Notifications

Dr Tadhg O'Sullivan

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 are 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 8 for assignments but your project is allowed target latest api.

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 / NotficiationCompat Builder

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

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

// Creates an explicit intent for an Activity in your app
Intent resultIntent = new Intent(this, ResultActivity.class);
```

Use Builder cont'd

mBuilder.setContentIntent(resultIntent);
NotificationManager mNotificationManager =
   (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

Int NotificationID = 1;

mNotificationManager.notify(NotificationID,
mBuilder.build());

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

```
NotificationManager mgr =
    (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

mgr.cancel(NotificationID);
```

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

Questions

- Please ask in the Student Forum