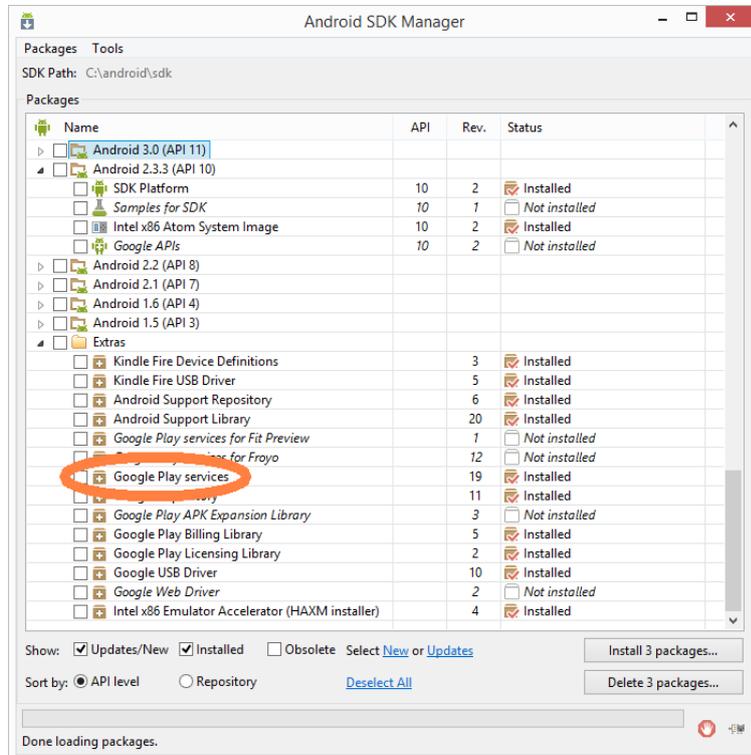
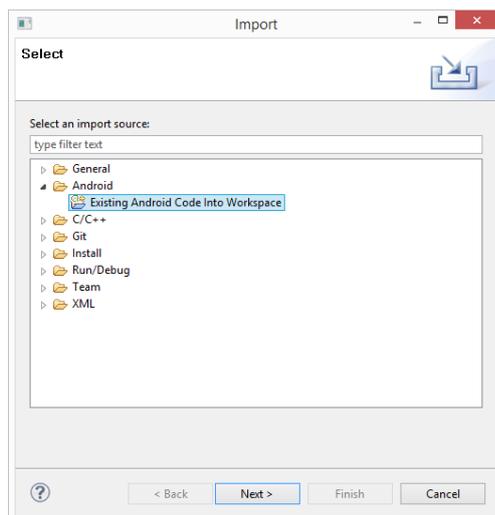


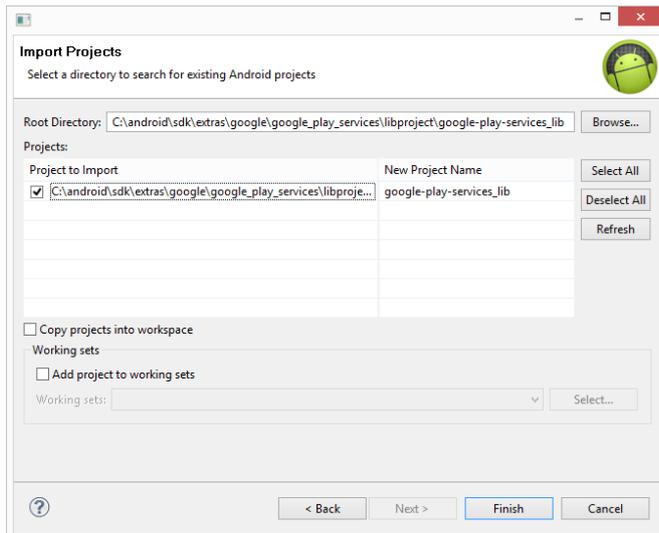
c't Longitude für Android – Aufsetzen des Projekts

Installieren Sie im Android SDK Manager die „Google Play services“:

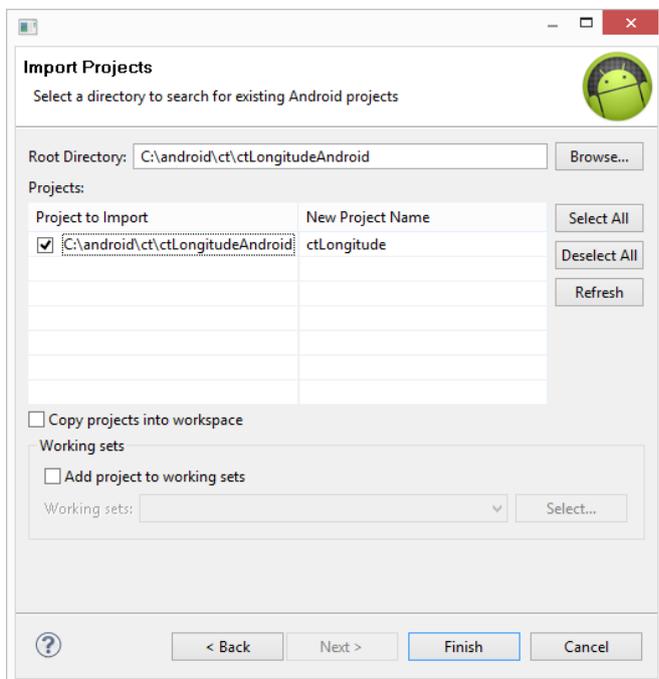


Starten Sie Eclipse und importieren dort mittels „File/Import/Existing Android Code Into Workspace“ das Projekt „google-play-services_lib“



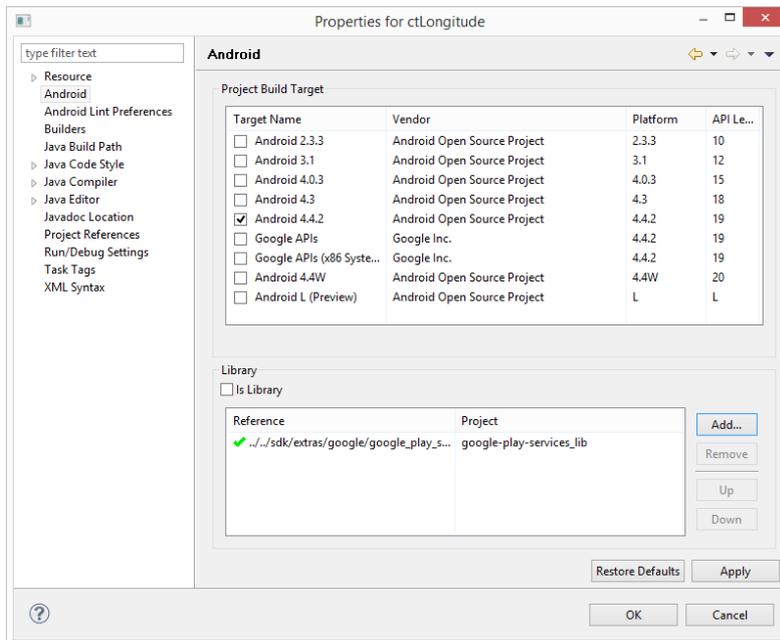


Importieren Sie nun das Beispielprojekt, ebenfalls via „File/Import/Existing Android Code Into Workspace“



Achtung, unter Windows müssen sich beide Projekte auf demselben Laufwerk befinden!

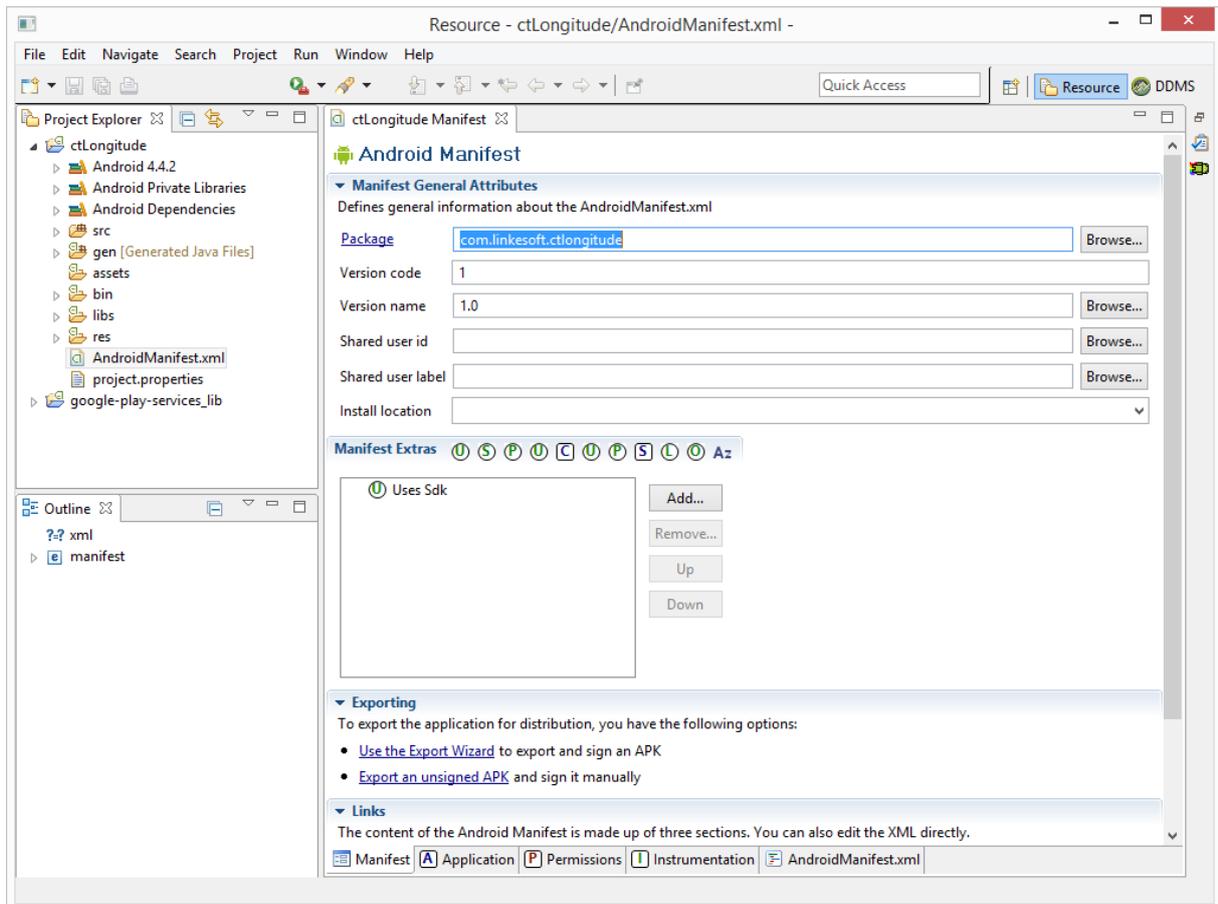
Wählen Sie das ctLongitude-Projekt aus. Vergewissern Sie sich unter „File/Properties/Android“, dass unter „Library“ die oben angelegte google-play-services_lib korrekt referenziert ist.



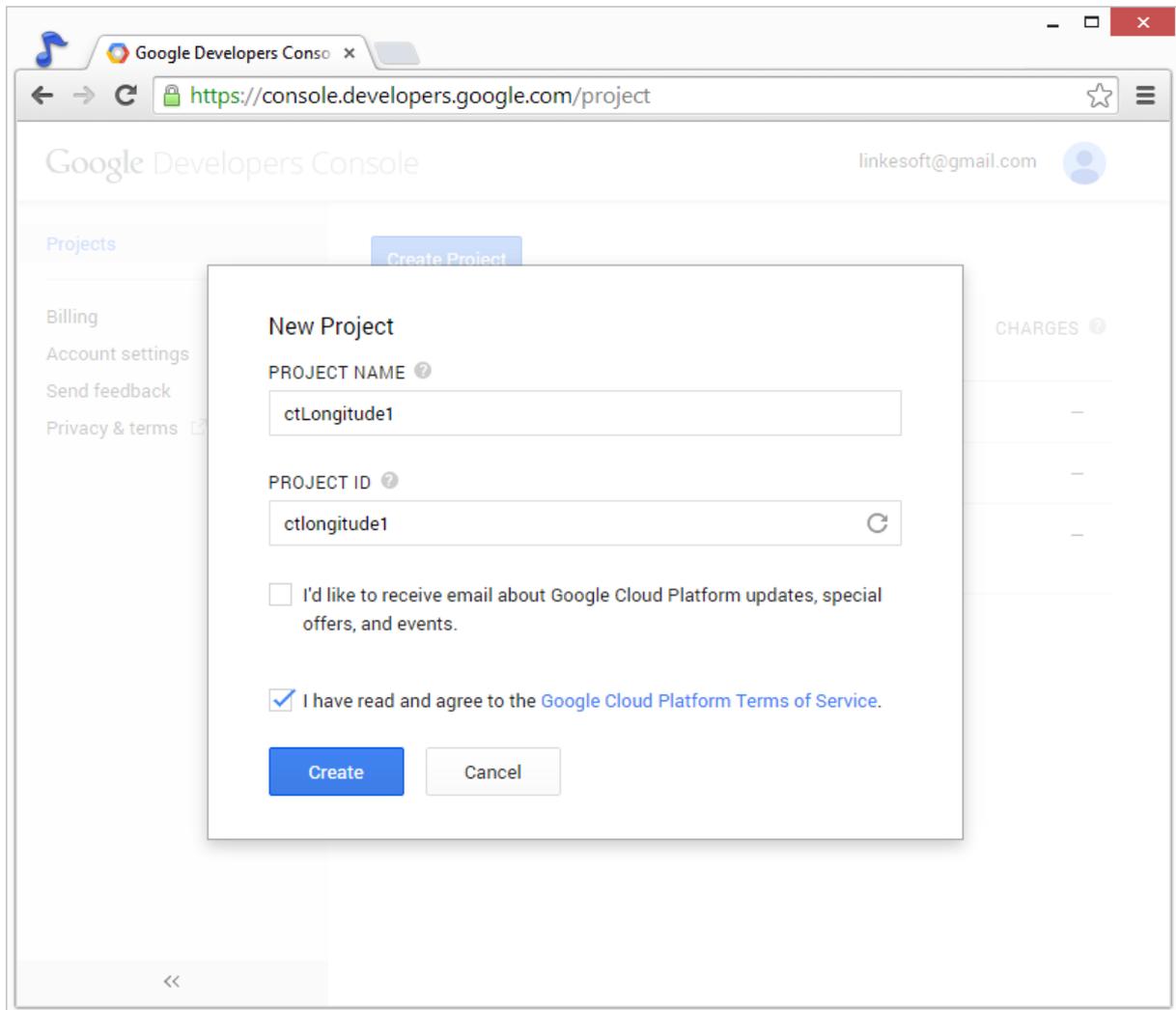
(Falls nicht, löschen Sie die Referenz und fügen Sie sie über Add... neu hinzu.)

Um die App selbst bauen zu können, müssen Sie sie in der Google Developer Console registrieren. Wie das geht, steht im c't-Artikel.

Ändern Sie dazu das Package der App im AndroidManifest, zum Beispiel auf org.<ihrname>.ctlongitude.



Gehen Sie nun zu <https://console.developers.google.com/> legen dort ein neues Projekt (mit beliebiger ID) an.



Entfernen Sie unter „APIs&auth/APIs“ alle vorausgewählten APIs und fügen stattdessen Google Maps Android API v2 hinzu.

The screenshot displays the Google Developers Console interface for the project 'ctLongitude1'. The left sidebar shows navigation options under 'APIs & auth', with 'APIs' selected. The main content area is divided into two sections: 'Enabled APIs' and 'Browse APIs'.

Enabled APIs

Some APIs are enabled automatically. You can disable them if you're not using their services.

| NAME | QUOTA | STATUS |
|--|-------|--------|
| Google Maps Android API v2 | | ON |

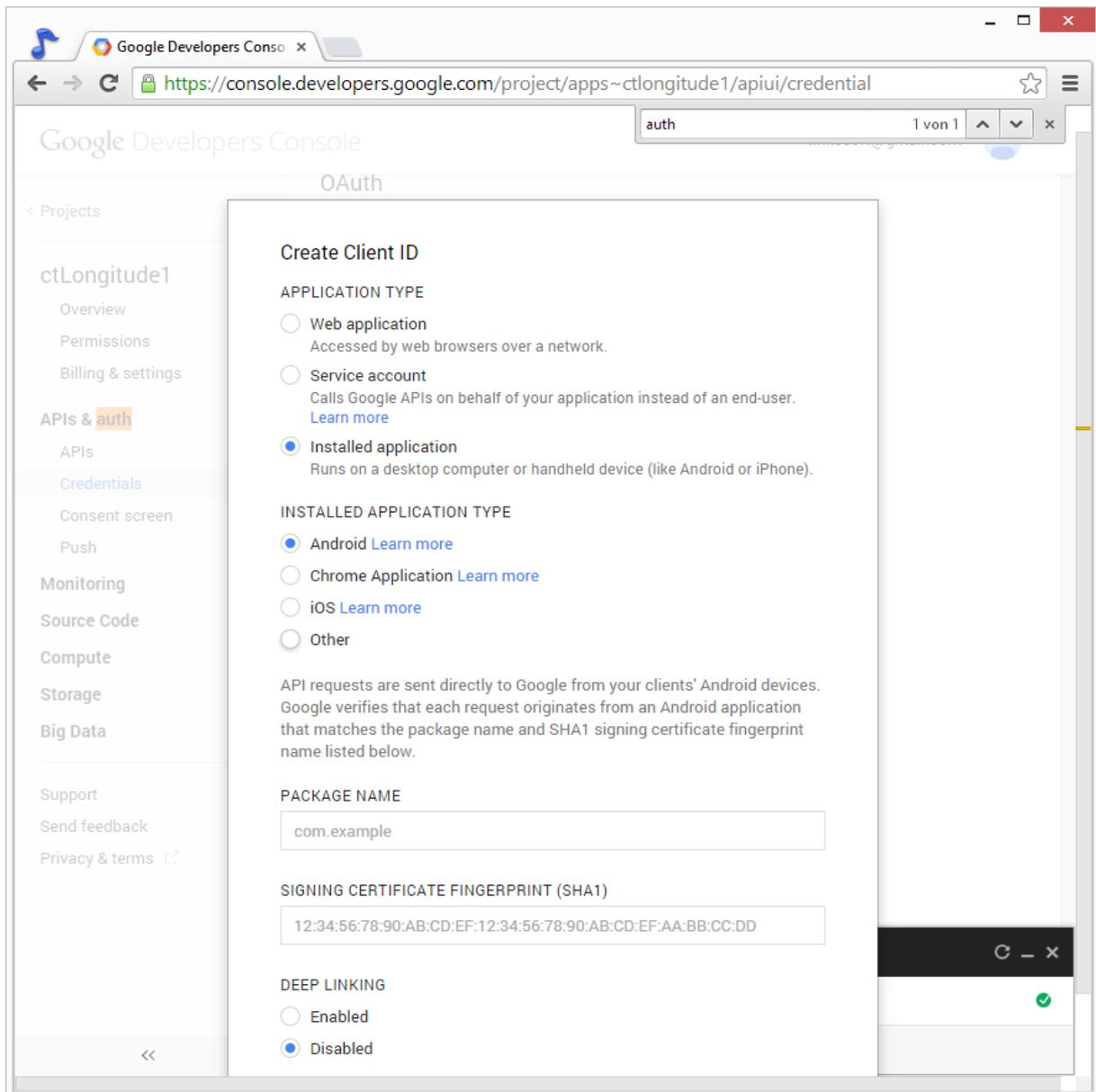
Browse APIs

Filter by API name or description

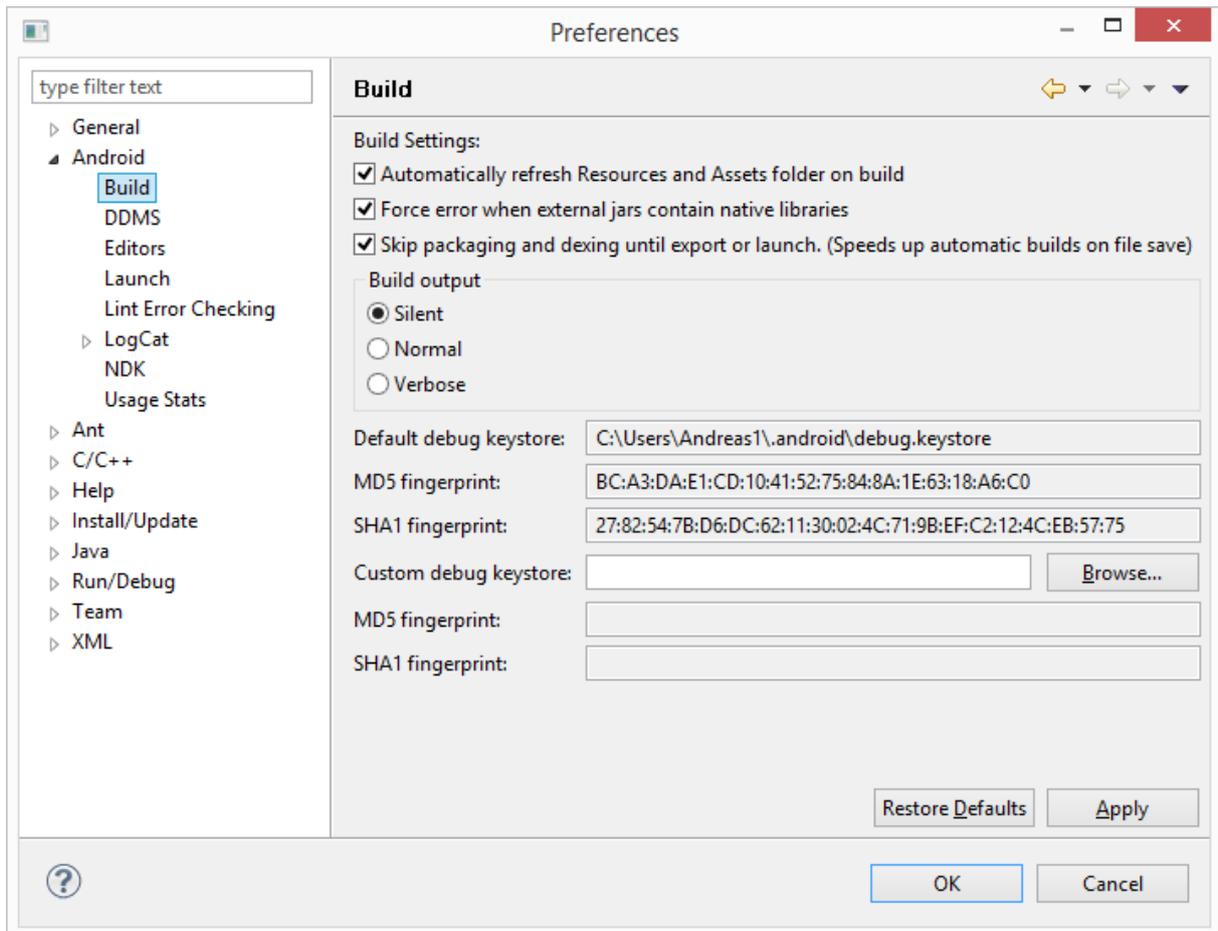
| NAME | QUOTA | STATUS |
|--|-------------------------|--------|
| Ad Exchange Buyer API | 1,000 requests/day | OFF |
| Ad Exchange Seller API | 10,000 requests/day | OFF |
| Admin SDK | 150,000 requests/day | OFF |
| AdSense Host API | 100,000 requests/day | OFF |
| AdSense Management API | 10,000 requests/day | OFF |
| Analytics API | 50,000 requests/day | OFF |
| Apps Activity API | 10,000,000 requests/day | OFF |
| Audit API | | |
| BigQuery API | | |
| Blogger API v3 | | |

An 'Activities (Idle)' window is open at the bottom, showing a successful event: 'Create Project: ctLongitude1' with a green checkmark. A 'See all activity' link is also visible.

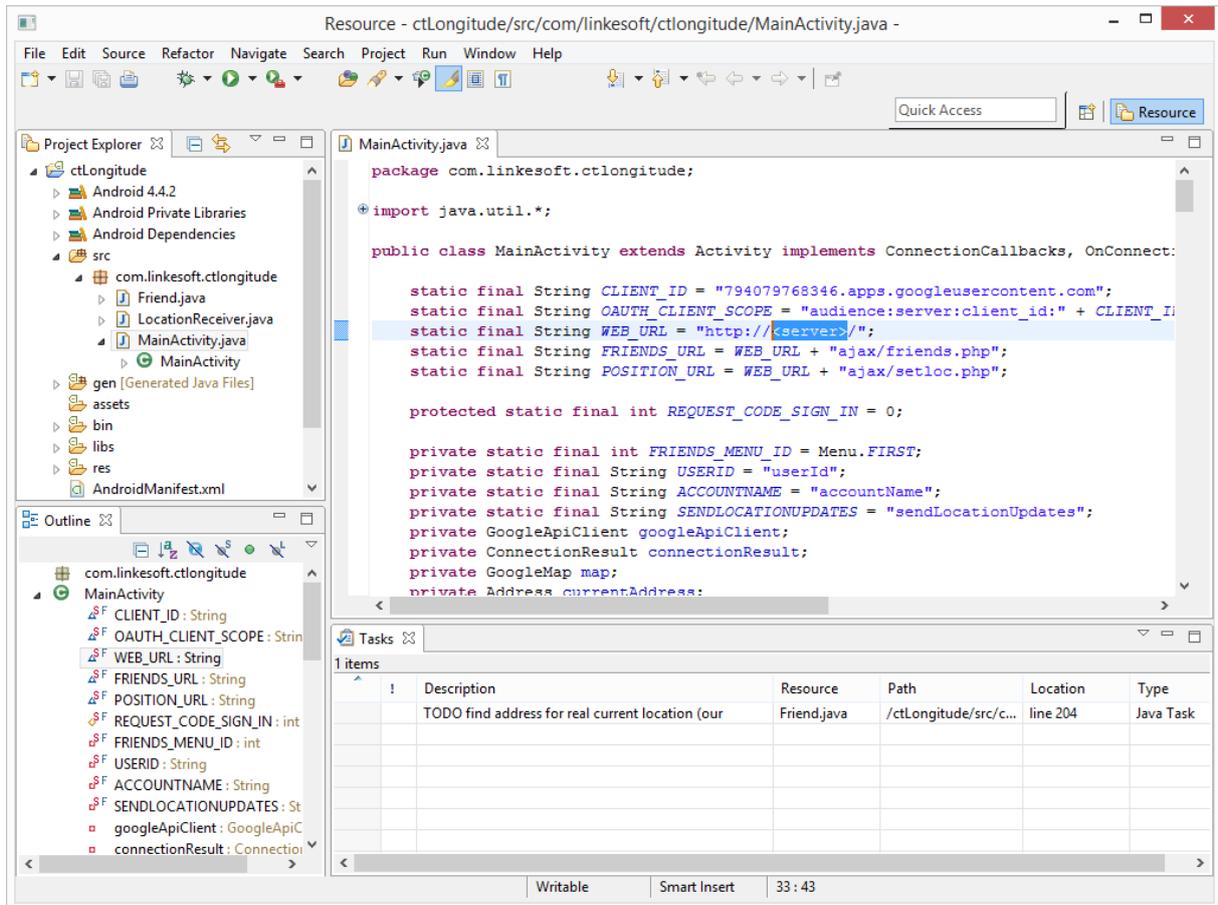
Unter Credentials müssen Sie nun wie im Artikel „Aufgeschlossen“ (c't 12/2014, S. 172) beschrieben für OAuth und Public APIs Ihre Android App mit dem Fingerprint Ihres Code-Signing-Zertifikats registrieren.



Den SHA1 Fingerprint für Eclipse-Builds finden Sie in Eclipse unter „Window/Preferences/Android/Build“.



Öffnen Sie nun in Eclipse die Klasse MainActivity und fügen in der Konstante WEB_URL die Adresse Ihres c't-Longitude-Servers ein.



The screenshot shows the Eclipse IDE with the following components:

- Project Explorer:** Shows the project structure for 'ctLongitude'. The 'src' folder contains 'com.linkesoft.ctlongitude', which includes 'Friend.java', 'LocationReceiver.java', and 'MainActivity.java'. The 'gen' folder contains 'Generated Java Files'. Other folders include 'assets', 'bin', 'libs', and 'res'.
- Outline:** Shows the class members of 'MainActivity':
 - CLIENT_ID : String
 - OAUTH_CLIENT_SCOPE : String
 - WEB_URL : String
 - FRIENDS_URL : String
 - POSITION_URL : String
 - REQUEST_CODE_SIGN_IN : int
 - FRIENDS_MENU_ID : int
 - USERID : String
 - ACCOUNTNAME : String
 - SENDLOCATIONUPDATES : String
 - googleApiClient : GoogleApiC
 - connectionResult : Connection
- MainActivity.java:** The code is as follows:

```
package com.linkesoft.ctlongitude;

import java.util.*;

public class MainActivity extends Activity implements ConnectionCallbacks, OnConnect:

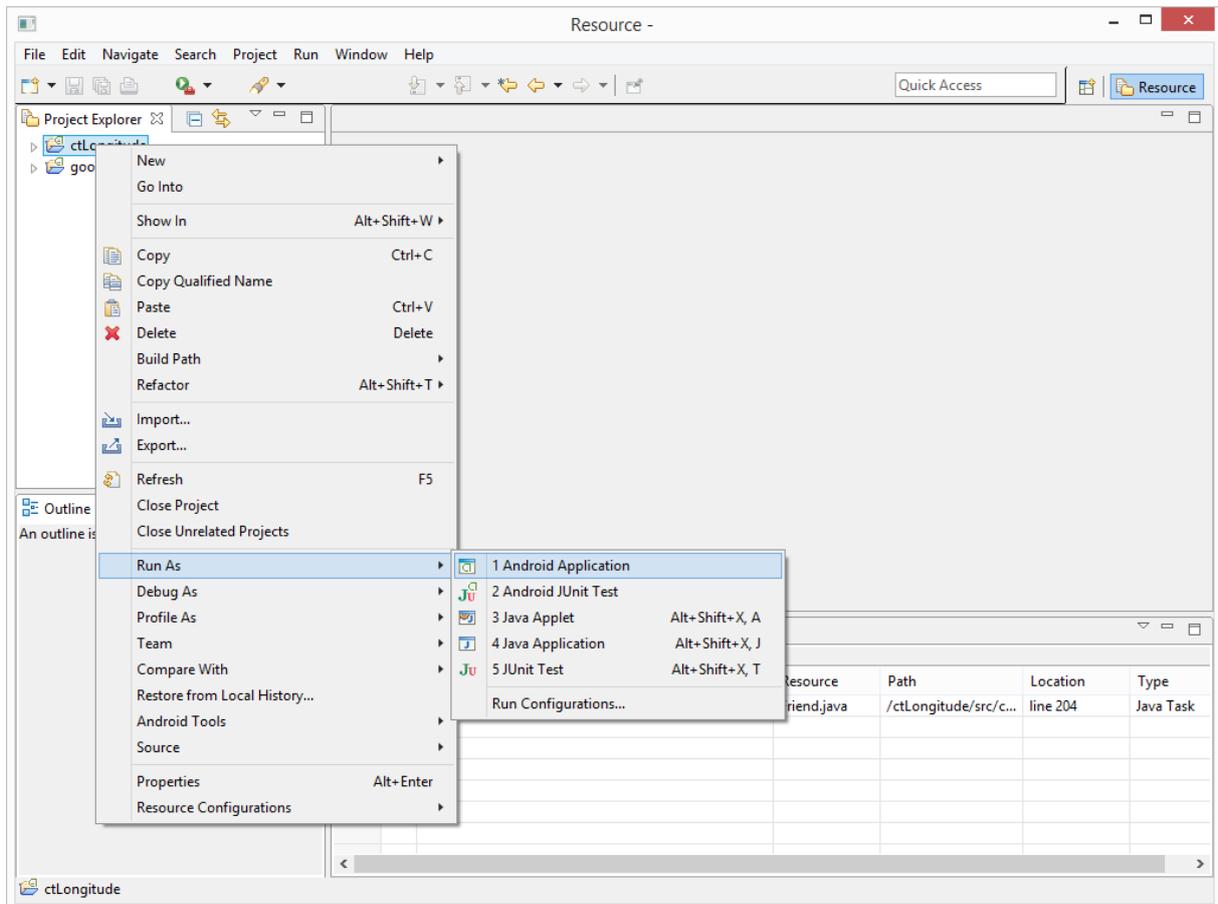
    static final String CLIENT_ID = "794079768346.apps.googleusercontent.com";
    static final String OAUTH_CLIENT_SCOPE = "audience:server:client_id:" + CLIENT_ID;
    static final String WEB_URL = "http://kserver/";
    static final String FRIENDS_URL = WEB_URL + "ajax/friends.php";
    static final String POSITION_URL = WEB_URL + "ajax/setloc.php";

    protected static final int REQUEST_CODE_SIGN_IN = 0;

    private static final int FRIENDS_MENU_ID = Menu.FIRST;
    private static final String USERID = "userId";
    private static final String ACCOUNTNAME = "accountName";
    private static final String SENDLOCATIONUPDATES = "sendLocationUpdates";
    private GoogleApiClient googleApiClient;
    private ConnectionResult connectionResult;
    private GoogleMap map;
    private Address currentAddress;
```
- Tasks:** Shows 1 item:

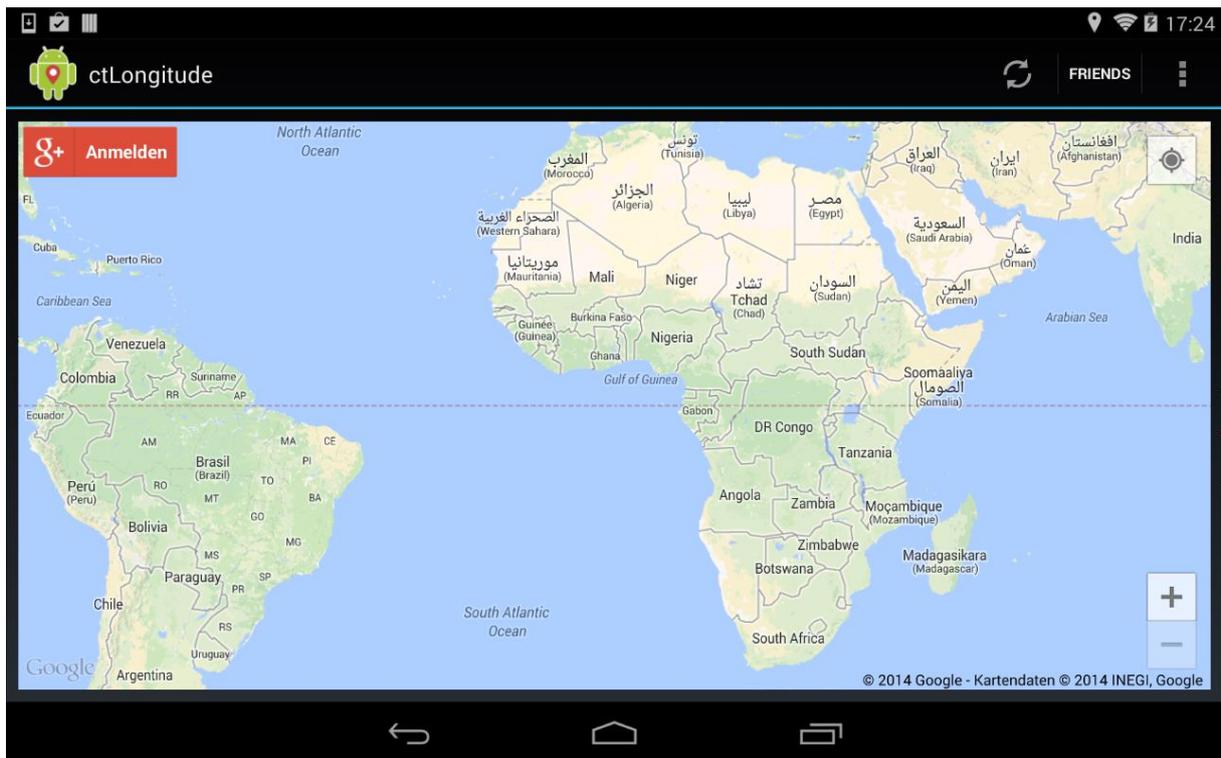
| ! | Description | Resource | Path | Location | Type |
|---|--|-------------|-----------------------|----------|-----------|
| | TODO find address for real current location (our | Friend.java | /ctLongitude/src/c... | line 204 | Java Task |

Anschließend starten Sie durch Rechtsklick auf das Projekt über „Run As/Android Application“ die App auf dem angeschlossenen Android-Gerät.



(Auf dem Simulator funktioniert die App nicht, da dort die google-play-services nicht verfügbar sind.)

Auf dem Gerät sollte die App wie folgt aussehen



Durch Tippen auf den roten G+-Anmeldeknopf starten Sie den Login bei Google+ und sehe dann nach kurzer Wartezeit die Positionen Ihrer Freunde auf der Karte.

