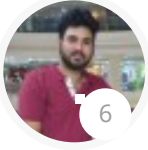




+91 9008 91550 | srinivas@androhub.com

[HOME](#) | [ANDROID BASICS](#) | [ANDROID ALARMMANAGER](#)

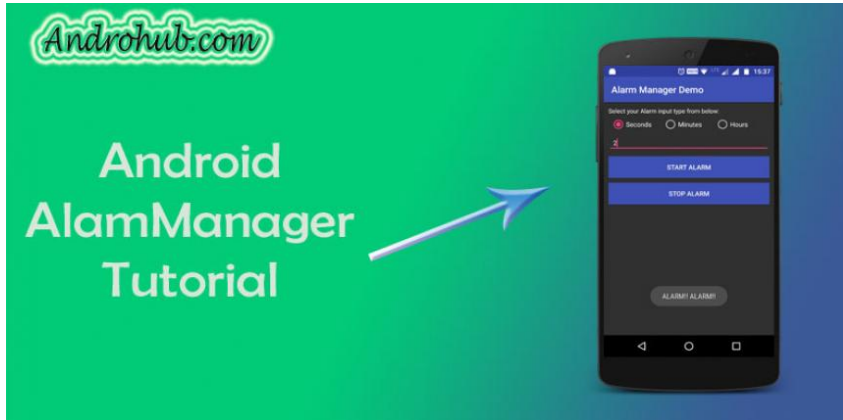


Android AlarmManager

April 10, 2017

[Android Basics](#)

By Dr. Droid



Daily we used to set Alarms in our phone to wake up in morning or for some meetings, etc. I used to set lot of alarms for morning but unfortunately i used to wake up late only. 😊 So today we are going to learn about Alarms. [Alarms](#) (based on the [AlarmManager](#) class) give you a way to perform time-based operations outside the lifetime of your application. For example, you could use an alarm to initiate a long-running operation, such as starting a service once a day to download a weather forecast. You can schedule your application to run at a specific time in the future. It works whether your phone is running or not.

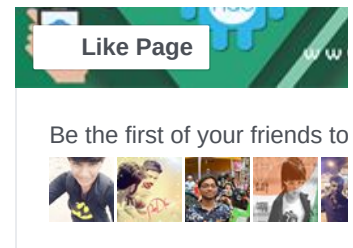
AlarmManager is having different characteristics mentioned below:

- AlarmManager runs outside the lifetime of your application. Once an alarm is scheduled, it will invoke even when your application is not running or in sleep mode.
- An scheduled alarm will execute unless it is stopped explicitly by calling **cancel()** method, or until device reboots.
- All scheduled alarms will be stopped when device reboots. This means, you need to re-schedule them explicitly when device boot completes.
- AlarmManger fires an Intent at given intervals. This can be used along with broadcast receivers to start a service to perform network operations.
- AlarmManager is different form java **Timer** and **TimerTask**.



[androhub.co](#)

242 likes



Subscribe to our mailing list and get interesting stuff and updates to your email inbox. We don't Spam.

Enter your email here

SIGN UP NOW

We respect your privacy and take protecting it seriously.

RECENT POSTS

- [Android Confirm Credential \(Security Lock\)](#)
- [Sectioned RecyclerView](#)

Android supports two clock types for alarm service; **Elapsed real time** and **real time clock (RTC)**. Elapsed real time uses the time since device last booted. Real time clock (RTC) uses UTC time for alarm service clock. RTC is most commonly used for setting alarm service in android. The following example, using RTC to schedule alarm.

Prerequisite:

1. [Radio Button](#).
2. [Notification Manager](#).
3. [Services](#).
4. [Broadcast Receiver](#).

Example

In this tutorial, we will learn how to use AlarmManager that runs after a specific time provided by user.

VIDEO DEMO

Android AlarmManager Demo



1. Create a new project in Android Studio by navigating to **File ⇒ New ⇒ New Project** and fill required details. By default my activity is *MainActivity.java*.

2. Open **res ⇒ values ⇒ strings.xml** and add below string values. These are some strings that we are going to use in our project.

```
1 <resources>
2     <string name="app_name">Alarm Manager Demo</string>
3     <string name="stop_alarm">Stop Alarm</string>
```

- [Android Material Design working with Tabs \(Advanced\)](#)

- [Swipe Dismiss](#)

[CardView with Coordinator Layout](#)

- [Swipe Dismiss](#)

[RecyclerView](#)

RECENT COMMENTS

- [Jagdish on Working with Android Marshmallow 6.0 Permissions](#)

- [Jagdish on Android Download Files & Save](#)

- [May 2017](#)

- [April 2017](#)

- [March 2017](#)

- [February 2017](#)

- [January 2017](#)

- [August 2016](#)

- [July 2016](#)

```

4     <string name="start_alarm">Start Alarm</string>
5     <string name="edit_text_hint">Input your Alarm time
6     <string name="hours">Hours</string>
7     <string name="minutes">Minutes</string>
8     <string name="seconds">Seconds</string>
9     <string name="label">Select your Alarm input type f
10 </resources>

```

3. Open your *activity_main.xml* layout and place the below code to it. The layout contains some RadioButton, Edit Text and Button.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:orientation="vertical"
7      android:padding="10dp"
8      tools:context="com.alarmmanager_demo.MainActivity">
9
10     <TextView
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="@string/label"
14         android:textColor="@android:color/white"
15         android:textSize="13sp" />
16
17     <RadioGroup
18         android:id="@+id/time_type_selection_radio_grou
19         android:layout_width="match_parent"
20         android:layout_height="wrap_content"
21         android:gravity="center"
22         android:orientation="horizontal"
23         android:padding="5dp">
24
25         <RadioButton
26             android:id="@+id/seconds_radio_button"
27             android:layout_width="match_parent"
28             android:layout_height="wrap_content"
29             android:layout_weight="1"
30             android:checked="true"
31             android:gravity="center_vertical"
32             android:text="@string/seconds"
33             android:textColor="@android:color/white" />
34
35         <RadioButton
36             android:id="@+id/minutes_radio_button"
37             android:layout_width="match_parent"
38             android:layout_height="wrap_content"
39             android:layout_weight="1"
40             android:gravity="center_vertical"
41             android:text="@string/minutes"
42             android:textColor="@android:color/white" />
43
44         <RadioButton
45             android:id="@+id/hours_radio_button"
46             android:layout_width="match_parent"
47             android:layout_height="wrap_content"
48             android:layout_weight="1"
49             android:gravity="center_vertical"
50             android:text="@string/hours"
51             android:textColor="@android:color/white" />
52     </RadioGroup>

```

- June 2016
- April 2016
- March 2016
- December 2015
- November 2015
- October 2015
- September 2015
- August 2015
- July 2015
- June 2015
- May 2015
- April 2015

CATEGORIES

- Android ActionBar
- Android Advanced
- Android Animations
- Android
- Authentication
- Android AutoComplete
- Android Basics
- Android Bluetooth
- Android CardView
- Android Clocks
- Android Contacts
- Android Content
- Provider
- Android Custom Font
- Android Custom View
- Android Database
- Android Dialogs
- Android D
- Files
- Android Drawer

```

53
54     <EditText
55         android:id="@+id/input_interval_time"
56         android:layout_width="match_parent"
57         android:layout_height="wrap_content"
58         android:hint="@string/edit_text_hint"
59         android:inputType="number"
60         android:padding="10dp"
61         android:textColor="@android:color/white"
62         android:textColorHint="@android:color/darker_gr
63         android:textSize="14sp" />
64
65     <Button
66         android:id="@+id/start_alarm_button"
67         android:layout_width="match_parent"
68         android:layout_height="wrap_content"
69         android:layout_marginTop="10dp"
70         android:background="@color/colorPrimary"
71         android:text="@string/start_alarm"
72         android:textColor="@android:color/white"
73         android:textSize="14sp" />
74
75     <Button
76         android:id="@+id/stop_alarm_button"
77         android:layout_width="match_parent"
78         android:layout_height="wrap_content"
79         android:layout_marginTop="10dp"
80         android:background="@color/colorPrimary"
81         android:text="@string/stop_alarm"
82         android:textColor="@android:color/white"
83         android:textSize="14sp" />
84 </LinearLayout>

```

4. Now create Broadcast Receiver java class naming

AlarmReceiver.java which will invoke by **AlarmManger**. Here you can do your task like I am running AlarmSound by starting a service and also displaying a Notification.

```

1  package com.alarmanager_demo;
2
3  import android.content.BroadcastReceiver;
4  import android.content.ComponentName;
5  import android.content.Context;
6  import android.content.Intent;
7  import android.widget.Toast;
8
9  import static android.support.v4.content.WakefulBroadca
10
11  /**
12   * Created by sonu on 09/04/17.
13   */
14
15  public class AlarmReceiver extends BroadcastReceiver {
16      @Override
17      public void onReceive(Context context, Intent inten
18          Toast.makeText(context, "ALARM!! ALARM!!", Toas
19
20          //Stop sound service to play sound for alarm
21          context.startService(new Intent(context, Alarms
22
23          //This will send a notification message and sho
24          ComponentName comp = new ComponentName(context.
25              AlarmNotificationService.class.getName(

```

- Android

ExpandableListView

- Android File

- Android FingerPrint

- Android Fragment

- Android Gestures

- Android GPS

- Android Image Picker

- Android Internet

Connection

- Android JSON

- Android ListView

- Android Material

Design

- Android

NavigationDrawer

- Android Notification

Bar

- Android Palette

- Android ProgressBar

- Android RatingBar

- Android RecyclerView

- Android SeekBar

- Android Spinner

- Android StackView

- Android SwipeRefresh

Layout

- Android Tabs

- Android TextSwitcher

- Android ToolBar

- Android U^T Design

- Android V tor

- Android Viewpager

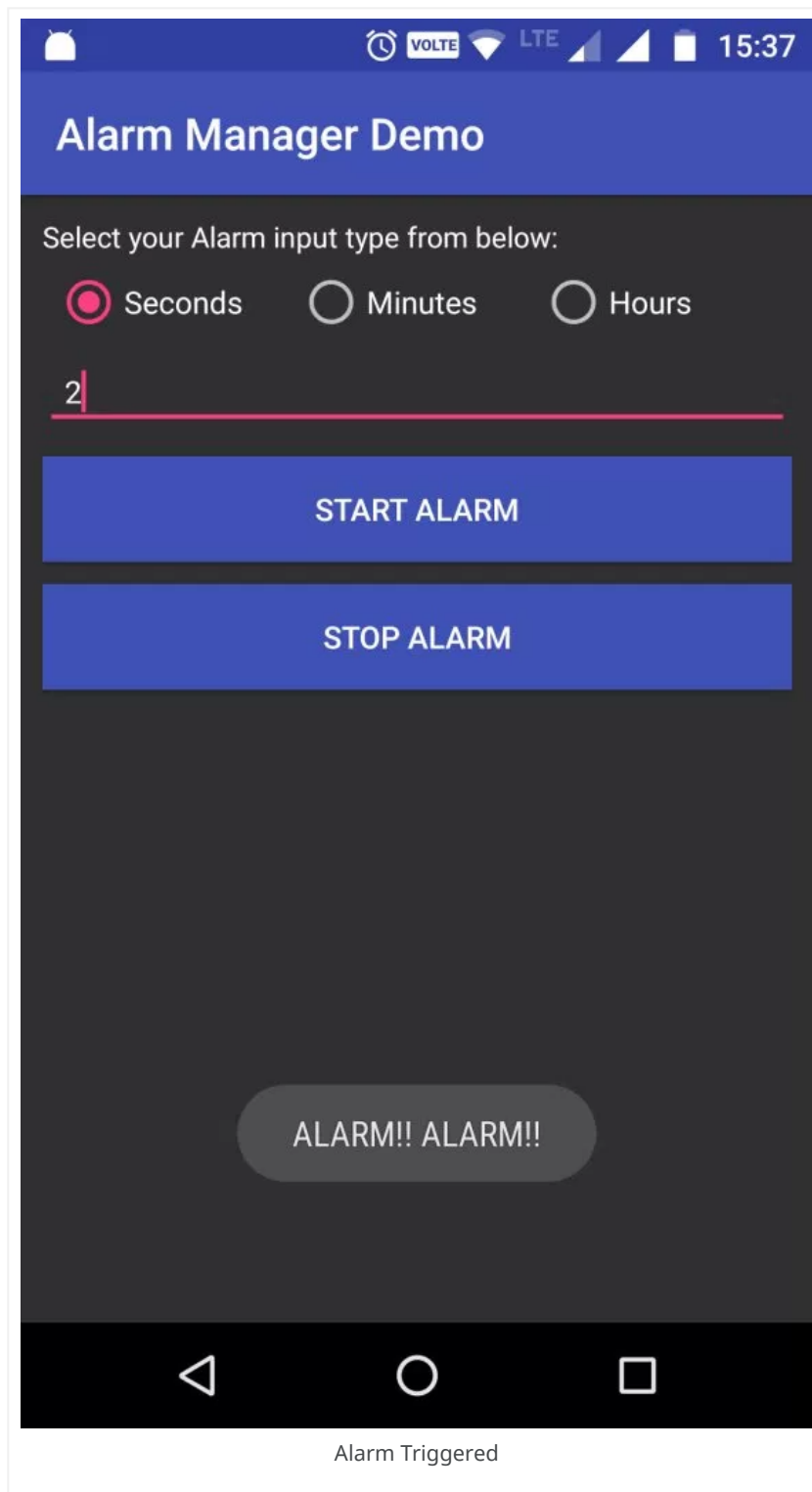
- Android ViewSwitcher

```

26         startWakefulService(context, (intent.setComponent
27
28     }
29
30
31 }

```

- Android WIFI
- Google Maps
- Http Requests
- Intent



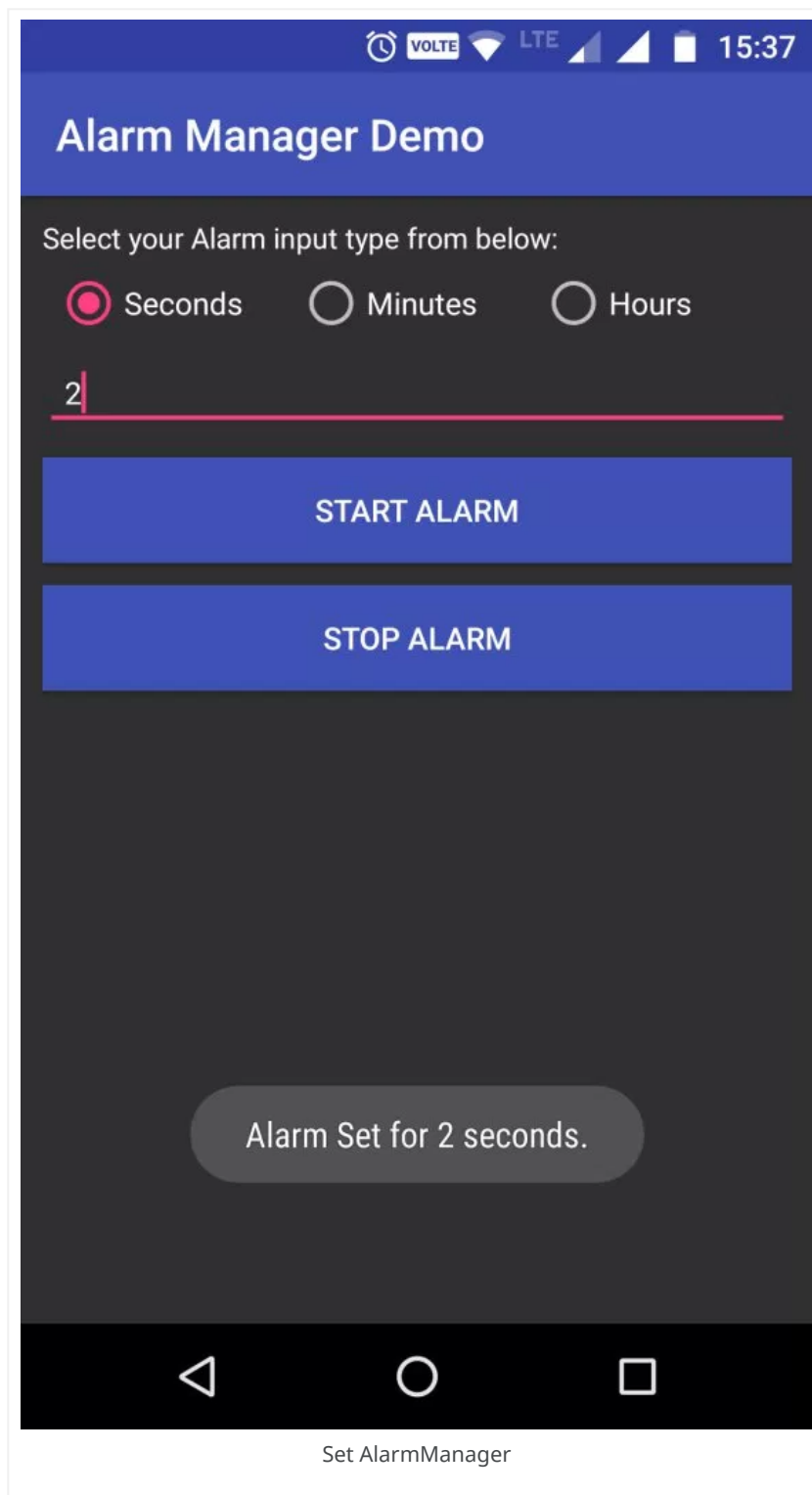
5. Let us have a look into *MainActivity.java* file. In this class, we have defined two simple methods *triggerAlarmManager(int alarmTriggerTime)* and *stopAlarmManager()*.

- **Retrieve PendingIntent** : These two lines of use to get the PendingIntent of *AlarmReceiver* broadcast receiver.

```
1      /* Retrieve a PendingIntent that will perform a broa
2          Intent alarmIntent = new Intent(MainActivity.thi
3          pendingIntent = PendingIntent.getBroadcast(MainA
```

- **Trigger AlarmManager** : *triggerAlarmManager(int alarmTriggerTime)* method use to set the AlarmManager at particular time that we pass in parameter.

```
1      //Trigger alarm manager with entered time interval
2      public void triggerAlarmManager(int alarmTriggerTim
3          // get a Calendar object with current time
4          Calendar cal = Calendar.getInstance();
5          // add alarmTriggerTime seconds to the calendar
6          cal.add(Calendar.SECOND, alarmTriggerTime);
7
8          AlarmManager manager = (AlarmManager) getSystem
9          manager.set(AlarmManager.RTC_WAKEUP, cal.getTim
10
11          Toast.makeText(this, "Alarm Set for " + alarmTr
12      }
```



- **Stop AlarmManager** :*stopAlarmManager()* method use to stop any running alarm. As I am playing sound and displaying notification as well so i am stopping the sound and cancelling the notification respectively.

```
1 //Stop/Cancel alarm manager
2 public void stopAlarmManager() {
3
4     AlarmManager manager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
5     manager.cancel(pendingIntent); //cancel the alarm
6
7
8     //Stop the Media Player Service to stop sound
```

```

9         stopService(new Intent(MainActivity.this, Alarm
10
11         //remove the notification from notification tra
12         NotificationManager notificationManager = (Noti
13         .getSystemService(Context.NOTIFICATION_
14         notificationManager.cancel(AlarmNotificationSer
15
16         Toast.makeText(this, "Alarm Canceled/Stop by Us
17     }

```

Final Source Code

MainActivity.java

```

2
3  import android.app.AlarmManager;
4  import android.app.NotificationManager;
5  import android.app.PendingIntent;
6  import android.content.Context;
7  import android.content.Intent;
8  import android.os.Bundle;
9  import android.support.v7.app.AppCompatActivity;
10 import android.view.View;
11 import android.widget.EditText;
12 import android.widget.RadioButton;
13 import android.widget.Toast;
14
15 public class MainActivity extends AppCompatActivity {
16
17     //Pending intent instance
18     private PendingIntent pendingIntent;
19
20     private RadioButton secondsRadioButton, minutesRad
21
22     //Alarm Request Code
23     private static final int ALARM_REQUEST_CODE = 133;
24
25
26     @Override
27     protected void onCreate(Bundle savedInstanceState) {
28         super.onCreate(savedInstanceState);
29         setContentView(R.layout.activity_main);
30
31         //Find id of all radio buttons
32         secondsRadioButton = (RadioButton) findViewById
33         minutesRadioButton = (RadioButton) findViewById
34         hoursRadioButton = (RadioButton) findViewById
35
36         /* Retrieve a PendingIntent that will perform
37         Intent alarmIntent = new Intent(MainActivity.t
38         pendingIntent = PendingIntent.getBroadcast(Mai
39
40         //Find id of Edit Text
41         final EditText editText = (EditText) findViewB
42
43         //Set On Click over start alarm button
44         findViewById(R.id.start_alarm_button).setOnCli
45         @Override
46         public void onClick(View v) {
47             String getInterval = editText.getText(
48
49             //check interval should not be empty a
50             if (!getInterval.equals("") && !getInt
51                 //finally trigger alarm manager
52                 triggerAlarmManager(getTimeInterva
53

```



```

54         }
55     });
56
57     //set on click over stop alarm button
58     findViewById(R.id.stop_alarm_button).setOnClickListener
59         @Override
60         public void onClick(View v) {
61             //Stop alarm manager
62             stopAlarmManager();
63         }
64     });
65
66 }
67
68 //get time interval to trigger alarm manager
69 private int getTimeInterval(String getInterval) {
70     int interval = Integer.parseInt(getInterval);
71
72     //Return interval on basis of radio button sel
73     if (secondsRadioButton.isChecked())
74         return interval;
75     if (minutesRadioButton.isChecked())
76         return interval * 60; //convert minute into
77     if (hoursRadioButton.isChecked()) return inter
78
79     //else return 0
80     return 0;
81 }
82
83
84 //Trigger alarm manager with entered time interval
85 public void triggerAlarmManager(int alarmTriggerTi
86     // get a Calendar object with current time
87     Calendar cal = Calendar.getInstance();
88     // add alarmTriggerTime seconds to the calenda
89     cal.add(Calendar.SECOND, alarmTriggerTime);
90
91     AlarmManager manager = (AlarmManager) getSystem
92     manager.set(AlarmManager.RTC_WAKEUP, cal.getTi
93
94     Toast.makeText(this, "Alarm Set for " + alarmT
95 }
96
97 //Stop/Cancel alarm manager
98 public void stopAlarmManager() {
99
100     AlarmManager manager = (AlarmManager) getSystem
101     manager.cancel(pendingIntent); //cancel the ala
102
103
104     //Stop the Media Player Service to stop sound
105     stopService(new Intent(MainActivity.this, Alar
106
107     //remove the notification from notification tr
108     NotificationManager notificationManager = (Not
109         .getSystemService(Context.NOTIFICATION
110     notificationManager.cancel(AlarmNotificationSe
111
112     Toast.makeText(this, "Alarm Canceled/Stop by U
113 }
114 }

```

6. As we started sound service in *AlarmReceiver.java* class, so now create a *AlarmSoundService.java* class extending with *Service*. This

class will use to play the alarm sound in **onCreate()** and stop the sound in **onDestroy()** when user stops the Alarm.

You can download the Alarm Sound mp3 file from this [link](#). Its a good website for Alarm Sounds.

```
1 package com.alarmanager_demo;
2
3 import android.app.Service;
4 import android.content.Intent;
5 import android.media.MediaPlayer;
6 import android.os.IBinder;
7 import android.support.annotation.Nullable;
8
9 /**
10  * Created by sonu on 10/04/17.
11  */
12
13 public class AlarmSoundService extends Service {
14     private MediaPlayer mediaPlayer;
15
16     @Nullable
17     @Override
18     public IBinder onBind(Intent intent) {
19         return null;
20     }
21
22     @Override
23     public void onCreate() {
24         super.onCreate();
25
26         //Start media player
27         mediaPlayer = MediaPlayer.create(this, R.raw.al
28         mediaPlayer.start();
29         mediaPlayer.setLooping(true); //set looping true
30     }
31
32     @Override
33     public void onDestroy() {
34         super.onDestroy();
35
36         //On destory stop and release the media player
37         if (mediaPlayer != null && mediaPlayer.isPlaying)
38             mediaPlayer.stop();
39             mediaPlayer.reset();
40             mediaPlayer.release();
41     }
42 }
43 }
```

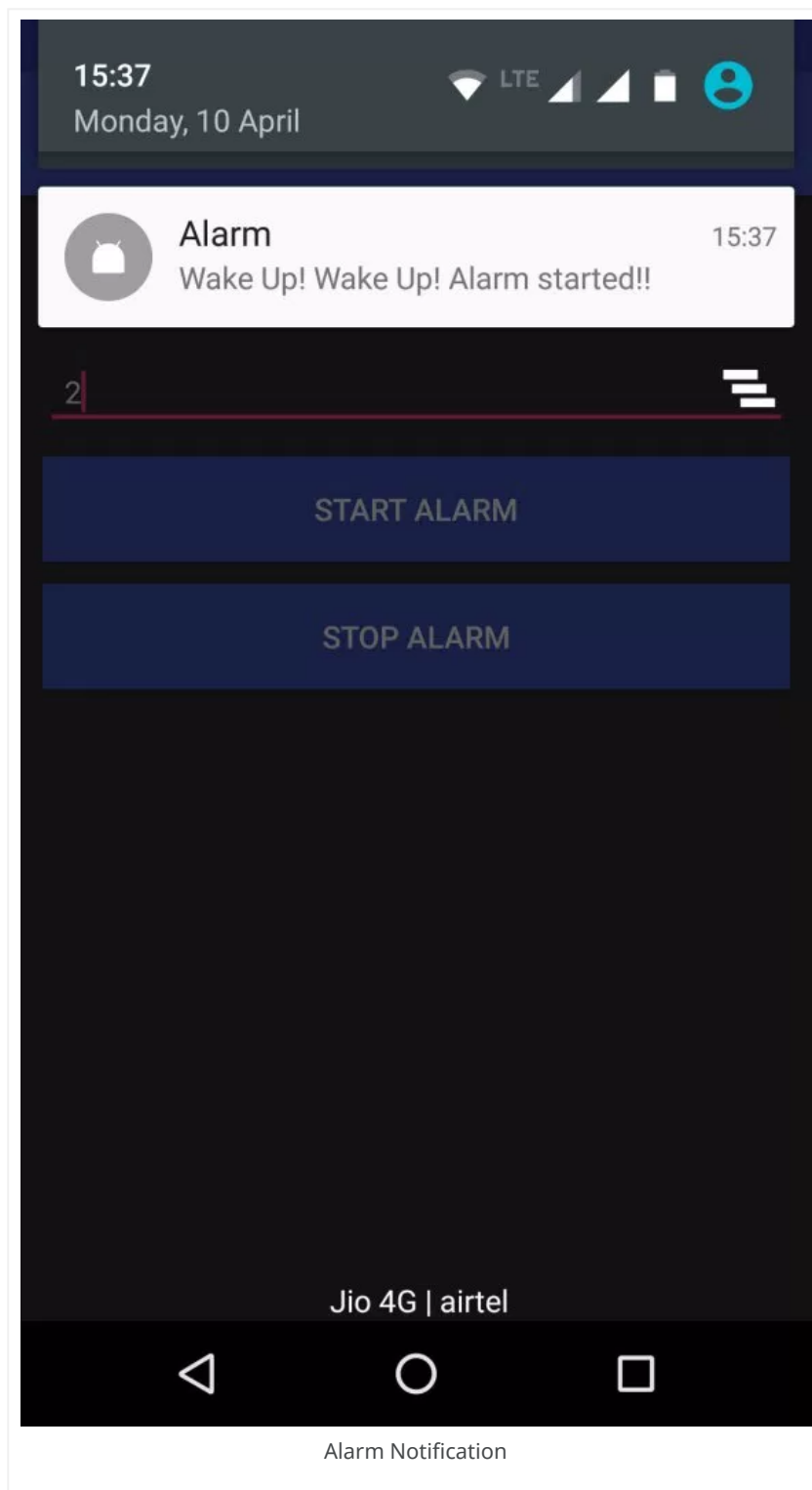
7. Also we are displaying Notification when Alarm starts so lets create new java class naming *AlarmNotificationService.java* and extend it with [IntentService](#). This class used to show notification.

```
1 package com.alarmanager_demo;
2
3 import android.app.IntentService;
4 import android.app.NotificationManager;
5 import android.app.PendingIntent;
6 import android.content.Context;
7 import android.content.Intent;
```

```

8  import android.support.v4.app.NotificationCompat;
9
10 /**
11  * Created by sonu on 10/04/17.
12  */
13
14 public class AlarmNotificationService extends IntentSer
15     private NotificationManager alarmNotificationManage
16
17     //Notification ID for Alarm
18     public static final int NOTIFICATION_ID = 1;
19
20     public AlarmNotificationService() {
21         super("AlarmNotificationService");
22     }
23
24     @Override
25     public void onHandleIntent(Intent intent) {
26
27         //Send notification
28         sendNotification("Wake Up! Wake Up! Alarm start
29     }
30
31     //handle notification
32     private void sendNotification(String msg) {
33         alarmNotificationManager = (NotificationManager
34             .getSystemService(Context.NOTIFICATION_
35
36         //get pending intent
37         PendingIntent contentIntent = PendingIntent.get
38             new Intent(this, MainActivity.class), P
39
40         //Create notification
41         NotificationCompat.Builder alamNotificationBuil
42             this).setContentTitle("Alarm").setSmall
43             .setStyle(new NotificationCompat.BigTex
44             .setContentText(msg).setAutoCancel(true
45         alamNotificationBuilder.setContentIntent(conten
46
47         //notiy notification manager about new notifica
48         alarmNotificationManager.notify(NOTIFICATION_ID
49     }
50
51
52 }

```



8. Also don't forget to declare BroadcastReceiver, Service and IntentService in Manifest file and as we are showing notification with WAKE_LOCK so use the **android.permission.WAKE_LOCK** permission. Your *AndroidManifest.xml* will look like as below.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk
3     package="com.alarmmanager_demo">
4
5     <!-- Permission required to Waking Lock if device
6     <uses-permission android:name="android.permission.W
```

```

7
8     <application
9         android:allowBackup="true"
10        android:icon="@mipmap/ic_launcher"
11        android:label="@string/app_name"
12        android:roundIcon="@mipmap/ic_launcher_round"
13        android:supportsRtl="true"
14        android:theme="@style/AppTheme">
15        <activity android:name=".MainActivity">
16            <intent-filter>
17                <action android:name="android.intent.ac
18
19                <category android:name="android.intent.
20            </intent-filter>
21        </activity>
22
23        <!-- Declare Alarm Broadcast receiver here -
24        <receiver android:name=".AlarmReceiver" />
25
26        <!-- Declare Alarm Notification Service here
27        <service android:name=".AlarmNotificationService
28
29        <!-- Declare Alarm Sound Service here -->
30        <service android:name=".AlarmSoundService" />
31    </application>
32
33 </manifest>

```

9. Finally all done, now you can also make apps with AlarmManager.

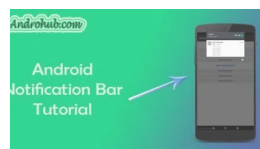
Thanks. 😊

[Download Source Code - >](#)

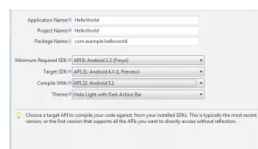
Share this:



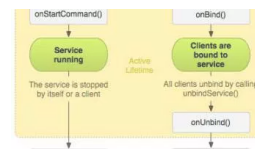
Related



Android Notification Bar
In "Android Notification Bar"



Hello World Example
In "Android Basics"



Android Services
In "Android Basics"

ABOUT THE AUTHOR



I am an Android developer for whom Android is the one and only love. I work as a freelancer which gets me and Android closer. I always love to share my learnings and love to learn from others knowledge too.

COMMENTS (6)



By Crsitofer

June 27, 2017at 1:48 pm [Reply](#)

What Android OS version are you using? because I added a reminder for one of my app, it is working well in Pre Nougat versión but in Nougat version the dayInterval is not working!



By [Dr. Droid](#)

June 27, 2017at 1:59 pm [Reply](#)

Hi Cristofer,

I am using Marshmallow OS Version for testing. I have not tested in Nougat OS Devices.

Thanks.



By Vijay

August 17, 2017at 5:00 pm [Reply](#)

How to trigger the alarm for every 5 seconds. I want to hit my server every 5 secs. Can you give me sample for that?



By [Dr. Droid](#)

August 18, 2017at 5:11 am [Reply](#)

Hi Vijay,

Please see the article by clicking below link:
<http://www.androhub.com/android-image-slider-using-viewpager/>

In this article I am using TIMER inside MainActivity.java to slide the images after 3 sec. So you can also use that TIMER

code to hit your server after 5 sec by changing the value.

Thanks



By vikas
joshi

September 24, 2017 at 11:19 am [Reply](#)

can you tell me
how to can pick image compressed to gallery
like whats up profile set gallery



By Dr.
Droid

September 25, 2017 at 5:59 am [Reply](#)

Hi Vikas Joshi,

For picking image from gallery see this article :
<http://www.androhub.com/select-and-share-multiple-images/>

To compress image visit this link :
<https://stackoverflow.com/questions/18545246/how-to-compress-image-size>

Thanks

LEAVE A COMMENT

Comment (required)



Object in the image



Name (required)

Email (required)

SENT

PREVIOUS POST

NEXT POST

Copyright © 2015 - 2017 Androhub.com | All rights reserved.

Powered by [PAC Developers](#)