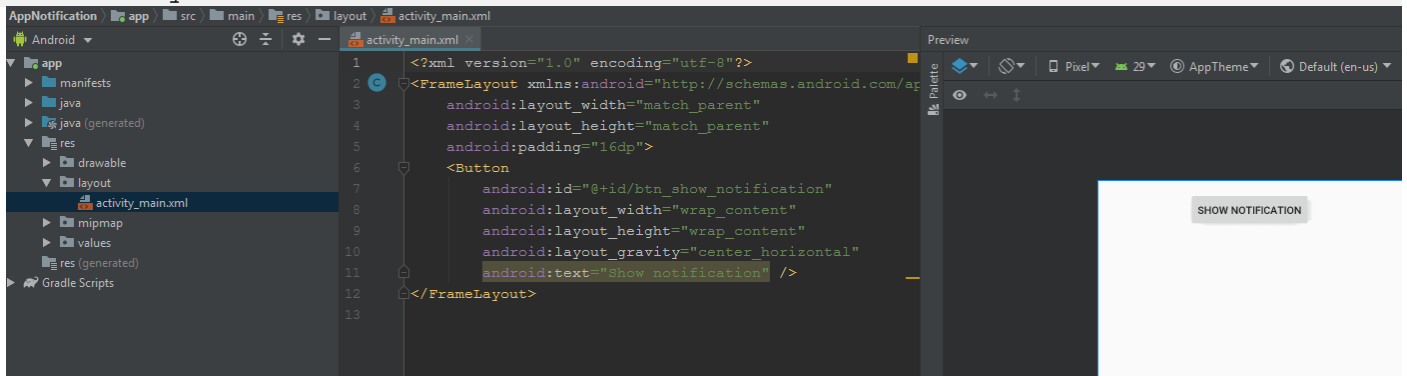


เราจะใช้ NotificationCompat ซึ่งอยู่ใน support library v4

เริ่มต้นออกแบบ layout ก่อน โดยจะมีปุ่มสักปุ่มหนึ่ง กดแล้วจะให้แจ้งเตือนขึ้นมา

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">
    <Button
        android:id="@+id/btn_show_notification"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Show notification" />
</FrameLayout>
```



มาที่ MainActivity.java

ให้กำหนด findViewById ของปุ่มให้เรียบร้อย พร้อมกำหนด onClick ของปุ่ม

ประกาศตัวแปร

```
Button btnShowNotification;
```

```
public class MainActivity extends AppCompatActivity {

    Button btnShowNotification;
```

เพิ่ม Code ไว้ใน onCreate

```
btnShowNotification = findViewById(R.id.btn_show_notification);
btnShowNotification.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        showNotification();
    }
});
```



```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    btnShowNotification = findViewById(R.id.btn_show_notification);
    btnShowNotification.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            showNotification();
        }
    });
}
```

จากนั้นให้เราทำการสร้างฟังก์ชัน `showNotification()` ขึ้นมา

```
private void showNotification() {
}
```

เราจะ implement ส่วนของการแสดง notification กัน โดยหลักๆ ก็ต้อง set พวก icon แล้วก็เนื้อหา

ไอคอนเล็ก สามารถใช้จาก resource ได้ โดยมันคือ ไอคอนที่แสดงบน notification bar จะถูกแปลงเป็นสีขาวอัตโนมัติ

ส่วนไอคอนใหญ่ต้องใช้ Bitmap ก็สามารถแปลง drawable resource มาเป็น Bitmap ได้

```
Bitmap bitmap = BitmapFactory.decodeResource(this.getResources(),
    R.mipmap.ic_launcher);
```

ทำการกำหนดค่าให้กับ Builder ซึ่งตัวนี้ คือตัวสำคัญของเรื่องนี้เลย

เสร็จแล้วก็เรียกคำสั่ง `.build()` จะได้ object ของ Notification

```
NotificationCompat.Builder notification = new
NotificationCompat.Builder(this)
    .setSmallIcon(R.mipmap.ic_launcher)
    .setLargeIcon(bitmap)
    .setContentTitle("Hello World")
    .setContentText("Good night.");
```

ส่ง Notification

ตั้งค่าไว้แล้วก็ต้องทำการส่ง notification ออกไป โดยจะทำโดย NotificationManager

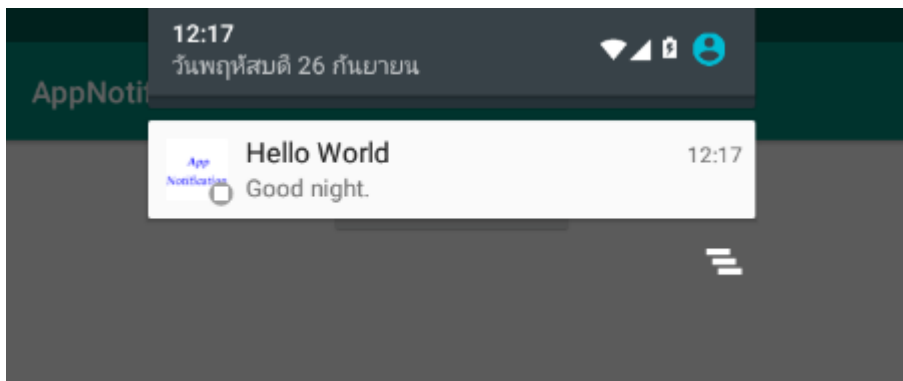
แล้วเรียก `notify()` ก็จะส่งออกไปแล้ว โพล์ให้ user เห็น



```
NotificationManager notificationManager =  
    (NotificationManager)  
this.getSystemService(this.NOTIFICATION_SERVICE);  
notificationManager.notify(1000, notification.build());
```

```
private void showNotification() {  
    Bitmap bitmap = BitmapFactory.decodeResource(this.getResources(),  
        R.mipmap.ic_launcher);  
  
    NotificationCompat.Builder notification = new NotificationCompat.Builder(context: this)  
        .setSmallIcon(R.mipmap.ic_launcher)  
        .setLargeIcon(bitmap)  
        .setContentTitle("Hello World")  
        .setContentText("Good night.");  
  
    NotificationManager notificationManager =  
        (NotificationManager) this.getSystemService(this.NOTIFICATION_SERVICE);  
    notificationManager.notify(id: 1000, notification.build());  
}
```

กดปุ่มก็จะมี Notification โผล่ออกมา แต่จะยังกดไม่ได้ เพราะเรายังไม่ได้เพิ่ม Action ให้มันนั่นเอง





เปิด Activity จาก notification

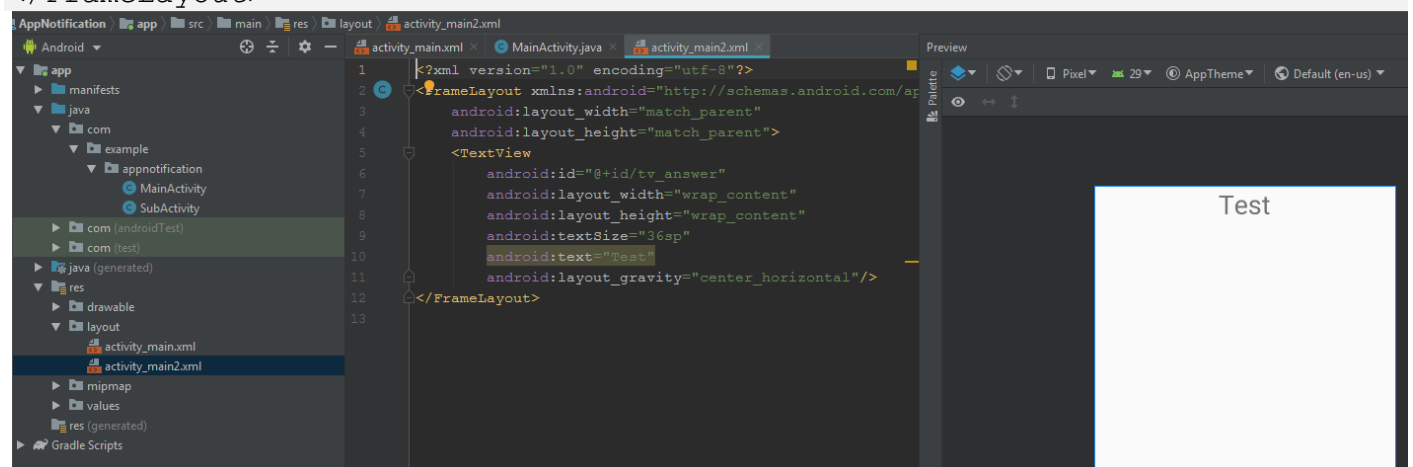
ต่อมาก่อนจะสร้าง action มาสร้างอีกหน้ารอกันก่อน สิ่งที่จะทำคือ จะให้กดที่ Notification แล้วให้มาหน้าที่ 2 นี้

พร้อมส่งค่ามาด้วย แล้วเราก็เอามาแสดง

สร้าง activity ใหม่ให้เรียบร้อย

activity_main2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/tv_answer"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="36sp"
        android:text="Test"
        android:layout_gravity="center_horizontal"/>
</FrameLayout>
```



สร้างไฟล์ SubActivity.java ขึ้นมา

SubActivity.java

```
public class SubActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        TextView tv = findViewById(R.id.tv_answer);
        Bundle bundle = getIntent().getExtras();
        if(bundle != null){
            tv.setText(bundle.getString("MESSAGE"));
        }
    }
}
```

```

}
}
}

AppNotification / app / src / main / java / com / example / appnotification / SubActivity
Android ▾
  app
    manifests
    java
      com
        example
          appnotification
            MainActivity
            SubActivity
        com (androidTest)
        com (test)
      java (generated)
      res
        drawable
        layout
          activity_main.xml
          activity_main2.xml
        mipmap
        values
        res (generated)
    Gradle Scripts
  activity_main.xml x MainActivity.java x activity_main2.xml x SubActivity.java x
1 package com.example.appnotification;
2
3 import ...
7
8 public class SubActivity extends AppCompatActivity {
9
10 @Override
11 protected void onCreate(Bundle savedInstanceState) {
12     super.onCreate(savedInstanceState);
13     setContentView(R.layout.activity_main2);
14
15     TextView tv = findViewById(R.id.tv_answer);
16     Bundle bundle = getIntent().getExtras();
17     if(bundle != null){
18         tv.setText(bundle.getString(key: "MESSAGE"));
19     }
20 }
21
22

```

เมื่อสร้างไฟล์ SubActivity.java แล้ว อย่าลืมประกาศ activity ใน AndroidManifest.xml ด้วย

```
<activity android:name=".SubActivity"></activity>
```

```

AppNotification / app / src / main / AndroidManifest.xml
Android ▾
  app
    manifests
      AndroidManifest.xml
    java
    java (generated)
    res
    res (generated)
  Gradle Scripts
  AndroidManifest.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.appnotification">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="AppNotification"
9         android:supportRtl="true"
10        android:theme="@style/AppTheme">
11
12        <activity android:name=".MainActivity">
13            <intent-filter>
14                <action android:name="android.intent.action.MAIN" />
15
16                <category android:name="android.intent.category.LAUNCHER" />
17            </intent-filter>
18        </activity>
19        <activity android:name=".SubActivity"></activity>
20    </application>
21 </manifest>

```

กลับมาที่ MainActivity.java

เราจะใช้คำสั่ง เพิ่มเข้าไป

```
.setContentIntent(getPendingIntent(this, "Hello World!!"))
```

PendingIntent คือ intent ที่แอปอื่นสามารถเอาไปใช้ได้

โดยการทำ pendingIntent ในกรณีนี้ ก็มีขั้นตอนนิดหน่อย คือ ต้องไปสร้าง TaskStackBuilder ก่อน แล้ว add Intent ที่เราต้องการเปิดลงไป จากนั้นค่อยดึงมันออกมา ก็จะได้ PendingIntent ละ



โดยขั้นตอนนี้ เราจะ put ค่าที่เตรียมจะส่งไปด้วยเลยกับ Intent ในที่นี้คือ message

```
private PendingIntent getPendingIntent(Context context, String message) {
    Intent intent = new Intent(context, SubActivity.class);
    intent.putExtra("MESSAGE", message);
    TaskStackBuilder stackBuilder = TaskStackBuilder.create(context);
    stackBuilder.addParentStack(SubActivity.class);
    stackBuilder.addNextIntent(intent);
    return stackBuilder.getPendingIntent(0,
    PendingIntent.FLAG_UPDATE_CURRENT);
}
```

```
private void showNotification() {
    Bitmap bitmap = BitmapFactory.decodeResource(this.getResources(),
        R.mipmap.ic_launcher);

    NotificationCompat.Builder notification = new NotificationCompat.Builder(context: this)
        .setSmallIcon(R.mipmap.ic_launcher)
        .setLargeIcon(bitmap)
        .setContentIntent(getPendingIntent(context: this, message: "Hello World!!"))
        .setAutoCancel(true)
        .setContentTitle("Hello World")
        .setContentText("Good night.");
    NotificationManager notificationManager =
        (NotificationManager) this.getSystemService(this.NOTIFICATION_SERVICE);
    notificationManager.notify(id: 1000, notification.build());
}

private PendingIntent getPendingIntent(Context context, String message) {
    Intent intent = new Intent(context, SubActivity.class);
    intent.putExtra(name: "MESSAGE", message);
    TaskStackBuilder stackBuilder = TaskStackBuilder.create(context);
    stackBuilder.addParentStack(SubActivity.class);
    stackBuilder.addNextIntent(intent);
    return stackBuilder.getPendingIntent(requestCode: 0, PendingIntent.FLAG_UPDATE_CURRENT);
}
```

รวมทั้งหมด ตั้งแต่สร้าง Notification แล้ว set ค่าให้มัน เพิ่ม action แล้วส่งออกไป

มาเขียนในส่วนของการรับค่าที่ส่งมาใน หน้าที 2

ก็รับค่าจาก Intent แล้วใส่ให้แสดงใน textView

ลองกดบน Notification ดู ก็จะเด้งไป activity นั้น

AppNotification

Hello World!!

เปิด Browser จาก notification

บางกรณีก็อาจจะอยากให้เกิด เว็บบแทนการเปิด activity ใหม่ ก็สามารทำได้เช่นกัน

เพียงเปลี่ยน pendingIntent นั้นเอง ไปใช้ Uri แทน

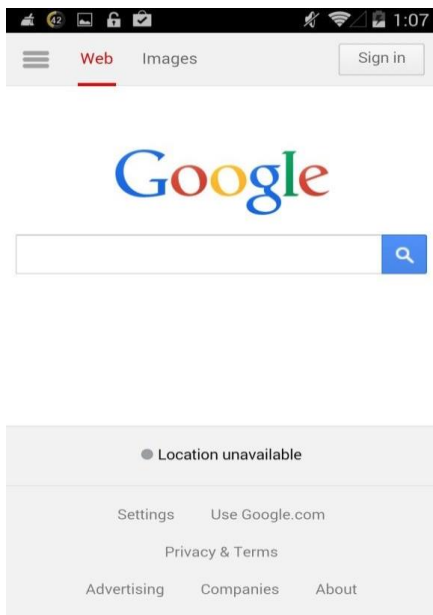
```
private PendingIntent getPendingIntentURL() {  
    String url = "https://onlinemarket101.com";  
    Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));  
    return PendingIntent.getActivity(MainActivity.this, 0, intent, 0);  
}
```

แล้วใส่ใน setContentIntent ลงไป

```
.setContentIntent(getPendingIntentURL())
```

```
private void showNotification() {  
    Bitmap bitmap = BitmapFactory.decodeResource(this.getResources(),  
        R.mipmap.ic_launcher);  
  
    NotificationCompat.Builder notification = new NotificationCompat.Builder(context: this)  
        .setSmallIcon(R.mipmap.ic_launcher)  
        .setLargeIcon(bitmap)  
        .setContentIntent(getPendingIntentURL())  
        .setAutoCancel(true)  
        .setContentTitle("Hello World")  
        .setContentText("Good night.");  
    NotificationManager notificationManager =  
        (NotificationManager) this.getSystemService(this.NOTIFICATION_SERVICE);  
    notificationManager.notify(id: 1000, notification.build());  
}  
  
private PendingIntent getPendingIntentURL() {  
    String url = "https://onlinemarket101.com";  
    Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));  
    return PendingIntent.getActivity(context: MainActivity.this, requestCode: 0, intent, flags: 0);  
}
```

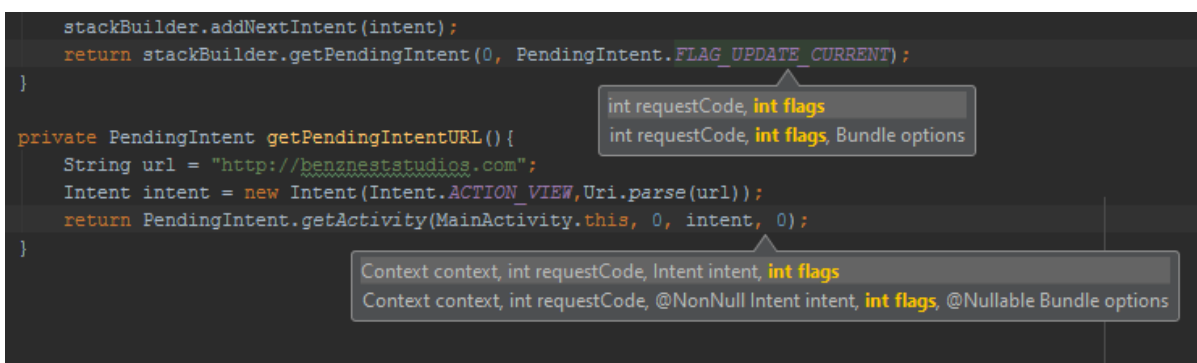
ลองกด Notification ก็จะไปเปิดเว็บทันที ตาม url ที่ใส่ไว้



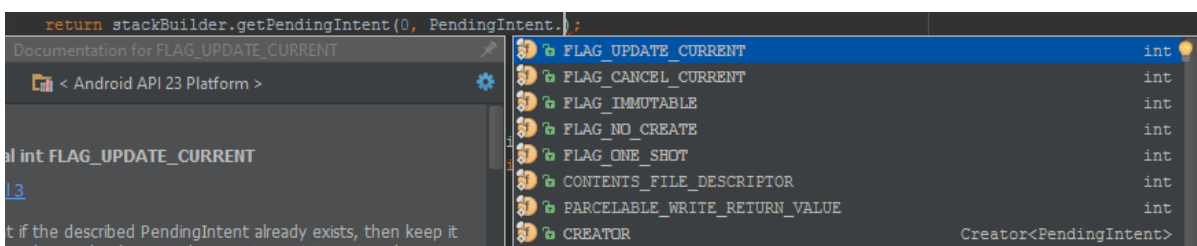
มารู้จัก Flags กันเถอะ

จะเห็นพารามิเตอร์หนึ่งที่น่าสนใจ ของ `getPendingIntent` , `getActivity` มันคือ `Flags`

หน้าที่ของมันก็คือกำหนด behavior ของ `pendingIntent` นั้นเอง



ลองกดให้มัน suggest ดู ว่ามี flags อะไรบ้าง โดยเอาเมาส์ไปจิ้มตรงนั้นแล้วกด (Ctrl + space bar)



`FLAG_ONE_SHOT` คือ ใช้ได้ครั้งเดียว ถ้าเรียกซ้ำมันจะไม่ทำงาน

`FLAG_NO_CREATE` คือ ไปเช็คก่อนว่า มีอยู่มั้ยถ้า มีอยู่แล้วมันจะไม่สร้างใหม่ พร้อมกับ return null

`FLAG_CANCEL_CURRENT` คือ อันก่อนหน้า ถ้ามีอยู่แล้วจะถูกยกเลิก แล้วสร้างอันใหม่

`FLAG_UPDATE_CURRENT` คือ ถ้ามีอยู่แล้ว จะทำการไปอัปเดต

ปุ่ม Action ใน notification

ใน notification นอกจากจะให้ user กดโดยตรงแล้ว เราสามารถเพิ่มปุ่ม action อื่นๆได้อีกด้วย เช่น แชร่เลย ตอบกลับทันที

โดยฟีเจอร์นี้รองรับเฉพาะ api 20 ขึ้นไป

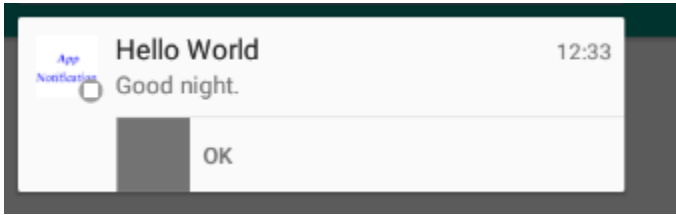
วิธีการคือ ใส่เพิ่มใน builder เช่นเคยโดยใช้คำสั่ง addAction

```
.addAction(action);
```

การสร้าง Action สามารถกำหนด icon ได้ และกำหนดคำที่แสดง

```
NotificationCompat.Action action =  
    new NotificationCompat.Action.Builder(R.mipmap.ic_launcher, "OK",  
getPendingIntent(this, "Hello World"))  
    .build();
```

```
private void showNotification() {  
    Bitmap bitmap = BitmapFactory.decodeResource(this.getResources(),  
        R.mipmap.ic_launcher);  
  
    NotificationCompat.Action action =  
        new NotificationCompat.Action.Builder(R.mipmap.ic_launcher, title: "OK", getPendingIntent(context: this, me  
        .build();  
  
    NotificationCompat.Builder notification = new NotificationCompat.Builder(context: this)  
        .setSmallIcon(R.mipmap.ic_launcher)  
        .setLargeIcon(bitmap)  
        .addAction(action)  
        .setAutoCancel(true)  
        .setContentTitle("Hello World")  
        .setContentText("Good night.");  
    NotificationManager notificationManager =  
        (NotificationManager) this.getSystemService(this.NOTIFICATION_SERVICE);  
    notificationManager.notify(id: 1000, notification.build());  
}
```



การลบ Notification

ในความเป็นจริง เมื่อผู้ใช้กดดูแอปจาก notification แล้ว เราก็ควรลบมันออกจาก Notification center ซะ ด้วย user จะรำคาญ

ถ้าตั้ง autoCancel ใน builder พอผู้ใช้กด มันจะลบให้อัตโนมัติ

แต่บางครั้งอาจจะมีหลาย notification ที่ต้องการลบ

คำสั่งลบทั้งหมด สามารถใช้ `cancelAll()` ได้เลย

```
.setAutoCancel(true)
```

หรือจะลบเฉพาะ notification ก็ได้เช่นกัน โดยใช้ id

```
NotificationManager notificationManager =
    (NotificationManager)
    context.getSystemService(Context.NOTIFICATION_SERVICE);
notificationManager.cancelAll();
```

หรือจะลบเฉพาะ notification ก็ได้เช่นกัน โดยใช้ id

```
notificationManager.cancel(id);
```

โดย id ก็มาจากตอนที่เราใส่ไปตอน notify นั่นเอง

```
NotificationManager notificationManager =
    (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
notificationManager.notify(1000, notification);
}
```

`int id, Notification notification`

การกำหนดให้ป้องกันการลบ

กรณีที่ต้องการให้ Notification นั้นคงอยู่ ลบไม่ได้ เราสามารถกำหนด Flag ให้ได้ แม้ว่าจะ Clear ก็จะไม่ถูกลบ

แต่ถ้าผู้ใช้ลบแอป ก็จะหายไป

```
NotificationCompat.Builder notificationBuilder = new NotificationCompat.Builder(context)
    .setSmallIcon(R.mipmap.ic_launcher)
    .setContentTitle(title)
    .setContentText(body)
    .setAutoCancel(false)
    .setContentIntent(pendingIntent);

Notification notification = notificationBuilder.build();
notification.flags |= Notification.FLAG_NO_CLEAR | Notification.FLAG_ONGOING_EVENT;

NotificationManager notificationManager = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
notificationManager.notify(100, notification);
```

