**Simple Android Service Example**

**What is Android Service ?**

**Service is another component of an Android application, which does not provide any type of GUI. It is a program that can run in the background for an indefinite period. That means if we want to do a long operation (example: download data from internet), then we need to create an Android service for this purpose. Please keep in your mind that Android Service is neither a Thread or nor a Process. I have written a very good article on basic Android fundamental/components and their lifecycle (Activity, Service, Broadcast Receiver, Content Provider etc). You might like that article. Please click here to read.**

**Generally There are two types of Android Services.**

**Started Service (Unbounded)**

**This type of service is created and called by Android Activities. There is no 2 way communication between Android Activity and Service. The Activity just started the service and does not care about the status of the service.**

**Bound Service (Bounded)**

**This type of Android Service is for two-way communication. Suppose an Android Activity has started a bound service, then an Activity can be notified the status at any given time from the service. A bound service is a client-server interface that allows two-way communication, send requests, get results, and even work for across processes using interprocess communication (IPC)**

**Create/Start /Stop an Android Service from an Activity.**

**Step 1: Create an Android Service**

**Create an Android service class (Example: MyService) by extending the Android inbuilt class Service like below.**

import android.app.Service;

import android.content.Intent;

import android.os.IBinder;

import android.util.Log;

import android.widget.Toast;

public class MyService extends Service{

}

**Step 2: Override Android Service Functions**

@Override/implement below virtual functions in your class like below.

@Override

public IBinder onBind(Intent arg0) {

return null;

}

@Override

public void onCreate() {

Toast.makeText(this, "Congrats! MyService Created", Toast.LENGTH\_LONG).show();

Log.d(TAG, "onCreate");

}

@Override

public void onStart(Intent intent, int startId) {

Toast.makeText(this, "My Service Started", Toast.LENGTH\_LONG).show();

Log.d(TAG, "onStart");

}

@Override

public void onDestroy() {

Toast.makeText(this, "MyService Stopped", Toast.LENGTH\_LONG).show();

Log.d(TAG, "onDestroy");

}

**Step 3: Register Android Service class in the Manifest.xml file**

<service android:enabled="true" android:name=".MyService" />

**Step 4: Start Android Service using startService() function**

**Start the service from an Activity:**

//start the service from here //MyService is your service class name

startService(new Intent(this, MyService.class));

**Step 5: Stop Android Service using stopService() function**

**Stop the service from an Activity:**

//Stop the running service from here//MyService is your service class name

//Service will only stop if it is already running.

stopService(new Intent(this, MyService.class));

**Project Creation: Android Service Example (Complete Code)**

**To make this example very simple, we just add 2 buttons (‘Start Service’ and ‘StopService’) on the main Activity GUI, We will start a new service when click on the button Start Service and stop the running service when click on the button Stop Service and show the service status using an Android toast message.**

**1. Create a new project with project name: ServiceExample**

**2. Fill Application name: ServiceExample**

**3. Fill Package Name as: com.mm214.serviceexample**

**4. Add below xml file (activity\_main.xml) into your project’s res/layout folder. or you can copy the xml file contents.**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity" >

<TextView

android:id="@+id/text\_view\_header"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:gravity="center\_horizontal"

android:textColor="#FF0000"

android:textSize="20sp"

android:text="Android Simple Service Example" />

<Button

android:id="@+id/start\_service"

android:layout\_marginTop="20dp"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:gravity="center\_horizontal"

android:text="Start Service"

android:layout\_below="@+id/text\_view\_header"

android:onClick="onClickStartServie" />

<Button

android:id="@+id/stop\_service"

android:layout\_marginTop="20dp"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:gravity="center\_horizontal"

android:text="Stop Service"

android:layout\_below="@+id/start\_service"

android:onClick="onClickStopService" />

</RelativeLayout>

**6. Now add a new java file for your Service Class code (MyService.Java) to your projects /src folder or just copy and paste the content.**

**Here We have created a new class called MySercive which is extended from Android inbuilt Service class. Then in that class we have @ Override some required virtual functions like ( onBind(), onCreate(), onStart(), onDestroy() etc).**

package com.mm214.serviceexample;

import android.app.Service;

import android.content.Intent;

import android.os.IBinder;

import android.util.Log;

import android.widget.Toast;

public class MyService extends Service{

private static final String TAG = "MyService";

@Override

public IBinder onBind(Intent arg0) {

return null;

}

@Override

public void onCreate() {

Toast.makeText(this, "Congrats! MyService Created", Toast.LENGTH\_LONG).show();

Log.d(TAG, "onCreate");

}

@Override

public void onStart(Intent intent, int startId) {

Toast.makeText(this, "My Service Started", Toast.LENGTH\_LONG).show();

Log.d(TAG, "onStart");

//Note: You can start a new thread and use it for long background processing from here.

}

@Override

public void onDestroy() {

Toast.makeText(this, "MyService Stopped", Toast.LENGTH\_LONG).show();

Log.d(TAG, "onDestroy");

}

}

**7. Register your service (MyService class) in the Manifest file. Add the below Manifest file in the root folder of your project or just copy and paste the content.**

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.mm214.serviceexample"

android:versionCode="1"

android:versionName="1.0" >

<uses-sdk

android:minSdkVersion="8"

android:targetSdkVersion="16" />

<application

android:allowBackup="true"

android:icon="@drawable/ic\_launcher"

android:label="@string/app\_name"

android:theme="@style/AppTheme" >

<activity

android:name="com.mm214.serviceexample.MainActivity"

android:label="@string/app\_name" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<service android:enabled="true" android:name=".MyService" />

</application>

</manifest>

**8. Call Start Service and Stop Service from the main activity class.**

package com.mm214.serviceexample;

import android.os.Bundle;

import android.app.Activity;

import android.content.Intent;

import android.view.Menu;

import android.view.View;

public class MainActivity extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

//start the service

public void onClickStartServie(View V)

{

//start the service from here //MyService is your service class name

startService(new Intent(this, MyService.class));

}

//Stop the started service

public void onClickStopService(View V)

{

//Stop the running service from here//MyService is your service class name

//Service will only stop if it is already running.

stopService(new Intent(this, MyService.class));

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.activity\_main, menu);

return true;

}

}

**Notes:**

**When we start an Android Service by calling the function startService(), then the execution flow in the Service class will be from onCreate() to onStart() function. Similarly when we call stopService() function, the onDestroy() function will be call from the service class.**

**stopService() will stop the service no matter how mnay times you called startService().**

**If the service is already running and you started the service again, then its onStartCommand() is also called. So your service class must be prepared that onStartCommand() can be called several times.**

**9. Build and Run the code on Emulator/Device. Then click Start Service and Stop Service buttons to test your Android Service Example.**