

CSc 317

Location and Notifications

Benjamin Dicken



Announcements

- Presentations next week!
- SCS (62.79% before class)
- Final exam - Thursday, May 12, 10:30-12:30

Getting user Location

- Use a **LocationListener** object
- Implement required callbacks
 - To be notified when the device location changes, override the **onLocationChanged** function

```
private final LocationListener mLocationListener = new LocationListener() {
    @Override
    public void onLocationChanged(final Location location) {
        double currLat = location.getLatitude();
        double currLng = location.getLongitude();
        // Do things with the Lat and Lng
    }

    @Override
    public void onStatusChanged(String s, int i, Bundle bundle) { }
    @Override
    public void onProviderEnabled(String s) { }
    @Override
    public void onProviderDisabled(String s) { }
};
```

```
mLocationManager.requestLocationUpdates(  
    LocationManager.GPS_PROVIDER,  
    100 /*LOCATION_REFRESH_TIME*/,  
    1 /*LOCATION_REFRESH_DISTANCE*/,  
    mLocationListener);
```

The goal

- Create an application that tracks a user's path
- Draws a marker when it gets a location update, and a path from last known location to updated location
- Generate your own key (or reuse from last time)



App Notifications

- Lots of apps give notifications
 - Messaging, email
 - Reminders for calendar event
 - Reminder to update phone
 - Download notification
- <https://developer.android.com/guide/topics/ui/notifiers/notifications>
- What elements in a notification?

Notification Anatomy

Notification anatomy

The design of a notification is determined by system templates—your app simply defines the contents for each portion of the template. Some details of the notification appear only in the expanded view.

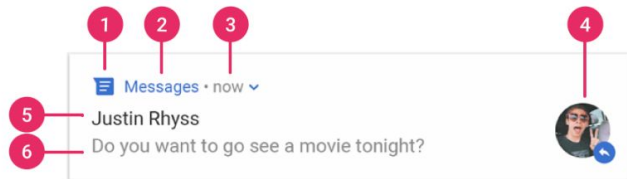


Figure 7. A notification with basic details

The most common parts of a notification are indicated in figure 7 as follows:

- 1 Small icon: This is required and set with `setSmallIcon()`.
- 2 App name: This is provided by the system.
- 3 Time stamp: This is provided by the system but you can override with `setWhen()` or hide it with `setShowWhen(false)`.
- 4 Large icon: This is optional (usually used only for contact photos; do not use it for your app icon) and set with `setLargeIcon()`.
- 5 Title: This is optional and set with `setContentTitle()`.
- 6 Text: This is optional and set with `setContentText()`.

Basic Notification Code

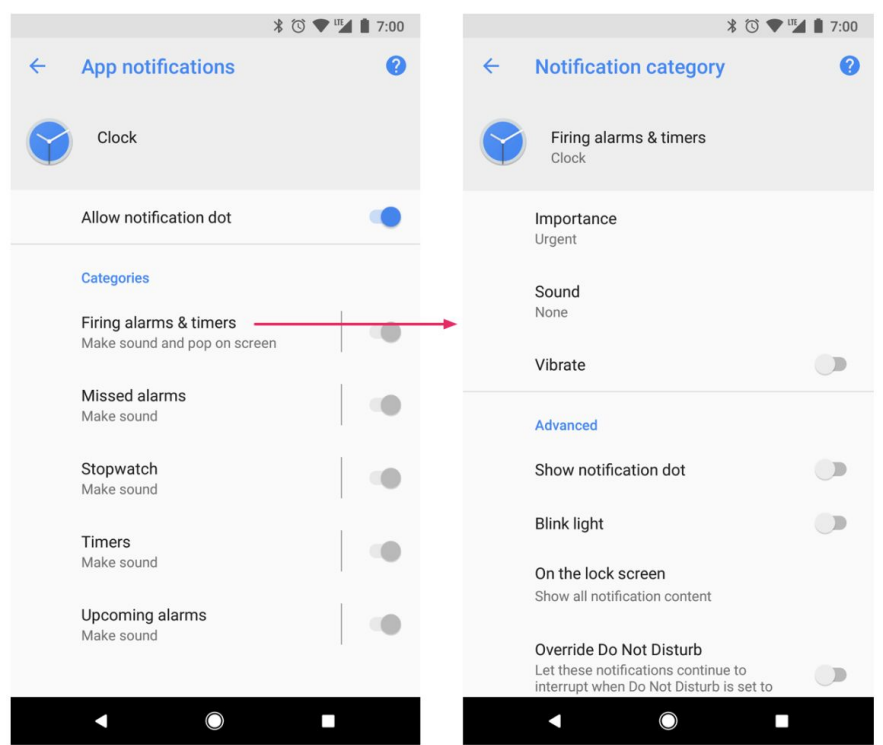
```
NotificationCompat.Builder builder =  
    new NotificationCompat.Builder(this, CHANNEL_ID) // string  
        .setSmallIcon(R.drawable.att)  
        .setContentTitle("title")  
        .setContentText("content")  
        .setPriority(NotificationCompat.PRIORITY_MAX);
```

```
NotificationManagerCompat notificationManager =  
    NotificationManagerCompat.from(this);
```

```
notificationManager.notify(notificationId++, builder.build());
```

App Notification Channels

- One app can have multiple notification channels
- Can give user more control
- Should setup on in API level 26+



Setting up a notification channel

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
    CharSequence name = getString(R.string.channel_name);
    String description = getString(R.string.channel_description);
    int importance = NotificationManager.IMPORTANCE_DEFAULT;

    NotificationChannel channel = new NotificationChannel(
        CHANNEL_ID, name, importance);
    channel.setDescription(description);

    NotificationManager notificationManager =
        getSystemService(NotificationManager.class);
    notificationManager.createNotificationChannel(channel);
}
```

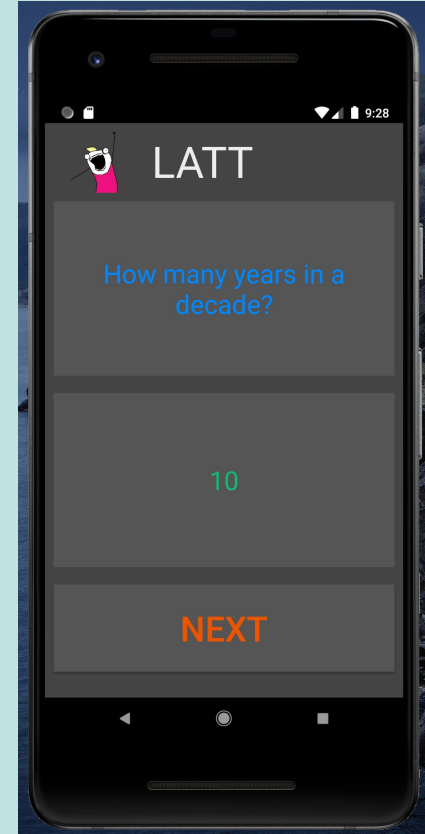
Basic Notification Code

```
Intent intent = new Intent(this, MainActivity.class);
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
PendingIntent pendingIntent = PendingIntent.getActivity(
    this, 0, intent, PendingIntent.FLAG_IMMUTABLE);
```

```
NotificationCompat.Builder builder =
    new NotificationCompat.Builder(this, CHANNEL_ID) // string
        .setSmallIcon(R.drawable.att)
        .setContentTitle("title")
        .setContentText("content")
        .setContentIntent(pendingIntent)
        .setPriority(NotificationCompat.PRIORITY_MAX);
NotificationManagerCompat notificationManager =
    NotificationManagerCompat.from(this);
notificationManager.notify(notificationId++, builder.build());
```

Create the Notification Channel

- Create the **Notification Channel**



Triggering a Service at a Later Time

```
Intent intent = new Intent(this, SomeService.class);
PendingIntent sender = PendingIntent.getService(this, 2, intent, 0);

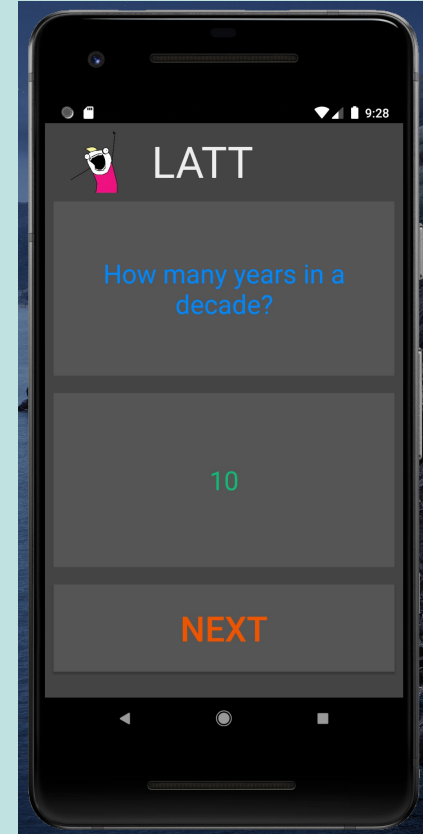
AlarmManager am = (AlarmManager) getSystemService(ALARM_SERVICE);

// One minute from now
long l = new Date().getTime() + 60000;

// Repeat every minute
am.setRepeating(AlarmManager.RTC_WAKEUP, l, 60000, sender);
```

Set the AlarmManager

```
Intent intent = new Intent(this, SomeService.class);  
PendingIntent sender = PendingIntent.getService(this, 2, intent, 0);  
  
AlarmManager am = (AlarmManager) getSystemService(ALARM_SERVICE);  
  
long l = new Date().getTime() + 60000;  
am.setRepeating(AlarmManager.RTC_WAKEUP, l, 60000, sender);
```



Notify

- Build the notification

