CS 495/595

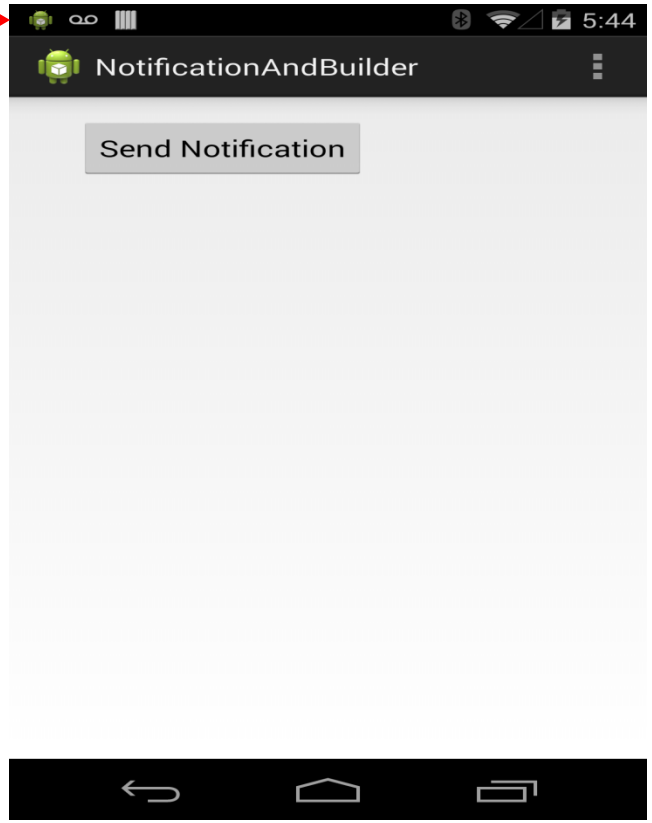
Notifications

Notifications

- Notices that do not require activity
- Handled by Notification Manager
  - Display a status bar icon
  - Flash Lights/LEDs
  - Vibrate Phone
  - Audible Alerts (Ringtones etc)
  - Display additional info in notification tray
  - Broadcast intents from notification tray

Notifications

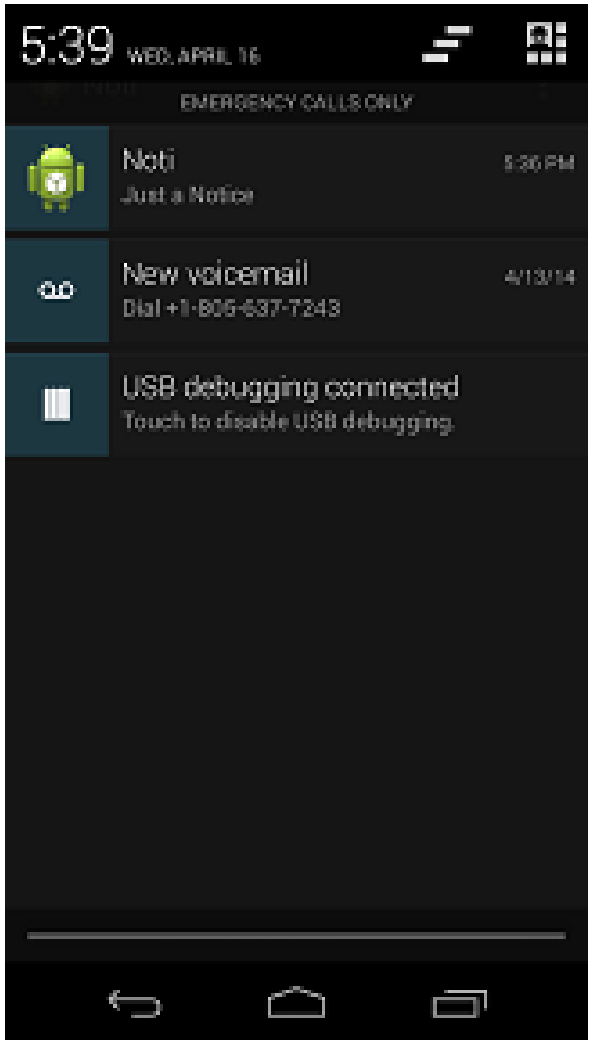
Messages and Information
That appear here



Notifications

Status Bar
Settings, time, date

Notification Tray
Drag down from top in Android 4.4



Notifications

- Get Reference to NotificationManager service

```
NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
```

- Create notification (Old way)

```
Notification notification = new Notification(R.drawable.ic_launcher,
        "This is the text",System.currentTimeMillis());
```

- Works but deprecated… Use a Builder instead

Aside Builder Pattern

- Objects sometimes have many optional fields
- Multiple Constructors? Works but is hard to read, easy to reverse params if types are the same. Leads to subtle bugs
- One Constructor for required fields and then setters? What if a setter fails? Cannot enforce consistency.
- Best – Use a builder – Build an object with all required data, use to construct final object
- Builder pattern – see ComplicatedObject and Builder in NotificationsAndBuilder project.

Notifications

- Builder Introduced in API 11 (3.0) although some methods added in API 16.
- Notifications have a lot of settings. How can you ensure they are correct before construction?
- Once built, use notificationManager object to send it.

Notifications

- Can show progress
  - The following snipped will indicate 30% complete

```
Notification noti = bldr.setContentTitle("New mail from " + "test@gmail.com")
      .setContentText("Progress").setSmallIcon(R.drawable.ic_launcher)
      .setContentIntent(pIntent)
      .setProgress(MAX, PROGRESS, false)
      .addAction(R.drawable.ic_launcher, "Call", pIntent)
      .addAction(R.drawable.ic_launcher, "More", pIntent)
      .addAction(R.drawable.ic_launcher, "And more", pIntent).build();
```

Notifications
Ongoing and Insistent

- Ongoing - In the builder .setOngoing(true)
  - Cant be canceled by user
  - Must be dismissed by app

Notifications
Retrigger and Cancel

- Retrigger
  - Pass in same ref ID with updated notification fields
- Cancel
  - notificationManager.cancel(NOTIFICATION_REF);

Notifications Example

```
public void doNotification(View v) {
    NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);

    Notification noti = new Notification.Builder(this)
    .setContentTitle(getString(R.string.app_name))
    .setContentText("Just a Notice")
    .setSmallIcon(R.drawable.ic_launcher)
    .setOngoing(true)                       // true only dismissable by app
    .build();

    noti.flags |= Notification.FLAG_INSISTENT;

    notificationManager.notify(MYNOTIFICATION, noti);
}

public void doCancelNotification(View v) {
    NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
    notificationManager.cancel(MYNOTIFICATION);
}
```

References

- ‘Android Notifications Tutorial’, Lars Vogel http://www.vogella.com/articles/AndroidNotifications/article.html