

Android Tutorials and Tips

FRIDAY, 7 JUNE 2013

Start Activity or Service on boot Android

Hi guys,

We have number of situations in our development that we want our service or our app should be run as soon as Device Start, or after Boot Completed.

After boot completes the Android system broadcasts an intent with the action `android.intent.action.BOOT_COMPLETED`. And now all we need is an `IntentReceiver`, called a `BroadcastReceiver`, to listen and act on it.

So here is simple example where you can run your Activity or Service as soon as device Start.

first create simple activity as our favorite activity `MainActivity` :P

```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
  
}
```

and Here its XML

```
<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" >  
  
    <TextView  
        android:id="@+id/textView1"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_alignParentTop="true"  
        android:layout_centerHorizontal="true"
```

```
android:layout_marginTop="156dp"
android:text="App Main Page"
android:textAppearance="?android:attr/textAppearanceMedium" />
```

```
</RelativeLayout>
```

And create simple service to show messages as it created and running etc

TestService.java

```
public class TestService extends Service {
    public TestService() {
    }

    @Override
    public IBinder onBind(Intent intent) {
        // TODO: Return the communication channel to the service.
        throw new UnsupportedOperationException("Not yet implemented");
    }

    @Override
    public void onCreate() {
        // TODO Auto-generated method stub
        Toast.makeText(getApplicationContext(), "Service Created",1).show();
        super.onCreate();
    }

    @Override
    public void onDestroy() {
        // TODO Auto-generated method stub
        Toast.makeText(getApplicationContext(), "Service Destroy",1).show();
        super.onDestroy();
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        // TODO Auto-generated method stub
        Toast.makeText(getApplicationContext(), "Service Working",1).show();
        return super.onStartCommand(intent, flags, startId);
    }
}
```

Now when the device finishes booting up, our BroadcastReceiver will receive the BOOT_COMPLETED action in the onReceive() method which is where we'll start our service

```

public class BootStartUpReciever extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO: This method is called when the BroadcastReceiver is receiving

        // Start Service On Boot Start Up
        Intent service = new Intent(context, TestService.class);
        context.startService(service);

        //Start App On Boot Start Up
        Intent App = new Intent(context, MainActivity.class);
        App.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        context.startActivity(App);

    }
}

```

AndroidManifest.xml

Now we have to modify the AndroidManifest.xml file:

- 1) adding the permission to capture the event of the boot completed:
- 2) registering the Service:
- 3) And The receiver needs to be declared in the manifest, e.g. with the following entry:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.bootstart"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.bootstart.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:name="com.example.bootstart.TestService"
    android:enabled="true"
    android:exported="true" >
</service>

<receiver
    android:name="com.example.bootstart.BootStartupReciever"
    android:enabled="true"
    android:exported="true" >
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</receiver>
</application>
</manifest>

```

You can download complete example Here
[Click Download](#)

Happy Coddddding :)

Khurram Ahmed at 05:08

Share  10



Add a comment

Top comments



prashant dhameliya

8 months ago

Shared publicly

my application will unfortunatly stop when
device boot up....



Knurram Anmed 8 months ago

just download code and run ,
Or just share your on onReceive
method code if you change
something there ?



Jesus Gualberto Palomo Garcia

1 month ago

Shared publicly

Wow!!! Thanks!!! Your Code is really Good!
Best Regards Form México!



Jim Czekaj

2 months ago

Shared publicly

Thanks for the example. I am new to Android
Apps. Based on this tutorial, it appears the
Android environment maintains a list of
receivers (apps who are listening for certain
[Read more](#)



Sisila Priyankara

4 months ago

Shared publicly

thanks ,nice tutorial.....!



Guillermo Pardos Vázquez

11 months ago

Shared publicly

Hello, I use this code to start an activity
when the phone starts, but my activity has
attribute
android:alwaysRetainTaskState=true, when
[Read more](#)



Gábor Gergely

1 year ago

Shared publicly

Hi Ahmed
My app has an activity (not main), that must
run regularly, so it uses the alarm service.
I want to start this activity at device boot.
When this activity start after boot, it can't
refer to the object's of the main activity, until
I start the app manually
How should I solve it?



pavan spl

1 year ago

Shared publicly

thanks.. It is working nice.



Alex Semykin

11 months ago

Shared publicly

For example I want to put a Button (Start
Service) and a Button (Stop Service)

Service) and a Button (Stop Service).
When I push "Start Service" it should start
the service and it will work even after boot.

[Read more](#)



Валерий Шанич 3 months ago
Добрый день, помогите сделать
приложение в автозагрузку.

[Translate](#)



[Home](#)



[View web version](#)

ABOUT ME



Khurram Ahmed



[Follow](#)

115

As for me, all I know is that I know nothing.

[View my complete profile](#)

Powered by [Blogger](#)
