

# **Notification**

Mobile Software  
2019 Fall

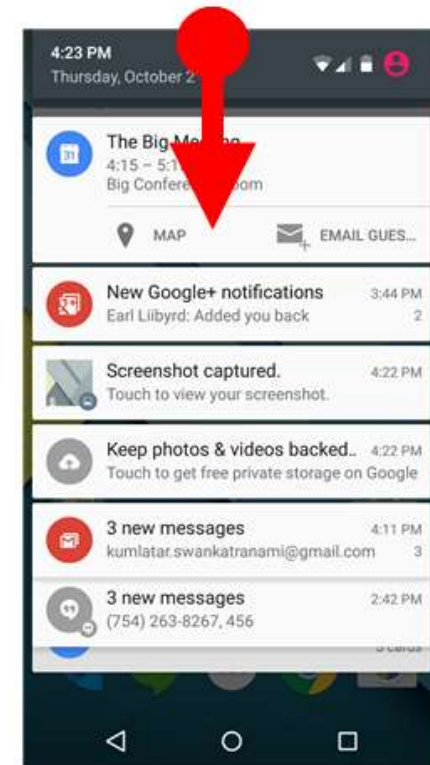
# Notification (알림)

- 기능

- App이 실행되고 있지 않거나 background에서 동작 중일 때, app이 user에게 메시지를 전달하기 위한 방법
  - Home 화면의 status bar에 알림 아이콘이 나타나거나 launcher icon에 notification dot가 나타남.
  - status bar를 아래로 drag해서 notification drawer를 열거나 launcher icon을 클릭해 메시지 확인
- 알림 창에서 직접 응답 메시지를 작성해서 보낼 수도 있음(**Direct reply**) – Android 7부터 가능

- 종류

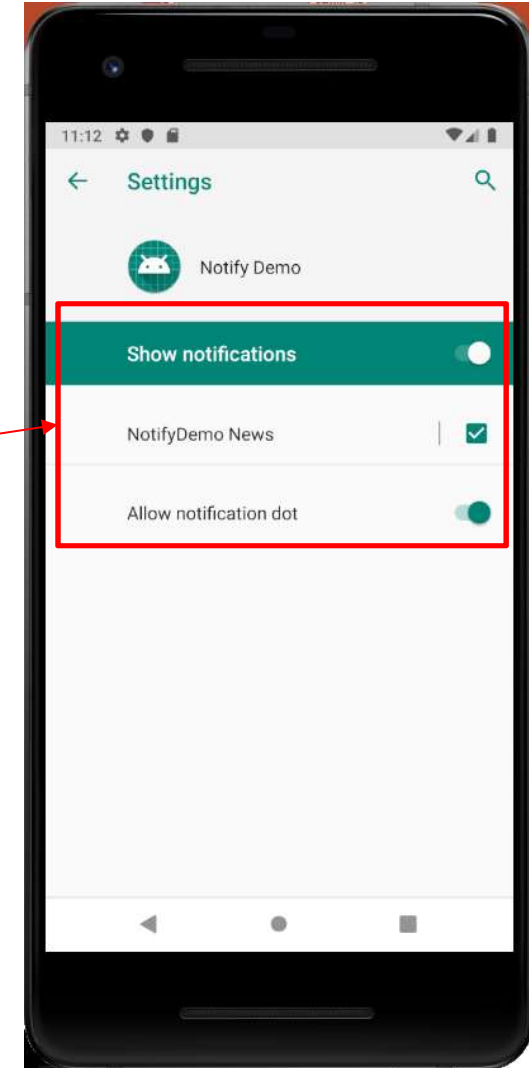
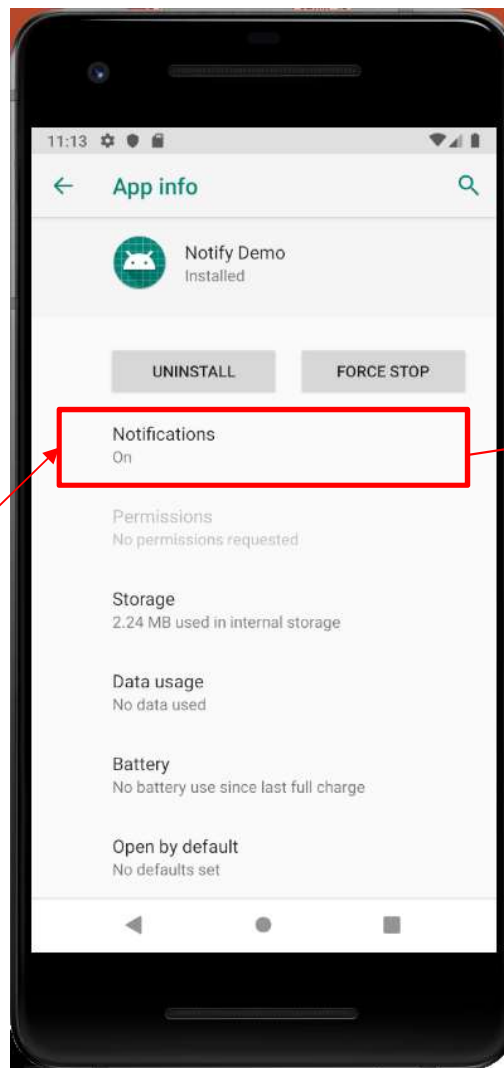
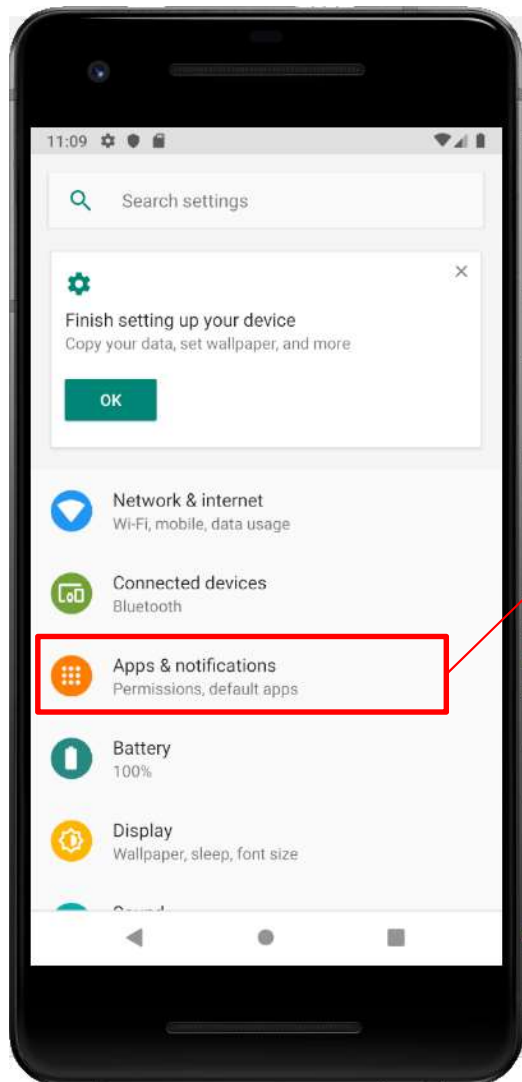
- 지역 알림
  - 사용 중인 device에 설치한 App에서 보내는 알림.
- 원격 알림
  - Remote server에서 device에 전달하는 알림.



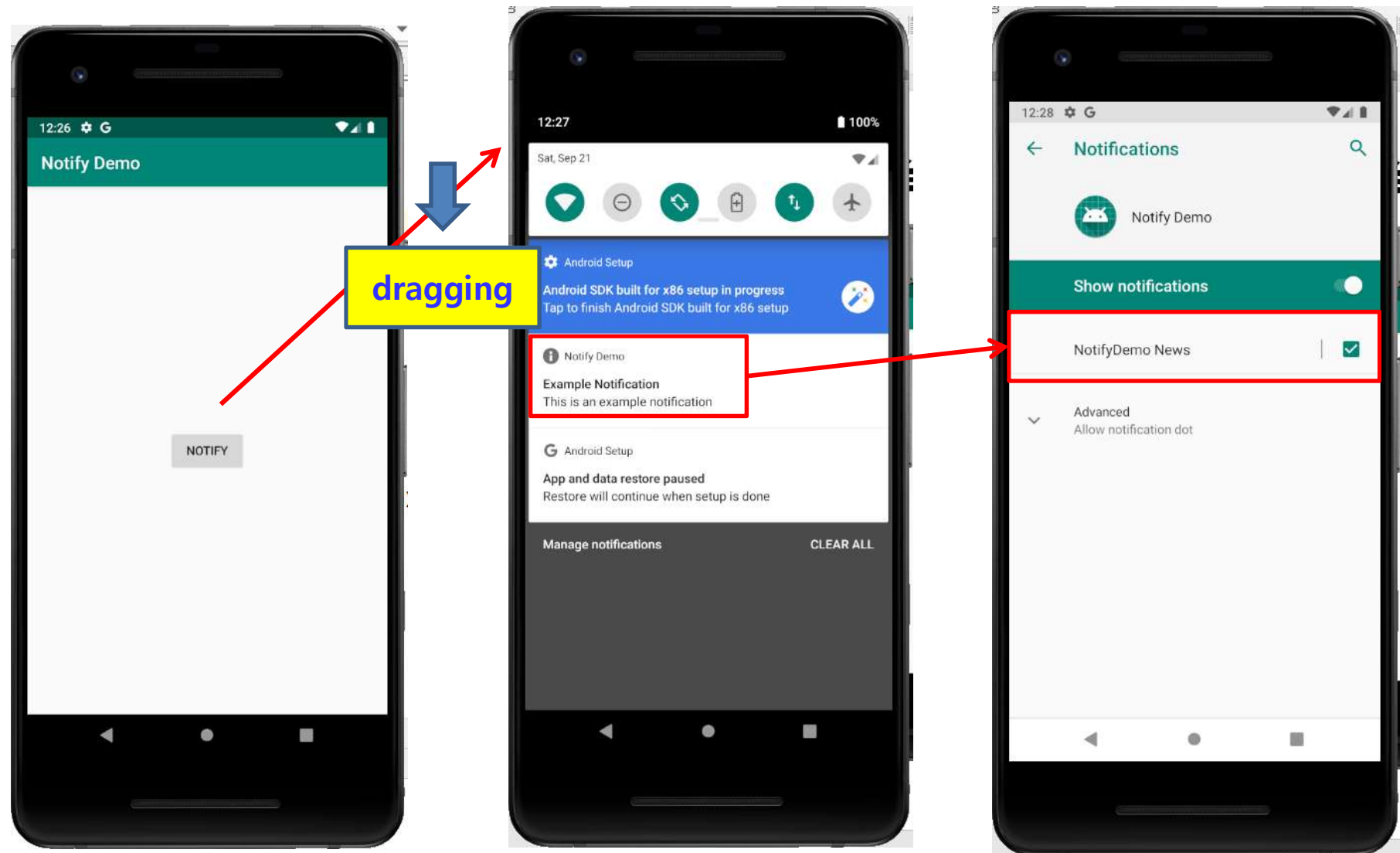
# 실습 내용

- 새 프로젝트 생성
  - Application name : **Notify Demo**
  - Activity : **Empty Activity**
  - Target Android Devices : **API 26 (8.0, Oreo)**
- 실습 내용
  - 코드 참조 : **7장-소스코드(notification).hwp**
  - 1) **AVD 설정**(Notification 허용, notification dot 허용)
  - 2) **실습 1**: 상태 창에 notification 나타내기
  - 3) **실습 2**
    - 3-1) **PendingIntent**를 사용해 새 activity로 이동
    - 3-2) Notification 창에 **action** 추가
    - 3-3) **Bundled Notification**

# AVD 설정



# 실습 1: Notification 생성



# 실습 1: Notification 생성 - Layout

## activity\_main.xml

```
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="sendNotification"
        android:text="Notify"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



# 실습 1: Notification 생성 – Activity (1/2)

```
private var notificationManager: NotificationManager? = null
```

```
override fun onCreate(savedInstanceState: Bundle?) {
```

```
    super.onCreate(savedInstanceState)
```

```
    setContentView(R.layout.activity_main)
```

```
    notificationManager =
```

```
        getSystemService(
```

```
            Context.NOTIFICATION_SERVICE) as NotificationManager
```

```
    createNotificationChannel(
```

```
        "edu.incheon.notifydemo.news",
```

```
        "NotifyDemo News",
```

```
        "Example News Channel"
```

```
    )
```

```
}
```

```
private fun createNotificationChannel(id: String, name: String,  
                                     description: String) {
```

```
    val importance = NotificationManager.IMPORTANCE_LOW
```

```
    val channel = NotificationChannel(id, name, importance)
```

```
    channel.description = description
```

```
    channel.enableLights(true)
```

```
    channel.lightColor = Color.RED
```

```
    channel.enableVibration(true)
```

```
    channel.vibrationPattern =
```

```
        longArrayOf(100, 200, 300, 400, 500, 400, 300, 200, 400)
```

```
    notificationManager?.createNotificationChannel(channel)
```

```
}
```

MainActivity.kt



# 실습 1: Notification 생성 – Activity (2/2)

```
fun sendNotification(view: View) {  
  
    val notificationID = 101  
  
    val channelId = "edu.incheon.notifydemo.news"  
  
    val notification = Notification.Builder(  
        this@MainActivity,  
        channelId  
    )  
        .setContentTitle("Example Notification")  
        .setContentText("This is an example notification")  
        .setSmallIcon(android.R.drawable.ic_dialog_info)  
        .setChannelId(channelId)  
        .build()  
  
    notificationManager?.notify(notificationID, notification)  
  
}
```

MainActivity.kt



# 알림 전송을 위한 Channel

- Oreo(API 26) 버전부터 알림 메시지 전송을 위해서는 channel을 등록해야 함
  - Channel이란?
    - TV채널과 유사 - 스포츠 채널, 홈쇼핑 채널, 교육 채널 등
    - channel 주제에 맞는 알림 메시지를 해당 채널에 전송

```
val importance = NotificationManager.IMPORTANCE_LOW
val channel = NotificationChannel(id, name, importance)

channel.description = description
channel.enableLights(true)
channel.lightColor = Color.RED
channel.enableVibration(true)
channel.vibrationPattern =
    longArrayOf(100, 200, 300, 400, 500, 400, 300, 200, 400)
notificationManager?.createNotificationChannel(channel)
```

# NotificationManager 클래스

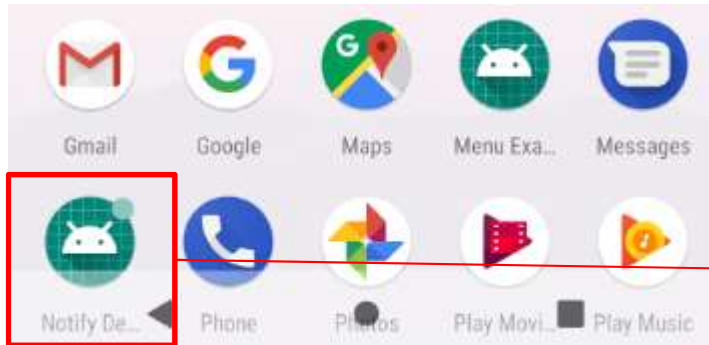
- ◆ You do not instantiate this class directly; instead, retrieve it through **getSystemService ( String )**.

```
notificationManager =  
    getSystemService(  
        Context.NOTIFICATION_SERVICE) as NotificationManager
```

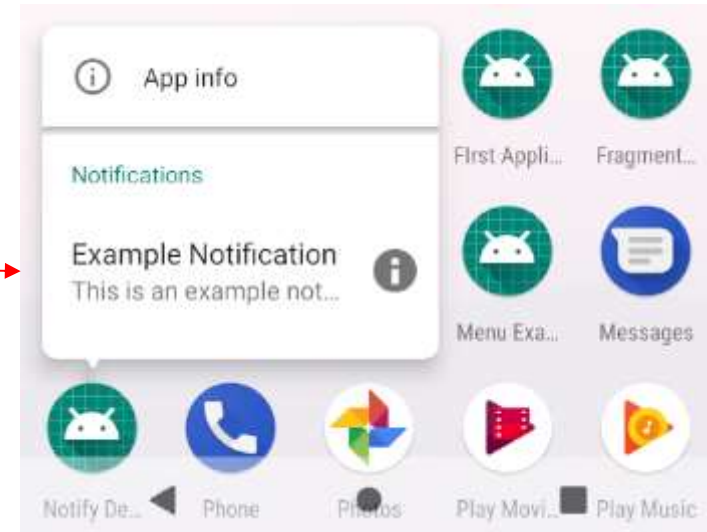
NotificationManager 객체를 가져 올 수 있는 유일한 방법!

```
val notification = Notification.Builder(  
    this@MainActivity,  
    channelId  
)  
    .setContentTitle("Example Notification")  
    .setContentText("This is an example notification")  
    .setSmallIcon(android.R.drawable.ic_dialog_info)  
    .setChannelId(channelId)  
    .build()  
  
notificationManager?.notify(notificationID, notification)
```

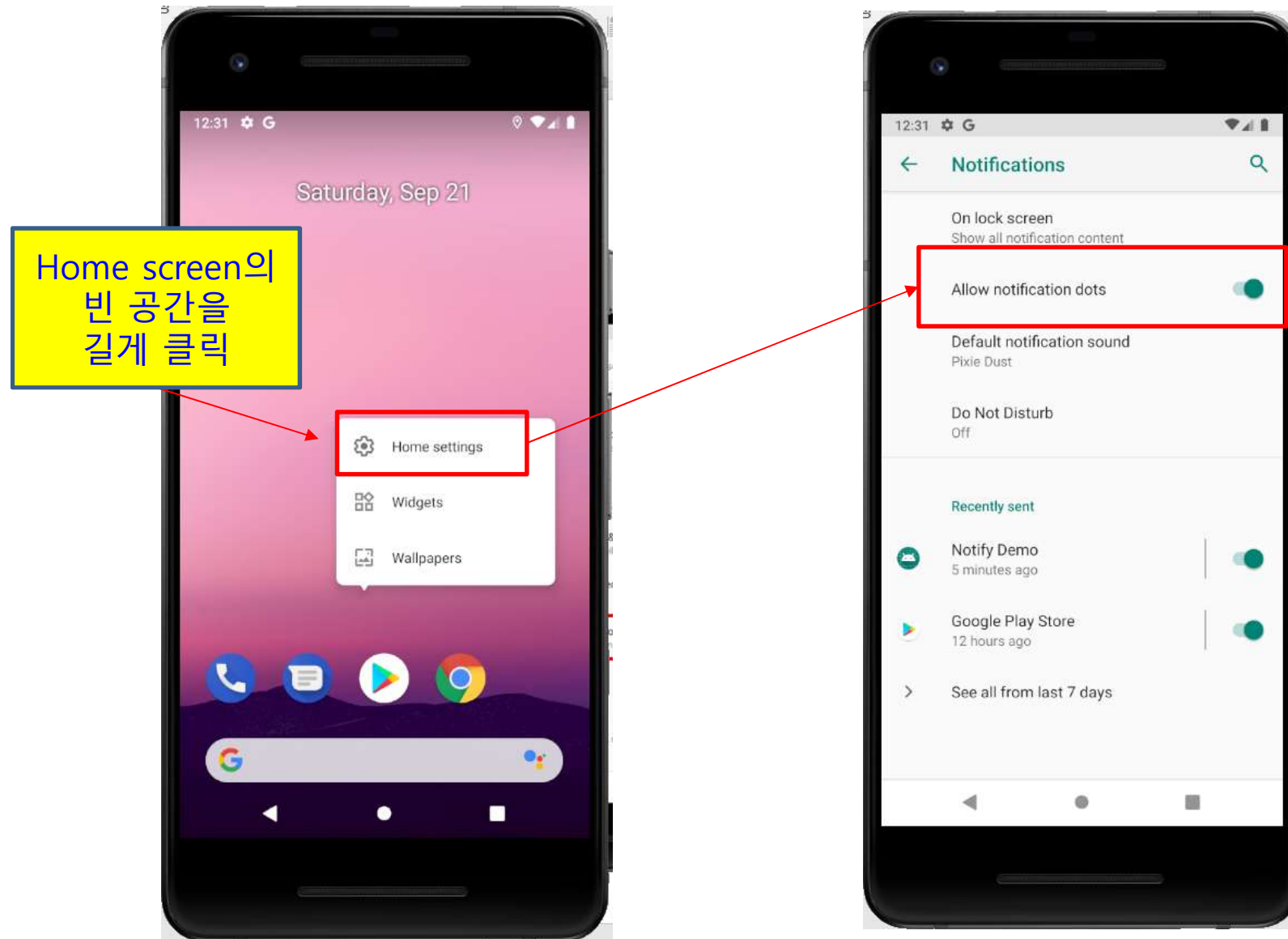
# Notification dot 실행



launcher icon  
길게 클릭



# 잠깐! Notification dot가 안 보여요



# PendingIntent 클래스

- 사용자가 (상태 창을 확장시켜) notification 객체를 선택했을 때의 동작을 지정
  - 일반 Intent와 기능 면에서 다름
    - 시스템이 직접 관리
      - 사용자 클릭 또는 tab하기 전에는 대기(pending) 상태
    - PendingIntent를 생성한 응용 프로그램이 종료되어도 유효
  - Constructor가 정의되어 있지 않아 객체를 직접 생성할 수 없음.
    - 다음 3개의 정적 메소드 중 하나를 사용하여 객체를 생성.
      - getActivity** (Context context , int requestCode, int flags)
      - getBroadcast** (Context context, int requestCode, int flags)
      - getService** (Context context, int requestCode, int flags)

# 잠깐! PendingIntent flag

```
PendingIntent pendingIntent =  
    PendingIntent.getActivity(MainActivity.this,  
        0, intent,  
        PendingIntent.FLAG_UPDATE_CURRENT);
```

## **FLAG\_CANCEL\_CURRENT :**

Flag indicating that if the described PendingIntent already exists, the current one should be canceled before generating a new one.

## **FLAG\_IMMUTABLE(변경 불가능) :**

Flag indicating that the created PendingIntent should be immutable.

## **FLAG\_NO\_CREATE :**

Flag indicating that if the described PendingIntent does not already exist, then simply return null instead of creating it.

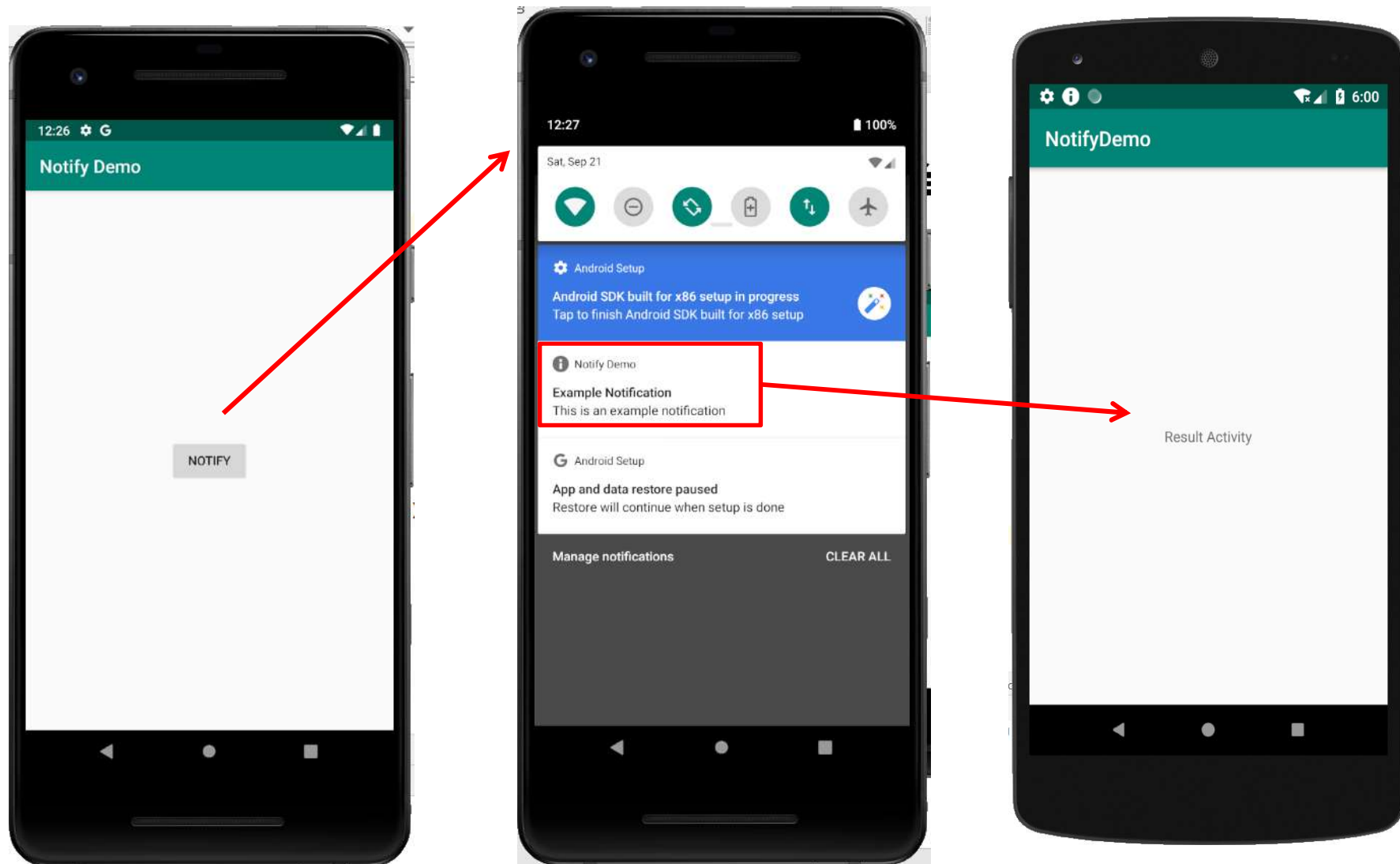
## **FLAG\_ONE\_SHOT :**

Flag indicating that this PendingIntent can be used only once.

## **FLAG\_UPDATE\_CURRENT :**

Flag indicating that if the described PendingIntent already exists, then keep it but replace its extra data with what is in this new Intent.

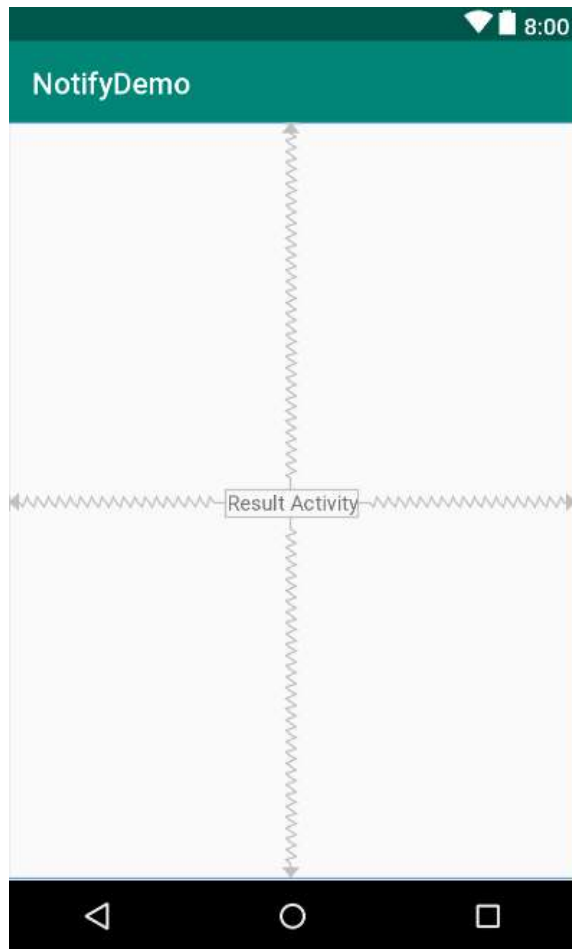
## 실습 2-1: PendingIntent를 사용해 activity 이동





## 실습 2-1: 알림에서 activity 이동 - ResultActivity

activity\_result.xml



ResultActivity.kt

```
class ResultActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_result)  
    }  
}
```

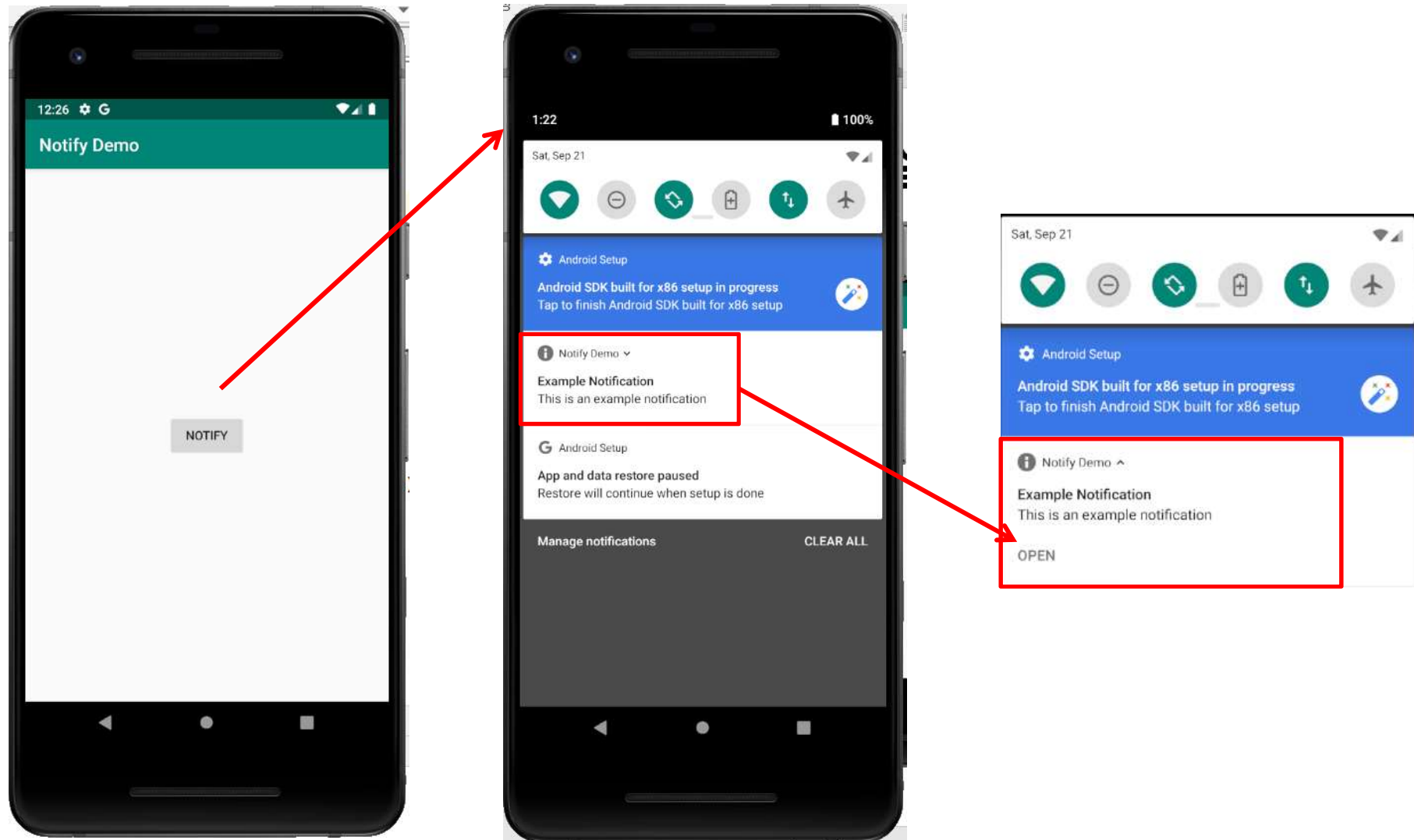
AndroidManifest.xml

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

## 실습 2-1: 알림에서 activity 이동 - Activity

```
fun sendNotification(view: View) {  
  
    val notificationID = 101  
  
    val resultIntent = Intent(this, ResultActivity::class.java)  
    val pendingIntent = PendingIntent.getActivity(  
        this,  
        0,  
        resultIntent,  
        PendingIntent.FLAG_UPDATE_CURRENT  
    )  
  
    val channelId = "edu.incheon.notifydemo.news"  
    val notification = Notification.Builder(  
        this@MainActivity,  
        channelId  
    )  
        .setContentTitle("Example Notification")  
        .setContentText("This is an example notification")  
        .setSmallIcon(android.R.drawable.ic_dialog_info)  
        .setChannelId(channelId)  
        .setContentIntent(pendingIntent)  
        .build()  
  
    notificationManager?.notify(notificationID, notification)  
}
```

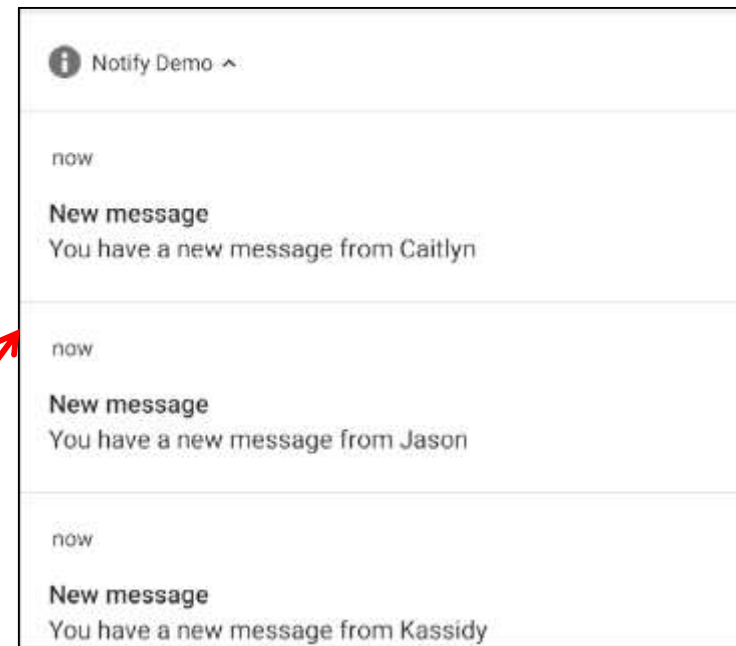
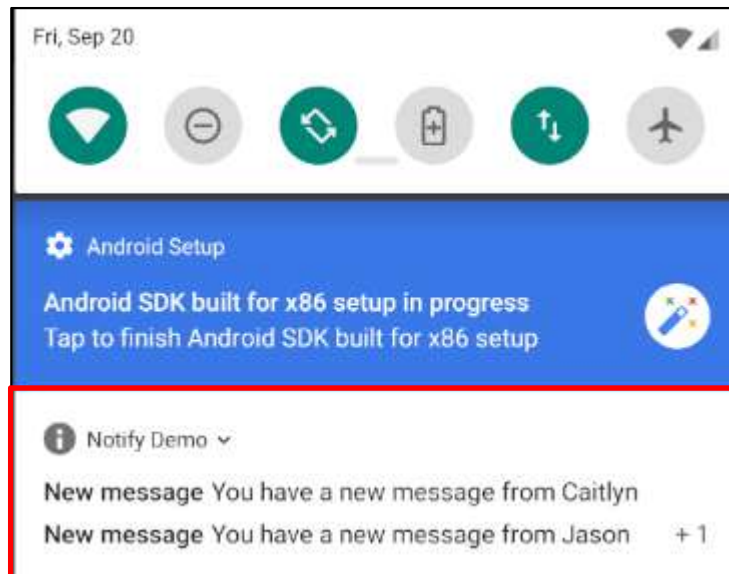
## 실습 2-2: 알림에 action 추가



## 실습 2-2: 알림에 action 추가

```
fun sendNotification(view: View) {  
  
    val notificationID = 101  
    val resultIntent = Intent(this, ResultActivity::class.java)  
    val pendingIntent = PendingIntent.getActivity(  
        this,  
        0,  
        resultIntent,  
        PendingIntent.FLAG_UPDATE_CURRENT  
    )  
  
    val icon: Icon = Icon.createWithResource(this, android.R.drawable.ic_dialog_info)  
    val action: Notification.Action =  
        Notification.Action.Builder(icon, "Open", pendingIntent).build()  
  
    val channelId = "edu.incheon.notifydemo.news"  
    val notification = Notification.Builder(  
        this@MainActivity,  
        channelId  
    )  
        .setContentTitle("Example Notification")  
        .setContentText("This is an example notification")  
        .setSmallIcon(android.R.drawable.ic_dialog_info)  
        .setChannelId(channelId)  
        .setContentIntent(pendingIntent)  
        .setActions(action)  
        .build()  
  
    notificationManager?.notify(notificationID, notification)  
}
```

# 실습 2-3: Bundled Notification





## 실습 2-3: Bundled Notification (1/2)

```
fun sendNotification(view: View) {  
  
    val channelId = "edu.incheon.notifydemo.news"  
  
    val GROUP_KEY_NOTIFY = "group_key_notify"  
    val builderSummary: Notification.Builder = Notification.Builder(this, channelId)  
        .setSmallIcon(android.R.drawable.ic_dialog_info)  
        .setContentTitle("A Bundle example")  
        .setContentText("You have 3 messages")  
        .setGroup(GROUP_KEY_NOTIFY)  
        .setGroupSummary(true)  
  
    val builder1: Notification.Builder = Notification.Builder(this, channelId)  
        .setSmallIcon(android.R.drawable.ic_dialog_info)  
        .setContentTitle("New message")  
        .setContentText("You have a new message from Kassidy")  
        .setGroup(GROUP_KEY_NOTIFY)  
  
    val builder2: Notification.Builder = Notification.Builder(this, channelId)  
        .setSmallIcon(android.R.drawable.ic_dialog_info)  
        .setContentTitle("New message")  
        .setContentText("You have a new message from Caitlyn")  
        .setGroup(GROUP_KEY_NOTIFY)  
  
    val builder3: Notification.Builder = Notification.Builder(this, channelId)  
        .setSmallIcon(android.R.drawable.ic_dialog_info)  
        .setContentTitle("New message")  
        .setContentText("You have a new message from Jason")  
        .setGroup(GROUP_KEY_NOTIFY)  
}
```

## 실습 2-3: Bundled Notification (2/2)

```
val notificationId0 = 100  
val notificationId1 = 101  
val notificationId2 = 102  
val notificationId3 = 103
```

```
notificationManager?.notify(notificationId1, builder1.build())  
notificationManager?.notify(notificationId2, builder2.build())  
notificationManager?.notify(notificationId3, builder3.build())  
notificationManager?.notify(notificationId0, builderSummary.build())
```