ANDROID FIND DISTANCE AND DURATION BETWEEN TWO POINTS ON GOOGLE MAP

In this tutorial, we are going to learn how to find the distance and duration between two points on Google Map API v2. There are situations were you might want your app to measure the distance and duration between two points on a Map, this can give an insight of how long the distance is and the time it will take to accomplish the journey.

Major part of this android tutorial on finding distance and duration between two points on a Map is adopted from my tutorial on **How to draw path between two points on a Map.** If you have not read the tutorial, I will suggest you read it first.

1. We are going to implement a Map

2. Select two points on the Map we want find their distance and duration

3. We draw a path between the two points and calculate the distance and duration between them

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 SCREENSHOT

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

Appointment Booking App (10%, 15 Votes)

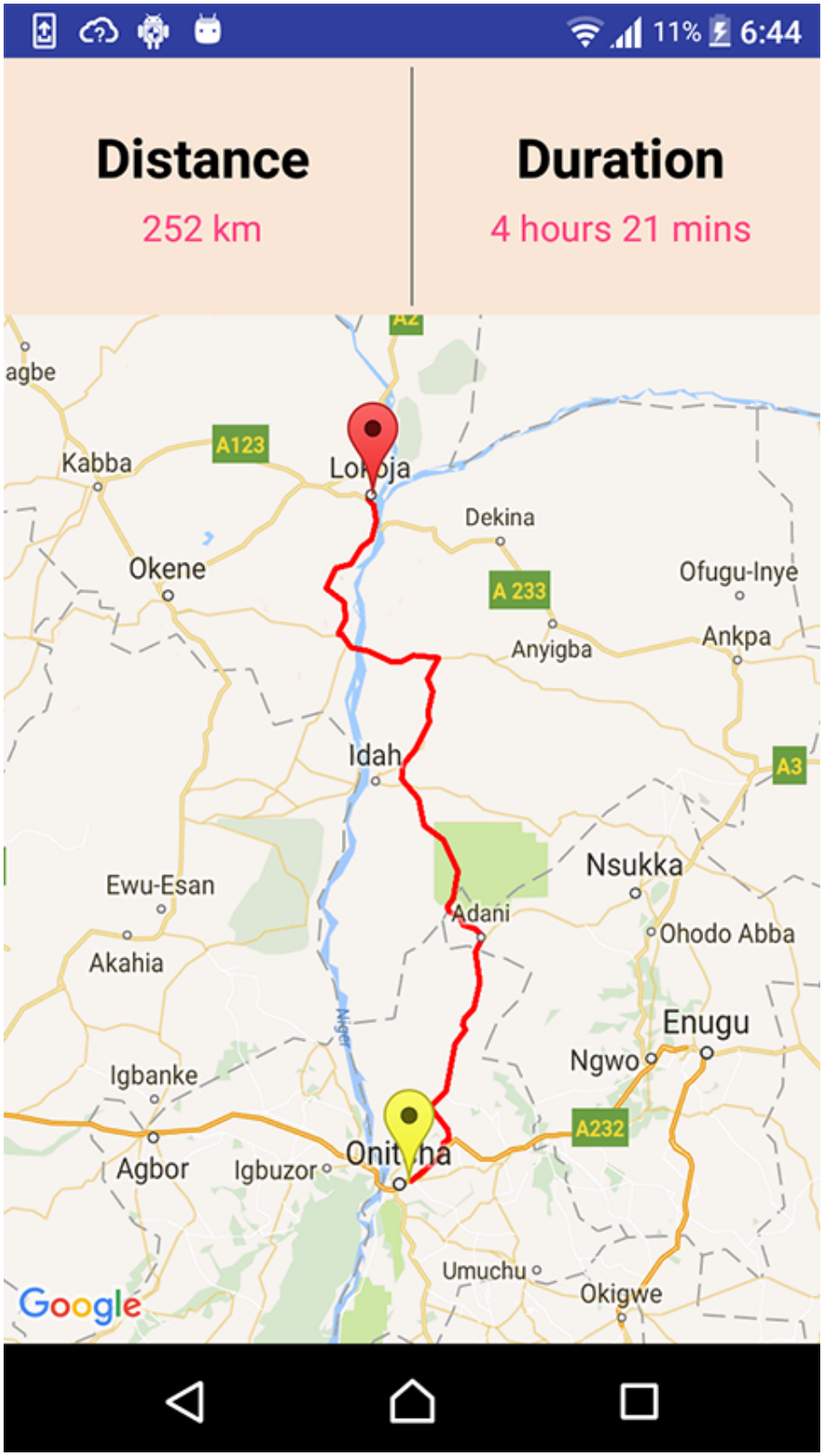
Business & Service Provider Finder (6%, 9 Votes)

Total Voters: **147**

LIKE US ON FACEBOOK

RECENT POSTS

- Camera Translator All Language App
- Why MVVM and How to execute MVVM combined with Data Binding and LiveData in Four Simple Steps
- Camera Translator All Languages – Android Project idea source code for students
- Note Scan Android App -Project idea source code for students
- How To Build Android WhatsApp Clone Chat Application
- Android Car Rental App with Web Admin Panel – Project Idea Source Code For Students



- 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

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: AndroidMapRouteDistance

Package: com.inducesmile.androidmaproutedistance

Select Map Activity

Name your activity: MapsActivity

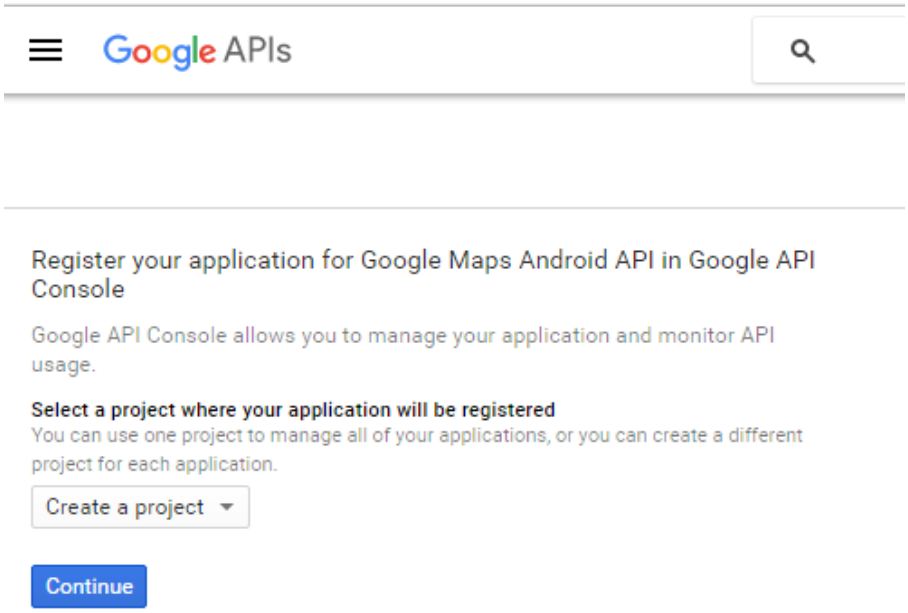
Keep other default selections

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.

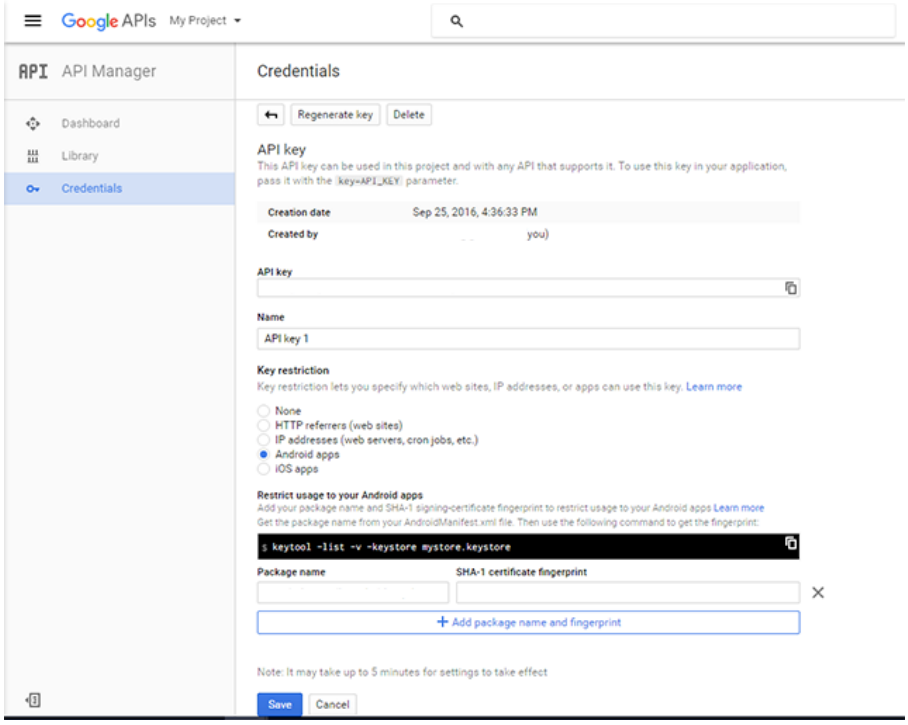
Copy for creating a Google Map Key and paste it in a web browser and hit enter. You will see a page like this.



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
```

```
    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
        }
```

```
    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. We are going to add some user permissions. 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.androidmaproutedistance">

    <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
```

```
</manifest>
```

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">AndroidMapRouteDistance</str
    <string name="title_activity_maps">Map</string>
    <string name="route_distance">Distance</string>
    <string name="route_distance_value">40.2 Km</string>
    <string name="route_duration">Duration</string>
    <string name="route_duration_value">48 mins</string>
    <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>
    <color name="colorPrimaryText">#000000</color>
    <color name="colorSecondaryText">#737373</color>
    <color name="colorIcons">#FFFFFF</color>
    <color name="colorDivider">#737373</color>
    <color name="colorMap">#FAE6D6</color>
</resources>
```

ANDROID GOOGLE DIRECTION API

This android tutorial on how to find distance and duration 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.

```
<LinearLayout xmlns:android="http://schemas.android.com/
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_width="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:orientation="horizontal"
        android:background="@color/colorMap"
```

```
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:gravity="center"
        android:orientation="vertical">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textStyle="bold"
            android:textSize="24sp"
            android:textColor="@color/colorPrima
            android:text="@string/route_distance

        <TextView
            android:id="@+id/distance_value"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="16sp"
            android:layout_marginTop="4dp"
            android:textColor="@color/colorAccen
            android:text="@string/route_distance

    </LinearLayout>

    <View
        android:layout_width="1dp"
        android:layout_height="match_parent"
        android:background="@color/colorDivider"
        android:layout_margin="4dp"/>

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:gravity="center"
        android:orientation="vertical">
```

```
                android:textSize="24sp"
                android:textColor="@color/colorPrima
                android:text="@string/route_duration

            <TextView
                android:id="@+id/duration_value"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="16sp"
                android:layout_marginTop="4dp"
                android:textColor="@color/colorAccen
                android:text="@string/route_duration

        </LinearLayout>

    </LinearLayout>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:orientation="vertical"
    android:layout_weight="8">

    <fragment
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.Su
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context="com.inducesmile.androidmaprou

</LinearLayout>

</LinearLayout>
```

MAPSACTIVITY CLASS

In the MapsActivity class, the Android Google Map is setup my Android

When the Map is load, the click event listener is attached to the Map View. Users can select a location of interest on the Map by clicking on the location. Once clicked a marker will appear there to let you know the location you have selected.

The selected location is added to a List Object, once a user has selected two different location, we will draw a path to between the two location points and calculate the distance and duration between the location.

Open the MapsActivity class and add the code below to the class.

```
import android.content.pm.ActivityInfo;
import android.graphics.Color;
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
import android.util.Log;
import android.widget.TextView;
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.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.BitmapDescripto
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.androidmaproutedistance.entityObj
import com.inducesmile.androidmaproutedistance.entityObj
import com.inducesmile.androidmaproutedistance.entityObj
import com.inducesmile.androidmaproutedistance.entityObj
import com.inducesmile.androidmaproutedistance.entityObj
import com.inducesmile.androidmaproutedistance.network.G
import com.inducesmile.androidmaproutedistance.network.V
import java.util.ArrayList;
import java.util.List;
```

```
    private TextView distanceValue;
    private TextView durationValue;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIE
        distanceValue = (TextView)findViewById(R.id.dist
        durationValue = (TextView)findViewById(R.id.dura
        latLngList = new ArrayList<LatLng>();
        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() > 1){
            refreshMap(mMap);
            latLngList.clear();
            distanceValue.setText("");
            durationValue.setText("");
        }
        latLngList.add(latLng);
        Log.d(TAG, "Marker number " + latLngList.size())
        createMarker(latLng, latLngList.size());
        if(latLngList.size() == 2){
            LatLng origin = latLngList.get(0);
            LatLng destination = latLngList.get(1);
            //use Google Direction API to get the route
            String directionApiPath = Helper.getUrl(Stri
                    String.valueOf(destination.latitude)
            Log.d(TAG, "Path " + directionApiPath);
            getDirectionFromDirectionApiServer(direction
        }
    }
    private void refreshMap(GoogleMap mapInstance){
        mapInstance.clear();
    }
```

```
        }else{
            mOptions.icon(BitmapDescriptorFactory.defaul
        }
        addCameraToMap(latLng);
        mMap.addMarker(mOptions);
    }
    private void addCameraToMap(LatLng latLng){
        CameraPosition cameraPosition = new CameraPositi
                .target(latLng)
                .zoom(8)
                .build();
        mMap.animateCamera(CameraUpdateFactory.newCamera
    }
    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();
                }
            };
```

Accept

```
        durationValue.setText(duration);
    }
    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){
                String routeDistance = leg.getDistance()
                String routeDuration = leg.getDuration()
                setRouteDistanceAndDuration(routeDistanc
                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);
    }
    /**
     * Method to decode polyline points
     * Courtesy : http://jeffreysambells.com/2010/05/27/
     * */
    private List<LatLng> decodePoly(String encoded) {
        List<LatLng> poly = new ArrayList<>();
```

Accept

```
            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",
```

```
      "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
```


Accept

```
                    },
                    "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

Accept

```
    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;

 private DistanceObject distance;

 private DurationObject duration;

 public LegsObject(DurationObject duration, DistanceObje
 this.duration = duration;
 this.distance = distance;
 this.steps = steps;
 }

 public List<StepsObject> getSteps() {
 return steps;
 }

 public DistanceObject getDistance() {
 return distance;
```

```
  }
}
```

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;
```

```
        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.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 {
```

```
        } 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);
            }
        });
    }
```

```
        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
    }
}
```

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:

Android Code Competition Announcement

Embed and Play YouTube Video in Android WebView

Android Fake Call Application Tutorial

Android Creating Tab Layout with TabHost and TabWigdet or with FragmentTabHost and Fragment

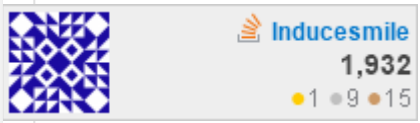
Android EasyFont Library Example

Android Room with LiveData Example Tutorial

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



13 COMMENTS

elmahdi | October 14, 2016 Log in to Reply

hi henry,
String routeDistance = leg.getDistance().getText();
String routeDuration = leg.getDuration().getText();
getDistance() and getDuration() aren't defined on LegsObject.

any advice?!

Henry Author | October 14, 2016
Log in to Reply

Wow, thanks for pointing it out. I think I mistakenly used a class from another project.

The class is pasted below and I have updated the source code

Accept

```
private DistanceObject distance;
private DurationObject duration;
public LegsObject(DurationObject duration, DistanceObject distance, List steps) {
this.duration = duration;
this.distance = distance;
this.steps = steps;
}
public List getSteps() {
return steps;
}
public DistanceObject getDistance() {
return distance;
}
public DurationObject getDuration() {
return duration;
}
}
```

elmahdi | October 28, 2016

Log in to Reply

hi Henry,

there is no implementation of :
1) DistanceObject class
2) DurationObject class

can you plz put the code.

jo_177 | January 21, 2018

Log in to Reply

please add the codes for

1) DistanceObject class

Accept

Brian | October 17, 2016 Log in to Reply

Hi Henry,
The update seems not to work. Are there two other classes missing or the methods are note clearly defined.

Rafelcf | October 19, 2016 Log in to Reply

The perfect place to learn and use enough for my application would fragments code, since it is based on many json files. You can download the code ?. thanks Henry

Rafelcf | October 20, 2016 Log in to Reply

Sorry, too complicated for me, GSON for me is new, only I pretended to renew my app but I find it very difficult. Thank you

Brian | November 11, 2016 Log in to Reply

Can you assist with the source code?

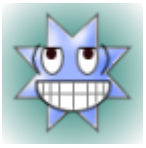
Sophie | September 11, 2017 Log in to Reply

Hello Henry,

Nice. Thanks for sharing.

Please have a look at this issue on SO:
https://stackoverflow.com/q/46103680/3585072

Accept



selo | September 13, 2017 Log in to Reply

hi Henry,
there is no implementation of :
1) DistanceObject class
2) DurationObject class
can you plz put the code

Naveena | October 26, 2017 Log in to Reply

Can you pls share the link for the source code?

saikrr | January 30, 2018 Log in to Reply

please provide implementation of : DistanceObject class, DurationObject class

kainat | March 22, 2018 Log in to Reply

hi your tutorial is very useful
i am beginner and i am working on map can u help me i want to store current latitude and longitude in database using volley library and then want to retrieve that latitude and longitude and display on google map plz help me its urgent thanks in advance

ADD A COMMENT

You must be logged in to post a comment.

Accept