**DoS (Denial of service)**

Description:

A denial of service (DoS) attack is an incident in which a user or organization is deprived of the services of a resource they would normally expect to have. In a distributed denial-of-service, large numbers of compromised systems (sometimes called a botnet) attack a single target.

The problem is that, Android have this problem in very command way, because number of automatic request that receiving by the server from different user’s app, like request for automatic check for new news in the server, or update user location in the server every one meter that user location change, imagine your app have 100,000 users, so there is to many automatic requests is going to the server every seconds, that need to many number of database instance, that lead to DoS because your server is not powerful server and have limit number of requests because the powerful server is cost to much and normal developer cannot buy it.

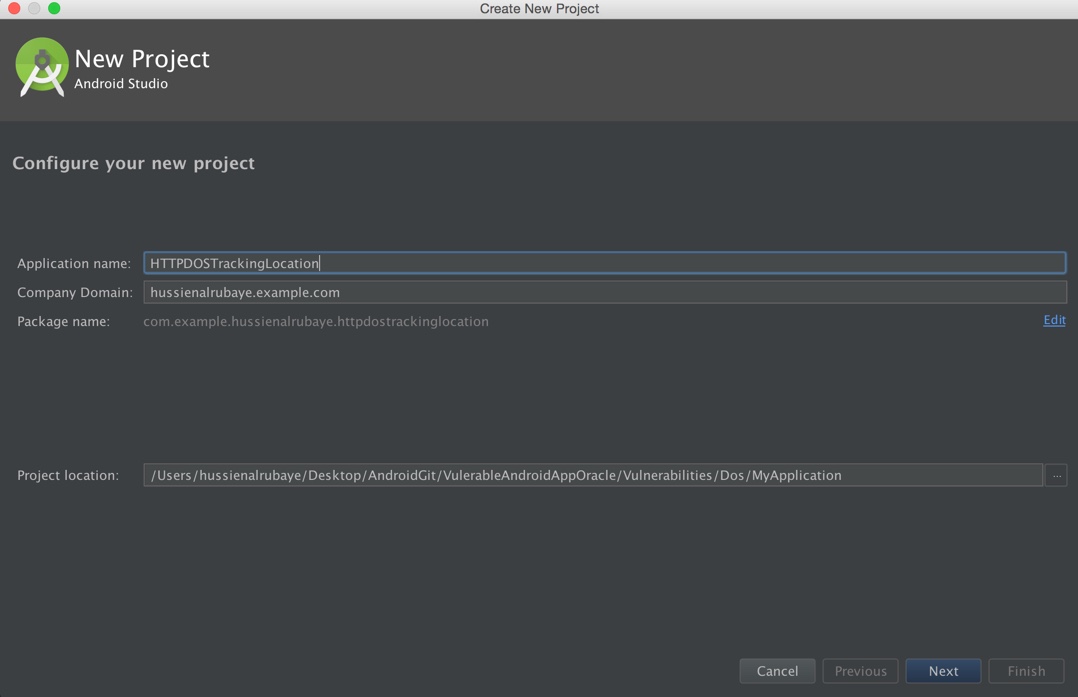
To fix this problem we have to manage number of request that send by user app to the server, and avoid many unnecessary requests, so instead sending check news every 10 sec make it 1 minute it reduce the traffic.

We will demonstrate example, showing app that tracking user location by send user location over HTTP to the server every one-meter change in user location, the problem happens if there is more than one person have the app and travel by car, so the change in user location is very fast and the request is to much this cause DoS in the server and the server could not receive user location. then we will fix the problem by manage the app requests

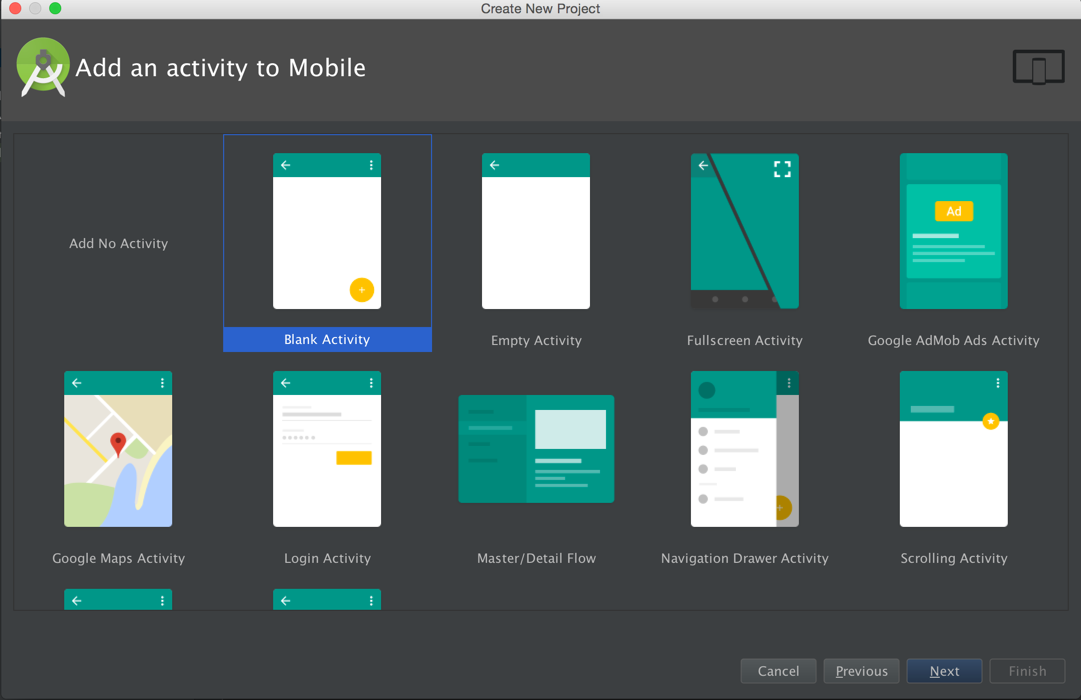
**Steps to work**

**Tracking Location App :** this app is track user location by send user location over HTTP to the server every one-meter change in user location.

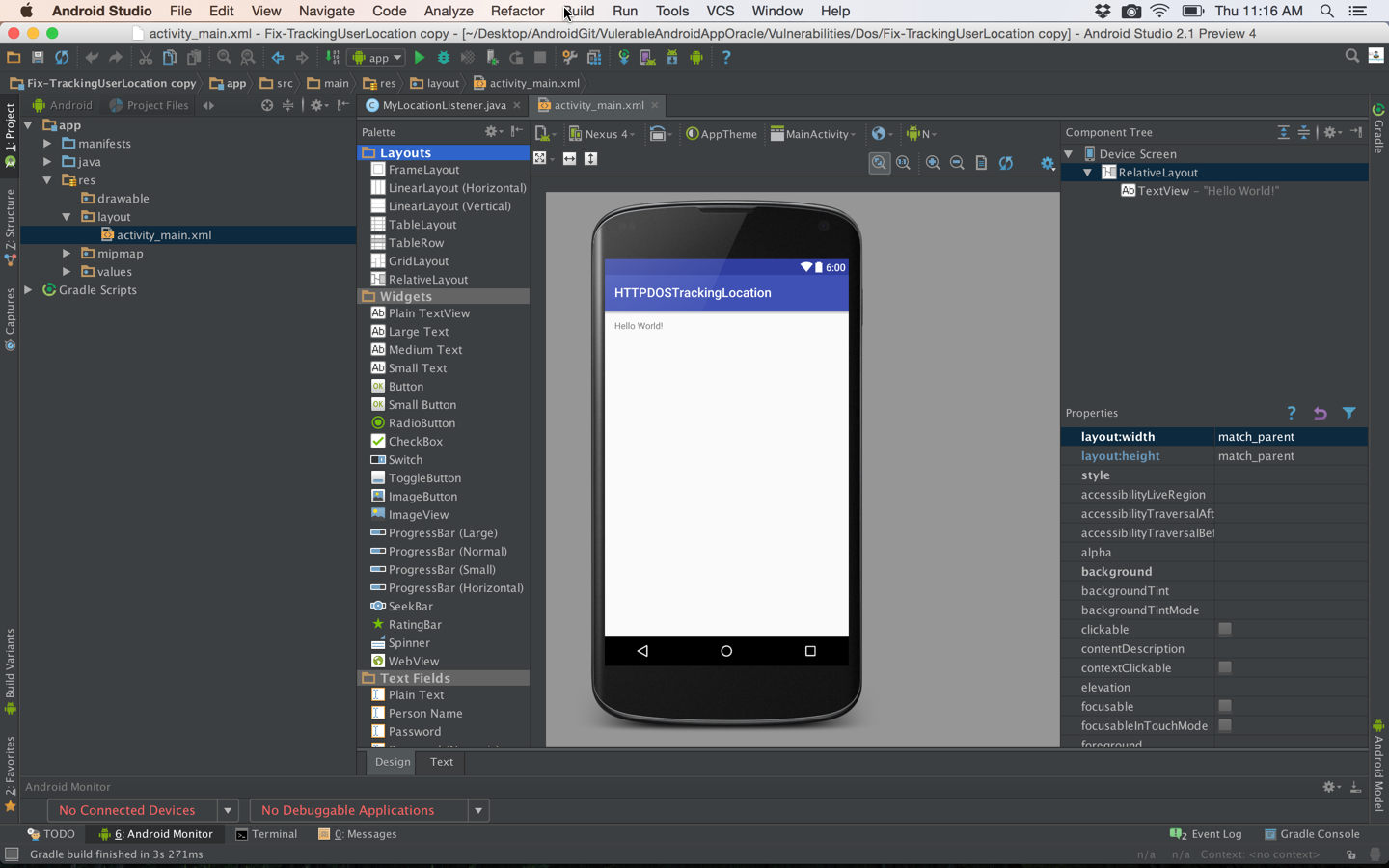
1-Create new project named “**HTTPDOSTrackingLocation**”, and the make sure to remember package name.



2- select project from type Blank Activity



3- Design the app to be like this



4- add class named “**MyLocationListener.java**”

|  |
| --- |
| Java |
| package com.example.hussienalrubaye.httpdostrackinglocation;  import android.content.Context; import android.location.Location; import android.location.LocationListener; import android.os.Bundle; import android.widget.Toast;  import java.io.BufferedInputStream; import java.io.IOException; import java.io.InputStream; import java.net.HttpURLConnection; import java.net.MalformedURLException; import java.net.URL; import java.text.SimpleDateFormat; import java.util.Date; import java.util.Locale;  import android.content.Context; import android.location.Location; import android.location.LocationListener; import android.os.Bundle; import android.widget.Toast;  import java.text.SimpleDateFormat; import java.util.Date; import java.util.Locale;  */\*\*  \* This service listen to change in user location every 1 meter  \* and send this data over http to server to save user location  \*/* public class MyLocationListener implements LocationListener {  // previous send date  Context context;  public static boolean *ISActive*=false;  String URL;  public MyLocationListener(Context context){  this.context =context ;  }  // this method calls every change in one meter in user location  public void onLocationChanged(Location location) {    //define the http url that recive location  URL = "http://news.alruabye.net/newfeeds.aspx?Latitude=" + Double.*toString*(location.getLatitude()) + "&Longitude=" + Double.*toString*(location.getLongitude());  Toast.*makeText*(context, URL, Toast.*LENGTH\_LONG*).show();   try {  //define url for send user location  URL url = new URL(URL);  //make http connection to send user location  HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();  urlConnection.setConnectTimeout(7000);//set timeout to 5 seconds  // InputStream in = new BufferedInputStream(urlConnection.getInputStream());    } catch (Exception e) {  e.printStackTrace();  } finally {  //urlConnection.disconnect();  }    }  public void onStatusChanged(String s, int i, Bundle b) {   }   public void onProviderDisabled(String s) {   }  public void onProviderEnabled(String s) {   } } |

5- Update **Manifest.xml** to be like this

|  |
| --- |
| Java |
| <?xml version="1.0" encoding="utf-8"?> <manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.example.hussienalrubaye.httpdostrackinglocation">  <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  <uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />  <uses-permission android:name="android.permission.INTERNET"/>  <application  android:allowBackup="true"  android:icon="@mipmap/ic\_launcher"  android:label="@string/app\_name"  android:supportsRtl="true"  android:theme="@style/AppTheme">  <activity android:name=".MainActivity">  <intent-filter>  <action android:name="android.intent.action.MAIN" />   <category android:name="android.intent.category.LAUNCHER" />  </intent-filter>  </activity>  </application>  </manifest> |

5- update **MainActivity.java** to be like this

|  |
| --- |
| Java |
| package com.example.hussienalrubaye.httpdostrackinglocation;  import android.Manifest; import android.content.Context; import android.content.pm.PackageManager; import android.location.LocationManager; import android.os.Build; import android.support.v4.app.ActivityCompat; import android.support.v7.app.AppCompatActivity; import android.os.Bundle;  public class MainActivity extends AppCompatActivity {  //setting send user location times  private static final long *MINIMUM\_DISTANCE\_CHANGE\_FOR\_UPDATES* = 1; // in Meters  private static final long *MINIMUM\_TIME\_BETWEEN\_UPDATES* = 1000; // in Milliseconds   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main*);   //check if the API>=23 to display runtime request permison  if ((int) Build.VERSION.*SDK\_INT* >= 23) {  // check if this permission is not grated yet  if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*) !=  PackageManager.*PERMISSION\_GRANTED*) {  //shouldShowRequestPermissionRationale(). This method returns true  // if the app has requested this permission previously and the user denied the request.  if (!shouldShowRequestPermissionRationale(Manifest.permission.*ACCESS\_FINE\_LOCATION*)) {  // display request permission  requestPermissions(new String[]{Manifest.permission.*ACCESS\_FINE\_LOCATION*,  Manifest.permission.*ACCESS\_COARSE\_LOCATION*},  REQUEST\_CODE\_ASK\_PERMISSIONS);  return;   }   return;  }  }  //call start track location  StartService();  }   //get access to mailbox  final private int REQUEST\_CODE\_ASK\_PERMISSIONS = 123;  //request permsion result  @Override  public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {  switch (requestCode) {  case REQUEST\_CODE\_ASK\_PERMISSIONS:  if (grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) {  //call start track location  StartService();    break;  default:  super.onRequestPermissionsResult(requestCode, permissions, grantResults);  }  } //This method start service for tracking location  void StartService() {  // run service for one time only  if (MyLocationListener.*ISActive* == false) {  MyLocationListener.*ISActive* = true;  //start listen service  LocationManager locationManage = (LocationManager) getSystemService(Context.*LOCATION\_SERVICE*);  locationManage.requestLocationUpdates(  LocationManager.*GPS\_PROVIDER*,  *MINIMUM\_TIME\_BETWEEN\_UPDATES*,  *MINIMUM\_DISTANCE\_CHANGE\_FOR\_UPDATES*,  new MyLocationListener(this)  );  }  }} |

**Fix The Problem:**

To fix the problem we will manage the sending of user location to the server time period, that is mean we send update in user location every one minute, so even the location changes very fast we don’t send the change in user location unless there is one minute left from last sending.

1- update class named “**MyLocationListener.java**”

|  |
| --- |
| Java |
| package com.example.hussienalrubaye.httpdostrackinglocation;  import android.content.Context; import android.location.Location; import android.location.LocationListener; import android.os.Bundle; import android.widget.Toast;  import java.io.BufferedInputStream; import java.io.IOException; import java.io.InputStream; import java.net.HttpURLConnection; import java.net.MalformedURLException; import java.net.URL; import java.text.SimpleDateFormat; import java.util.Date; import java.util.Locale;  import android.content.Context; import android.location.Location; import android.location.LocationListener; import android.os.Bundle; import android.widget.Toast;  import java.text.SimpleDateFormat; import java.util.Date; import java.util.Locale;  */\*\*  \* This service listen to change in user location every 1 meter  \* and send this data over http to server to save user location  \*/* public class MyLocationListener implements LocationListener {  // previous send date  Date interestingDate = new Date();  Context context;  public static boolean *ISActive*=false;  String URL;  long DiffDate;  public MyLocationListener(Context context){  this.context =context ;  }  // this method calls every change in one meter in user location  public void onLocationChanged(Location location) {   //check if there is more than 1 minute past from last send  DiffDate= (new Date()).getTime() - interestingDate.getTime();   if(DiffDate>60000) {  interestingDate = new Date();  //define the http url that recive location  URL = "http://news.alruabye.net/newfeeds.aspx?Latitude=" + Double.*toString*(location.getLatitude()) + "&Longitude=" + Double.*toString*(location.getLongitude());  Toast.*makeText*(context, URL, Toast.*LENGTH\_LONG*).show();   try {  //define url for send user location  URL url = new URL(URL);  //make http connection to send user location  HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();  urlConnection.setConnectTimeout(7000);//set timeout to 5 seconds  // InputStream in = new BufferedInputStream(urlConnection.getInputStream());    } catch (Exception e) {  e.printStackTrace();  } finally {  //urlConnection.disconnect();  }  }   }   public void onStatusChanged(String s, int i, Bundle b) {   }   public void onProviderDisabled(String s) {   }   public void onProviderEnabled(String s) {   }   } |