

04

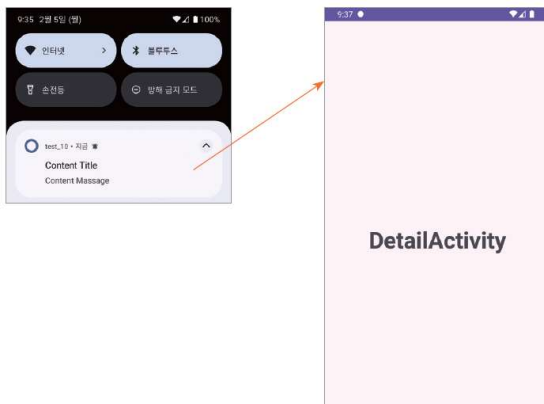
### 이4-3. 다양한 Notification 활용

**브로드캐스트 리시버**

## 10-5 알림 띄우기

### 알림 구성

- 알림 터치 이벤트
  - 인텐트를 준비한 후 Notification 객체에 담아서 이벤트가 발생할 때 인텐트를 실행해 달라고 시스템에 의뢰
  - `static fun getActivity(context: Context!, requestCode: Int, intent: Intent!, flags: Int): PendingIntent!`
  - `static fun getBroadcast(context: Context!, requestCode: Int, intent: Intent!, flags: Int): PendingIntent!`
  - `static fun getService(context: Context!, requestCode: Int, intent: Intent!, flags: Int): PendingIntent!`



#### • 알림 객체에 액티비티 실행 정보 등록

```
val intent = Intent(this, DetailActivity::class.java)
val pendingIntent =
    PendingIntent.getActivity(this, 10, intent,
        PendingIntent.FLAG_IMMUTABLE)
builder.setContentIntent(pendingIntent) // 터치 이벤트 등록
```

## 10-5 알림 띄우기

- 액션
  - 액션을 최대 3개까지 추가



### • 액션 등록하기

```
val actionIntent = Intent(this, OneReceiver::class.java)
val actionPendingIntent = PendingIntent.getBroadcast(this, 20, actionIntent,
    PendingIntent.FLAG_IMMUTABLE)
builder.addAction(
    NotificationCompat.Action.Builder(
        android.R.drawable.stat_notify_more,
        "Action",
        actionPendingIntent
    ).build()
)
```

### ▶ 실행 결과



## 10-5 알림 띄우기

- 원격 입력
  - 알림에서 사용자 입력을 직접 받는 기법



### • 원격 입력

```
val KEY_TEXT_REPLY = "key_text_reply"
var replyLabel: String = "답장"
var remoteInput: RemoteInput = RemoteInput.Builder(KEY_TEXT_REPLY).run {
    setLabel(replyLabel)
    build()
}
```

- PendingIntent를 준비

### • 인텐트 준비

```
val replyIntent = Intent(this, ReplyReceiver::class.java)
val replyPendingIntent = PendingIntent.getBroadcast(this, 30, replyIntent,
    PendingIntent.FLAG_MUTABLE)
```

## 10-5 알림 띄우기

- 알림에 액션을 등록

• 원격 입력 액션 등록하기

```
builder.addAction(  
    NotificationCompat.Action.Builder(  
        R.drawable.send,  
        "답장",  
        replyPendingIntent  
    ).addRemoteInput(remoteInput).build()  
)
```

▶ 실행 결과



## 10-5 알림 띄우기

---

- 사용자가 입력한 글을 받을 때 코드

• 브로드캐스트 리시버에서 사용자가 입력한 글을 받는 코드

```
val replyTxt = RemoteInput.getResultsFromIntent(intent)
    ?.getCharSequence("key_text_reply")
```

## 10-5 알림 띄우기

- 프로그레스
  - 진행 상황을 프로그레스에 바로 알려 줍니다.

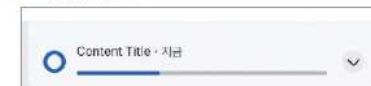


• 프로그레스 바의 진행값을 증가시키는 스레드

```
builder.setProgress(100, 0, false)
manager.notify(11, builder.build())

thread {
    for (i in 1..100) {
        builder.setProgress(100, i, false)
        manager.notify(11, builder.build())
        SystemClock.sleep(100)
    }
}
```

▶ 실행 결과



## 10-5 알림 띄우기

### 알림 스타일

- 큰 이미지 스타일



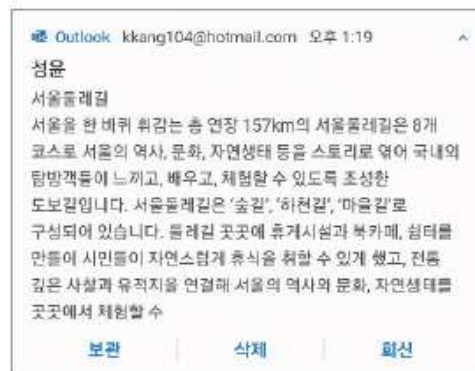
- 큰 이미지 스타일

```
val bigPicture = BitmapFactory.decodeResource(resources, R.drawable.test)
val bigStyle = NotificationCompat.BigPictureStyle()
bigStyle.bigPicture(bigPicture)
builder.setStyle(bigStyle)
```



## 10-5 알림 띄우기

- 긴 텍스트 스타일



- 긴 텍스트 스타일

```
val bigTextStyle = NotificationCompat.BigTextStyle()  
bigTextStyle.bigText(resources.getString(R.string.long_text))  
builder.setStyle(bigTextStyle)
```

## 10-5 알림 띄우기

- 상자 스타일

### • 상자 스타일

```
val style = NotificationCompat.InboxStyle()  
style.addLine("1코스 - 수락.불암산코스")  
style.addLine("2코스 - 용마.아차산코스")  
style.addLine("3코스 - 고덕.일자산코스")  
style.addLine("4코스 - 대모.우면산코스")  
builder.setStyle(style)
```

### ▶ 실행 결과



## 10-5 알림 띄우기

---

- 메시지 스타일
  - Person은 알림에 출력될 한 사람의 정보를 담는 클래스

• Person 객체 생성

```
val sender1: Person = Person.Builder()
    .setName("kkang")
    .setIcon(IconCompat.createWithResource(this, R.drawable.person1))
    .build()

val sender2: Person = Person.Builder()
    .setName("kim")
    .setIcon(IconCompat.createWithResource(this, R.drawable.person2))
    .build()
```

## 10-5 알람 띄우기

- 메시지 스타일

- 메시지 객체 생성

```
val message1 = NotificationCompat.MessagingStyle.Message(  
    "hello",  
    System.currentTimeMillis(),  
    sender1  
)  
  
val message2 = NotificationCompat.MessagingStyle.Message(  
    "world",  
    System.currentTimeMillis(),  
    sender2  
)
```

- 메시지 스타일 만들기

```
val messageStyle = NotificationCompat.MessagingStyle(sender1)  
    .addMessage(message1)  
    .addMessage(message2)  
builder.setStyle(messageStyle)
```

▶ 실행 결과





# 감사합니다

단단히 마음먹고 떠난 사람은  
산꼭대기에 도착할 수 있다.  
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어  
William Shakespeare