CS 475/575

Alarm Manager

Notifications

Alarm Manager

- Fire Intents at predetermined intervals or times
- Do not need app running to work
- Bit more overhead than Timers
- Trigger whatever (broadcast intents, services, or Activities)
- Allow effective resource management

Alarm Manager-Create, Set, Cancel

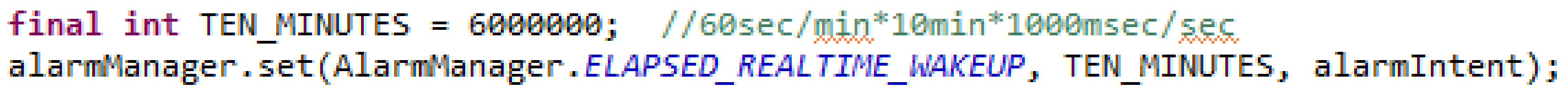
- Get Reference to AlarmManager service

```
AlarmManager alarmManager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
```

- Decide type
  - RTC_WAKEUP – wakes and fires pending intent at time specified
  - RTC – same as RTC_WAKEUP **no** wakeup
  - ELAPSED_REALTIME_WAKEUP - wakes and fires pending intent after specific amount of time has elapsed since system boot
  - ELAPSED_REALTIME - same as ELAPSED_REALTIME_WAKEUP **no** wakeup

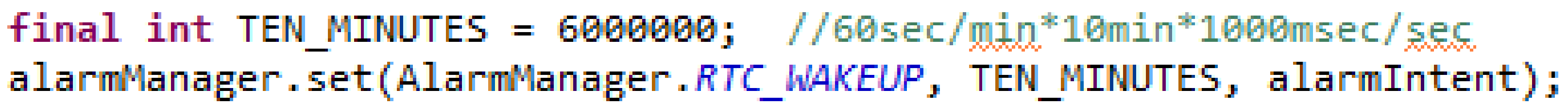
Alarm Manager-Create, Set, Cancel

- ELAPSED_REALTIME_WAKEUP
- ELAPSED_REALTIME
  - Starts specific time after device boots



```
final int TEN_MINUTES = 6000000;  //60sec/min*10min*1000msec/sec
alarmManager.set(AlarmManager.ELAPSED_REALTIME_WAKEUP, TEN_MINUTES, alarmIntent);
```

- RTC_WAKEUP
- RTC
  - Starts specific time from now



```
final int TEN_MINUTES = 6000000;  //60sec/min*10min*1000msec/sec
alarmManager.set(AlarmManager.RTC_WAKEUP, TEN_MINUTES, alarmIntent);
```

Alarm Manager-Create, Set, Cancel

Wake up the device to fire a one-time (non-repeating) alarm in one minute

```
private AlarmManager alarmMgr;
private PendingIntent alarmIntent;
...
alarmMgr = (AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
Intent intent = new Intent(context, AlarmReceiver.class);
alarmIntent = PendingIntent.getBroadcast(context, 0, intent, 0);

alarmMgr.set(AlarmManager.ELAPSED_REALTIME_WAKEUP,
        SystemClock.elapsedRealtime() +
        60 * 1000, alarmIntent);
```

Intents fired <u>for</u> your app by <u>another</u> app at a <u>later</u> time

Execute with same identity and permissions as your app

Cancel alarm

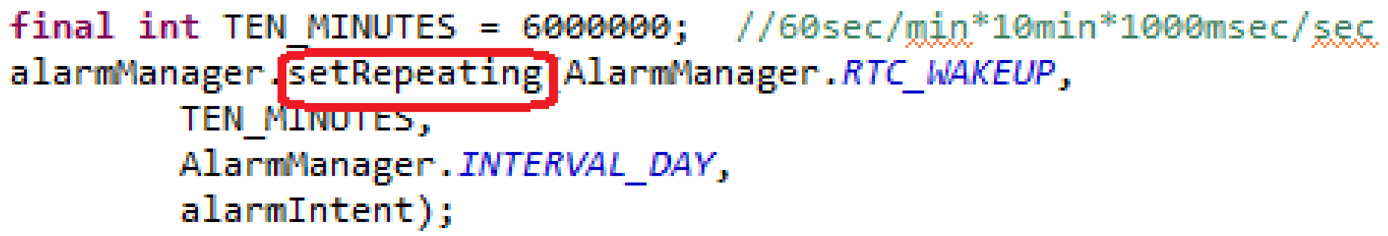
```
// If the alarm has been set, cancel it.
if (alarmMgr!= null) {
    alarmMgr.cancel(alarmIntent);
}
```

Alarm Manager-Repeat Alarm

- setRepeating – for precise control over exact alarm interval
- setInExactRepeating – OS schedules multiple inExactRepeating alarms to execute at same time
  - INTERVAL_FIFTEEN_MINUTES, INTERVAL_HALF_HOUR, INTERVAL_HOUR, INTERVAL_HALF_DAY, INTERVAL_DAY
  - If do not require precision, then wake all alarms that are ‘close together’ at same time (verses multiple wakeups around same time)
  - Reduces battery drain

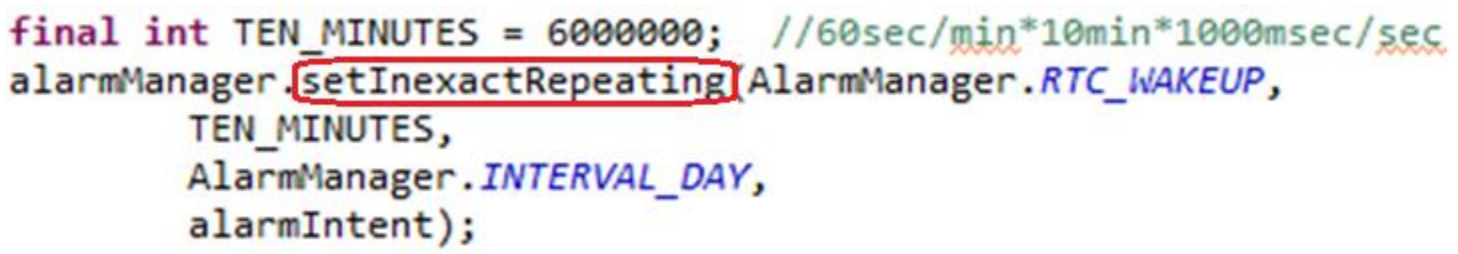
Alarm Manager-Repeat Alarm

- setRepeating



```
final int TEN_MINUTES = 6000000;  //60sec/min*10min*1000msec/sec
alarmManager.setRepeating(AlarmManager.RTC_WAKEUP,
        TEN_MINUTES,
        AlarmManager.INTERVAL_DAY,
        alarmIntent);
```

- setInExactRepeating



```
final int TEN_MINUTES = 6000000;  //60sec/min*10min*1000msec/sec
alarmManager.setInexactRepeating(AlarmManager.RTC_WAKEUP,
        TEN_MINUTES,
        AlarmManager.INTERVAL_DAY,
        alarmIntent);
```

Pending Intent

- Intents fired <u>for</u> your app by <u>another</u> app at a <u>later</u> time
- Execute with same identity and permissions as your app

CS 475/575

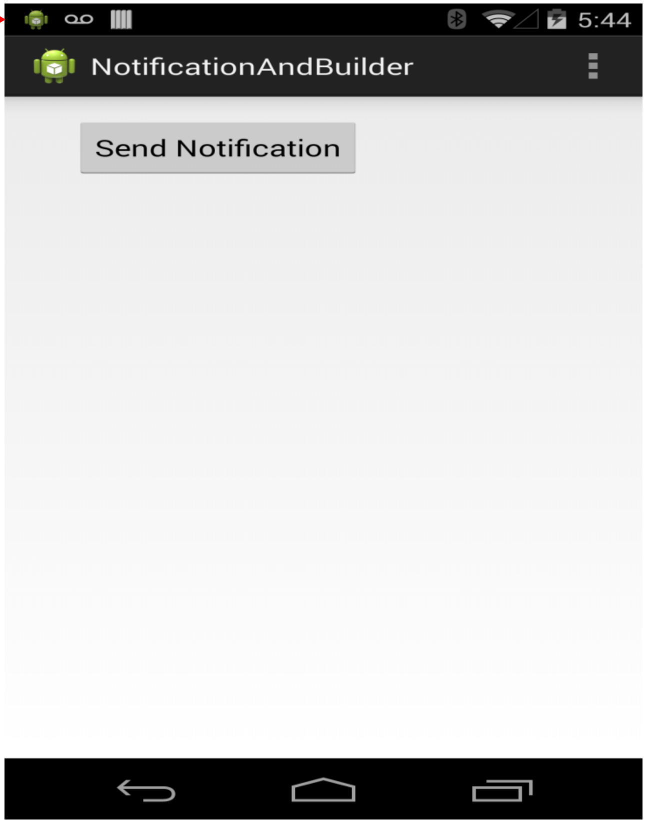
Notifications

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

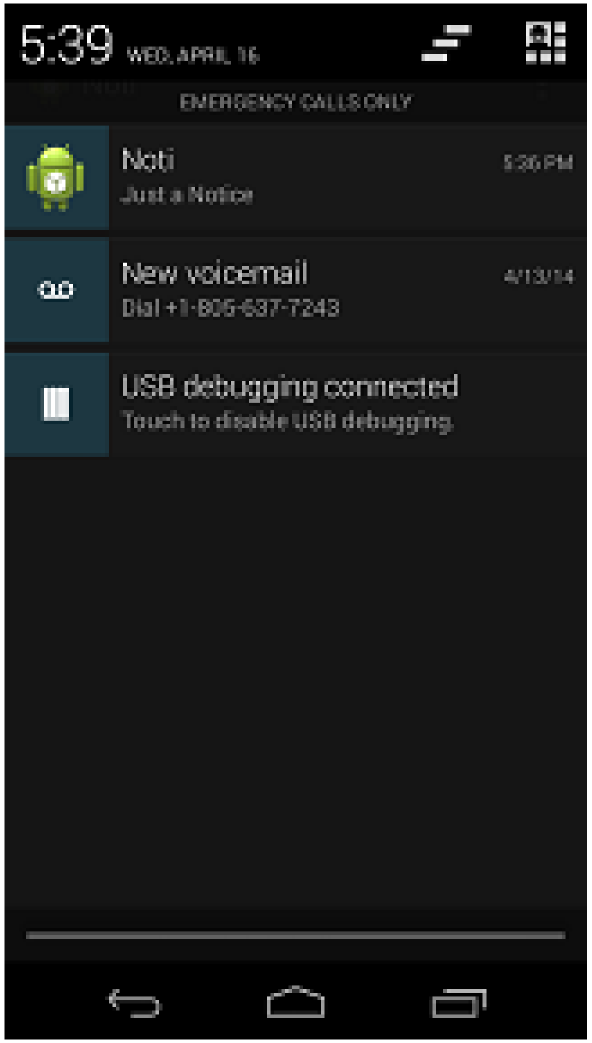
Messages and Information
That appear here →



Notifications

Status Bar
Settings, time, date

Notification Tray
Drag down from top


5:39 WED, APRIL 16

EMERGENCY CALLS ONLY

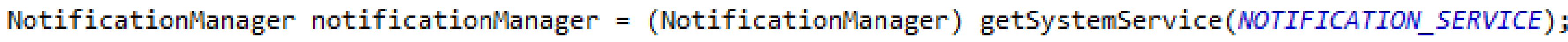
Noti 5:36 PM
Just a Notice

New voicemail 4/13/14
Dial +1-805-637-7243

USB debugging connected
Touch to disable USB debugging.

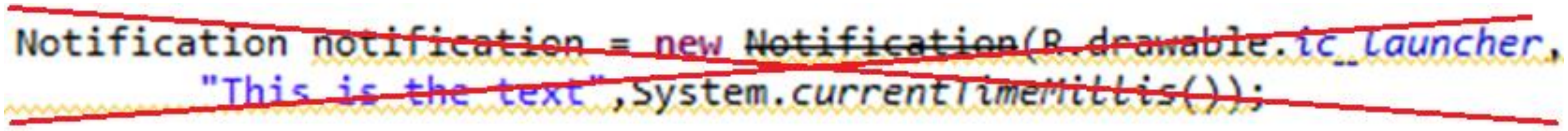
Notifications

- 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());
```

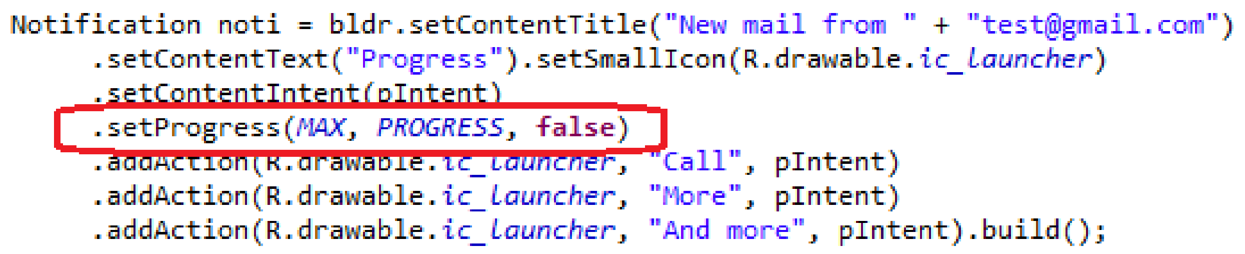
- Works but deprecated… Use a Builder instead

Notifications

- Builder Introduced in API 11 (3.0) although some methods added in API 16.
- Notifications have a lot of settings. Builder ensures 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();
```

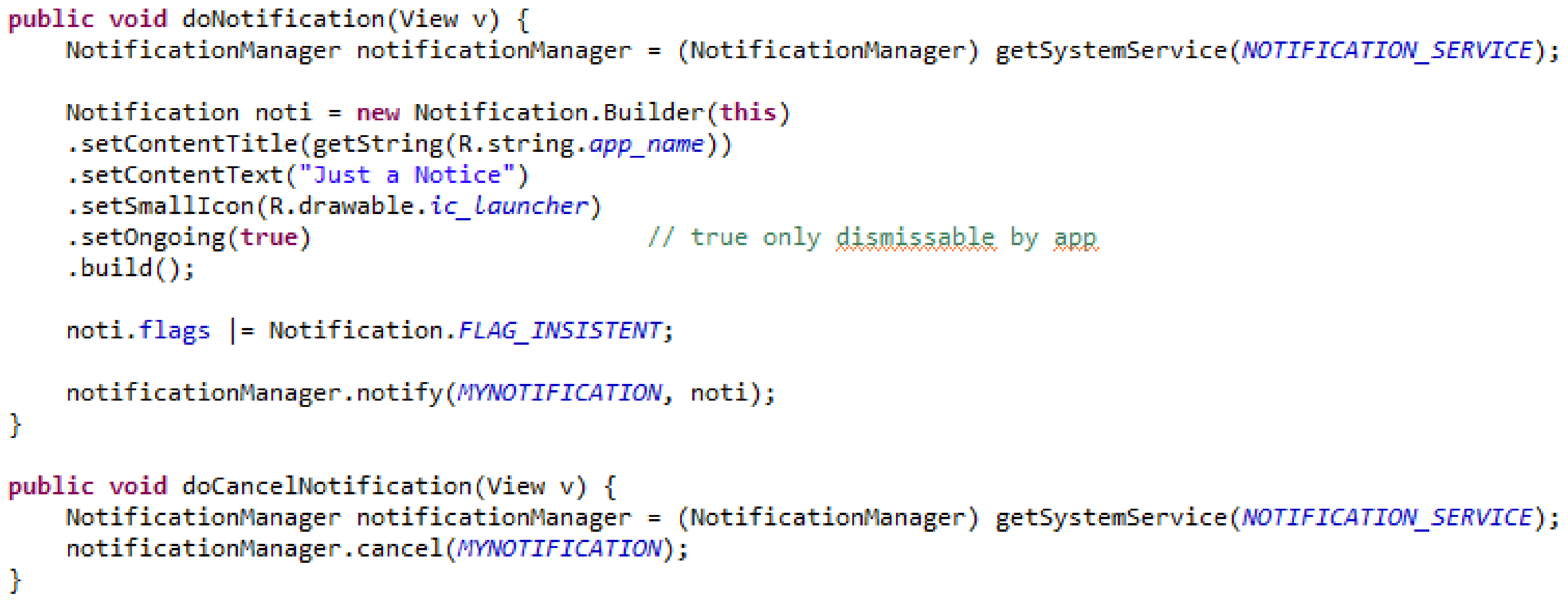
Notifications
Ongoing and Insistent

- 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);
}
```