

Manual: How to integrate Google Analytics with
Android Apps

Version 1.0

May 16, 2015

Alexander J. Padin
Web/Mobile Developer
May 16, 2015

Introduction

Google Analytics is one of the major parts in applications that drive revenue. Mobile App Analytics helps you set and track the goal conversions you want most: purchases, clicks, or simply time spent on your app. It is required to make business inclined towards user's interest.

With Google Analytics we get the following information:

1. Number of users/active users from across the world
2. Most viewed products, categories, brands, etc.
3. Number of users placing request, etc.

The above information is required in order to improve the product and make the service is better.

Current supported features

- 1 Google Play Integration
- 2 Crash and Exception Reporting
- 3 Custom Reports
- 4 Event Tracking
- 5 Flow Visualization
- 6 Real-Time Reporting

How to Integrate Google Analytics

This manual will contain detailed steps to install Google Analytics for the Android application.

Before begin with Analytics integration the following items are needed:

- 1 Android developer SDK and required configurations.
Please see: <http://developer.android.com/sdk>
- 2 Google Analytics SDK for Android v3, with **libGoogleAnalyticsServices.jar** file.
Please see:
<https://developers.google.com/analytics/devguides/collection/android/v3/>
Copy this file in your project's /libs directory.
- 3 Codebase of an Android app that you like to implement the Google Analytics
- 4 Create a Google Analytics account (Requires Gmail), and then create a new app property and view in the Analytics console.

Integrating Google Analytics in Android Application

- 1 Login to your Google Analytics Account. Then press “Admin” button on the right side of your screen.
- 2 Create a new Account, if you don’t have one and then create a new property. While creating a new property choose Mobile App.
- 3 After property is created, you will get the **tracking Id**. This tracking id will be used as configuration in android app.
- 4 Add analytics.xml configuration file

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

    <!-- Replace placeholder ID with your tracking ID -->
    <string name="ga_trackingId">UA-xxxxxxxxxx</string>

    <!-- Enable automatic activity tracking -->
    <bool name="ga_autoActivityTracking">true</bool>

    <!-- Enable automatic exception tracking -->
    <bool name="ga_reportUncaughtExceptions">true</bool>

</resources>
```

- 5 Android Activity Layout (activity_main.xml) For this manual, I have created an sample Activity with two buttons for tracking event and exceptions.

```
<LinearLayout 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"
    android:orientation="vertical"
    android:padding="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/info" />

    <Button
        android:id="@+id/trackEvent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Track Event" />

    <Button
        android:id="@+id/trackCrash"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Track Crash" />

</LinearLayout>
```

- 6 Android Activity (MainActivity.java) You need to write the tracking code in your activity class. For that simply add the `activityStart()` and `activityStop()` methods to the `onStart()` and `onStop()` methods of each of your Activities as in the following example. This two method enables basic activity tracking.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    easyTracker = EasyTracker.getInstance(MainActivity.this);

    findViewById(R.id.trackEvent).setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View view) {
            easyTracker.send(MapBuilder.createEvent("your_action",
                "envet_name", "button_name/id", null).build());
        }
    });

    findViewById(R.id.trackCrash).setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View view) {

            try {
                int a[] = new int[2];
                int num = a[4];
            } catch (ArrayIndexOutOfBoundsException e) {
                easyTracker.send(MapBuilder.createException(
                    new StandardExceptionParser(MainActivity.this, null)
                        .getDescription(Thread.currentThread().getName(), e), false).build());
            }

        }

    });
}

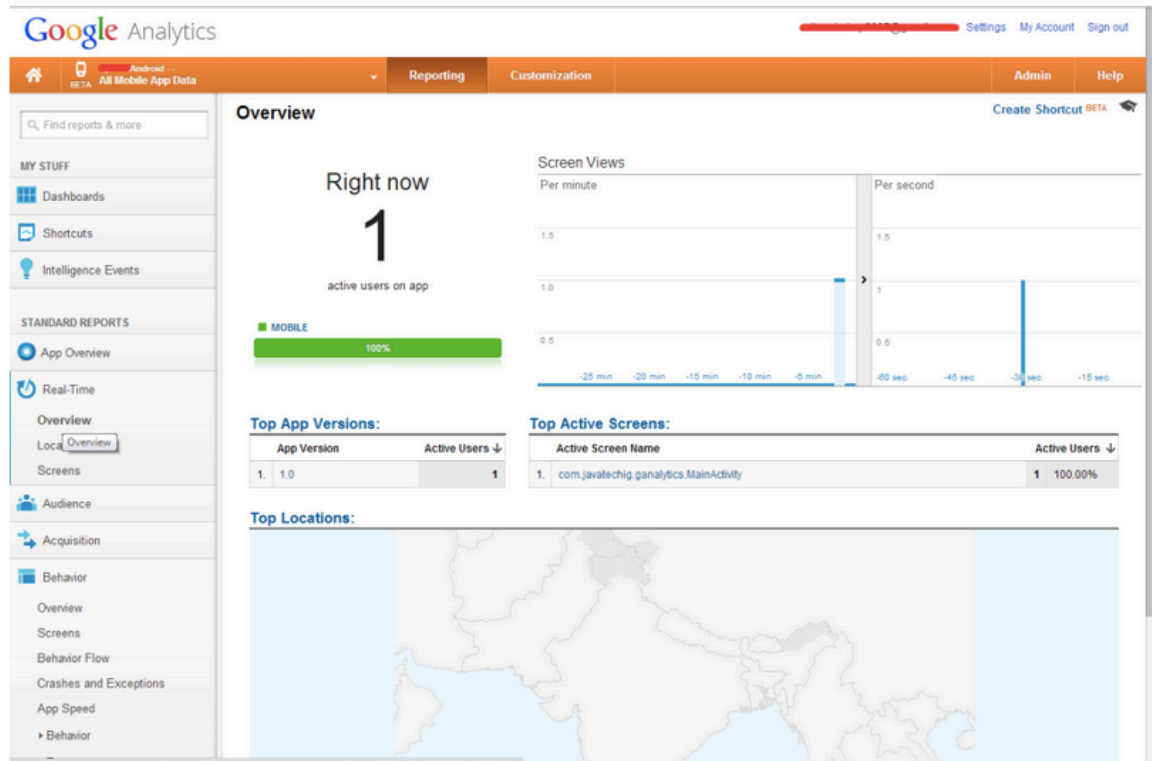
@Override
public void onStart() {
    super.onStart();
    EasyTracker.getInstance(this).activityStart(this);
}

@Override
public void onStart() {
    super.onStart();
    EasyTracker.getInstance(this).activityStart(this);
}

@Override
public void onStop() {
    super.onStop();
    EasyTracker.getInstance(this).activityStop(this);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
```

7 Google Analytics Output



NOTE: For more information please see:

<https://developers.google.com/android/reference/com/google/android/gms/analytics/GoogleAnalytics>