How to draw path as I move starting from my current location using Google Maps

I am trying to draw route as I move from my current location. I am facing a big problem in drawing route dynamically please help me to solve it. I am having marker at my current location in my map. As soon as I start moving I want the map to start drawing lines in the path I move. I do not have two fixed points. Can any one please provide me a solution to over come this. I have seen lot of answers in SO which draws path between two fixed points. But here only my initial point is fixed. I am able to get my current location in my app currently. I tried with the following code but getLocationManager() is resulting in error. i am using Android Studio.

Updated Code:

My Activity:

```
import android.content.Context;
import android.content.SharedPreferences;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
import android.util.Log;
import android.util.Xml;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesUtil;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.PendingResult;
import com.google.android.gms.common.api.Status;
import com.google.android.gms.location.LocationListener;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.maps.android.ui.IconGenerator;

import org.xmlpull.v1.XmlSerializer;

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.StringWriter;
import java.text.DateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import java.util.Locale;

public class MainActivity extends FragmentActivity implements
        LocationListener,
        GoogleApiClient.ConnectionCallbacks,
        GoogleApiClient.OnConnectionFailedListener {
```

```
    private static final String TAG = "MainActivity";
    private static final long INTERVAL = 1000 * 60 * 1; //1 minute
    private static final long FASTEST_INTERVAL = 1000 * 60 * 1; // 1 minute
    private LocationRequest mLocationRequest;
    private GoogleApiClient mGoogleApiClient;
    private Location mCurrentLocation;
    private String mLastUpdateTime;
    private String city = "";
    private String country = "";
    private String area = "";
    private String title;
    private String requiredArea = "";
    private GoogleMap googleMap;
    private List<Address> addresses;


    protected void createLocationRequest() {
        mLocationRequest = new LocationRequest();
        mLocationRequest.setInterval(INTERVAL);
        mLocationRequest.setFastestInterval(FASTEST_INTERVAL);
        mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.d(TAG, "onCreate ............................");
        //show error dialog if GoolglePlayServices not available
        if (!isGooglePlayServicesAvailable()) {

            Toast.makeText(this, "Google Play Services is not available",
Toast.LENGTH_LONG).show();

            finish();
        }
        createLocationRequest();
        mGoogleApiClient = new GoogleApiClient.Builder(this)
                .addApi(LocationServices.API)
                .addConnectionCallbacks(this)
                .addOnConnectionFailedListener(this)
                .build();

        setContentView(R.layout.activity_main);
        SupportMapFragment fm = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);

        googleMap = fm.getMap();
        googleMap.setMyLocationEnabled(true);
        googleMap.setOnMyLocationButtonClickListener(new
GoogleMap.OnMyLocationButtonClickListener() {
            @Override
            public boolean onMyLocationButtonClick() {

                Toast.makeText(getApplicationContext(), "Location button has been
clicked", Toast.LENGTH_LONG).show();
                return true;
            }
        });
        googleMap.getUiSettings().setZoomControlsEnabled(true);
        googleMap.getUiSettings().setAllGesturesEnabled(true);


    }

    @Override
    public void onStart() {
        super.onStart();
        Log.d(TAG, "onStart fired ..............");
        mGoogleApiClient.connect();
    }

    @Override
    public void onStop() {
        super.onStop();
        Log.d(TAG, "onStop fired ..............");
        mGoogleApiClient.disconnect();
        Log.d(TAG, "isConnected ...............: " + mGoogleApiClient.isConnected());
    }

    private boolean isGooglePlayServicesAvailable() {
        int status = GooglePlayServicesUtil.isGooglePlayServicesAvailable(this);
        if (ConnectionResult.SUCCESS == status) {
            return true;
        } else {
            GooglePlayServicesUtil.getErrorDialog(status, this, 0).show();
            Toast.makeText(getApplicationContext(), "Google Play Services is not
Available", Toast.LENGTH_LONG).show();
            return false;
        }
    }

    @Override
    public void onConnected(Bundle bundle) {
        Log.d(TAG, "onConnected - isConnected ...............: " +
```

```
mGoogleApiClient.isConnected());
        startLocationUpdates();
    }

    protected void startLocationUpdates() {
        PendingResult<Status> pendingResult =
LocationServices.FusedLocationApi.requestLocationUpdates(
                mGoogleApiClient, mLocationRequest, this);
        Log.d(TAG, "Location update started .............: ");
    }

    @Override
    public void onConnectionSuspended(int i) {

    }

    @Override
    public void onConnectionFailed(ConnectionResult connectionResult) {
        Log.d(TAG, "Connection failed: " + connectionResult.toString());
    }

    @Override
    public void onLocationChanged(Location location) {
        Log.d(TAG, "Firing
onLocationChanged..........................................");
        mCurrentLocation = location;
        mLastUpdateTime = DateFormat.getTimeInstance().format(new Date());
        addMarker();
        float accuracy = location.getAccuracy();
        Log.d("iFocus", "The amount of accuracy is " + accuracy);
        double latitude = location.getLatitude();
        double longitude = location.getLongitude();
        Bundle extras = location.getExtras();
        Boolean has = location.hasAccuracy();
        String provider = location.getProvider();
        Long time = location.getTime();

//        Location locationB = new Location("Begur");
//        double lati = 12.8723;
//        double longi =  77.6329;
//        locationB.setLatitude(lati);
//        locationB.setLongitude(longi);
//        Float distance = location.distanceTo(locationB);

        Calendar calendar = Calendar.getInstance();
        calendar.setTimeInMillis(time);

        int mYear = calendar.get(Calendar.YEAR);
        int mMonth = calendar.get(Calendar.MONTH) + 1;
        int mDay = calendar.get(Calendar.DAY_OF_MONTH);

        String formattedTime = mDay + ":" + mMonth + ":" + mYear;
        Log.d("iFocus", "The name of provider is " + provider);
        Log.d("iFocus", "The value of has is " + has);
        Log.d("iFocus", "The value of extras is " + extras);
        Log.d("iFocus", "The value of Month is " + mMonth);
        Log.d("iFocus", "The value of Day is " + mDay);
        Log.d("iFocus", "The value of Year is " + mYear);
        Log.d("iFocus", "The value of Time is " + formattedTime);
        //Log.d("iFocus", "The value of distance is "+distance);

        LatLng latLng = new LatLng(latitude, longitude);

        Geocoder geocoder = new Geocoder(this, Locale.getDefault());

        try {
            addresses = geocoder.getFromLocation(latitude, longitude, 1);
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        String cityName = addresses.get(0).getAddressLine(0);
        String stateName = addresses.get(0).getAddressLine(1);
        String countryName = addresses.get(0).getAddressLine(2);

        String[] splittedStateName = stateName.split(",");
        requiredArea = splittedStateName[2];
        Log.d("iFocus", "The value of required area is " + requiredArea);

        city = addresses.get(0).getLocality();
        area = addresses.get(0).getSubLocality();
        String adminArea = addresses.get(0).getAdminArea();
        String premises = addresses.get(0).getPremises();
        String subAdminArea = addresses.get(0).getSubAdminArea();
        String featureName = addresses.get(0).getFeatureName();
        String phone = addresses.get(0).getPhone();
        country = addresses.get(0).getCountryName();
        Log.d("iFocus", "The name of city is " + city);
        Log.d("iFocus", "The name of area is " + area);
        Log.d("iFocus", "The name of country is " + country);
        Log.d("iFocus", "The value of cityName is " + cityName);
        Log.d("iFocus", "The value of StateName is " + stateName);
        Log.d("iFocus", "The value of CountryName is " + countryName);
```

```
        Toast.makeText(this, cityName + " " + stateName + " " + countryName,
Toast.LENGTH_LONG).show();

        SharedPreferences sharedPreferences = getSharedPreferences("MyValues",
MODE_PRIVATE);
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putString("CITY", cityName);
        editor.putString("STATE", stateName);
        editor.putString("COUNTRY", countryName);
        editor.commit();

        TextView mapTitle = (TextView) findViewById(R.id.textViewTitle);

        if (requiredArea != "" && city != "" && country != "") {
            title = mLastUpdateTime.concat(", " + requiredArea).concat(", " +
city).concat(", " + country);
        }
        else {
            title = mLastUpdateTime.concat(", " + area).concat(", " + city).concat(", " +
country);
        }
        mapTitle.setText(title);
        addMarker();// newly added

        final String xmlFile = "userData.xml";

        try {
            // FileOutputStream fos = new  FileOutputStream("userData.xml");
            FileOutputStream fos = openFileOutput(xmlFile, Context.MODE_PRIVATE);
            XmlSerializer xmlSerializer = Xml.newSerializer();
            StringWriter writer = new StringWriter();
            xmlSerializer.setOutput(writer);
            xmlSerializer.startDocument("UTF-8", true);
            xmlSerializer.startTag(null, "userData");
            xmlSerializer.startTag(null, "Time");
            xmlSerializer.text(mLastUpdateTime);
            xmlSerializer.endTag(null, "Time");
            xmlSerializer.startTag(null, "Area");
            if (requiredArea != "") {
                xmlSerializer.text(requiredArea);
            }
            else {
                xmlSerializer.text(area);
            }
            xmlSerializer.endTag(null, "Area");
            xmlSerializer.startTag(null, "City");
            xmlSerializer.text(city);
            xmlSerializer.endTag(null, "City");
            xmlSerializer.endTag(null, "userData");
            xmlSerializer.endDocument();
            xmlSerializer.flush();
            String dataWrite = writer.toString();
            fos.write(dataWrite.getBytes());
            fos.close();
        } catch (FileNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IllegalArgumentException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IllegalStateException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        String dir = getFilesDir().getAbsolutePath();
        Log.d("Pana", "The value of Dir is "+dir);

    }

    private void addMarker() {
        MarkerOptions options = new MarkerOptions();

        // following four lines requires 'Google Maps Android API Utility Library'
        // https://developers.google.com/maps/documentation/android/utility/
        // I have used this to display the time as title for location markers
        // you can safely comment the following four lines but for this info
        IconGenerator iconFactory = new IconGenerator(this);
        iconFactory.setStyle(IconGenerator.STYLE_PURPLE);
        //
options.icon(BitmapDescriptorFactory.fromBitmap(iconFactory.makeIcon(mLastUpdateTime +
requiredArea + city)));
        options.icon(BitmapDescriptorFactory.fromBitmap(iconFactory.makeIcon(requiredArea
+ ", " + city)));
        options.anchor(iconFactory.getAnchorU(), iconFactory.getAnchorV());
        LatLng currentLatLng = new LatLng(mCurrentLocation.getLatitude(),
mCurrentLocation.getLongitude());
        options.position(currentLatLng);
        Marker mapMarker = googleMap.addMarker(options);
        long atTime = mCurrentLocation.getTime();
        mLastUpdateTime = DateFormat.getTimeInstance().format(new Date(atTime));
```

```
        String title = mLastUpdateTime.concat(", " + requiredArea).concat(", " +
city).concat(", " + country);
        mapMarker.setTitle(title);


        TextView mapTitle = (TextView) findViewById(R.id.textViewTitle);
        mapTitle.setText(title);

        Log.d(TAG, "Marker added.............................");
        googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(currentLatLng,
                13));
        Log.d(TAG, "Zoom done.............................");
    }

    @Override
    protected void onPause() {
        super.onPause();
        stopLocationUpdates();
    }

    protected void stopLocationUpdates() {
        LocationServices.FusedLocationApi.removeLocationUpdates(
                mGoogleApiClient, this);
        Log.d(TAG, "Location update stopped .......................");
    }

    @Override
    public void onResume() {
        super.onResume();
        if (mGoogleApiClient.isConnected()) {
            startLocationUpdates();
            Log.d(TAG, "Location update resumed .....................");
        }
    }
}
```

I am trying to add this method in my code to draw line but its giving error in getLocationManager();

```
       private void addLocationListener(LocationListener locationListener) {
        LocationProvider locationProvider =
getLocationManager().getProvider(LocationManager.GPS_PROVIDER);

        getLocationManager().requestLocationUpdates(locationProvider.getName(),
LOCATION_UPDATE_INTERVAL,
                LOCATION_UPDATE_MIN_DISTANCE, locationListener);
    }

    private LocationManager getLocationManager() {
        return (LocationManager) context.getSystemService(Context.LOCATION_SERVICE);
    }

    private void startGpsListening(Location start) {
       this.startLocation = start;
       addLocationListener(new MyLocationListener());
    }

    private Location startLocation = new Location("");

    private class MyLocationListener extends LocationListener {

        public void onLocationChanged(Location location) {

        }
        ...
    }
```

android google-maps geolocation location locationlistener

edited May 15 '15 at 4:21

asked May 15 '15 at 1:04

Keshav1234
1,052 2 6 29

Welcome to Stack Overflow! Unlike forum sites, we don't use "Thanks", or "Any help appreciated", or signatures on Stack Overflow. See "Should 'Hi', 'thanks,' taglines, and salutations be removed from posts?. – John Saunders May 15 '15 at 1:35

Ok sir. Can you please help with the solution if you know. – Keshav1234 May 15 '15 at 1:38

If I knew the solution and if I wanted to help you, then I would do that. – John Saunders May 15 '15 at 1:40

@DanielNugent: I did not get answer for that. i am struggling with this problem from two weeks. So I posted the question again.. Since that question was old. Sorry for my mistake. – Keshav1234 May 15 '15 at 2:52

@DanielNugent it says cannot be resolved into a variable. – Keshav1234 May 15 '15 at 4:02

1 Answer

It seems that the best implementation would be to just use an `ArrayList<LatLng>` to store each point given in `onLocationChanged()`. Then, each time you get a new point, re-draw the line.

First, import what you need for drawing the lines:

```
import com.google.android.gms.maps.model.Polyline;
import com.google.android.gms.maps.model.PolylineOptions;
```

Create member variables for the ArrayList and the Polyline:

```
private ArrayList<LatLng> points; //added
Polyline line; //added
```

Initialize `points` in `onCreate()`:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    points = new ArrayList<LatLng>(); //added
    //...............
```

Then, in `onLocationChanged()`, add each point you get to the ArrayList:

```
@Override
public void onLocationChanged(Location location) {
      double latitude = location.getLatitude();
      double longitude = location.getLongitude();
      LatLng latLng = new LatLng(latitude, longitude); //you already have this

      points.add(latLng); //added

      redrawLine(); //added

}
```

Taking from this answer, define your `redrawLine()` method.
Remove all other calls to `addMarker()`, since you will be calling `clear()` on your map, which removes all Markers and Polylines.

```
private void redrawLine(){

    googleMap.clear();  //clears all Markers and Polylines

    PolylineOptions options = new
PolylineOptions().width(5).color(Color.BLUE).geodesic(true);
    for (int i = 0; i < points.size(); i++) {
        LatLng point = points.get(i);
        options.add(point);
    }
    addMarker(); //add Marker in current position
    line = googleMap.addPolyline(options); //add Polyline
}
```

Edit: You will also probably want to dial in the minimum distance in meters between location changed callbacks.

```
private static final String TAG = "MainActivity";
private static final long INTERVAL = 1000 * 60 * 1; //1 minute
private static final long FASTEST_INTERVAL = 1000 * 60 * 1; // 1 minute
private static final float SMALLEST_DISPLACEMENT = 0.25F; //quarter of a meter
```

Call `setSmallestDisplacement()`:

```
protected void createLocationRequest() {
    mLocationRequest = new LocationRequest();
    mLocationRequest.setInterval(INTERVAL);
    mLocationRequest.setFastestInterval(FASTEST_INTERVAL);
    mLocationRequest.setSmallestDisplacement(SMALLEST_DISPLACEMENT); //added
    mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
}
```

That should be enough to get you started. You may need to fine-tune the frequency of location changed callbacks to get your desired result. There's probably more to it than that, but you can find the edge cases and fix them after testing.

edited yesterday

answered May 15 '15 at 5:13

Daniel Nugent
22.7k 6 45 76

Thanks a lot for this beautiful answer. I was searching all day for an answer. – Keshav1234 May 15 '15 at 6:35

I am unable to draw path for two fixed points can you help me? – Keshav1234 May 18 '15 at 5:00

@Daniel Nugent Hi Daniel, can you please see this question and can you please let me know where am I going wrong: stackoverflow.com/questions/34120829/… – Keshav1234 Dec 7 '15 at 9:22

@Daniel I am seriously struck please help me. – Keshav1234 Dec 7 '15 at 9:23