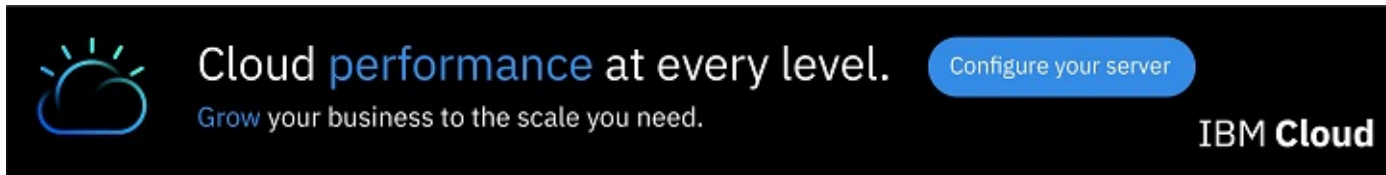


Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

Implementing LocationListener on a service thread



I'm trying to get the location updates running in a background service. The service is running a workerthread of its own doing a lot of other stuff already, like socket communication. I'd like it to also handle location updates but this seems to only work on an activity. As far as I can read this is due to the message loop missing on a workerthread. I think I need to use `Looper.prepare()` somewhere, but maybe I need another thread just to handle locations requests? I can't seem to get the emulator to respond to any geo fix events, so I must be doing something wrong.

Below is the service code, stripped for all the non-relevant parts.

```
public class MyService extends Service implements LocationListener {
    private Thread runner;
    private volatile boolean keepRunning;
    private LocationManager locationManager = null;

    @Override
    public void onCreate() {
        keepRunning = true;

        runner = new Thread(null, new Runnable() {
            public void run() { workerLoop(); }
        });
        runner.start();
        startGps();
    }

    @Override
    public void onDestroy() {
        keepRunning = false;
    }
}
```

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

 Google

Facebook



Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
//Looper.prepare();

// Main worker loop for the service
while (keepRunning) {
    try {
        if (commandQueue.notEmpty()) {
            executeJob();
        } else {
            Thread.sleep(1000);
        }
    } catch (Exception e) {
    }
}
stopGps();
}

public void onLocationChanged(Location location) {
    // Doing something with the position...
}
public void onProviderDisabled(String provider) {}
public void onProviderEnabled(String provider) {}
public void onStatusChanged(String provider, int status, Bundle extras) {}

private void startGps() {
    if (locationManager == null)
        locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
    if (locationManager != null) {
        Criteria criteria = new Criteria();
        criteria.setAccuracy(Criteria.ACCURACY_FINE);
        criteria.setAltitudeRequired(false);
        criteria.setBearingRequired(true);
        criteria.setCostAllowed(false);
        criteria.setSpeedRequired(true);
        String provider = locationManager.getBestProvider(criteria, true);
        if (provider != null)
            locationManager.requestLocationUpdates(provider, 10, 5,
(LocationListener) this);
    }
}

private void stopGps() {
    if (locationManager != null)
        locationManager.removeUpdates((LocationListener) this);
    locationManager = null;
}
```

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

 android service gps loop

edited Oct 12 '12 at 14:04



jnthnjs

7,980 4 34 59

asked Mar 11 '12 at 20:59



user1262805

16 1 2

1 Answer

The LocationListener doesn't care whether it is in the context of an activity or a service. You want to use the location updates in your workerLoop(), right? The location updates (the call to the LocationListener) and the workerLoop() are both acting independently from each other. To get the location update to the workerLoop() you need to join the two threads. One method to do this is to use the blackboard pattern: The LocationListener writes the new location to a field of your class (this is easy since both are in the context of the same class):

```
private Location blackboard = null;
public void onLocationChanged(final Location location) {
    if( location != null )
        this.blackboard = location;
}

private void workerLoop() {
    ...
    if( this.blackboard != null ) {
        final Location locationUpdate = this.blackboard;
        this.blackboard = null;
        // .. do something with the location
    }
    ...
}
```

With the above code you may get race conditions when the LocationListener writes to the blackboard while the workerLoop() is reading or erasing the location. This can be solved by surrounding the access to the blackboard by a synchronized block like this:

```
synchronized(this) {
    this.blackboard = location
}
```

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
private volatile Location blackboard = null;
```

Alternatively you may use a Lock, confer to the docs for more details:

<http://docs.oracle.com/javase/tutorial/essential/concurrency/newlocks.html>

answered Mar 12 '12 at 9:21



Stefan

4,006 1 13 28

Yes I want the service to handle everything in case the UI thread is not active. There seem to be some sort of difference between implementing it in a service vs an activity. It is like events doesn't get through to the emulator when I telnet with the geo fix command. I'm using a rather old Google Api v 1.6 though, maybe the newer versions are better or perhams I am missing something... I have ACCESS_mock_LOCATION in the manifest and everything works fine for the activity... only the service seems dead (nothing in the LogCat either). It does pop up the gps disc notification icon though.. — [user1262805](#) Mar 12 '12 at 21:46
