Handling Screen OFF and Screen ON Intents

JANUARY 24, 2010

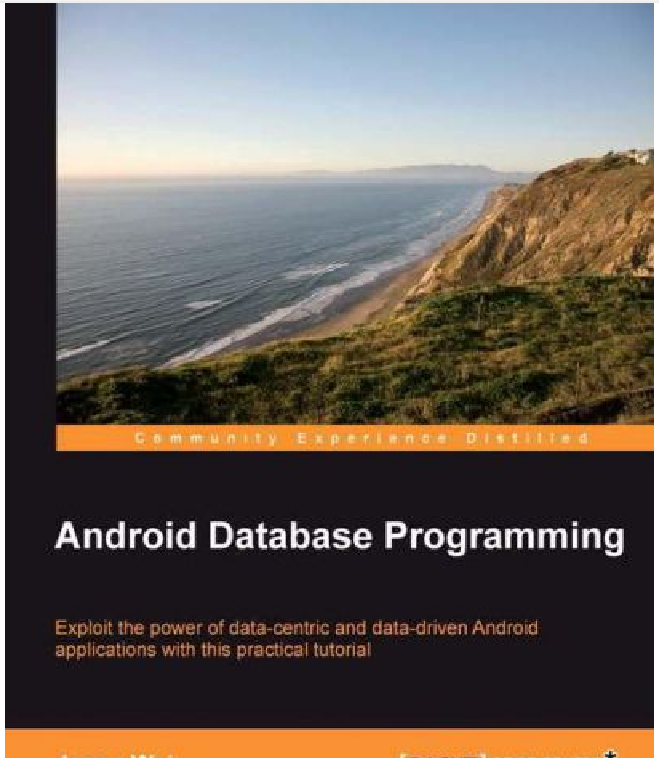
tags: android, broadcast receiver, example, intent, intent.action_screen_off, intent.action_screen_on

Hey everyone,

Haven't posted in a while – sorry school has been busy. Any who, this little code snippet/example will be on how to deal with the Intent.ACTION_SCREEN_OFF and the Intent.ACTION_SCREEN_ON, which will come in nifty if you're making an application that might need to save state or respond to the user's screen going to sleep/waking up, etc.

First, unlike other broad casted intents, for Intent.ACTION_SCREEN_OFF and Intent.ACTION_SCREEN_ON you CANNOT declare them in your Android Manifest! I'm not sure exactly why, but they must be registered in an IntentFilter in your JAVA code. And so, for this example we are going to have a receiver called ScreenReceiver, and I'm going to walk you

CHECKOUT MY NEW BOOK!


Community Experience Distilled
Android Database Programming
Exploit the power of data-centric and data-driven Android applications with this practical tutorial

through the differences between implementing it in a Service vs. in an Activity.

So, the receiver will simply look like:

```
public class ScreenReceiver extends BroadcastReceiver {

    // THANKS JASON
    public static boolean wasScreenOn = true;

    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals(Intent.ACTION_SCREEN_OFF)) {
            // DO WHATEVER YOU NEED TO DO HERE
            wasScreenOn = false;
        } else if (intent.getAction().equals(Intent.ACTION_SCREEN_ON)) {
            // AND DO WHATEVER YOU NEED TO DO HERE
            wasScreenOn = true;
        }
    }

}
```

Now, the first example will be for an Activity. Because of the life-cycle of an Activity, an Activity is actually easier to deal with as right before the screen turns off onPause() is called and right when the screen turns on onResume() is called, and so naturally we will handle the screen on/off events here:

```
public class ExampleActivity extends Activity {

   @Override
    protected void onCreate() {
        // INITIALIZE RECEIVER
        IntentFilter filter = new IntentFilter(Intent.ACTION_SCREEN_ON);
        filter.addAction(Intent.ACTION_SCREEN_OFF);
        BroadcastReceiver mReceiver = new ScreenReceiver();
        registerReceiver(mReceiver, filter);
        // YOUR CODE
    }

    @Override
    protected void onPause() {
        // WHEN THE SCREEN IS ABOUT TO TURN OFF
        if (ScreenReceiver.wasScreenOn) {
```

Android Database Programming

SUBSCRIBE TO THINK ANDROID

Enter your email address to subscribe to this blog and receive the latest examples, code snippets, and tutorials from Think Android.

Join 672 other followers

Let's Think Android

SEARCH FOR EXAMPLES

type and press enter

CATEGORIES

- Android Examples
  - Activity and Layout Examples
  - Android 1.6 Examples
  - Android 2.0 Examples
  - Intents
  - Misc Examples
  - Web Examples
  - Working with Images

```
            // THIS IS THE CASE WHEN ONPAUSE() IS CALLED BY THE SYSTEM DUE TO A SCREEN STATE CHANGE
            System.out.println("SCREEN TURNED OFF");
        } else {
            // THIS IS WHEN ONPAUSE() IS CALLED WHEN THE SCREEN STATE HAS NOT CHANGED
        }
        super.onPause();
    }

    @Override
    protected void onResume() {
        // ONLY WHEN SCREEN TURNS ON
        if (!ScreenReceiver.wasScreenOn) {
            // THIS IS WHEN ONRESUME() IS CALLED DUE TO A SCREEN STATE CHANGE
            System.out.println("SCREEN TURNED ON");
        } else {
            // THIS IS WHEN ONRESUME() IS CALLED WHEN THE SCREEN STATE HAS NOT CHANGED
        }
        super.onResume();
    }

}
```

Now, note that in my onPause() and onResume() methods I run a check to see that the method was called DUE TO A SCREEN STATE CHANGE. This is important as often onPause() or onResume() will get called because of other reasons – i.e. a new activity is being started on top of this one, or an incoming call might be coming in, etc – and you want to make sure that your screen change logic is only called when the screen has actually changed.

Now, something to keep in mind, is that the order of events before the system screen turns off is:

ExampleActivity.onPause() –> ScreenReceiver.onReceive()

Which is a little unintuitive as you'd think the receiver would get hit first – and so when you play around with setting booleans, etc, be aware of this little fact, and likewise when the screen turns on the order of events is:

ExampleActivity.onResume() –> ScreenReceiver.onReceive()

- Android Tutorials
  - ContentProvider Tutorials
  - CursorAdapter Tutorials
  - Layout Tutorials
  - Media Tutorial
  - Receiver Tutorial
- Google App Engine Series
- News and Announcements
- Uncategorized

TOP POSTS

- Incorporating Socket Programming into your Applications
- How to Position Views Properly in Layouts
- SimpleCursorAdapters and ListViews

WHAT I'M THINKING

Check out my new post on JSON parsing! wp.me/pKB6B-58 #android #json 4 months ago

Just published a book! "Android Database Programming" link.packtpub.com/Cs4hK9

And so again the order of events seems a little “backwards”.

Now, for a Service, it’s a little bit different since there is no onResume() or onPause() that gets called as the Service is always “running” in the background, and so instead what you’re going to have to do is modify your receiver a little to look like:

```java
public class ScreenReceiver extends BroadcastReceiver {

    private boolean screenOff;

    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals(Intent.ACTION_SCREEN_OFF)) {
            screenOff = true;
        } else if (intent.getAction().equals(Intent.ACTION_SCREEN_ON)) {
            screenOff = false;
        }
        Intent i = new Intent(context, UpdateService.class);
        i.putExtra("screen_state", screenOff);
        context.startService(i);
    }

}
```

And your service will look like:

```java
public static class UpdateService extends Service {

        @Override
        public void onCreate() {
            super.onCreate();
            // REGISTER RECEIVER THAT HANDLES SCREEN ON AND SCREEN OFF LOGIC
            IntentFilter filter = new IntentFilter(Intent.ACTION_SCREEN_ON);
            filter.addAction(Intent.ACTION_SCREEN_OFF);
            BroadcastReceiver mReceiver = new ScreenReceiver();
            registerReceiver(mReceiver, filter);
        }

        @Override
        public void onStart(Intent intent, int startId) {
            boolean screenOn = intent.getBooleanExtra("screen_state", false);
            if (!screenOn) {
                // YOUR CODE
```

Check it out! #android #androiddev
7 months ago

New Post! Lazy Loading Images: From URLs to ListViews wp.me/pKB6B-4p #android #androiddev 7 months ago

Some final thoughts on my GAE series - wp.me/pKB6B-4d #Android 1 year ago

My 40th Tutorial and the last of my GAE series - wp.me/pKB6B-45 - enjoy! #Android
1 year ago

DEL.ICIO.US

BLOG STATS

1,430,620 hits

```
            } else {
                // YOUR CODE
            }
        }
}
```

And so this is pretty self explanatory. When the screen state changes, it will notify your ScreenReceiver and from there you can set the state information into an Intent and send that data to your Service which can then handle it appropriately.

Hopefully this was useful. Let me know if you have questions.

Happy coding.

- jwei

Share this: Share

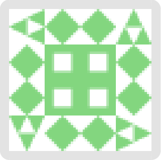
Like this: ★Like One blogger likes this.

from → Intents, Receiver Tutorial

← Retrieving Contact Information (Name, Number, and Profile Picture)

Debugging Applications by Emailing Error Reports →

49 Comments leave one →



Jason PERMALINK

April 1, 2010 1:52 pm

Great idea, but the onPause code in the Activity is broken. wasScreenOn will always be true on entering the function. As you indicated, "ExampleActivity.onPause() –> ScreenReceiver.onReceive()" is the order of events. Thus, whether you entered onPause as a result of letting the screen turn off, or you entered onPause by putting another activity on top of this one, wasScreenOn will always be true.

REPLY

jwei512 PERMALINK *

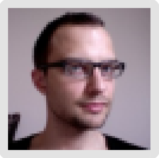
April 5, 2010 2:04 am

Hey Jason,

Thanks for the catch. I think the fix is pretty simple – just initialize wasScreenOn to be TRUE. Then the rest of the logic works (correct me if I'm wrong).

- jwei

REPLY



marko PERMALINK

April 22, 2010 12:47 pm

How about monitoring user inactivity? Is there a system internal inactivity timer or do I need to poke a countdown timer thread from regular methods using event listeners?

Many thanks for your article by the way! Very intuitive and clear.

REPLY

Nam Il-Woo PERMALINK

May 26, 2010 6:27 pm

It’s a useful article~~

REPLY

Solution 9420 PERMALINK

June 2, 2010 3:44 am

Wow, your article is really come in handly. I was looking to turn off the shake sensor on my 9420 Thai Keyboard when the phone is locked and/or screen is off.

Thank you so much,

Solution 9420

REPLY

cx PERMALINK

August 4, 2010 12:39 am

very good ! thank you (谢谢） from china ！

REPLY

Jerome PERMALINK

October 7, 2010 3:06 am

Thank you very much. It is a useful article.

REPLY

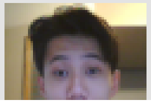
E PERMALINK

October 11, 2010 7:56 pm

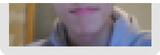
Thank you very much! Your effort contribute to my effort!

^.^

REPLY



Morris Lee PERMALINK



November 17, 2010 10:55 pm

For the life of me, I can't get this to work, I only need a simple check if the screen is currently on or off, is there something simpler to this? I tried your instructions, didn't work, copying the codes, didn't work, I am not sure where I am doing wrong…

Can I just do a constant loop in my onCreate() under my Activity/Main class?

such as

```
onCreate()
{
while (true)
{
if (ScreenOn(or whatever is necessary to check it)
//do what i want here
else
//do anything else when screen is off
}
}
```

Makes sense??? :S
Morris Lee

REPLY

jwei512 PERMALINK *

February 19, 2011 1:49 am

Hey Morris,

No you shouldn’t do that for several reasons. First the while(true) will be run on the main UI thread and will basically cause your application to get stuck – at which point the OS will make the user force close your app.

I would suggest you read about the Activity life cycle – this is important to understand since it will allow you to check for various things (in this case a screen on boolean) at the appropriate times without having to do weird hacks like the one you proposed:

http://developer.android.com/reference/android/app/Activity.html

Hope this helps!

- jwei

REPLY

Kyle PERMALINK

February 12, 2011 3:35 am

This all works for most cases. I am making a game where if you press the HOME button(onPause) the game saves. Then when you bring the game back to the front the onResume will load it. (I do this so in case Android wants to erase all the variables i need to run the game while the user is using some other app)

However, if the screen turns off i do not want onPause to run any code. And when the screen turns back on i don’t want onResume to run any code. (Because Android will not erase the game’s variables when the screen just gets turned off)

Because you initialize wasScreenOn to be TRUE, only one part of the onPause code above will be used whether you press the HOME key or turn the screen off.
It does not make a difference if you initialize wasScreenOn to be false or true. The same problem happens.

I solved the problem by making a few changes to onCreate() and onPause()
Everything else stays the same as above

```
@Override
protected void onCreate() {
// initialize receiver
IntentFilter filter = new IntentFilter(Intent.ACTION_SCREEN_ON);
filter.addAction(Intent.ACTION_SCREEN_OFF);
BroadcastReceiver mReceiver = new ScreenReceiver();
registerReceiver(mReceiver, filter);

//NEW
PowerManager pm =(PowerManager)
getSystemService(Context.POWER_SERVICE);

// your code
}

@Override
protected void onPause() {
// when the screen is about to turn off
// Use the PowerManager to see if the screen is turning off
if (pm.isScreenOn() == false) {
// this is the case when onPause() is called by the system due to the screen
turning off
System.out.println("SCREEN TURNED OFF");
} else {
// this is when onPause() is called when the screen has not turned off
}
super.onPause();
}
```

Hope this helps

REPLY

swapnil PERMALINK

February 14, 2011 12:03 pm

Service example above seems wrong. As You are registering receiver in Oncreate of service. And starting service from receiver… 😀 .

REPLY

jwei512 PERMALINK *

February 19, 2011 1:47 am

Hey swapnil,

Hm not sure what you mean? I don't think I'm starting any "services" (I presume you mean Activities in my case) in my receiver… I'm just toggling a boolean.

- jwei

REPLY

chaplu PERMALINK

March 20, 2011 1:34 pm

Hi,

I cannot run the service, though my receiver is working fine, can you help?

REPLY

fugui PERMALINK

April 18, 2011 6:01 am

OK, Now it can run. Yesterday I make a mistake of Manifest file.

REPLY

dave PERMALINK

April 29, 2011 3:39 pm

how would one go about doing this from an AppWidgetProvider.

I don’t use Service, instead i opted for java.util.timer for scheduling my updates.

cheers
Dave.

REPLY

dave PERMALINK

April 29, 2011 3:56 pm

damn i miss C++, multiple inheritance just gets things done

i'll use a inner class

ty anyway

REPLY

sjava PERMALINK

May 19, 2011 9:22 am

I implemented your code in my application. Intent.ACTION_SCREEN_OFF fires when I manually hit the power button to turn off the screen but it does not fire when the screen saver timed out. Does this happen to anybody else?

REPLY

beargreaves PERMALINK

June 26, 2011 1:42 pm

My Eclipse reports an error with the following line

BroadcastReceiver myReceiver = new ScreenReceiver();

It gives an error cannot convert from ScreenReceiver to BroadcastReceiver??? Surly the line of code is fine, I don't see anything wrong with it. The only change I made to ScreenReceiver was remove the static keyword as it's not permitted. Any ideas?

REPLY

beargreaves PERMALINK

June 26, 2011 1:45 pm

I've fixed it now, strange problem with eclipse???

REPLY

Michael PERMALINK

August 22, 2011 10:22 pm

I think the way it works is that your app has to be running when your app receives Intent.ACTION_SCREEN_OFF intent. But how can you invoke your app when the screen_off?

Someone said there is no way due to google's reason, is this true?

REPLY

Zink PERMALINK

September 13, 2011 4:33 pm

Great article. I used it for my app. Thank you very much from Spain !

REPLY

guest@guest.guest PERMALINK

September 14, 2011 3:33 am

thank you!

REPLY

Lili Park PERMALINK

October 6, 2011 12:20 am

I follow this code and it run ok.

Thanks a lot!

REPLY

Rahul PERMALINK

November 30, 2011 10:40 pm

Thanks, I was looking for something else. But I learned a new thing.

REPLY

Daniel Szasz PERMALINK

December 1, 2011 2:51 am

Wonderful job…
I need something else…maybe someone can help me…

I have an application that have to keep the screen on and avoid screen off and lock…itţs a timer…

how can I do this ?

thanks

daniel

REPLY

jwei512 PERMALINK *

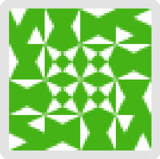
December 8, 2011 8:05 pm

Hey Daniel,

You’re probably looking for something along the lines of http://stackoverflow.com/questions/4263868/disabling-screen-sleep-programatically-in-android

- jwei

REPLY



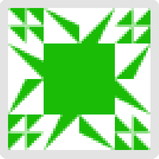
Nikhil PERMALINK

January 1, 2012 10:29 pm

Please unregister the receiver that you have registered programmatically to avoid window leakage error.

kindly unregister the receiver in onDestroy method.

REPLY

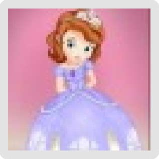


Pradeep PERMALINK

January 11, 2012 11:30 pm

Thank u really use full for me

REPLY



Pallavi PERMALINK

March 14, 2012 10:41 pm

It totally worked for me! thanks a lot sir!!!!

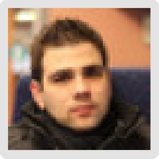
REPLY

dugrhill PERMALINK

March 18, 2012 7:04 pm

EXACTLY what I needed. Thanks!

REPLY



Christos Bacharakis (@bacharakis) PERMALINK

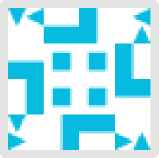
April 18, 2012 1:25 am

Hello!
I am trying to build an app that starts the time the phone is idle before the screen turns off.

Do you know how can I determine the screen is idle?

Thanks

REPLY



shareef PERMALINK

April 18, 2012 7:59 pm

Thank you for great information, from INDIA

REPLY

Guen PERMALINK

May 31, 2012 5:54 am

Thank you kindly!!! It was concise and clear.

I think its better app architecture to have the logic in a class outside the activity, in which case the same class can extend the BroadcastReceiver saving the information ping-pong.

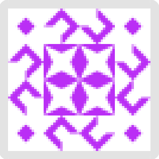
REPLY

dmmh77 PERMALINK

June 26, 2012 11:59 am

I don’t understand the Service approach, why not simply put the code to run in the receiver, that’s easier 😉

REPLY



Alberto PERMALINK

August 29, 2012 1:19 am

Hello, I have a problem with the thread of the service. Hopefully you could help me with it.
I wanna stop the service (this service has its own thread) when the screen goes off and start it again when the screen turns on.
As the services cannot be "resume" I think I have to destroy it and create it again when I switched the screen. But when I create the service again it launch a NullPointerException ("Unable to start service com.example.daemon3.DaemonService@40ff0dfo with intent {cmp=com.example.daemon3/.DaemonService (has extras) }: java.lang.NullPointerException")
Really hope you can help me.

…

Thank you so much.

REPLY

jwei512 PERMALINK *

September 4, 2012 9:55 am

Hi Alberto,

It seems like the issue is not with my example but with the way that you're re-starting your service.

Please see http://stackoverflow.com/questions/4679654/unable-to-start-service-service-name-with-null for the cause of your error.

Hope this helps.

- jwei

REPLY

Noureen PERMALINK

September 26, 2012 8:35 am

hi!
I am developing an app in which i have create three activities
Main_Activity Second_Activity CustomView_Activity

here is the process in which these Activties called:

MainActivity calls Second_Activtiy and in OnCreate of Second_Activtiy I have created object of CustomView_Activity.

In Main_Activity I have a button on a LinearLayout which onClick calls Second_Activity in Second_activity OnCreate i have to keep my Screen On through Keep_Screen_On and declare other things like buttons layouts etc.

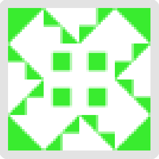
in CustomView_Activity i have creates canvas and onCanvas i have draw a Rectangle.

Now when my Second_Activity calls after Main_Activity and calling OnPause on pressing Power button my activity calls OnPause OnStop OnCreate OnResume OnRestart OnDestroy and when i press the power button to on the screen OnResume onCreate On Restart calls.

Due to calling of OnCreate again and again my activity not performing tasks correctly.

Please anyone who can help??

REPLY



shubham PERMALINK

November 9, 2012 3:02 pm

first example is very clear but i am not bale to understand how the second one works. As we are starting a service from broadcastreceiver class. There should be a activity class along which a broadcastreiver is made.

REPLY

jwei512 PERMALINK *

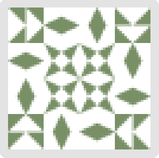
November 12, 2012 8:11 am

Hi Shubham,

I'm not entirely sure what you mean – I presume by second example you mean the example where you are registering the broadcast receiver in a Service as opposed to an Activity. If that's the case you'll notice that upon starting your Service (which is essentially an Activity that runs in the background), just as how you would register a receiver in an Activity's onCreate() method, you can do the same for a Service.

- jwei

REPLY



Peter Jackson PERMALINK

December 14, 2012 7:12 am

Modifier static not allowed here (public static class UpdateService extends Service {)
Why?

REPLY

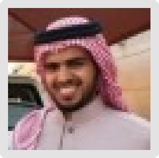
jwei512 PERMALINK *

December 17, 2012 7:12 am

What's the error?

- jwei

REPLY



aljuaid86 PERMALINK

January 20, 2013 6:12 am

I found this tutorial,, very helpful also http://michael.theirwinfamily.net/articles/android/starting-activity-sleeping-device

REPLY

Trackbacks

1. Auto Log-out Timer Research « Team Daily Do
2. lock screen sometime not display android | Build Future Repository
3. Orientation listener in a service | PHP Developer Resource
4. [wip][rom][cm7][zvc] lg-ls855 marquee - Page 3 - Android Forums
5. check if wifi is connected every 5 minutes after screen is off : Android Community - For Application Development
6. BroadCaste reciever invoking if screen is off : Android Community - For Application Development

Leave a Reply

Enter your comment here...