

Notifications

OVERVIEW NOTES

Dr. Abraham Campbell
University College Dublin
Abey.campbell@ucd.ie

Notifications

- Alerting users without using an Activity
- The preferred way to alert users that some event has happened that requires attention
- Used by invisible application components:
 - Broadcast Receivers
 - Services
 - Inactive Activities

Notifications Cont'd

- Notifications in Android v 2.x can:
 - Create new status bar icons
 - Display additional information and launch an intent
 - Flash the light/LEDs
 - Vibrate the phone
 - Sound an audible alert
- Keep in mind that hardware differs!
- Notifications were considerably expanded in v4

Notifications v4

- From 4.1 on
- Notifications can include actions that enable the user to immediately act on a notification from the notification drawer.
- Notifications are now more flexible in size and layout. They can be expanded to show additional information details.
- A priority flag was introduced that helps to sort notifications by importance rather than time only
- Improvements in allowing interaction

Using Improved Notifications from V4

- **NotificationCompat.Builder**
- Allows backwards compatibility
- Depending on Manufacture of a device some additional features will just be ignored.
- This course still aims for API 15 for assignments so you do not need to use this builder but be aware of how Android can use such an additional library to allow older Android devices to still have the same functionality as the latest versions.

Working with Notification Manager

- Get reference to Notification Manager service instance by calling
 - `getSystemService(NOTIFICATION_SERVICE)`
- Create and fill Notification object
- Raise notification on it with `notify()`
- Cancel notifications with `cancel()` and `cancelAll()`

Use Notification Builder

- In general you should use a builder in conjunction with the Notification manager here is an example

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

Use Builder cont'd

```
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
    );

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


Notification ID

- Gives you control over a notification if you need to address it later.
- Allows you to update a notification later on
- Allows you to cancel it as well

```
Notification.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());
```

Notifications: Other

- Can be expanded with remote views
- Additional permissions might be required for some notifications (*i.e.* VIBRATE)
- Notifications can be very well integrated with Alarm Manager
- Android v4 has significantly expanded notification system