**package** com.example.navya.googlemapapi;  
  
 **import** android.Manifest;  
 **import** android.app.ProgressDialog;  
 **import** android.content.pm.PackageManager;  
 **import** android.graphics.Color;  
 **import** android.support.v4.app.ActivityCompat;  
 **import** android.support.v4.app.FragmentActivity;  
 **import** android.os.Bundle;  
 **import** android.view.View;  
 **import** android.widget.Button;  
 **import** android.widget.EditText;  
 **import** android.widget.TextView;  
 **import** android.widget.Toast;  
  
 **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.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.android.gms.maps.model.Polyline;  
 **import** com.google.android.gms.maps.model.PolylineOptions;  
  
 **import** java.io.UnsupportedEncodingException;  
 **import** java.util.ArrayList;  
 **import** java.util.List;  
  
 **import** com.example.navya.googlemapapi.DirectionFinderListener;  
 **import** com.example.navya.googlemapapi.Route;  
  
**public class** MapsActivity **extends** FragmentActivity **implements** OnMapReadyCallback, DirectionFinderListener {  
  
 **private** GoogleMap **mMap**;  
 **private** Button **btnFindPath**;  
 **private** EditText **etOrigin**;  
 **private** EditText **etDestination**;  
 **private** List<Marker> **originMarkers** = **new** ArrayList<>();  
 **private** List<Marker> **destinationMarkers** = **new** ArrayList<>();  
 **private** List<Polyline> **polylinePaths** = **new** ArrayList<>();  
 **private** ProgressDialog **progressDialog**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_maps***);  
 *// Obtain the SupportMapFragment and get notified when the map is ready to be used.* SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()  
 .findFragmentById(R.id.***map***);  
 mapFragment.getMapAsync(**this**);  
  
 **btnFindPath** = (Button) findViewById(R.id.***btnFindPath***);  
 **etOrigin** = (EditText) findViewById(R.id.***etOrigin***);  
 **etDestination** = (EditText) findViewById(R.id.***etDestination***);  
  
 **btnFindPath**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 sendRequest();  
 }  
 });  
 }  
  
 **private void** sendRequest() {  
 String origin = **etOrigin**.getText().toString();  
 String destination = **etDestination**.getText().toString();  
 **if** (origin.isEmpty()) {  
 Toast.*makeText*(**this**, **"Please enter origin address!"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 **if** (destination.isEmpty()) {  
 Toast.*makeText*(**this**, **"Please enter destination address!"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
  
 **try** {  
 **new** DirectionFinder(**this**, origin, destination).execute();  
 } **catch** (UnsupportedEncodingException e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 **public void** onMapReady(GoogleMap googleMap) {  
 **mMap** = googleMap;  
 LatLng hcmus = **new** LatLng(10.762963, 106.682394);  
 **mMap**.moveCamera(CameraUpdateFactory.*newLatLngZoom*(hcmus, 18));  
 **originMarkers**.add(**mMap**.addMarker(**new** MarkerOptions()  
 .title(**"Đại học Khoa học tự nhiên"**)  
 .position(hcmus)));  
  
 **if** (ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {  
 *//* ***TODO: Consider calling*** *// ActivityCompat#requestPermissions  
 // here to request the missing permissions, and then overriding  
 // public void onRequestPermissionsResult(int requestCode, String[] permissions,  
 // int[] grantResults)  
 // to handle the case where the user grants the permission. See the documentation  
 // for ActivityCompat#requestPermissions for more details.* **return**;  
 }  
 **mMap**.setMyLocationEnabled(**true**);  
 }  
  
  
 @Override  
 **public void** onDirectionFinderStart() {  
 **progressDialog** = ProgressDialog.*show*(**this**, **"Please wait."**,  
 **"Finding direction..!"**, **true**);  
  
 **if** (**originMarkers** != **null**) {  
 **for** (Marker marker : **originMarkers**) {  
 marker.remove();  
 }  
 }  
  
 **if** (**destinationMarkers** != **null**) {  
 **for** (Marker marker : **destinationMarkers**) {  
 marker.remove();  
 }  
 }  
  
 **if** (**polylinePaths** != **null**) {  
 **for** (Polyline polyline:**polylinePaths** ) {  
 polyline.remove();  
 }  
 }  
 }  
  
 @Override  
 **public void** onDirectionFinderSuccess(List<Route> routes) {  
 **progressDialog**.dismiss();  
 **polylinePaths** = **new** ArrayList<>();  
 **originMarkers** = **new** ArrayList<>();  
 **destinationMarkers** = **new** ArrayList<>();  
  
 **for** (Route route : routes) {  
 **mMap**.moveCamera(CameraUpdateFactory.*newLatLngZoom*(route.**startLocation**, 16));  
 ((TextView) findViewById(R.id.***tvDuration***)).setText(route.**duration**.**text**);  
 ((TextView) findViewById(R.id.***tvDistance***)).setText(route.**distance**.**text**);  
  
 **originMarkers**.add(**mMap**.addMarker(**new** MarkerOptions()  
 .icon(BitmapDescriptorFactory.*fromResource*(R.drawable.***start\_blue***))  
 .title(route.**startAddress**)  
 .position(route.**startLocation**)));  
 **destinationMarkers**.add(**mMap**.addMarker(**new** MarkerOptions()  
 .icon(BitmapDescriptorFactory.*fromResource*(R.drawable.***end\_green***))  
 .title(route.**endAddress**)  
 .position(route.**endLocation**)));  
  
 PolylineOptions polylineOptions = **new** PolylineOptions().  
 geodesic(**true**).  
 color(Color.***BLUE***).  
 width(10);  
  
 **for** (**int** i = 0; i < route.**points**.size(); i++)  
 polylineOptions.add(route.**points**.get(i));  
  
 **polylinePaths**.add(**mMap**.addPolyline(polylineOptions));  
 }  
 }  
}