

Building a Notification

This lesson teaches you to

- Create a Notification Builder (#builder)

- Define the Notification's Action (#action)

- Set the Notification's Click Behavior (#click)

- Issue the Notification (#notify)

This lesson explains how to create and issue a notification.

The examples in this class are based on the `NotificationCompat.Builder`

(<https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html>) class.

`NotificationCompat.Builder`

(<https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html>) is in the Support Library (<https://developer.android.com/>). You should use `NotificationCompat`

(<https://developer.android.com/reference/android/support/v4/app/NotificationCompat.html>) and its subclasses, particularly `NotificationCompat.Builder`

(<https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html>), to provide the best notification support for a wide range of platforms.

Create a Notification Builder

When creating a notification, specify the UI content and actions with a `NotificationCompat.Builder`

(<https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html>) object. At bare

minimum, a `Builder` (<https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html>) object must include the following:

- A small icon, set by `setSmallIcon()`
([https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html#setSmallIcon\(int\)](https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html#setSmallIcon(int)))
- A title, set by `setContentTitle()`
([https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html#setContentTitle\(java.lang.CharSequence\)](https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html#setContentTitle(java.lang.CharSequence)))

```
a.lang.CharSequence))
```

- Detail text, set by `setContentText()`

```
(https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html#setContentText\(java  
.lang.CharSequence))
```

For example:

```
NotificationCompat.Builder mBuilder =  
    new NotificationCompat.Builder(this)  
        .setSmallIcon(R.drawable.notification_icon)  
        .setTitle("My notification")  
        .setContentText("Hello World!");
```

Define the Notification's Action

Although actions are optional, you should add at least one action to your notification. An action takes users directly from the notification to an `Activity` (<https://developer.android.com/reference/android/app/Activity.html>) in your application, where they can look at the event that caused the notification or do further work. Inside a notification, the action itself is defined by a `PendingIntent` (<https://developer.android.com/reference/android/app/PendingIntent.html>) containing an `Intent` (<https://developer.android.com/reference/android/content/Intent.html>) that starts an `Activity` (<https://developer.android.com/reference/android/app/Activity.html>) in your application.

How you construct the `PendingIntent` (<https://developer.android.com/reference/android/app/PendingIntent.html>) depends on what type of `Activity` (<https://developer.android.com/reference/android/app/Activity.html>) you're starting. When you start an `Activity` (<https://developer.android.com/reference/android/app/Activity.html>) from a notification, you must preserve the user's expected navigation experience. In the snippet below, clicking the notification opens a new activity that effectively extends the behavior of the notification. In this case there is no need to create an artificial back stack (see [Preserving Navigation when Starting an Activity](https://developer.android.com/training/notify-user/navigation.html) (<https://developer.android.com/training/notify-user/navigation.html>) for more information):

```
Intent resultIntent = new Intent(this, ResultActivity.class);  
...  
// Because clicking the notification opens a new ("special") activity, there's  
// no need to create an artificial back stack.  
PendingIntent resultPendingIntent =  
    PendingIntent.getActivity(  
        this,  
        0,  
        resultIntent,  
        PendingIntent.FLAG_UPDATE_CURRENT  
    );
```

Set the Notification's Click Behavior

To associate the `PendingIntent` (<https://developer.android.com/reference/android/app/PendingIntent.html>) created in the previous step with a gesture, call the appropriate method of `NotificationCompat.Builder` (<https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html>). For example, to start an activity when the user clicks the notification text in the notification drawer, add the `PendingIntent` (<https://developer.android.com/reference/android/app/PendingIntent.html>) by calling `setContentIntent()` ([https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html#setContentIntent\(android.app.PendingIntent\)](https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html#setContentIntent(android.app.PendingIntent))). For example:

```
PendingIntent resultPendingIntent;  
...  
mBuilder.setContentIntent(resultPendingIntent);
```

Issue the Notification

To issue the notification:

- Get an instance of `NotificationManager` (<https://developer.android.com/reference/android/app/NotificationManager.html>).
- Use the `notify()` ([https://developer.android.com/reference/java/lang/Object.html#notify\(\)](https://developer.android.com/reference/java/lang/Object.html#notify())) method to issue the notification. When you call `notify()` ([https://developer.android.com/reference/java/lang/Object.html#notify\(\)](https://developer.android.com/reference/java/lang/Object.html#notify())), specify a notification ID. You can use this ID to update the notification later on. This is described in more detail in [Managing Notifications](https://developer.android.com/training/notify-user/managing.html) (<https://developer.android.com/training/notify-user/managing.html>).
- Call `build()` ([https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html#build\(\)](https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder.html#build())), which returns a `Notification` (<https://developer.android.com/reference/android/app/Notification.html>) object containing your specifications.

For example:

```
NotificationCompat.Builder mBuilder;  
...  
// Sets an ID for the notification  
int mNotificationId = 001;  
// Gets an instance of the NotificationManager service
```

```
NotificationManager mNotifyMgr =  
    (NotificationManager) getSystemService(NOTIFICATION_SERVICE);  
// Builds the notification and issues it.  
mNotifyMgr.notify(mNotificationId, mBuilder.build());
```

