

Alarm



Chapter 14

Do Trong Tuan

Introduction

AM 8:00

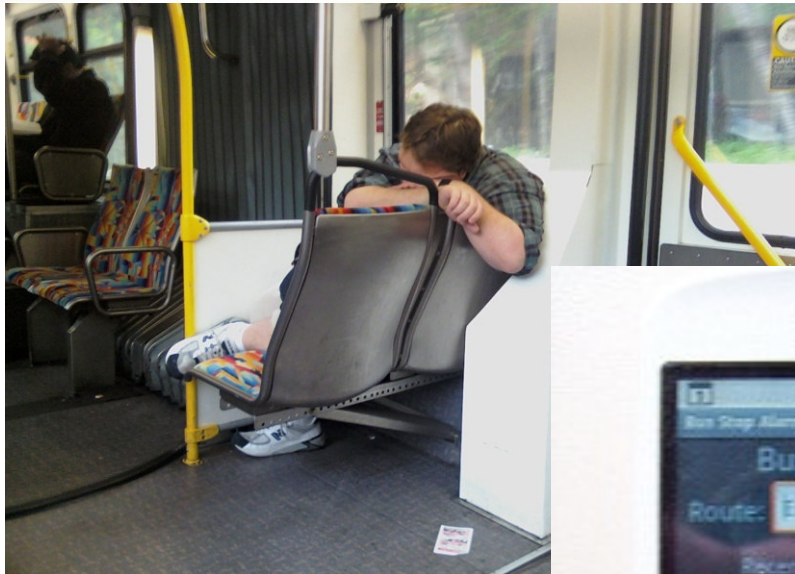


PM 2:30



AM 9:15

Introduction



Alarm Service

❖ Fires an Intent in the future

❖ Get it with

```
AlarmManager am = (AlarmManager)  
                    getSystemService(Context.ALARM_SERVICE);  
// set(int type, long triggerAtTime, PendingIntent operation);
```

❖ type is one of:

- ELAPSED_REALTIME
- ELAPSED_REALTIME_WAKEUP

SystemClock.elapsedRealTime()

- RTC
- RTC_WAKEUP

System.currentTimeMillis()



Alarm Service

❖ More methods

- `setRepeating(int type, long triggerAtTime, long interval, PendingIntent operation);`
 - Can use `INTERVAL_HOUR`, `INTERVAL_HALF_DAY`
- `cancel(PendingIntent operation);`
 - Match with `filterEquals(Intent anotherIntent);`

Alarm Manager

AlarmManager class provides access to the system alarm services. These allow you to schedule your application to be run at some point in the future. When an alarm goes off, the Intent that had been registered for it is broadcast by the system, automatically starting the target application if it is not already running.

Example: A Simple Alarm



In this exercise, a scheduled alarm of 10 seconds will start a service, MyAlarmService.

Example: A Simple Alarm

```
private PendingIntent pendingIntent;
```

```
buttonStart.setOnClickListener(new Button.OnClickListener() {
```

```
@Override
```

```
public void onClick(View arg0) {
```

```
// TODO Auto-generated method stub
```

```
Intent myIntent = new Intent(AndroidAlarmService.this, MyAlarmService.class);
```

```
pendingIntent = PendingIntent.getService(AndroidAlarmService.this, 0, myIntent, 0);
```

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

```
Calendar calendar = Calendar.getInstance();
```

```
calendar.setTimeInMillis(System.currentTimeMillis());
```

```
calendar.add(Calendar.SECOND, 10);
```

```
alarmManager.set(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(), pendingIntent);
```

```
Toast.makeText(AndroidAlarmService.this, "Start Alarm", Toast.LENGTH_LONG).show();
```

```
}});
```



Example: A Simple Alarm

```
buttonCancel.setOnClickListener(new Button.OnClickListener() {  
  
    @Override  
    public void onClick(View arg0) {  
        // TODO Auto-generated method stub  
        AlarmManager alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);  
        alarmManager.cancel(pendingIntent);  
  
        // Tell the user about what we did.  
        Toast.makeText(AndroidAlarmService.this, "Cancel!", Toast.LENGTH_LONG).show();  
  
    }  
});
```

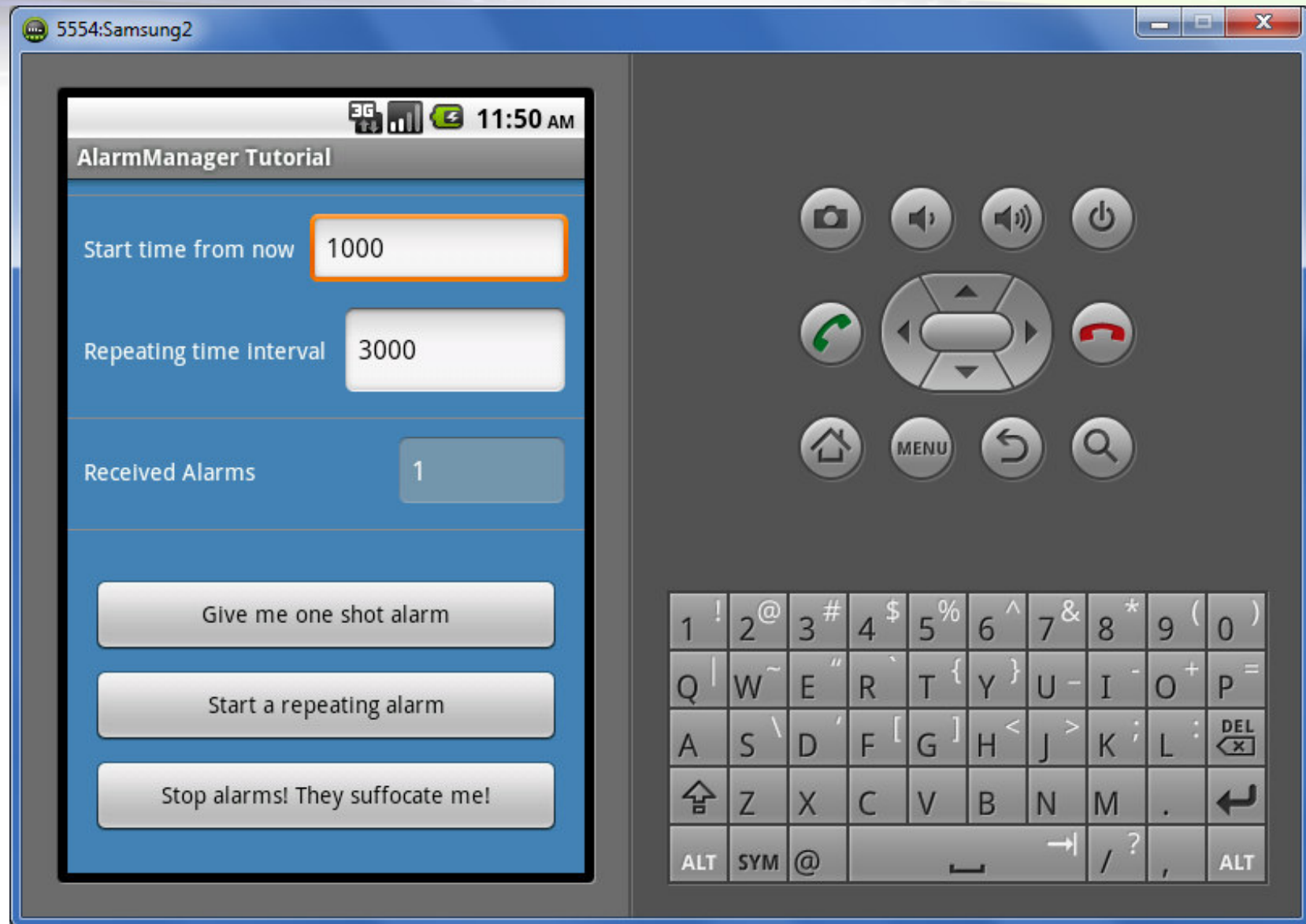


Example: A Simple Alarm

```
public class MyAlarmService extends Service {  
    @Override  
    public void onCreate() {  
        // TODO Auto-generated method stub  
        Toast.makeText(this, "MyAlarmService.onCreate()", Toast.LENGTH_LONG).show();  
    }  
  
    @Override  
    public IBinder onBind(Intent intent) {  
        // TODO Auto-generated method stub  
        Toast.makeText(this, "MyAlarmService.onBind()", Toast.LENGTH_LONG).show();  
        return null;  
    }  
  
    @Override  
    public void onStart(Intent intent, int startId) {  
        // TODO Auto-generated method stub  
        super.onStart(intent, startId);  
        Toast.makeText(this, "MyAlarmService.onStart()", Toast.LENGTH_LONG).show();  
    }  
}
```



Example: A Fancier Alarm



Main.xml

- ❖ `<?xml version="1.0" encoding="utf-8"?>`
- ❖ `<LinearLayout`
`xmlns:android="http://schemas.android.com/apk/res/android"`
- ❖ `android:orientation="vertical" android:layout_width="fill_parent"`
- ❖ `android:layout_height="fill_parent" android:background="#4682b4">`
- ❖ `<TextView android:layout_width="fill_parent"`
- ❖ `android:layout_height="1dp" android:background="#8b8989"`
- ❖ `android:layout_marginTop="10dp" />`

Main.xml

```
<TableRow android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:layout_marginTop="10dp">
<TextView android:id="@+id/textView1" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Start time from now"
    android:layout_marginLeft="10dp"
    android:textColor="#FFFFFF"></TextView>
<EditText android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:id="@+id/millsStartEditText"
    android:inputType="number" android:layout_weight="1"
    android:layout_marginLeft="10dp" android:layout_marginRight="10dp"
    android:hint="Mills from click" android:textSize="16dp">
    <requestFocus></requestFocus>
</EditText>
</TableRow>
```


Main.xml

```
<TableRow android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:layout_marginTop="10dp">
<TextView android:id="@+id/textView2" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Repeating time interval"
    android:layout_marginLeft="10dp"
    android:textColor="#FFFFFF"></TextView>
<EditText android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/millsIntervalsEditText"
    android:inputType="number" android:layout_weight="1"
    android:layout_marginLeft="10dp" android:layout_marginRight="10dp"
    android:textSize="16dp" android:hint="Mills for intervals"></EditText>
</TableRow>
<TextView android:layout_width="fill_parent"
    android:layout_height="1dp" android:background="#8b8989"
    android:layout_marginTop="10dp" />
```

Main.xml

```
<TableRow android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:layout_marginTop="10dp">
<TextView android:id="@+id/textView3" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Received Alarms"
    android:layout_marginLeft="10dp" android:layout_weight="1"
    android:textColor="#FFFFFF"></TextView>
<EditText android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:inputType="number"
    android:layout_weight="1"
    android:layout_marginLeft="10dp" android:layout_marginRight="10dp"
    android:textSize="16dp" android:text="0" android:enabled="false"
    android:id="@+id/alarmsReceivedEditText"
    android:textColor="#FFFFFF"></EditText>
</TableRow>
<TextView android:layout_width="fill_parent"
    android:layout_height="1dp" android:background="#8b8989"
    android:layout_marginTop="10dp" />
```

Main.xml

```
<LinearLayout android:layout_width="fill_parent"
    android:layout_height="fill_parent" android:orientation="vertical"
    android:gravity="center">
<Button android:id="@+id/oneShotAlarmButton"
    android:layout_height="wrap_content" android:text="Give me one shot alarm"
    android:layout_width="fill_parent" android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp" android:layout_marginTop="8dp"
    android:onClick="startOneShot"/>
<Button android:id="@+id/repeatingAlarmbutton"
    android:layout_height="wrap_content" android:text="Start a repeating alarm"
    android:layout_width="fill_parent" android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp" android:layout_marginTop="8dp"
    android:onClick="startRepeating" />
<Button android:id="@+id/stopAlarmbutton"
    android:layout_height="wrap_content" android:text="Stop alarms! They suffocate me!"
    android:layout_width="fill_parent" android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp" android:layout_marginTop="8dp"
    android:onClick="stopAlarms" />
</LinearLayout>
```

AlarmManagerTutorialActivity.java

```
private static final String ALARM_REFRESH_ACTION =  
    "apt.tutorial.alarmmanager.ALARM_REFRESH_ACTION";  
private static final int ALARM_CODE = 20;  
  
private static final long DEFAULT_START_DELAY = 1000;  
private static final long DEFAULT_INTERVAL_DELAY = 3000;  
  
private BroadcastReceiver alarmReceiver;  
private PendingIntent pendingIntent;  
  
private AlarmManager alarmManager;  
  
private int alarmCounted;  
  
// UI references  
private EditText millsStartEditText, millsIntervalEditText, alarmsReceived;
```

AlarmManagerTutorialActivity.java

// The handler that manage the UI updates

```
private Handler myHandler = new Handler() {  
    public void handleMessage(Message msg) {  
        switch (msg.what) {  
            case ALARM_CODE:  
                alarmsReceived.setText(String.valueOf(alarmCounted));  
                Toast.makeText(AlarmManagerTutorialActivity.this,  
                    "Alarm Received", Toast.LENGTH_SHORT).show();  
                break;  
            default:  
                break;  
        }  
    }  
};
```


AlarmManagerTutorialActivity.java

❖ onCreate

```
super.onCreate(savedInstanceState);  
setContentView(R.layout.main);  
// We retrieve UI references  
millsStartEditText = (EditText) findViewById(R.id.millsStartEditText);  
millsIntervalEditText = (EditText)  
    findViewById(R.id.millsIntervalsEditText);  
alarmsReceived = (EditText)  
    findViewById(R.id.alarmsReceivedEditText);
```

AlarmManagerTutorialActivity.java

❖ onStart

```
super.onStart();
```

```
// We get the AlarmManager
```

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

```
// We set default parameters for alarms
```

```
millsStartEditText.setText(String.valueOf(DEFAULT_START_DELAY));  
millsIntervalEditText.setText(String.valueOf(DEFAULT_INTERVAL_DE  
    LAY));
```

```
// We prepare the pendingIntent for the AlarmManager
```

```
Intent intent = new Intent(ALARM_REFRESH_ACTION);  
pendingIntent = PendingIntent.getBroadcast(this, 0, intent,  
    PendingIntent.FLAG_CANCEL_CURRENT);
```

AlarmManagerTutorialActivity.java

onStart

// We create and register a broadcast receiver for alarms

```
alarmReceiver = new BroadcastReceiver() {
```

```
    @Override
```

```
    public void onReceive(Context context, Intent intent) {
```

```
        // We increment received alarms
```

```
        alarmCounted++;
```

```
        // We notify to handler the arrive of alarm
```

```
        Message msg = myHandler.obtainMessage(ALARM_CODE,  
        intent);
```

```
        myHandler.sendMessage(msg);
```

```
    }
```

```
};
```

// We register dataReceiver to listen *ALARM_REFRESH_ACTION*

```
IntentFilter filter = new IntentFilter(ALARM_REFRESH_ACTION);
```

```
registerReceiver(alarmReceiver, filter);
```



AlarmManagerTutorialActivity.java

```
public void startOneShot(View v) {  
    // We get value for one shot alarm  
    int startTime = Integer.parseInt(millsStartEditText.getText()  
        .toString());  
    // We have to register to AlarmManager  
    Calendar calendar = Calendar.getInstance();  
    calendar.setTimeInMillis(System.currentTimeMillis());  
    calendar.add(Calendar.MILLISECOND, startTime);  
    // We set a one shot alarm  
    alarmManager.set(AlarmManager.RTC_WAKEUP, calendar  
        .getTimeInMillis(), pendingIntent);  
}
```

AlarmManagerTutorialActivity.java

```
public void startRepeating(View v) {  
    // We get value for repeating alarm  
    int startTime =  
        Integer.parseInt(millsStartEditText.getText().toString());  
    long intervals =  
        Long.parseLong(millsIntervalEditText.getText().toString());  
    // We have to register to AlarmManager  
    Calendar calendar = Calendar.getInstance();  
    calendar.setTimeInMillis(System.currentTimeMillis());  
    calendar.add(Calendar.MILLISECOND, startTime);  
    // We set a repeating alarm  
    alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, calendar  
        .getTimeInMillis(), intervals, pendingIntent);  
}
```


AlarmManagerTutorialActivity.java

```
public void stopAlarms(View v) {  
    // We cancel alarms that matches with pending intent  
    alarmManager.cancel(pendingIntent);  
    Toast.makeText(this, "Alarms stopped",  
        Toast.LENGTH_SHORT).show();  
}
```

Tutorials

1. TUTORIAL 19 : Is it Luchtime yet ?
2. TUTORIAL 20 : More Subtle Lunch Alarms ?

End of Lecture

