

CS 696 Emerging Web and Mobile Technologies  
Spring Semester, 2011  
Doc 24 Android Services  
Apr 26, 2011

Copyright ©, All rights reserved. 2011 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

# Service

Runs in the background

No user interaction

Runs indefinitely

Runs in thread of hosting process

    Create new thread to do work

# Service Lifecycle

## Local Service

Started when `Context.startService(Intent)` is called

Then service's methods are called

`onCreate()`

`onStart(Intent, int)`

Service runs until one of following is called

`Context.stopService(Intent)`

`stopSelf()`

Only one of a service runs

Implement `onDestroy()` to clean up

# Service Lifecycle

## Remote Service

Started when `bindService (Intent, ServiceConnection, int flags)` is called

Then service's method `onCreate()` is called if need

Client received `IBinder` object to communicate with service

Service runs until no clients are bound to the service

Implement `onDestroy()` to clean up

# Process Lifecycle

Services are killed if low on memory

If currently executing onCreate(), onStart(), or onDestroy()  
service not killed

If service has been started

process is less important than currently visible processes

process is more important than processes not visible & don't have service

If clients are bound to the service

process is as important as the most important client

# IntentService

IntentService

Subclass of Service

Creates worker thread for you

onHandleIntent() is called for each intent request

Service is stopped when all requests are handled

Default implementation of

onBind()

onStartCommand()

# Permissions

Need to declare service in manifest

```
<service android:name=".AvitarService" />
```

Other applications need `<uses-permission>` to use service

# Services & Battery life

Waking up a service 6 times an hour for 8 seconds each time

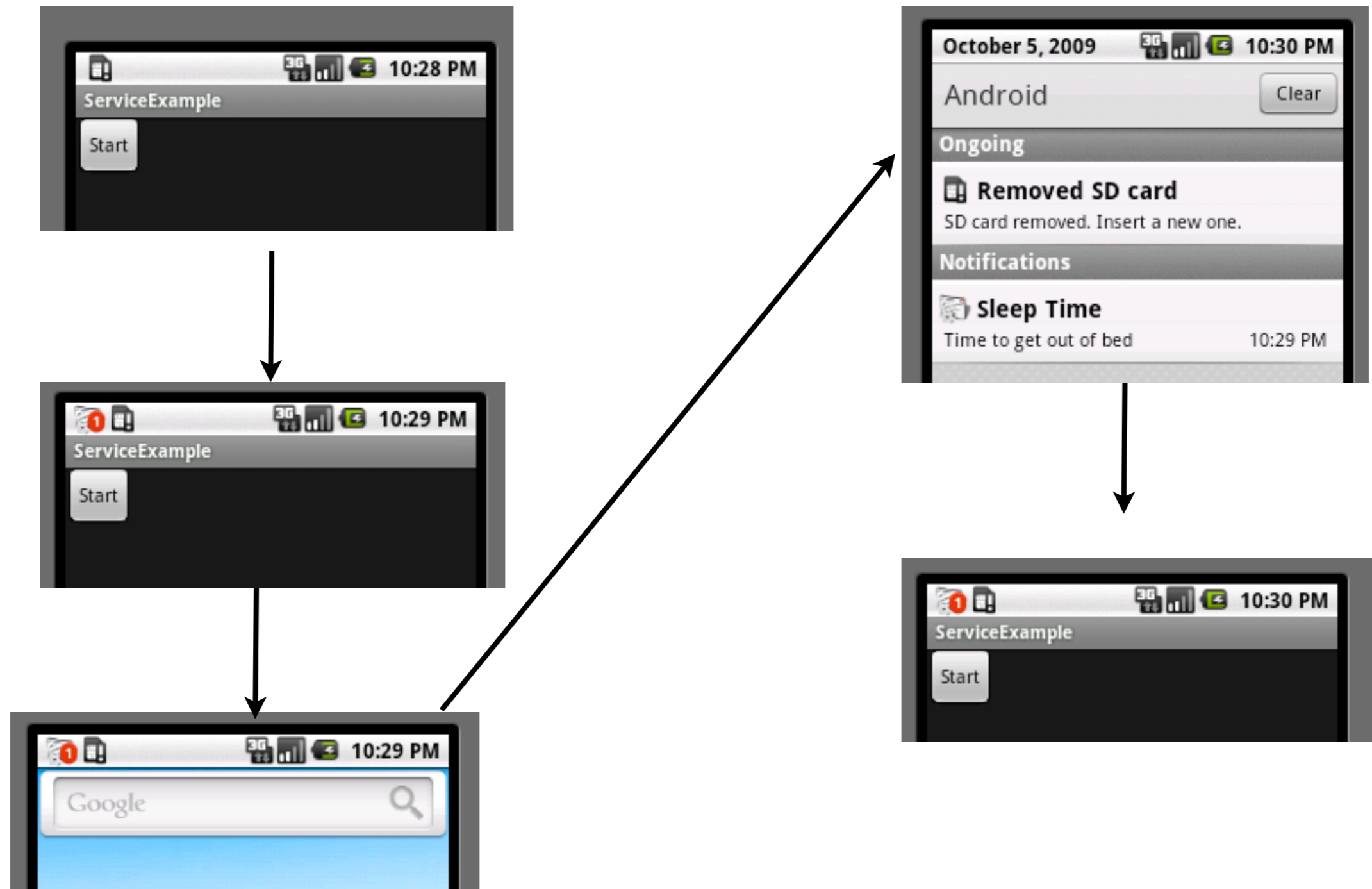
Cuts the battery life in half

Can schedule updates at approximate times so system can run multiple requests at same time



# Notification without Service

# Notification Without Service



# Send Notification on Button Click

```
public class ServiceExample extends Activity implements View.OnClickListener {
    private static final int NOTIFY_ID = 1123;

    private int count = 0;

    public void onClick(View v) {
        TimerTask task = new TimerTask() {
            public void run() {
                sendNotification();
            }
        };
        new Timer().schedule(task, 5000);
    }

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button ok = (Button) findViewById(R.id.start);
        ok.setOnClickListener(this);
    }
}
```

# The Notification

```
private void sendNotification() {
    Notification note = new Notification(R.drawable.icon, "Wake Up!",
        System.currentTimeMillis());
    PendingIntent intentToStart = PendingIntent.getActivity(this, 0,
        new Intent(this, ServiceExample.class),
        Intent.FLAG_ACTIVITY_NEW_TASK);

    note.setLatestEventInfo(this, "Sleep Time", "Time to get out of bed",
        intentToStart);
    note.number = ++count;
    note.defaults = Notification.DEFAULT_VIBRATE;
    NotificationManager manager = (NotificationManager)
        getSystemService(NOTIFICATION_SERVICE);
    manager.notify(NOTIFY_ID, note);
}
}
```

# Need permission to Vibrate

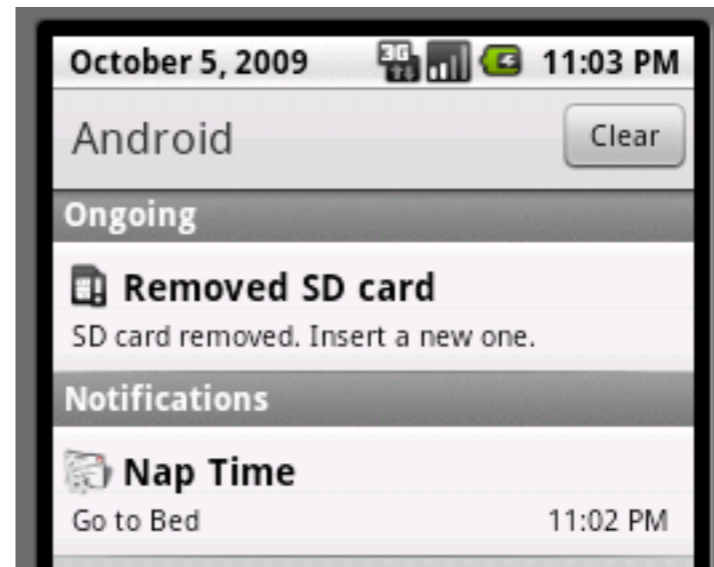
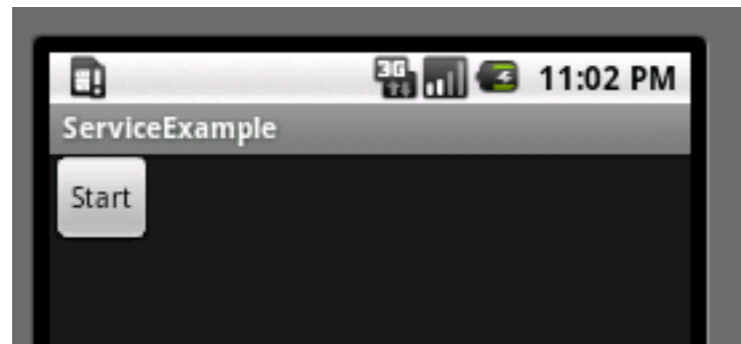
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.sdsu.cs.whitney"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".ServiceExample"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="3" />
    <uses-permission android:name="android.permission.VIBRATE"></uses-permission>
</manifest>
```

# Start an activity when clearing Notifications

```
private void sendNotification() {  
    Notification note = new Notification(R.drawable.icon, "Wake Up!",  
        System.currentTimeMillis());  
    PendingIntent intentToStart = PendingIntent.getActivity(this, 0,  
        new Intent(this, ServiceExample.class),  
        Intent.FLAG_ACTIVITY_NEW_TASK);  
  
    note.setLatestEventInfo(this, "Sleep Time", "Time to get out of bed",  
        intentToStart);  
    note.number = ++count;  
    note.deleteIntent = intentToStart;  
    note.defaults = Notification.DEFAULT_VIBRATE;  
    NotificationManager manager = (NotificationManager)  
        getSystemService(NOTIFICATION_SERVICE);  
    manager.notify(NOTIFY_ID, note);  
}
```

# Service Sending Notification

# Service that Sends Notifications





# Start Service when Click button

```
public class ServiceExample extends Activity implements View.OnClickListener {  
  
    public void onClick(View v) {  
        Intent serviceIntent = new Intent(this, AvitarService.class);  
        startService(serviceIntent);  
    }  
  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
        Button ok = (Button) findViewById(R.id.start);  
        ok.setOnClickListener(this);  
    }  
}
```

# AvitarService

```
public class AvitarService extends IntentService {
    private static final int NOTIFY_ID = 1123;

    public AvitarService() {
        super("AvitarServiceThreadName");
    }

    @Override
    protected void onHandleIntent(Intent intent) {
        TimerTask task = new TimerTask() {
            @Override
            public void run() {
                sendNotification();
            }
        };
        new Timer().schedule(task, 1000);
    }
}
```

# The Notification

```
private void sendNotification() {  
    Notification note = new Notification(R.drawable.icon, "Nap time!",  
        System.currentTimeMillis());  
    PendingIntent intentToStart = PendingIntent.getActivity(this, 0,  
        new Intent(this, ServiceExample.class),  
        Intent.FLAG_ACTIVITY_NEW_TASK);  
  
    note.setLatestEventInfo(this, "Nap Time", "Go to Bed", intentToStart);  
    NotificationManager manager =  
(NotificationManager) getSystemService(NOTIFICATION_SERVICE);  
    manager.notify(NOTIFY_ID, note);  
}
```

# Manifest File

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.sdsu.cs.whitney"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".ServiceExample"
            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=".AvitarService" />
    </application>
    <uses-sdk android:minSdkVersion="3" />
</manifest>
```

Service that actually does something  
Notify user when a web page is updated

# URL Checking Service

Checks webpage in background

Notifies user when page changes

Need Internet permission

Uses

- IntentService

- Schedules service at approximate time intervals

# Setting Recurrent Checks

```
public class ServiceExample extends Activity implements View.OnClickListener {
    public void onClick(View v) {
        Intent serviceIntent = new Intent(this, AvitarService.class);
        PendingIntent pendingIntent = PendingIntent.getService(this, 0, serviceIntent, 0);
        long interval = DateUtils.MINUTE_IN_MILLIS * 30;
        long firstWake = System.currentTimeMillis() + interval;
        AlarmManager alarm =
            (AlarmManager) getSystemService(Context.ALARM_SERVICE);
        alarm.setInexactRepeating(AlarmManager.RTC, firstWake, interval,
pendingIntent);
    }

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button ok = (Button) findViewById(R.id.start);
        ok.setOnClickListener(this);
    }
}
```

# The Service

```
public class AvitarService extends IntentService {
    private static final int NOTIFY_ID = 1123;

    public AvitarService() {
        super("AvitarSErviceThreadName");
    }

    protected void onHandleIntent(Intent intent) {
        try {
            URL address = new URL("http://www.eli.sdsu.edu");
            URLConnection connect = address.openConnection();
            long timeModified = connect.getLastModified();
            long currentTime = System.currentTimeMillis();
            long cutoffTime = currentTime - 1000 * 60 * 45;
            if (cutoffTime < timeModified) {
                Log.i("rew", "changed");
                sendNotification();
            }
        } catch (Exception error) {
            Log.i("rew", error.toString());
        }
    }
}
```



# Notification

```
private void sendNotification() {  
    Notification note = new Notification(R.drawable.icon,  
        "Web Page Changed!", System.currentTimeMillis());  
    PendingIntent intentToStart = PendingIntent.getActivity(this, 0,  
        new Intent(this, ServiceExample.class),  
        Intent.FLAG_ACTIVITY_NEW_TASK);  
  
    note.setLatestEventInfo(this, "Web Page Changes",  
        "http://www.eli.sdsu.edu", intentToStart);  
    NotificationManager manager =  
        (NotificationManager) getSystemService(NOTIFICATION_SERVICE);  
    manager.notify(NOTIFY_ID, note);  
}  
}
```

# Service With Braodcast

# Service with Broadcast

```
public class ServiceExample extends Activity implements View.OnClickListener {
    private Intent serviceIntent;

    private final BroadcastReceiver receiver = new BroadcastReceiver() {
        public void onReceive(Context context, Intent intent) {
            handleBroadcast();
        }
    };

    void handleBroadcast() {
        Toast.makeText(this, "Got the message", Toast.LENGTH_SHORT).show();
    }

    public void onClick(View v) {
        serviceIntent = new Intent(this, AvitarService.class);
        startService(serviceIntent);
    }
}
```

# Service with Broadcast

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    Button ok = (Button) findViewById(R.id.start);
    ok.setOnClickListener(this);
}

public void onDestroy() {
    super.onDestroy();
    stopService(serviceIntent);
}

public void onPause() {
    super.onPause();
    unregisterReceiver(receiver);
}
}
```

# Service with Broadcast

```
public void onResume() {  
    super.onResume();  
    registerReceiver(receiver, new IntentFilter(  
        AvitarService.BROADCAST_ACTION));  
}
```

# The Service

```
public class AvitarService extends Service {
    public static final String BROADCAST_ACTION =
        "edu.sdsu.cs.whitney.sampleBroadcast";

    private void notifyActivity() {
        Intent broadcast = new Intent(BROADCAST_ACTION);
        sendBroadcast(broadcast);
    }

    public IBinder onBind(Intent arg0) {
        return null;
    }

    public void onCreate() {
        super.onCreate();
    }
}
```

# The Service

```
public void onDestroy() {  
    super.onDestroy();  
}
```

```
public void onStart(Intent intent, int startId) {  
    TimerTask task = new TimerTask() {  
        @Override  
        public void run() {  
            notifyActivity();  
        }  
    };  
    new Timer().schedule(task, 1000);  
}
```

```
}
```

# Sending Data from service to Activity



# Service Sending Data to Activity

## The Service

```
public class AvitarService extends Service {
    private class SampleTask extends AsyncTask<Void, String, Void> {
        String[] items = { "Gautama Buddha", "Kalki", "Krishna", "Kurma",
            "Matsya", "Narasimha", "Parashurama", "Rama", "Vamana",
            "Varaha" };

        protected void doInBackground(Void... notused) {
            for (String word : items) {
                notifyActivity(word);
                SystemClock.sleep(1000);
            }
            return (null);
        }
    }
}
```

# Service Continued

```
SampleTask listNames;
```

```
public static final String BROADCAST_ACTION =  
    "edu.sdsu.cs.whitney.sampleBroadcast";
```

```
private void notifyActivity(String message) {  
    Intent broadcast = new Intent(BROADCAST_ACTION);  
    broadcast.putExtra("name", message);  
    sendBroadcast(broadcast);  
}
```

```
public IBinder onBind(Intent arg0) {  
    return null;  
}
```

```
public void onCreate() {  
    super.onCreate();  
}
```

# Service Continued

```
public void onDestroy() {  
    super.onDestroy();  
    listNames.cancel(true);  
}
```

```
public void onStart(Intent intent, int startId) {  
    listNames = new SampleTask();  
    listNames.execute();  
}
```

```
}
```

# ServiceExample

```
public class ServiceExample extends Activity implements View.OnClickListener {
    private Intent serviceIntent;

    private final BroadcastReceiver receiver = new BroadcastReceiver() {
        public void onReceive(Context context, Intent intent) {
            handleBroadcast(intent.getCharSequenceExtra("name"));
        }
    };

    void handleBroadcast(CharSequence name) {
        Toast.makeText(this, name, Toast.LENGTH_SHORT).show();
    }

    public void onClick(View v) {
        serviceIntent = new Intent(this, AvitarService.class);
        startService(serviceIntent);
    }
}
```

# Singleton for Activity Access

```
public class AvitarService extends Service {  
    public static AvitarService singleton = null;  
  
    public void onCreate() {  
        super.onCreate();  
        singleton = this;  
    }  
  
    public void onDestroy() {  
        super.onDestroy();  
        listNames.cancel(true);  
        singleton = null;  
    }  
}
```

# Activity

Can access service directly

```
void handleBroadcast() {  
    String name = AvitarService.singleton.avitar();  
    Toast.makeText(this, name, Toast.LENGTH_SHORT).show();  
}
```

```
public void onDestroy() {  
    super.onDestroy();  
    stopService(serviceIntent);  
}
```