package com.example.directionsparsing;

import java.io.BufferedReader;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.List;

import org.apache.http.HttpEntity;

import org.apache.http.HttpResponse;

import org.apache.http.client.HttpClient;

import org.apache.http.client.methods.HttpPost;

import org.apache.http.impl.client.DefaultHttpClient;

import org.json.JSONArray;

import org.json.JSONObject;

import android.app.Activity;

import android.os.Bundle;

import android.view.Menu;

import android.widget.TextView;

import com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.MapFragment;

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;

public class MainActivity extends Activity {

private static final String TAG\_LAT="lat";

private static final String TAG\_LONG="lng";

private static final String TAG\_ROUTES="routes";

private static final String TAG\_LEGS="legs";

private static final String TAG\_STEPS="steps";

private static final String TAG\_END="end\_location";

List<LatLng> Loc = new ArrayList<LatLng>();

GoogleMap map;

JSONArray steps;

TextView tv;

Thread th;

boolean flag=false;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

tv =(TextView)findViewById(R.id.tv);

map = ((MapFragment) getFragmentManager().findFragmentById(R.id.map)).getMap();

th = new Thread(new Runnable() {

@Override

public void run() {

// TODO Auto-generated method stub

parse();

}

});

th.start();

//th2.start();

try{

th.join();

//Draw\_PolyLine(Loc);

}

catch(Exception e){

System.err.println(e);

}

System.out.println("Finished Parsing :D");

}

private void parse()

{

try{

tv.append("Processing...");

System.out.println("Processing...");

HttpClient client = new DefaultHttpClient();

HttpPost post = new HttpPost("http://maps.googleapis.com/maps/api/directions/json?origin=Toronto&destination=Montreal&sensor=false&avoid=highways&mode=driving");

HttpResponse response = client.execute(post);

HttpEntity entity = response.getEntity();

InputStream is = entity.getContent();

BufferedReader reader = new BufferedReader(new InputStreamReader(is, "UTF-8"), 8);

StringBuilder sb = new StringBuilder();

String line = null;

String result;

while ((line = reader.readLine()) != null)

{

sb.append(line + "\n");

}

result = sb.toString();

//System.out.println("JSON" + result);

JSONObject json = new JSONObject(result);

JSONArray rt = json.getJSONArray(TAG\_ROUTES);

int i=0,j=0,k=0;

JSONArray st=null,leg=null;

for( i = 0; i < rt.length(); i++){

JSONObject c = rt.getJSONObject(i);

leg = c.getJSONArray(TAG\_LEGS);

for( j=0;j<leg.length();j++){

JSONObject ob = leg.getJSONObject(j);

st = ob.getJSONArray(TAG\_STEPS);

for(k=0;k<st.length();k++)

{

JSONObject loc = st.getJSONObject(k);

JSONObject end = loc.getJSONObject(TAG\_END);

final String lat = end.getString(TAG\_LAT);

final String longitude = end.getString(TAG\_LONG);

runOnUiThread(new Runnable() {

public void run() {

try{

Double d1 = Double.parseDouble(lat);

Double d2 = Double.parseDouble(longitude);

LatLng l1 = new LatLng(d1,d2);

Loc.add(l1);

System.out.println(lat+" "+longitude+"\n");

}

catch(Exception e){

System.err.println("From UIThread" + e);

}

}

});

}

}

}

if( k>=st.length())

{flag=true;

System.out.println("FLAG:" + flag);}

runOnUiThread(new Runnable() {

public void run() {

try{

Draw\_PolyLine(Loc);

}

catch(Exception e){

System.err.println("From UIThread 2" + e);

}

}

});

}

catch(Exception e){

System.err.println(e);

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.main, menu);

return true;

}

private void Draw\_PolyLine(List<LatLng> Loc)

{

map.setMapType(GoogleMap.MAP\_TYPE\_NORMAL);

System.out.println("Inside Polyline");

map.addMarker(new MarkerOptions()

.position(Loc.get(0))

.title("Origin"));

map.moveCamera(CameraUpdateFactory.newLatLngZoom(Loc.get(0), 10));

Polyline PL = map.addPolyline(new PolylineOptions()

.geodesic(false)

package com.example.directionsparsing;

import java.io.BufferedReader;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.List;

import org.apache.http.HttpEntity;

import org.apache.http.HttpResponse;

import org.apache.http.client.HttpClient;

import org.apache.http.client.methods.HttpPost;

import org.apache.http.impl.client.DefaultHttpClient;

import org.json.JSONArray;

import org.json.JSONObject;

import android.app.Activity;

import android.os.Bundle;

import android.view.Menu;

import android.widget.TextView;

import com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.MapFragment;

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;

public class MainActivity extends Activity {

private static final String TAG\_LAT="lat";

private static final String TAG\_LONG="lng";

private static final String TAG\_ROUTES="routes";

private static final String TAG\_LEGS="legs";

private static final String TAG\_STEPS="steps";

private static final String TAG\_END="end\_location";

List<LatLng> Loc = new ArrayList<LatLng>();

GoogleMap map;

JSONArray steps;

TextView tv;

Thread th;

boolean flag=false;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

tv =(TextView)findViewById(R.id.tv);

map = ((MapFragment) getFragmentManager().findFragmentById(R.id.map)).getMap();

th = new Thread(new Runnable() {

@Override

public void run() {

// TODO Auto-generated method stub

parse();

}

});

th.start();

//th2.start();

try{

th.join();

//Draw\_PolyLine(Loc);

}

catch(Exception e){

System.err.println(e);

}

System.out.println("Finished Parsing :D");

}

private void parse()

{

try{

tv.append("Processing...");

System.out.println("Processing...");

HttpClient client = new DefaultHttpClient();

HttpPost post = new HttpPost("http://maps.googleapis.com/maps/api/directions/json?origin=Toronto&destination=Montreal&sensor=false&avoid=highways&mode=driving");

HttpResponse response = client.execute(post);

HttpEntity entity = response.getEntity();

InputStream is = entity.getContent();

BufferedReader reader = new BufferedReader(new InputStreamReader(is, "UTF-8"), 8);

StringBuilder sb = new StringBuilder();

String line = null;

String result;

while ((line = reader.readLine()) != null)

{

sb.append(line + "\n");

}

result = sb.toString();

//System.out.println("JSON" + result);

JSONObject json = new JSONObject(result);

JSONArray rt = json.getJSONArray(TAG\_ROUTES);

int i=0,j=0,k=0;

JSONArray st=null,leg=null;

for( i = 0; i < rt.length(); i++){

JSONObject c = rt.getJSONObject(i);

leg = c.getJSONArray(TAG\_LEGS);

for( j=0;j<leg.length();j++){

JSONObject ob = leg.getJSONObject(j);

st = ob.getJSONArray(TAG\_STEPS);

for(k=0;k<st.length();k++)

{

JSONObject loc = st.getJSONObject(k);

JSONObject end = loc.getJSONObject(TAG\_END);

final String lat = end.getString(TAG\_LAT);

final String longitude = end.getString(TAG\_LONG);

runOnUiThread(new Runnable() {

public void run() {

try{

Double d1 = Double.parseDouble(lat);

Double d2 = Double.parseDouble(longitude);

LatLng l1 = new LatLng(d1,d2);

Loc.add(l1);

System.out.println(lat+" "+longitude+"\n");

}

catch(Exception e){

System.err.println("From UIThread" + e);

}

}

});

}

}

}

if( k>=st.length())

{flag=true;

System.out.println("FLAG:" + flag);}

runOnUiThread(new Runnable() {

public void run() {

try{

Draw\_PolyLine(Loc);

}

catch(Exception e){

System.err.println("From UIThread 2" + e);

}

}

});

}

catch(Exception e){

System.err.println(e);

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.main, menu);

return true;

}

private void Draw\_PolyLine(List<LatLng> Loc)

{

map.setMapType(GoogleMap.MAP\_TYPE\_NORMAL);

System.out.println("Inside Polyline");

map.addMarker(new MarkerOptions()

.position(Loc.get(0))

.title("Origin"));

CameraPosition cameraPosition = new CameraPosition.Builder()

.target(Loc.get(0)) // Sets the center of the map to Mountain View

.zoom(17) // Sets the zoom

.bearing(90) // Sets the orientation of the camera to east

.tilt(30) // Sets the tilt of the camera to 30 degrees

.build(); // Creates a CameraPosition from the builder

map.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPosition));

// map.moveCamera(CameraUpdateFactory.newLatLngZoom(Loc.get(0), 10));

Polyline PL = map.addPolyline(new PolylineOptions()

.geodesic(false)

);

PL.setPoints(Loc);

}

}

);

PL.setPoints(Loc);

}

}