stack overflow discussion

device id solution

```
final TelephonyManager tm = (TelephonyManager)
  getBaseContext().getSystemService(Context.TELEPHONY_SERVICE);

final String tmDevice, tmSerial, androidId;
tmDevice = "" + tm.getDeviceId();
tmSerial = "" + tm.getSimSerialNumber();
androidId = "" + android.provider.Settings.Secure.getString(
  getContentResolver(),
android.provider.Settings.Secure.ANDROID_ID);

UUID deviceUuid = new UUID(
  androidId.hashCode(), ((long)tmDevice.hashCode() << 32) |
tmSerial.hashCode());
String deviceId = deviceUuid.toString();
```

pseudo id

```
/**
 * Return pseudo unique ID
 * @return ID
 */
public static String getUniquePsuedoID() {
    // If all else fails, if the user does have lower than API 9
(lower
    // than Gingerbread), has reset their device or
'Secure.ANDROID_ID'
    // returns 'null', then simply the ID returned will be
solely based
    // off their Android device information. This is where the
collisions
    // can happen.
    // Thanks http://www.pocketmagic.net/?p=1662!
    // Try not to use DISPLAY, HOST or ID - these items could
change.
    // If there are collisions, there will be overlapping data
    String m_szDevIDShort = "35" + (Build.BOARD.length() % 10) +
(Build.BRAND.length() % 10) + (Build.CPU_ABI.length() % 10) +
(Build.DEVICE.length() % 10) + (Build.MANUFACTURER.length() %
10) + (Build.MODEL.length() % 10) + (Build.PRODUCT.length() %
10);

    // Thanks to @Roman SL!
    // http://stackoverflow.com/a/4789483/950427
    // Only devices with API >= 9 have android.os.Build.SERIAL
    //
http://developer.android.com/reference/android/os/Build.html#SERIAL

    // If a user upgrades software or roots their device, there
will be a duplicate entry
    String serial = null;
    try {
        serial =
android.os.Build.class.getField("SERIAL").get(null).toString();

        // Go ahead and return the serial for api => 9
        return new UUID(m_szDevIDShort.hashCode(),
serial.hashCode()).toString();
    } catch (Exception exception) {
        // String needs to be initialized
        serial = "serial"; // some value
    }

    // Thanks @Joe!
    // http://stackoverflow.com/a/2853253/950427
    // Finally, combine the values we have found by using the
UUID class to create a unique identifier
    return new UUID(m_szDevIDShort.hashCode(),
serial.hashCode()).toString();
}
```

google ad

extra permission `xml <uses-permission`

```
android:name="android.permission.INTERNET" />
```

```
import
com.google.android.gms.ads.identifier.AdvertisingIdClient;
import
com.google.android.gms.ads.identifier.AdvertisingIdClient.Info;
import
com.google.android.gms.common.GooglePlayServicesAvailabilityExcepti

import
com.google.android.gms.common.GooglePlayServicesNotAvailableExcepti

import java.io.IOException;
...

// Do not call this function from the main thread. Otherwise,
// an IllegalStateException will be thrown.
public void getIdThread() {

  Info adInfo = null;
  try {
    adInfo = AdvertisingIdClient.getAdvertisingIdInfo(mContext);

  } catch (IOException exception) {
    // Unrecoverable error connecting to Google Play services
(e.g.,
    // the old version of the service doesn't support getting
AdvertisingId).

  } catch (GooglePlayServicesAvailabilityException exception) {
    // Encountered a recoverable error connecting to Google Play
services.

  } catch (GooglePlayServicesNotAvailableException exception) {
    // Google Play services is not available entirely.
  }
  final String id = adInfo.getId();
  final boolean isLAT = adInfo.isLimitAdTrackingEnabled();
}
```

google io with save pref

```
private static String uniqueID = null;
private static final String PREF_UNIQUE_ID = "PREF_UNIQUE_ID";

public synchronized static String id(Context context) {
    if (uniqueID == null) {
        SharedPreferences sharedPrefs =
context.getSharedPreferences(
                PREF_UNIQUE_ID, Context.MODE_PRIVATE);
        uniqueID = sharedPrefs.getString(PREF_UNIQUE_ID, null);
        if (uniqueID == null) {
            uniqueID = UUID.randomUUID().toString();
            Editor editor = sharedPrefs.edit();
            editor.putString(PREF_UNIQUE_ID, uniqueID);
            editor.commit();
        }
    }
    return uniqueID;
}
```