Extract text on image and translate android app source code is available now.

See example in Play Store

ANDROID – HOW TO DRAW PATH BETWEEN 2 POINTS ON GOOGLE MAP

In this tutorial, we are going to learn to how to draw path between 2 points on Google Map API V2. Drawing a route on android Google Map API v2 can be challenging but in this tutorial we will see what is need to draw a path from a user current location to a point in the map that as serves as the user destination.

If you have not used Android Location API before to obtain user current location, I will suggest you read my tutorial on **Android Location Service API using Google Play Services.**

WHAT WE ARE GOING TO ACHIEVE

Scan & Translate Android App
Source code
VIEW APP SAMPLE

WRITE FOR US

ANDROID SOURCE CODE POLL

Which of the below listed source codes should we publish next month?

Multi Restaurant Food Ordering App (27%, 39 Votes)

Stock and Inventory App (18%, 26 Votes)

Android Dating App (16%, 23

Accept

3. Draw a path overlay on the map between this locations

Before we start it is important that that we understand what we are planning to achieve in this tutorial. I have add some screen-shot from this tutorial

APP SCREEN-SHOT

Android E-Book App (12%, 17 Votes)

Appointment Booking App (10%, 15 Votes)

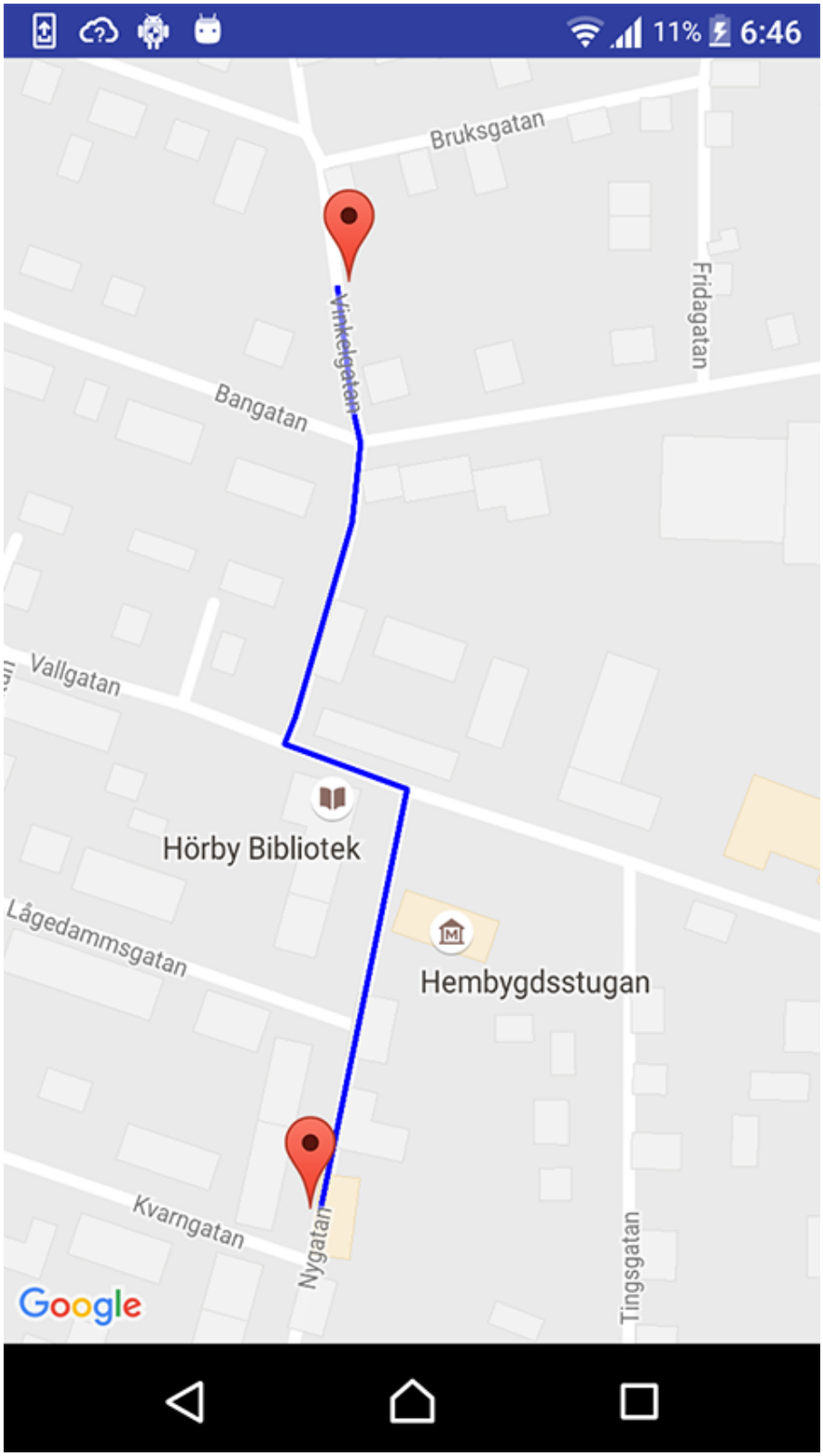
Business & Service Provider Finder (6%, 9 Votes)

Total Voters: **147**

LIKE US ON FACEBOOK

Induce smile
5,305 likes
Like Page
Be the first of your friends to like this

RECENT POSTS



- Android Retrofit 2 with JSON API Example
- Android Room with LiveData Example Tutorial
- Android UI App for Food Ordering and Delivery from Multiple Restaurants

CREATE NEW ANDROID PROJECT

Lets start to soil our hands in code. Start up your IDE. For this tutorial, I am using the following tools and environment, feel free to use what works for you.

Windows 10

Android Studio

Sony Xperia ZL

Min SDK 14

Target SDK 23

To create a new android application project, follow the steps as stipulated below.

Go to File menu

Click on New menu

Click on Android Application

Enter Project name: AndroidMapDrawRoute

Package: com.inducesmile.androidmapdrawroute

Select Map Activity

Name your activity: MapsActivity

Keep other default selections

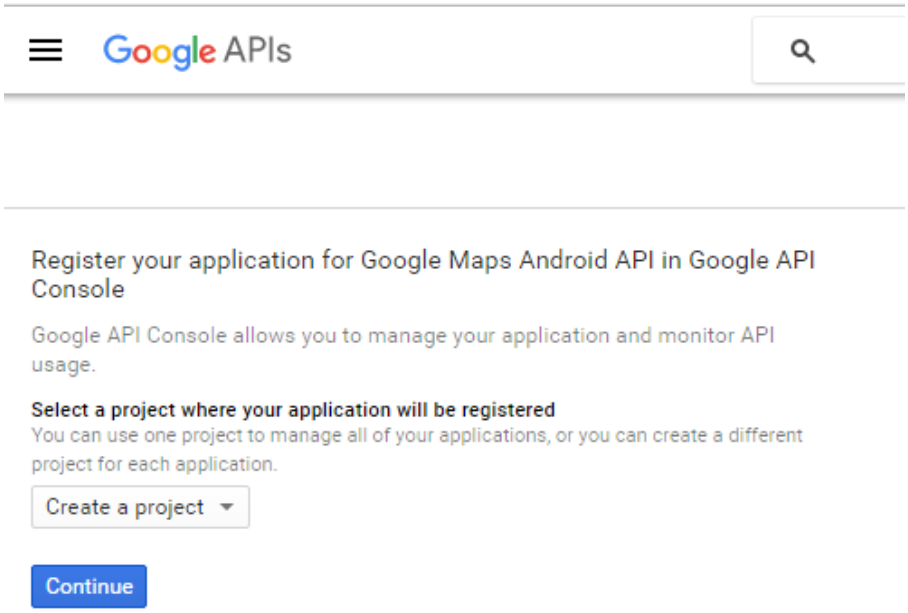
Continue to click on next button until Finish button is active, then click on Finish Button.

Since we selected the default android Map Activity template, Android Studio will add an xml file name ***google_maps_api.xml.*** This file is stored in the values folder inside the res folder of your android project.

When you open the file, it contains instruction on how to obtain a Google Map Key. Every request your application send to Google Map Server requires a unique key that will be used to identify your application.

Also, there is a limit to the number of request you can send in a day if your are using the free service. Your google map kep also helps Google to keep track of the number of request coming from your app.

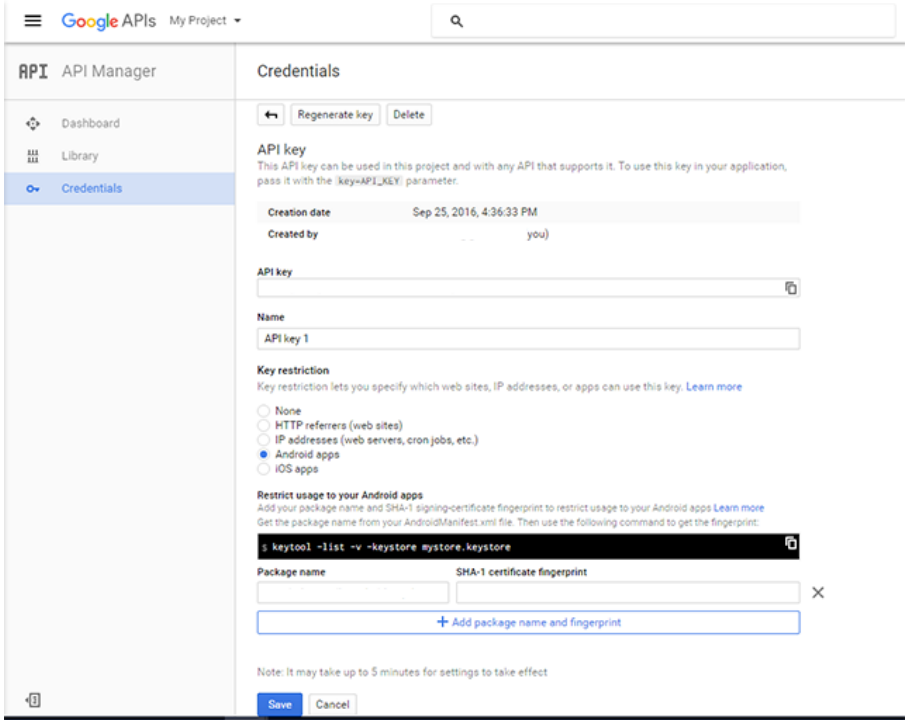
Copy the text for creating a Google Map Key and paste it in a web browser and hit enter. You will see a page like this.



You can create a new project or use an existing project. Click the continue button to proceed.

Click the Create AOI Key button that will appear in the next page to move over to Google API Manager page.

In the Google API Manager page, click on credentials and the key link to open the page.



Add a name for your key.

Select Android apps to restrict all the request from android apps

Add the your application package name

You can generate a SHA-1 certificate fingerprint. You will find the process on the page.

Click the Save button when you are done.

Finally, copy your application Google map key to the generated ***google_maps_api.xml*** as shown.

```
apply plugin: 'com.android.application'
```

```
<resources>
    <!--
    TODO: Before you run your application, you need a Go
    To get one, follow this link, follow the directions
    https://console.developers.google.com/flows/enableap
    You can also add your credentials to an existing key
    57:A8:DD:42:FF:9E:88:E6:EC:69:71:B7:0D:2C::46;com.in
    Alternatively, follow the directions here:
    https://developers.google.com/maps/documentation/and
    Once you have your key (it starts with "AIza"), repl
```

```
    string in this file.
    -->
    <string name="google_maps_key" templateMergeStrategy
</resources>
```

BUILD.GRADLE

In android, since we are going to make use of user location in drawing path between two points in Google Map API, we are going to use Google Play Services. Android Location Service API is part of Google Play Services.

Since the library is too big and to avoid going beyond 64000 methods which will force use to multiDexEnabled true in the defaultConfig, we will use Location and Map libraries alone.

Inaddition to these libraries, we are going to make use of Volley network library and Gson library.

Open your application build.gradle and add the code below.

```
apply plugin: 'com.android.application'
android {
    compileSdkVersion 24
    buildToolsVersion "24.0.1"
    defaultConfig {
        applicationId "com.inducesmile.androidmapdrawrou
        minSdkVersion 14
        targetSdkVersion 24
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('progua
        }
    }
}
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
```

```
    compile 'com.android.support:appcompat-v7:24.2.1'
    compile 'com.google.android.gms:play-services-maps:9
    compile 'com.google.android.gms:play-services-locati
    compile 'com.google.code.gson:gson:2.6.1'
    compile 'com.mcxiaoke.volley:library:1.0.19'
}
```

ANDROIDMANIFEST.XML

We are going to update our application androidmanifest.xml. Using Android Location requires that our application must request for user permission before it can access their location. Starting from android 6, location request are run time permission which the user will grant or deny while using the app.

Open your AndroidManifest.xml file and add the code below.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/
    package="com.inducesmile.androidmapdrawroute">
    <uses-permission android:name="android.permission.AC
    <uses-permission android:name="android.permission.AC
    <uses-permission android:name="android.permission.IN
    <application
        android:name=".network.CustomApplication"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <meta-data
            android:name="com.google.android.geo.API_KEY
            android:value="@string/google_maps_key" />
        <activity
            android:name=".MapsActivity"
            android:label="@string/title_activity_maps">
            <intent-filter>
                <action android:name="android.intent.act
                <category android:name="android.intent.c
            </intent-filter>
        </activity>
```

```
    </application>
</manifest>
```

The meta-data retrieves the Google Map Key that we obtained before.

STRINGS.XML

We are going to update our project strings.xml file located in the values folder inside the res folder. Open the file and add the code below to it.

```
<resources>
    <string name="app_name">AndroidMapDrawRoute</string>
    <string name="title_activity_maps">Map</string>
    <string name="permission_refused">Permission was not
    <string name="permission_request_title">Special Perm
    <string name="app_permission_notice">This permission
    <string name="failed_draw">Empty line path return</s
    <string name="server_error">Server error. Something
</resources
```

COLORS.XML

Open the colors.xml file in the same location as the strings.xml file and add the code below to the file.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

ANDROID GOOGLE DIRECTION API

This android tutorial on how to draw path between two points on Google Map API v2 is going to make use of android google direction

api.

Android Google direction API is a service that calculates directions between locations using an HTTP request.

So we are going to get the user current location as the origin and we will implement onMapClickListener which will mark the user destination.

Thereafter, we will use the Android Google Direction API to calculate the direction between the two locations. The Android Polyline is used to draw an overlay line between the two directions.

Lets move on to create a layout interface file

ACTIVITY_MAPS.XML

The Map Activity uses Fragment with android:name="com.google.android.gms.maps.SupportMapFragment".

The complete code for the layout file is as shown.

```
<fragment xmlns:android="http://schemas.android.com/apk/
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMap
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.inducesmile.androidmapdrawroute.M
```

MAPSACTIVITY CLASS

In the MapsActivity class, the Android Google Map is setup my Android Studio. We need to add some interfaces that the class will implement.

GoogleMap.OnMapClickListener – it listens to Map event click

GoogleApiClient.ConnectionCallbacks – Connection callback for GoogleApiClient

In the onConnected() callback method, we will get the user current location and save it in a List object.

When a user click on a destination location on the Map, it will get the location and store it on a List object.

We check if there is more that one location object then we will clear the list and add the present destination location in it.

When the two locations are obtained, we will use Volley to make a request to Google Direction API.

The complete code for the MapsActivity class is as shown.

```
import android.Manifest;
import android.content.DialogInterface;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.location.Location;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.FragmentActivity;
import android.support.v7.app.AlertDialog;
import android.util.Log;
import android.widget.Toast;
import com.android.volley.DefaultRetryPolicy;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.google.android.gms.common.api.GoogleApiClient
import com.google.android.gms.common.api.PendingResult;
import com.google.android.gms.common.api.ResultCallback;
import com.google.android.gms.common.api.Status;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.location.LocationSettingsR
import com.google.android.gms.location.LocationSettingsR
import com.google.android.gms.location.LocationSettingsS
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
```

```
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.CameraPosition;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.Polyline;
import com.google.android.gms.maps.model.PolylineOptions
import com.inducesmile.androidmapdrawroute.entityObjects
import com.inducesmile.androidmapdrawroute.entityObjects
import com.inducesmile.androidmapdrawroute.entityObjects
import com.inducesmile.androidmapdrawroute.entityObjects
import com.inducesmile.androidmapdrawroute.entityObjects
import com.inducesmile.androidmapdrawroute.network.GsonR
import com.inducesmile.androidmapdrawroute.network.Volle
import java.util.ArrayList;
import java.util.List;
public class MapsActivity extends FragmentActivity imple
    private static final String TAG = MapsActivity.class
    private GoogleApiClient mGoogleApiClient;
    private Location mLastLocation;
    private LocationRequest mLocationRequest;
    private double latitudeValue = 0.0;
    private double longitudeValue = 0.0;
    private GoogleMap mMap;
    private static final int PERMISSION_LOCATION_REQUEST
    private List<LatLng> latLngList;
    private MarkerOptions yourLocationMarker;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        latLngList = new ArrayList<LatLng>();
        if (mGoogleApiClient == null) {
            mGoogleApiClient = new GoogleApiClient.Build
                    .addConnectionCallbacks(this)
                    .addApi(LocationServices.API)
                    .build();
        }
        mLocationRequest = createLocationRequest();
        SupportMapFragment mapFragment = (SupportMapFrag
        mapFragment.getMapAsync(this);
    }
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        mMap.setOnMapClickListener(this);
```

```
    }
    @Override
    public void onMapClick(LatLng latLng) {
        if(latLngList.size() > 0){
            refreshMap(mMap);
            latLngList.clear();
        }
        latLngList.add(latLng);
        Log.d(TAG, "Marker number " + latLngList.size())
        mMap.addMarker(yourLocationMarker);
        mMap.addMarker(new MarkerOptions().position(latL
        LatLng defaultLocation = yourLocationMarker.getP
        LatLng destinationLocation = latLng;
        //use Google Direction API to get the route betw
        String directionApiPath = Helper.getUrl(String.v
                String.valueOf(destinationLocation.latit
        Log.d(TAG, "Path " + directionApiPath);
        getDirectionFromDirectionApiServer(directionApiP
    }
    @Override
    public void onConnected(@Nullable Bundle bundle) {
        LocationSettingsRequest.Builder builder = new Lo
        PendingResult<LocationSettingsResult> result = L
        result.setResultCallback(new ResultCallback<Loca
            @Override
            public void onResult(@NonNull LocationSettin
                final Status status = result.getStatus()
                switch (status.getStatusCode()) {
                    case LocationSettingsStatusCodes.SUC
                        Log.d(TAG, "Connection method ha
                        if (ActivityCompat.checkSelfPerm
                                && ActivityCompat.checkS
                            mLastLocation = LocationServ
                            assignLocationValues(mLastLo
                            setDefaultMarkerOption(new L
                        }else{
                            ActivityCompat.requestPermis
                        }
                        break;
                    case LocationSettingsStatusCodes.SET
                        break;
                }
            }
        });
    }
    @Override
```

```
    public void onConnectionSuspended(int i) {
    }
    @Override
    public void onRequestPermissionsResult(int requestCo
        switch (requestCode) {
            case PERMISSION_LOCATION_REQUEST_CODE: {
                // If request is cancelled, the result a
                if (grantResults[0] == PackageManager.PE
                    // permission was denied, show alert
                    showPermissionAlert();
                }else{
                    //permission is granted now start a
                    if (ActivityCompat.checkSelfPermissi
                            && ActivityCompat.checkSelfP
                        mLastLocation = LocationServices
                        assignLocationValues(mLastLocati
                        setDefaultMarkerOption(new LatLn
                    }
                }
            }
        }
    }
    private void assignLocationValues(Location currentLo
        if (currentLocation != null) {
            latitudeValue = currentLocation.getLatitude(
            longitudeValue = currentLocation.getLongitud
            Log.d(TAG, "Latitude: " + latitudeValue + "
            markStartingLocationOnMap(mMap, new LatLng(l
            addCameraToMap(new LatLng(latitudeValue, lon
        }
    }
    private void addCameraToMap(LatLng latLng){
        CameraPosition cameraPosition = new CameraPositi
                .target(latLng)
                .zoom(16)
                .build();
        mMap.animateCamera(CameraUpdateFactory.newCamera
    }
    private void showPermissionAlert(){
        AlertDialog.Builder builder = new AlertDialog.Bu
        builder.setTitle(R.string.permission_request_tit
        builder.setMessage(R.string.app_permission_notic
        builder.create();
        builder.setPositiveButton("OK", new DialogInterf
            @Override
            public void onClick(DialogInterface dialog,
```

```
                if (ActivityCompat.checkSelfPermission(M
                        && ActivityCompat.checkSelfPermi
                    ActivityCompat.requestPermissions(Ma
                }
            }
        });
        builder.setNegativeButton("Cancel", new DialogIn
            @Override
            public void onClick(DialogInterface dialog,
                Toast.makeText(MapsActivity.this, R.stri
            }
        });
        builder.show();
    }
    private void markStartingLocationOnMap(GoogleMap map
        mapObject.addMarker(new MarkerOptions().position
        mapObject.moveCamera(CameraUpdateFactory.newLatL
    }
    private void refreshMap(GoogleMap mapInstance){
        mapInstance.clear();
    }
    protected LocationRequest createLocationRequest() {
        LocationRequest mLocationRequest = new LocationR
        mLocationRequest.setInterval(5000);
        mLocationRequest.setFastestInterval(3000);
        mLocationRequest.setPriority(LocationRequest.PRI
        return mLocationRequest;
    }
    private void setDefaultMarkerOption(LatLng location)
        if(yourLocationMarker == null){
            yourLocationMarker = new MarkerOptions();
        }
        yourLocationMarker.position(location);
    }
    @Override
    protected void onStart() {
        mGoogleApiClient.connect();
        super.onStart();
    }
    @Override
    protected void onStop() {
        mGoogleApiClient.disconnect();
        super.onStop();
    }
    private void getDirectionFromDirectionApiServer(Stri
        GsonRequest<DirectionObject> serverRequest = new
```

```
                Request.Method.GET,
                url,
                DirectionObject.class,
                createRequestSuccessListener(),
                createRequestErrorListener());
        serverRequest.setRetryPolicy(new DefaultRetryPol
                Helper.MY_SOCKET_TIMEOUT_MS,
                DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
                DefaultRetryPolicy.DEFAULT_BACKOFF_MULT)
        VolleySingleton.getInstance(getApplicationContex
    }
    private Response.Listener<DirectionObject> createReq
        return new Response.Listener<DirectionObject>()
            @Override
            public void onResponse(DirectionObject respo
                try {
                    Log.d("JSON Response", response.toSt
                    if(response.getStatus().equals("OK")
                        List<LatLng> mDirections = getDi
                        drawRouteOnMap(mMap, mDirections
                    }else{
                        Toast.makeText(MapsActivity.this
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }
            };
        };
    }
    private List<LatLng> getDirectionPolylines(List<Rout
        List<LatLng> directionList = new ArrayList<LatLn
        for(RouteObject route : routes){
            List<LegsObject> legs = route.getLegs();
            for(LegsObject leg : legs){
                List<StepsObject> steps = leg.getSteps()
                for(StepsObject step : steps){
                    PolylineObject polyline = step.getPo
                    String points = polyline.getPoints()
                    List<LatLng> singlePolyline = decode
                    for (LatLng direction : singlePolyli
                        directionList.add(direction);
                    }
                }
            }
        }
        return directionList;
```

```
    }
    private Response.ErrorListener createRequestErrorLis
        return new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError erro
                error.printStackTrace();
            }
        };
    }
    private void drawRouteOnMap(GoogleMap map, List<LatL
        PolylineOptions options = new PolylineOptions().
        options.addAll(positions);
        Polyline polyline = map.addPolyline(options);
        CameraPosition cameraPosition = new CameraPositi
                .target(new LatLng(positions.get(1).lati
                .zoom(17)
                .build();
        map.animateCamera(CameraUpdateFactory.newCameraP
    }
    /**
     * Method to decode polyline points
     * Courtesy : http://jeffreysambells.com/2010/05/27/
     * */
    private List<LatLng> decodePoly(String encoded) {
        List<LatLng> poly = new ArrayList<>();
        int index = 0, len = encoded.length();
        int lat = 0, lng = 0;
        while (index < len) {
            int b, shift = 0, result = 0;
            do {
                b = encoded.charAt(index++) - 63;
                result |= (b & 0x1f) << shift;
                shift += 5;
            } while (b >= 0x20);
            int dlat = ((result & 1) != 0 ? ~(result >>
            lat += dlat;
            shift = 0;
            result = 0;
            do {
                b = encoded.charAt(index++) - 63;
                result |= (b & 0x1f) << shift;
                shift += 5;
            } while (b >= 0x20);
            int dlng = ((result & 1) != 0 ? ~(result >>
            lng += dlng;
            LatLng p = new LatLng((((double) lat / 1E5))
```

```
                    (((double) lng / 1E5)));
            poly.add(p);
        }
        return poly;
    }
}
```

RETURNED JSON RESPONSE OBJECT FROM GOOGLE DIRECTION API

The returned response object is a Json object and the direction path is stored in the Polyline class. This is the class structure of the Json response object. We are going to use the Gson library to convert it to plain Java object class.

```
{
   "geocoded_waypoints" : [
      {
         "geocoder_status" : "OK",
         "place_id" : "ChIJZymwbofzU0YRvgugqrVwH8Q",
         "types" : [ "street_address" ]
      },
      {
         "geocoder_status" : "OK",
         "place_id" : "EiJOeWdhdGFuIDMyLCAyNDIgMzEgSMO2c
         "types" : [ "street_address" ]
      }
   ],
   "routes" : [
      {
         "bounds" : {
            "northeast" : {
               "lat" : 55.8541564,
               "lng" : 13.661235
            },
            "southwest" : {
               "lat" : 55.85187149999999,
               "lng" : 13.660381
            }
         },
         "copyrights" : "Map data ©2016 Google",
```

```
"legs" : [
   {
      "distance" : {
         "text" : "0.3 km",
         "value" : 260
      },
      "duration" : {
         "text" : "1 min",
         "value" : 84
      },
      "end_address" : "Nygatan 32, 242 31 Hörby
      "end_location" : {
         "lat" : 55.85187149999999,
         "lng" : 13.660381
      },
      "start_address" : "Nygatan 12B, 242 31 Hö
      "start_location" : {
         "lat" : 55.8541564,
         "lng" : 13.661235
      },
      "steps" : [
         {
            "distance" : {
               "text" : "0.3 km",
               "value" : 260
            },
            "duration" : {
               "text" : "1 min",
               "value" : 84
            },
            "end_location" : {
               "lat" : 55.85187149999999,
               "lng" : 13.660381
            },
            "html_instructions" : "Head \u003cb
            "polyline" : {
               "points" : "o_|sIwekrAVHxBj@|Bh@
            },
            "start_location" : {
               "lat" : 55.8541564,
               "lng" : 13.661235
            },
            "travel_mode" : "DRIVING"
         }
      ],
      "traffic_speed_entry" : [],
```

```
               "via_waypoint" : []
            }
         ],
         "overview_polyline" : {
            "points" : "o_|sIwekrApCt@|Bh@nBr@hAV"
         },
         "summary" : "Nygatan",
         "warnings" : [],
         "waypoint_order" : []
      }
   ],
   "status" : "OK"
}
```

ENTITYOBJECT CLASSES

We will create the following classes to mimic the structure of the Json response object. The classes are

DIRECTIONOBJECT.JAVA

```
import java.util.List;
public class DirectionObject {
    private List<RouteObject> routes;
    private String status;
    public DirectionObject(List<RouteObject> routes, Str
        this.routes = routes;
        this.status = status;
    }
    public List<RouteObject> getRoutes() {
        return routes;
    }
    public String getStatus() {
        return status;
    }
}
```

LEGSOBJECT.JAVA

```
import java.util.List;
public class LegsObject {
    private List<StepsObject> steps;
    public LegsObject(List<StepsObject> steps) {
        this.steps = steps;
    }
    public List<StepsObject> getSteps() {
        return steps;
    }
}
```

POLYLINEOBJECT.JAVA

```
public class PolylineObject {
    private String points;
    public PolylineObject(String points) {
        this.points = points;
    }
    public String getPoints() {
        return points;
    }
}
```

ROUTEOBJECT.JAVA

```
import java.util.List;
public class RouteObject {
    private List<LegsObject> legs;
    public RouteObject(List<LegsObject> legs) {
        this.legs = legs;
    }
    public List<LegsObject> getLegs() {
        return legs;
    }
}
```

STEPSOBJECTS.JAVA

```
public class StepsObject {
    private PolylineObject polyline;
    public StepsObject(PolylineObject polyline) {
        this.polyline = polyline;
    }
    public PolylineObject getPolyline() {
        return polyline;
    }
}
```

USING ANDROID VOLLEY FOR NETWORK CALL

We are going to extends the Application class. Create a java class and name it CustomApplication.java.

We will create an instance of our Volley object in the class. By using a custom application class, we can access the Volley object anywhere in our application.

Add the following code to the class

```
import android.app.Application;
import com.android.volley.RequestQueue;
public class CustomApplication extends Application{
    private RequestQueue requestQueue;
    @Override
    public void onCreate() {
        super.onCreate();
        requestQueue = VolleySingleton.getInstance(getAp
    }
    public RequestQueue getVolleyRequestQueue(){
        return requestQueue;
    }
}
```

GSONREQUEST.JAVA

Create a java file and name it GsonRequest.java. Open the file and add the code below to it.

```
import com.android.volley.AuthFailureError;
import com.android.volley.NetworkResponse;
import com.android.volley.ParseError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.toolbox.HttpHeaderParser;
import com.google.gson.Gson;
import com.google.gson.JsonSyntaxException;
import java.io.UnsupportedEncodingException;
import java.util.Map;
public class GsonRequest<T> extends Request<T> {
    // create variables
    private Gson mGson = new Gson();
    private Class<T> tClass;
    private Map<String, String> headers;
    private Map<String, String> params;
    private Response.Listener<T> listener;
    public GsonRequest(int method, String url, Class<T>
        super(method, url, errorListener);
        this.tClass = tClass;
        this.listener = listener;
        mGson = new Gson();
    }
    public GsonRequest(int method, String url, Class<T>
        super(method, url, errorListener);
        this.tClass = tClass;
        this.params = params;
        this.listener = listener;
        this.headers = null;
        mGson = new Gson();
    }
    @Override
    public Map<String, String> getHeaders() throws AuthF
        return headers != null ? headers : super.getHead
    }
    @Override
    protected Map<String, String> getParams() throws Aut
        return params;
    }
    protected void deliverResponse(T response) {
```

```
        listener.onResponse(response);
    }
    @Override
    protected Response<T> parseNetworkResponse(NetworkRe
        try {
            String json = new String(response.data, Http
            return Response.success(mGson.fromJson(json,
        } catch (UnsupportedEncodingException e) {
            return Response.error(new ParseError(e));
        } catch (JsonSyntaxException e) {
            return Response.error(new ParseError(e));
        }
    }
}
```

VOLLEYSINGLETON.JAVA

Create a new Java class and name it VolleySingleton.java. Open the file and add the code below to it.

```
import android.content.Context;
import android.graphics.Bitmap;
import android.util.LruCache;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.Volley;
public class VolleySingleton {
    private static VolleySingleton mInstance;
    private RequestQueue mRequestQueue;
    private ImageLoader mImageLoader;
    private static Context mCtx;
    private VolleySingleton(Context context) {
        mCtx = context;
        mRequestQueue = getRequestQueue();
        mImageLoader = new ImageLoader(mRequestQueue, ne
            private final LruCache<String, Bitmap> cache
            @Override
            public Bitmap getBitmap(String url) {
                return cache.get(url);
            }
            @Override
```

```
            public void putBitmap(String url, Bitmap bit
                cache.put(url, bitmap);
            }
        });
    }
    public static synchronized VolleySingleton getInstan
        if (mInstance == null) {
            mInstance = new VolleySingleton(context);
        }
        return mInstance;
    }
    public RequestQueue getRequestQueue() {
        if (mRequestQueue == null) {
            mRequestQueue = Volley.newRequestQueue(mCtx.
        }
        return mRequestQueue;
    }
    public <T> void addToRequestQueue(Request<T> req) {
        getRequestQueue().add(req);
    }
    public ImageLoader getImageLoader() {
        return mImageLoader;
    }
}
```

HELPER.JAVA

We will create a new Java file and we will name it Helper.java. Open the file and add the code below.

```
import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
public class Helper {
    private static final String DIRECTION_API = "https:/
    public static final String API_KEY = "AIzaSyCuZCfoPP
    public static final int MY_SOCKET_TIMEOUT_MS = 5000;
    public static String getUrl(String originLat, String
        return Helper.DIRECTION_API + originLat+","+orig
    }
    public static boolean isNetworkAvailable(Context con
        ConnectivityManager connectivityManager = (Conne
```

```
        NetworkInfo activeNetworkInfo = connectivityMana
        return activeNetworkInfo != null && activeNetwor
    }
}
```

This brings us to the end of this tutorial. I hope that you have learn something. Run your app and see for yourself.

You can download the code for this tutorial below. If you are having hard time downloading the tutorial, kindly contact me.

Remember to subscribe with your email address to be among the first to receive my new android blog post once it is published.

OTHER INTERESTING POSTS:

Play YouTube video in Android using YouTube player API

Android RecyclerView with Material Design

Android Fragment Master/Detail Flow Tutorial in Android Studio

Android Light Sensor - Environment Sensors

Android Video Streamimg with Android VideoView and MediaController

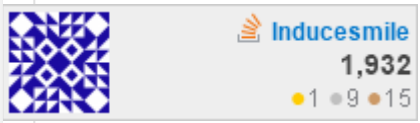
Android Sugar ORM Database

Tags: Android - How to draw path between 2 points on Google Map, Draw Route on Map in Android, Google Maps Draw Route between two points using Google, How to draw road directions between two geocodes in android google

ABOUT THE AUTHOR

Inducesmile

I learn and write about Android, iOS, Javascript, Php, Node.js, React Native, Mobile Game, Virtual Reality and Internet of Things



7 COMMENTS

rharj | May 18, 2017 Log in to Reply

The download link is missing

daffiew | September 13, 2017 Log in to Reply

code plz getting error

Rohit Singh | September 22, 2017 Log in to Reply

hi sir
i am getting one problem like whenever i use your API_KEY it successfully run but when i use my API_KEY it shows :——

{
“error_message” : “This IP, site or mobile application is not authorized to use this API key. Request received from IP address 146.196.37.156, with empty referer”,
“routes” : [],
“status” : “REQUEST_DENIED”
}

please tell me how to resolve it ??

Rohit Singh | September 22, 2017 Log in to Reply

one more question

The API_key you have mentioned in helper.java and google_map_api.xml both are same or different ??
because whenever i am using your key i am not getting any error in json ??

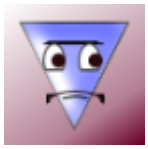
Dinu | January 8, 2018 Log in to Reply

Hi,

Please share a download link for this tutorial

hassam33$ | January 20, 2018 Log in to Reply

link to download

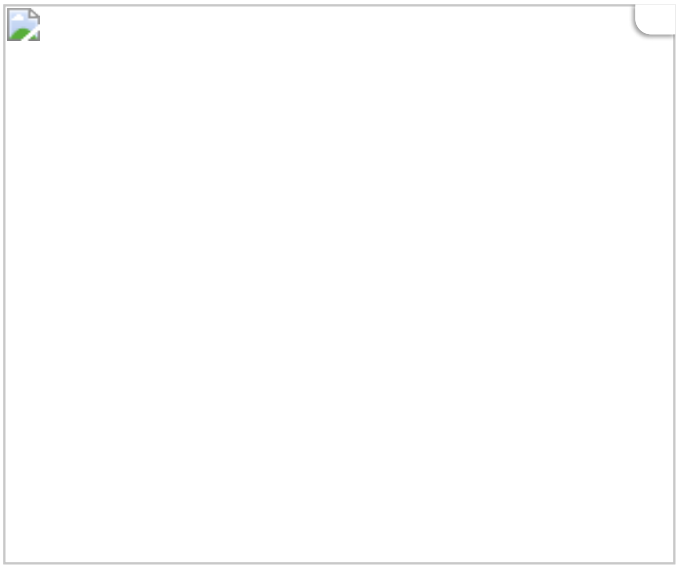


saikrr | January 30, 2018 Log in to Reply

error:
DistanceObject,DurationObject .
please share the implementation code.

ADD A COMMENT

You must be logged in to post a comment.



INDUCESMILE – ANDROID TUTORIAL, ANDROID APPS, ANDROID STUDIO, ANDROID SDK, ANDROID DEVELOPMENT COPYRIGHT © 2018.