

日本Androidの会 金沢支部 ハンズオン Oct 23, 2010

# Google Maps との 連携

有山圭二(日本Androidの会 関西支部)

# ■ Android と Google Maps

- Androidプラットフォーム搭載端末の中でも、キャリアが販売している携帯電話には、Google MapsやGmailなど、Googleのサービスと直接連携するアプリが同梱されている
- 特にGoogle Mapsは、Androidのフレームワークにも統合されており、開発者がアプリの部品としてGoogle Mapsを表示できる。



と、その前に...

# Interlude

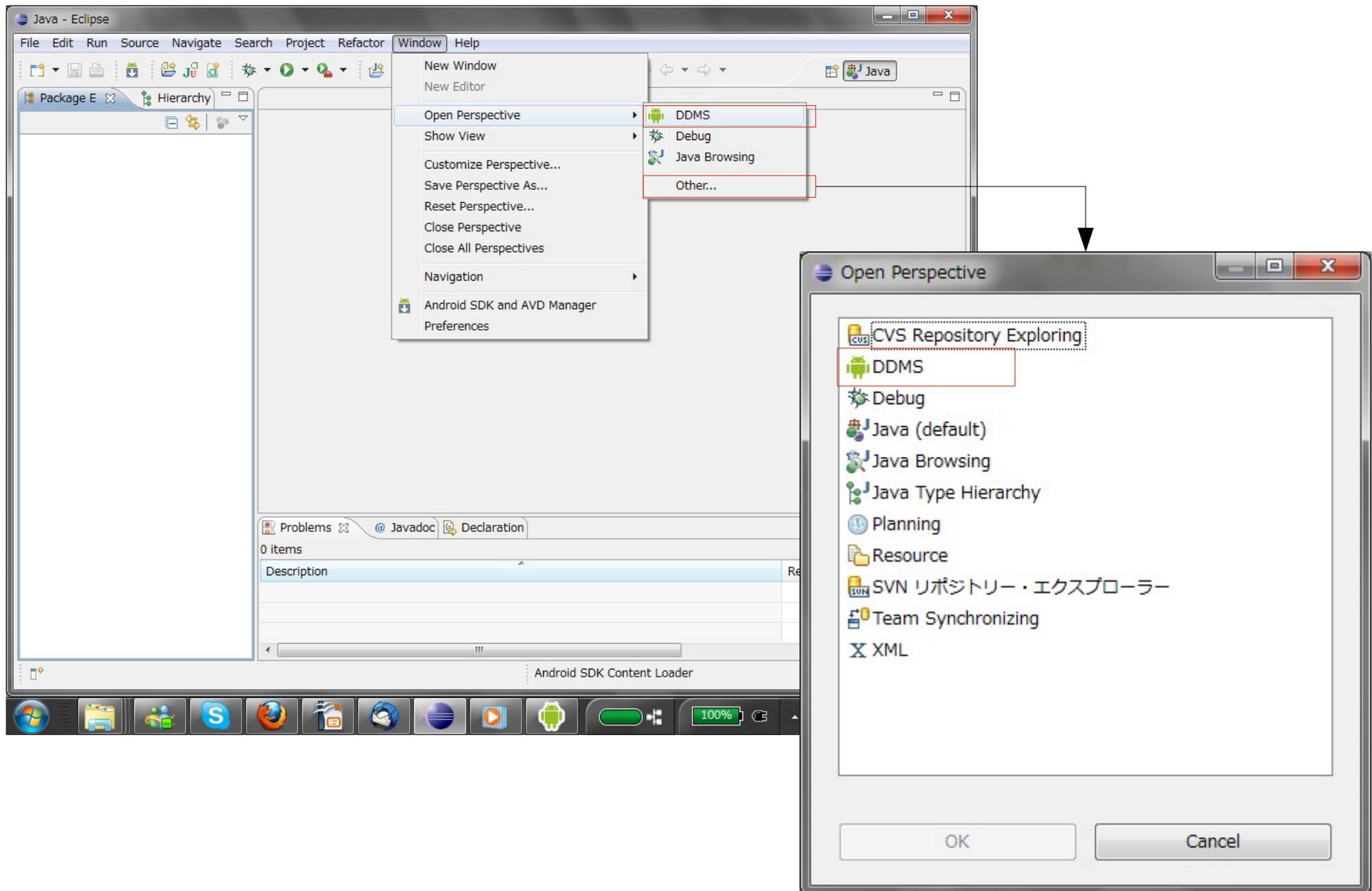
## Androidアプリの デバッグ方法

## ■ エラーの発生



原因は？

# ■ DDMSの表示



# ■ DDMS – Dalvik Debug Monitor Server

## ○ Androidアプリのデバッグ用情報表示画面

The screenshot shows the DDMS interface in Eclipse. The 'Logcat' panel is highlighted with a red box and contains the following log entries:

Time	pid	tag	Message
07-08 00:34...	I 75	ActivityManager	Process android.process.media (pid 616) has died.
07-08 00:34...	D 633	dalvikvm	GC freed 6521 objects / 352016 bytes in 171ms
07-08 00:34...	D 633	dalvikvm	GC freed 1684 objects / 201208 bytes in 212ms
07-08 00:34...	D 75	NetworkLocationPr...	onCellLocationChanged [11775.7917647]
07-08 00:34...	D 633	dalvikvm	GC freed 10405 objects / 643992 bytes in 167ms
07-08 00:35...	D 75	NetworkLocationPr...	onCellLocationChanged [11775.7917643]
07-08 00:35...	D 131	dalvikvm	GC freed 689 objects / 36920 bytes in 231ms

- Devices  
動作しているエミュレータ。接続している端末の一覧を表示する
- EmulatorControl  
エミュレータへの位置情報の設定や、電話の着信等を制御する
- Logcat  
ログを表示する
- File Explorer  
Devicesで選択したエミュレータ(端末)のファイル情報を表示する

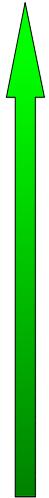
# ■ ログ出力 (Log)

## ○ ログレベル+タグ+内容

- Log.e( タグ, 内容 );
- Log.w
- Log.i
- Log.d
- Log.v

ログレベル 重要度高

- Error
- Warning
- Info
- Debug
- Verbose

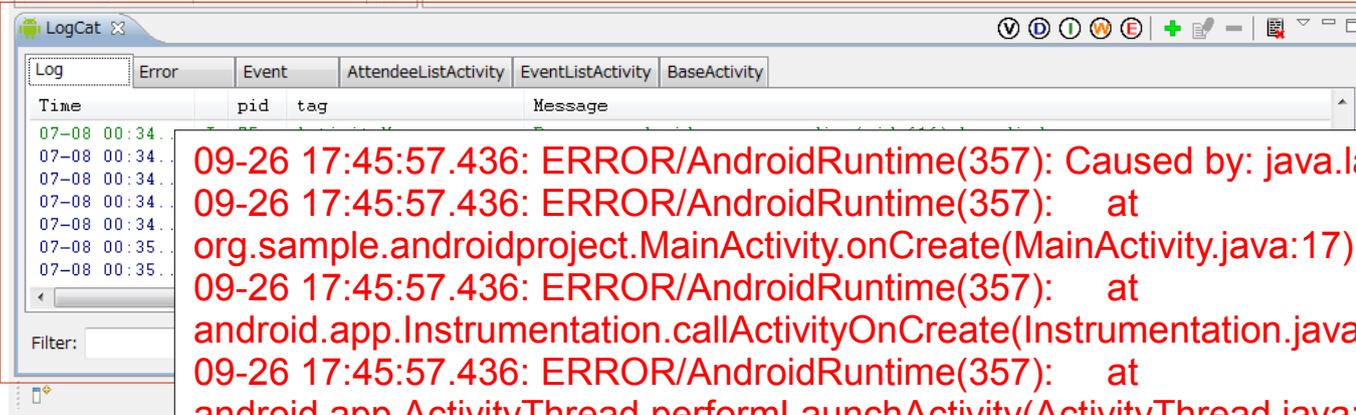
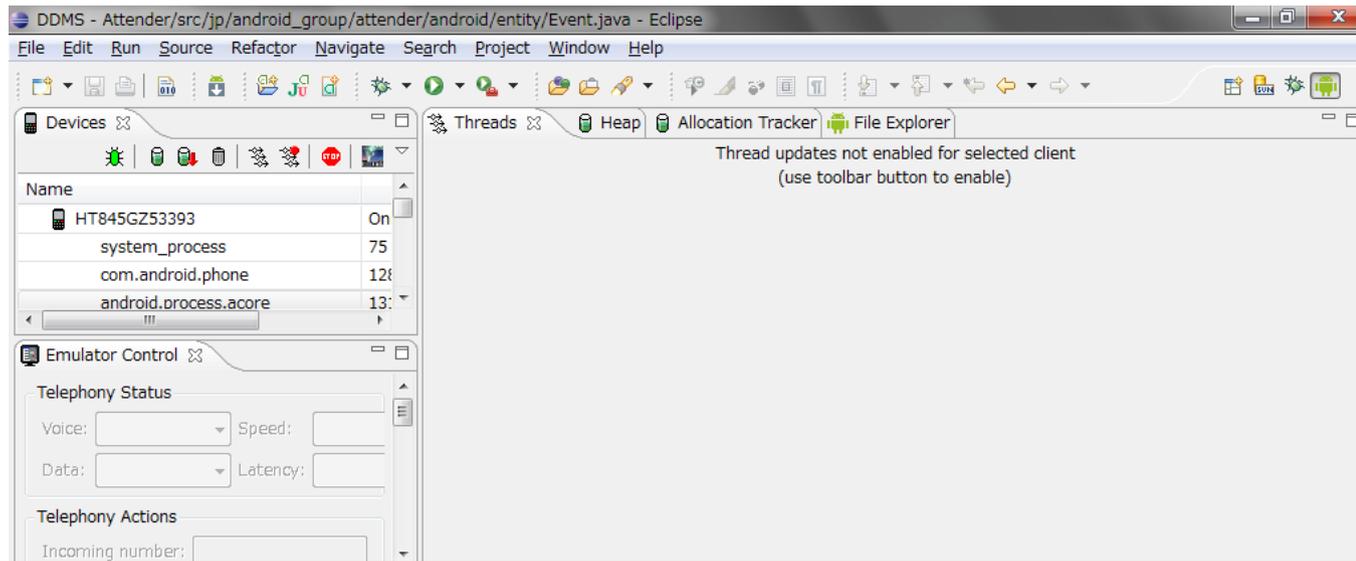


```
public class MainActivity extends Activity {
    public static String LOG_TAG = "HelloWorld";

    /*
     * (non-Javadoc)
     * @see android.app.Activity#onCreate(android.os.Bundle)
     */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // ログの出力
        Log.e(LOG_TAG, "onCreate");
        Log.w(LOG_TAG, "onCreate");
        Log.i(LOG_TAG, "onCreate");
        Log.d(LOG_TAG, "onCreate");
        Log.v(LOG_TAG, "onCreate");
    }
}
```

# ■ ログの確認



09-26 17:45:57.436: ERROR/AndroidRuntime(357): Caused by: java.lang.NullPointerException  
09-26 17:45:57.436: ERROR/AndroidRuntime(357): at  
org.sample.androidproject.MainActivity.onCreate(MainActivity.java:17)  
09-26 17:45:57.436: ERROR/AndroidRuntime(357): at  
android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1123)  
09-26 17:45:57.436: ERROR/AndroidRuntime(357): at  
android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2364)  
09-26 17:45:57.436: ERROR/AndroidRuntime(357): ... 12 more

## MainActivityの17行目

## ■ ログフィルタの登録

- 必要なログだけをフィルタして確認出来る

A dialog box titled "Log Filter" with a close button (X) in the top right corner. It contains four input fields: "Filter Name" with the text "SampleFilter", "by Log Tag" with the text "HelloWorld", "by pid" which is empty, and "by Log level" with a dropdown menu showing "<none>". At the bottom are "OK" and "Cancel" buttons.

- Filter Name  
フィルタ名

- Log Tag  
ログタグでのフィルタ条件

- pid  
プロセスIDでのフィルタ条件

- Log level  
ログレベルでのフィルタ条件

## ■ Log出力時の注意点

- ログの出力は、オーバーヘッドが大きい。フラグで一括ON/OFFの切り替えが出来るようにしておく
- 下の例では、DEBUG\_FLG定数をfalseに設定すれば、ログを出力しなくなる。

```
public class MainActivity extends Activity {
    public static String LOG_TAG = "HelloWorld";
    public static boolean DEBUG_FLG = true;

    /*
     * (non-Javadoc)
     *
     * @see android.app.Activity#onCreate(android.os.Bundle)
     */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // ログの出力
        if(DEBUG_FLG) Log.e(LOG_TAG, "onCreate");
        if(DEBUG_FLG) Log.w(LOG_TAG, "onCreate");
        if(DEBUG_FLG) Log.i(LOG_TAG, "onCreate");
        if(DEBUG_FLG) Log.d(LOG_TAG, "onCreate");
        if(DEBUG_FLG) Log.v(LOG_TAG, "onCreate");
    }
}
```

# Google Maps との連携

# ■ AndroidManifest.xml の変更

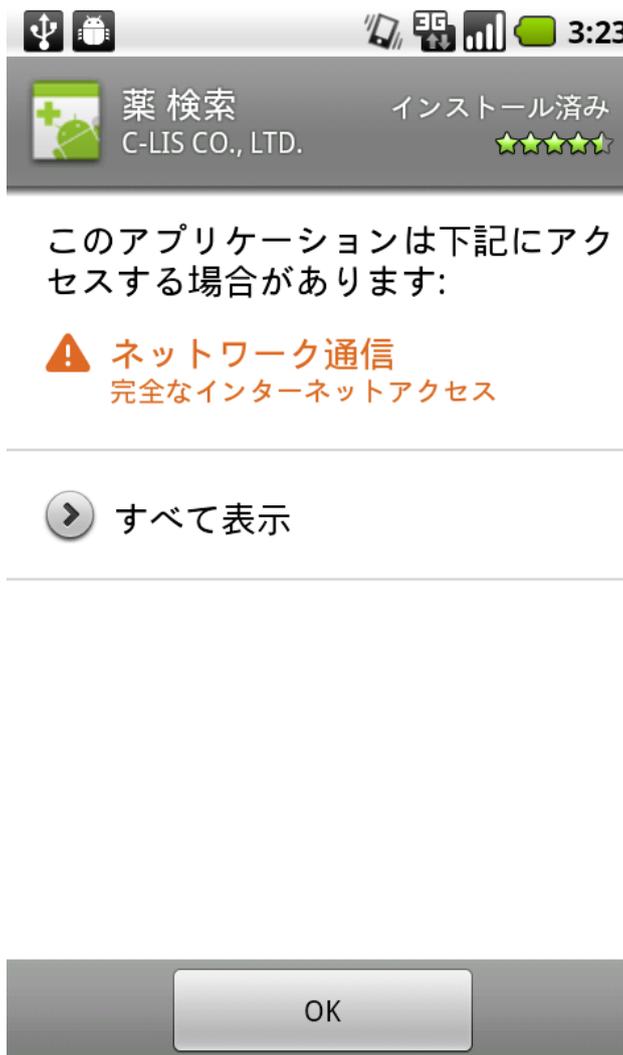
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="1"
    android:versionName="1.0" package="org.sample.androidproject">
    <uses-sdk android:minSdkVersion="4" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name="MainActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <uses-library android:name="com.google.android.maps" />
    </application>
</manifest>
```

追記

- パーMISSIONの追加
  - Google Mapsは、サーバーから地図データを取得しているため、ネットワークにアクセスするパーMISSION (INTERNET) が必要。
- ライブラリの利用を宣言
  - Androidでは、Google Mapsは拡張ライブラリという位置づけにある。

## ■ インストール時の表示

- アプリケーションが利用する機能を一覧表示する



## ■ 代表的なパーミッション

名前	説明
CALL_PHONE	電話をかける
CAMERA	カメラプレビュー、写真／動画撮影を行う
CHANGE_CONFIGURATION	端末の設定を変更する
INSTALL_PACKAGES	端末に別のアプリケーションをインストールする
INTERNET	インターネットに接続して、情報を送受信する
READ_CALENDAR	カレンダー情報を読み込む
READ_CONTACTS	電話帳情報を読み込む
VIBRATE	端末のバイブレーションを制御する

## ■ レイアウトファイルの作成

- res/layout/map\_activity.xml を作成

```
<?xml version="1.0" encoding="utf-8"?>
<com.google.android.maps.MapView
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/map_activity_map"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:apiKey="[APIKey]"
 />
```

- APIKeyを取得する必要がある（方法は後述）

# ■ Mapを表示する

## ○ MapActivityを継承

- MainActivityの親クラスを、ActivityからMapActivityに変更
- isRouteDisplayed() メソッドをオーバーライド

```
public class MainActivity extends MapActivity {  
    private static final boolean DEBUG_FLAG = true;  
    private static final String LOG_TAG = "MapActivity"  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.map_activity)  
    }  
  
    @Override  
    protected boolean isRouteDisplayed() {  
        return false;  
    }  
}
```

変更

追記

## ■ API-Keyを取得する

- MapViewを利用するには、APIKeyを取得する必要がある。
- APIKeyの取得は、Googleの用意したWEBのフォームから行う。
- 申請には、アプリ開発に使用している開発用の証明書のフィンガープリント(MD5)が必要となる。

## ■ 証明書のフィンガープリントを取得

- 開発用の証明書は、  
「ユーザーのホームディレクトリ/.android/debug.keystore」  
にある
- ユーザーのホームディレクトリは、Windowsの場合、  
コマンドプロンプトから下記のように入力して確認出来る

```
> echo %USERPROFILE%
```

## ■ フィンガープリントを取得

- フィンガープリントの取得には、JDKに含まれるkeytoolを使う。

```
> cd [jdk_path]¥bin  
> keytool -list -keystore [開発用証明書のパス]
```

```
キーストアのパスワードを入力してください:[Enterを入力]
```

```
***** 警告 警告 警告 *****  
* キーストアに保存された情報の完全性は検証されて *  
* いません! 完全性を検証するには、キーストアの *  
* パスワードを入力する必要があります。 *  
***** 警告 警告 警告 *****
```

```
キーストアのタイプ: JKS  
キーストアのプロバイダ: SUN
```

```
キーストアには 1 エントリが含まれます。
```

```
androiddebugkey, 2009/08/20, PrivateKeyEntry,  
証明書のフィンガープリント (MD5): D3:39:53:F2:BF:85:44:F2:37:C7:DA:73:44:09:DE:AF
```

```
----
```

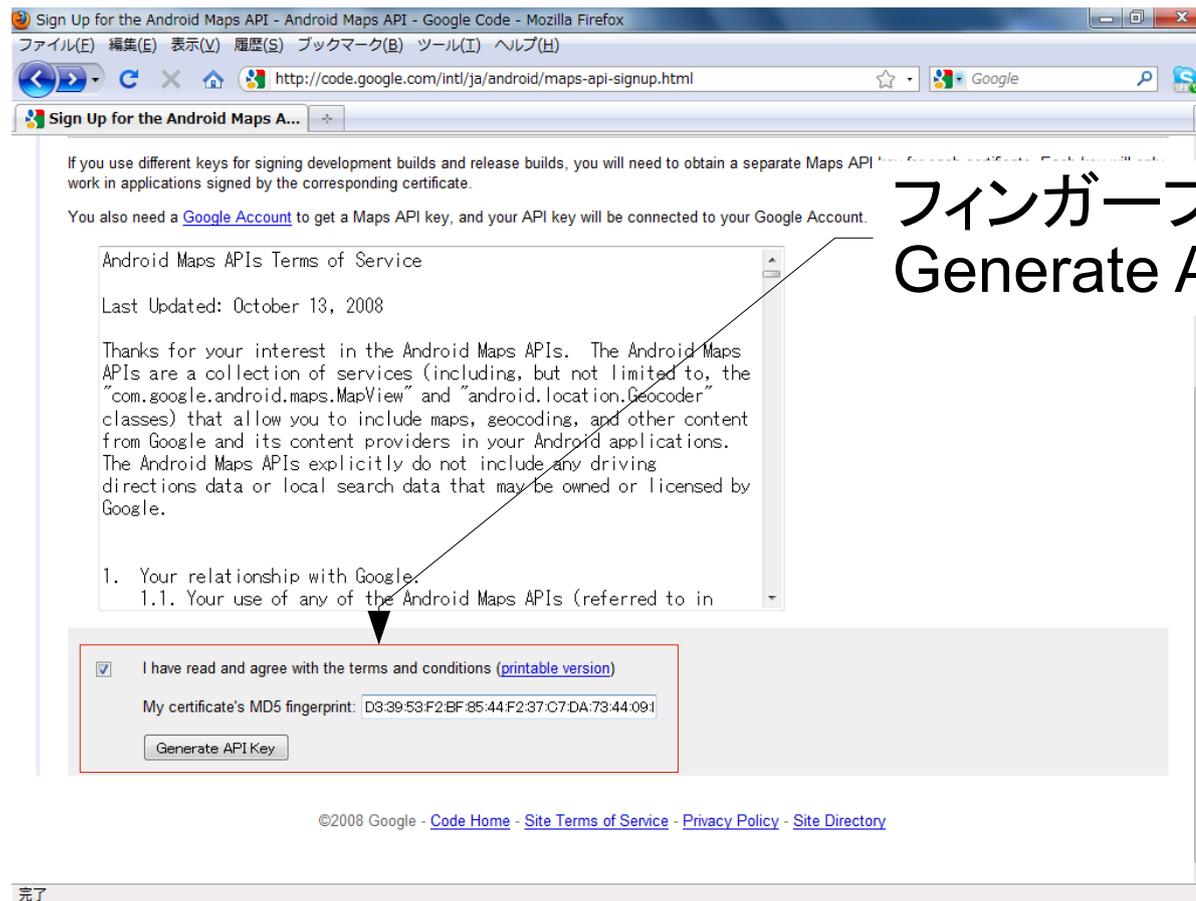


この文字列を保存

# ■ API-Keyの発行

- API-Key発行用のページを開く

http://code.google.com/intl/ja/android/maps-api-signup.html



フィンガープリントを入力して  
Generate APIKey ボタンを押す

完了

# API-Keyの発行

Android Maps API - ありがとうございます - Mozilla Firefox

ファイル(E) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(I) ヘルプ(H)

http://www.google.com/glm/mmap/a/api?fp=D3%3A39%3A53%3AF2%3ABF%3A85%3A44%3AF2%3A37%3AC7%3ADA%3A73%3A44%3A09%3ADE%3AAF

Google Maps API

Google Codeホーム > Google Maps API > Google Maps APIのアカウント登録

Android Maps APIキーにサインアップしていただき、ありがとうございます。

あなたのキーは次のとおりです:

**0x87A3uvHU02X89rs8gj8va3f9X1TelaYGWkizg** ← **保存する**

このキーは、以下のフィンガープリントによる認証を使用したすべてのアプリケーションで有効です:

D3:39:53:F2:BF:85:44:F2:37:C7:DA:73:44:09:DE:AF

以下に、地図を活用するためのxmlレイアウトの例を示します:

```
<com.google.android.maps.MapView
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:apiKey="0x87A3uvHU02X89rs8gj8va3f9X1TelaYGWkizg"
/>
```

詳細については、[APIのドキュメント](#)を参照してください。

©2008 Google - [プライバシーポリシー](#) - [採用情報](#) - [利用規約](#) - [Googleについて](#)

完了

## ■ API-Keyの設定

- res/layout/map\_view.xml に記載

```
<?xml version="1.0" encoding="utf-8"?>  
<com.google.android.maps.MapView  
  xmlns:android="http://schemas.android.com/apk/res/android"  
  android:id="@+id/map_view_map"  
  android:layout_width="fill_parent"  
  android:layout_height="fill_parent"  
  android:apiKey="0x87A3uvHU02X89rs8gj8va3f9X1TelaYGWkizg"  
 />
```

- プロジェクトを実行する

## ■ Mapsの注意事項

- API-Keyは、証明書毎に必要
- 複数人数で開発をする場合は、開発用証明書を共有すると、スムーズに開発出来る
- Android Marketでアプリを公開する場合、公開用の証明書を作成する必要がある。その場合は、公開用の証明書に応じたAPI-Keyの取得と設定が必要。

# ■ 地図を操作する

- 指定した座標を表示
  - なお、座標は経度・緯度ではなく、  
経度・緯度の値に1E6(10000000)乗算したものを指定する

```
public class MainActivity extends MapActivity {
    private static final boolean DEBUG_FLAG = true;
    private static final String LOG_TAG = "MapActivity";

    MapController mMapController = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.map_view);

        MapView mapView = (MapView) findViewById(R.id.map_view_map);
        mMapController = mapView.getController();

        // 指定した座標を表示
        mMapController.animateTo(new GeoPoint(34994732, 135740154));
    }

    @Override
    protected boolean isRouteDisplayed() {
        return false;
    }
}
```

# ■ 地図を操作する

## ○ 拡大率の変更

```
public class MainActivity extends MapActivity {
    private static final boolean DEBUG_FLAG = true;
    private static final String LOG_TAG = "MapActivity";

    MapController mMapController = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.map_view);

        MapView mapView = (MapView) findViewById(R.id.map_view_map);
        mMapController = mapView.getController();

        // ズームレベルを12へ変更
        mMapController.setZoom(12);
    }

    @Override
    protected boolean isRouteDisplayed() {
        return false;
    }
}
```

# ■ 指定の座標にアイコンを追加する

```
public class MainActivity extends com.google.android.maps.MapActivity {
    private static final boolean DEBUG_FLAG = true;
    private static final String LOG_TAG = "MainActivity";

    // Mapに重ねるオーバーレイ
    private static class IconOverlay extends ItemizedOverlay<OverlayItem> {

        List<GeoPoint> mPointList = null;
        public void setPointList(List<GeoPoint> list) {
            mPointList = list;

            // OverlayItemリストの再構築
            if(mPointList != null) populate();
        }

        public IconOverlay(Drawable defaultMarker) {
            super(defaultMarker);
        }

        @Override
        protected OverlayItem createItem(int i) {
            GeoPoint p = mPointList.get(i);
            String title = "title " + i;
            String snippet = "snippet " + i;
            OverlayItem item = new OverlayItem(p, title, snippet);
            return item;
        }

        @Override
        public int size() {
            if(mPointList == null) return 0;
            return mPointList.size();
        }
    }
}
```

# ■ 指定座標にアイコンを追加する

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

    MapView mapView = (MapView) findViewById(R.id.map_view_map);
    MapController mapController = mapView.getController();

    List<Overlay> overlays = mapView.getOverlays();

    // 指定した座標を表示
    mapController.animateTo(new GeoPoint(34994732, 135740154));

    // 画像を、リソースから取得
    Drawable pin = getResources().getDrawable(R.drawable.icon);
    pin.setBounds(0, 0, pin.getMinimumWidth(), pin.getMinimumHeight());

    // リソースからアイコン画像の取得
    IconOverlay iconOverlay = new IconOverlay(pin);

    // アイコンを表示する場所を設定
    LinkedList<GeoPoint> list = new LinkedList<GeoPoint>();
    list.add(new GeoPoint(34994732, 135740154));
    iconOverlay.setPointList(list);

    // オーバーレイに追加
    overlays.add(iconOverlay);
}

@Override
protected boolean isRouteDisplayed() {
    return false;
}
}
```

# 質疑応答