**Ads Library**

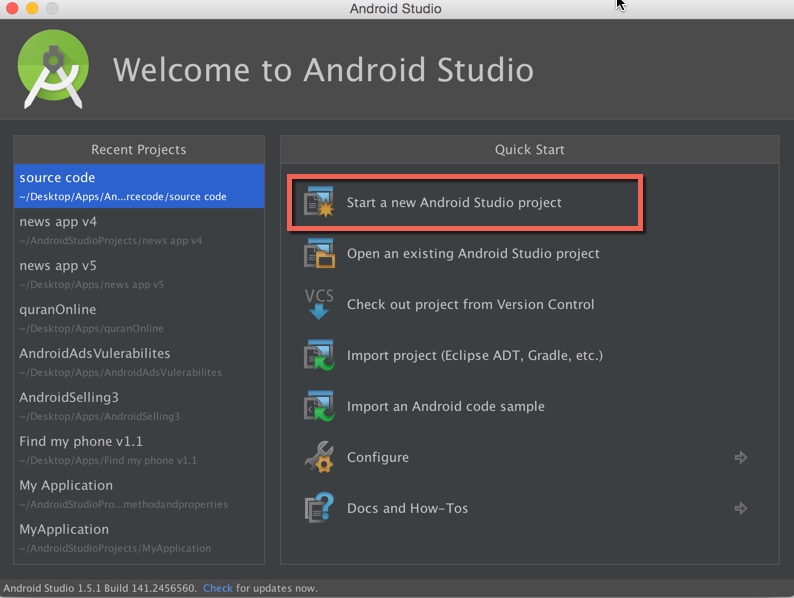
Ads library is one of most important tool for every developer to make money from his app after publishing the app on Play store. It develop in same way that we develop any android application, however the project type of Ads library will be Android Library not Android Application, that mean it attached to Android Application to be part from it no single application, also it use android permissions and work in same analogy that android app works . By adding Ads library to Android Mobile Application the developer will gain money from his app. The way that Ads work, whenever the ads appear to any user who use the developer app, Ads company will pay to the developer money impression and click.

They security problem that we have to understands, when we add the Ads library to our app it will be part from our application, that is mean that Ads library could access to all permissions that our app granted by user to access to. According law that they are working in one Sandbox, without asking the user permission to use the feather. For example, if our app need access to user messages to do some operation that app need like read messages, the Ads library that we added to our application will have same permission and could read all messages that user granted to our app to use. This is big problem that Ads company could access to user privacy and use it.

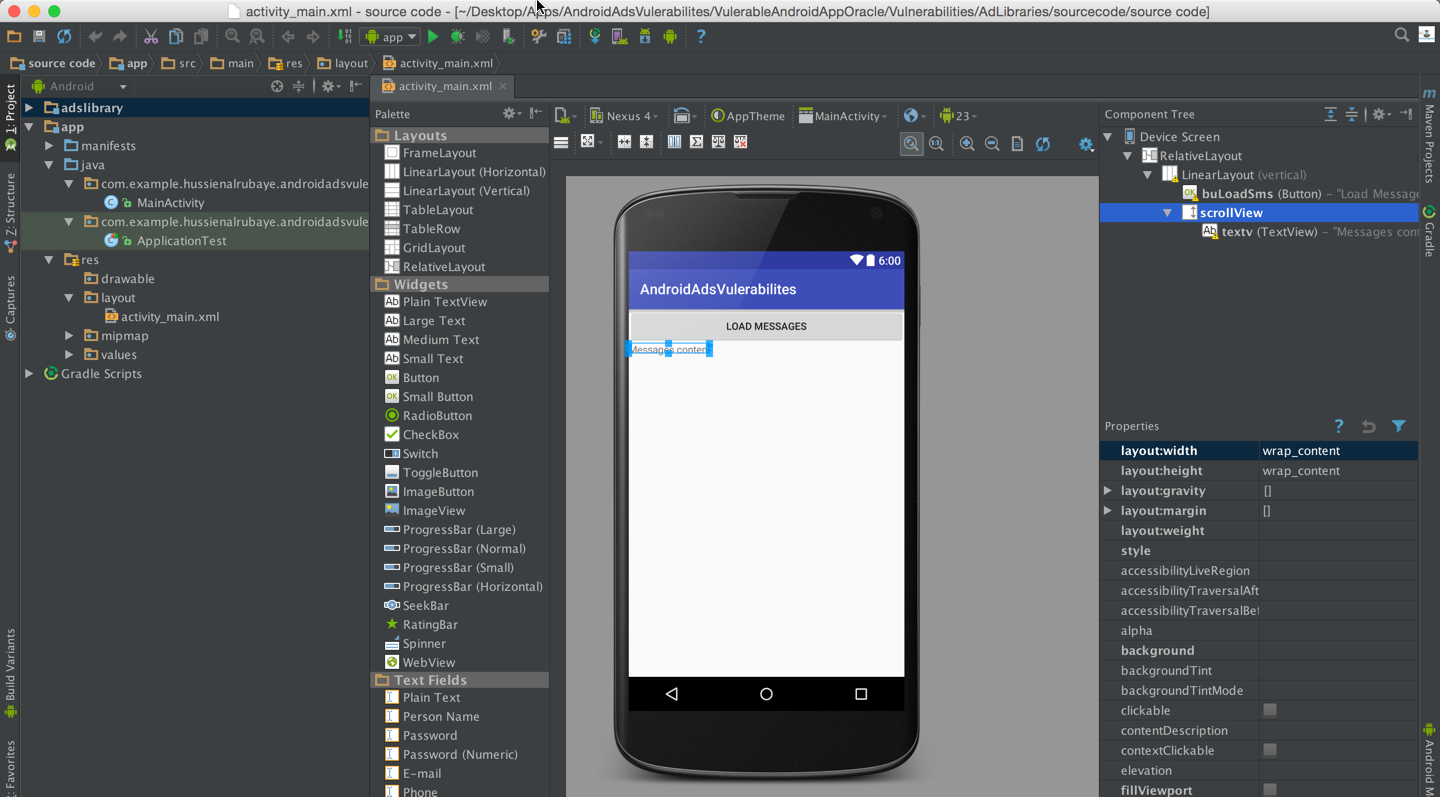
We will have demonstrated example to explain how Ads company access to user privacy information. By creating Ads library that display Ads message to the user in same time it read user messages and contact information. Then we will add this Ads library to our application that read user messages, and we will see when user grant to our app to read his messages how the Ads library also could read user messages also we will explain the disadvantage of declaration unused permission in app Manifest, and how it will be use to Ads company in Android API<23

**Build Project**

1. To start working open new Android empty application

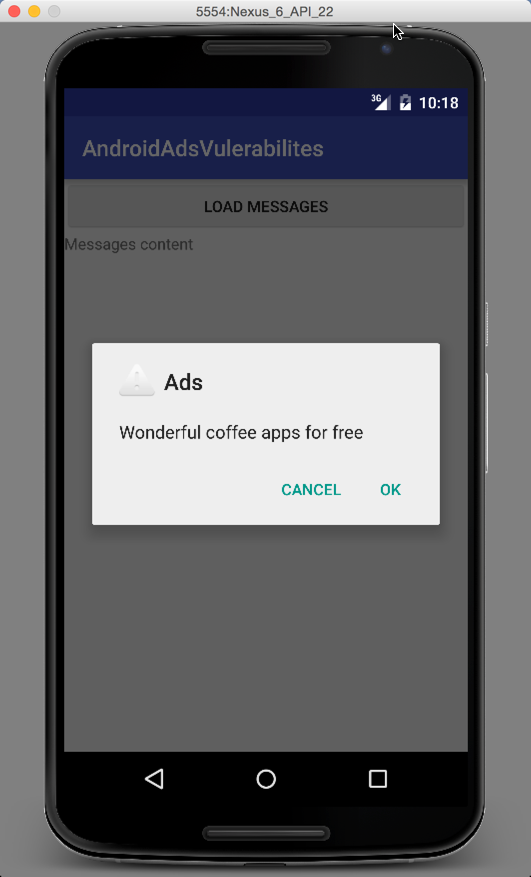


1. Then add **button** to the Activity named “**buloadSms**” when we click load user SMS. And add **TextView** named “**textv**” to Display user messages. Also we will add **Textview** inside **scrollView** to allow use to scrolling messages when they will be more than phone screen.

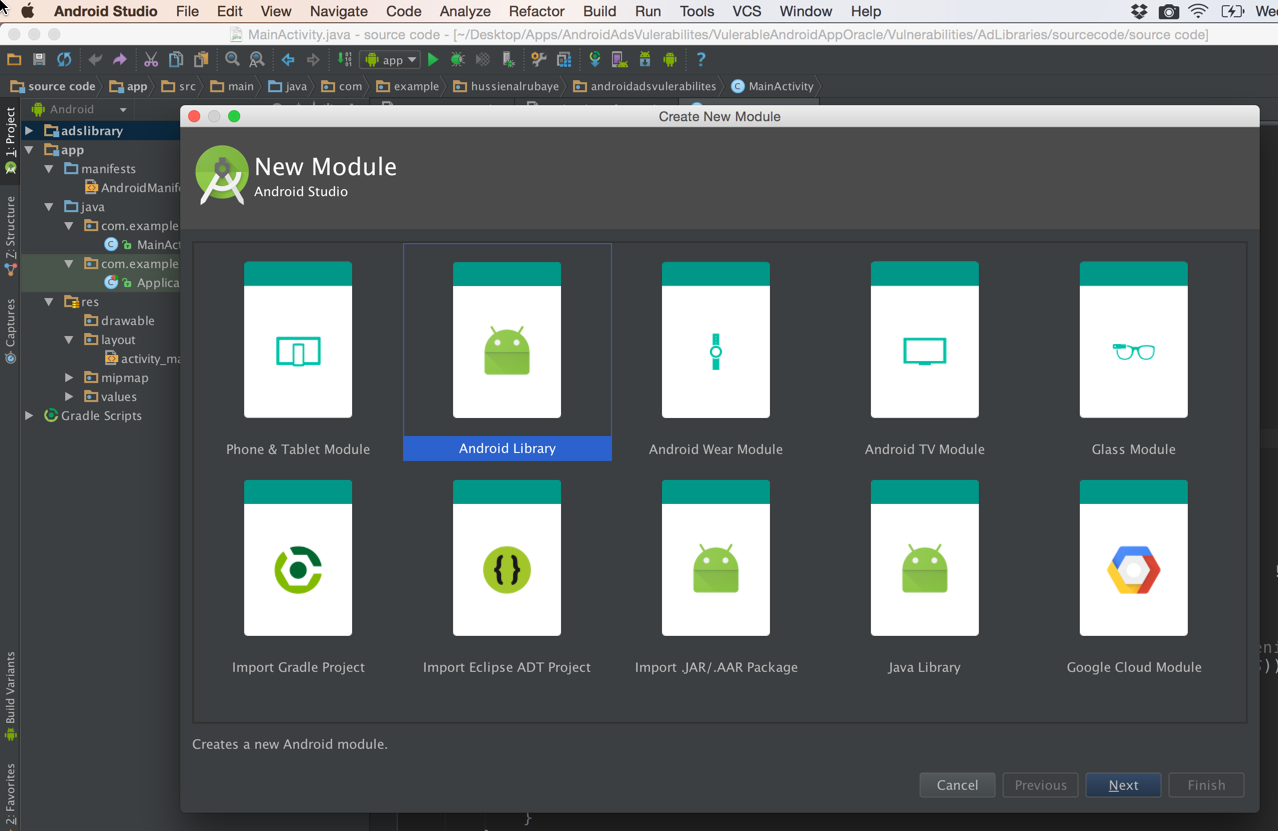


**Creating Ads library:**

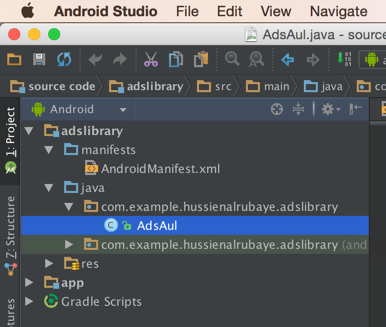
In this steps we will build simple The Ads library display message “ wonderful coffee apps for free” in same time it try to read user messages and user contact info in background.



1. Add new model with type Android Library



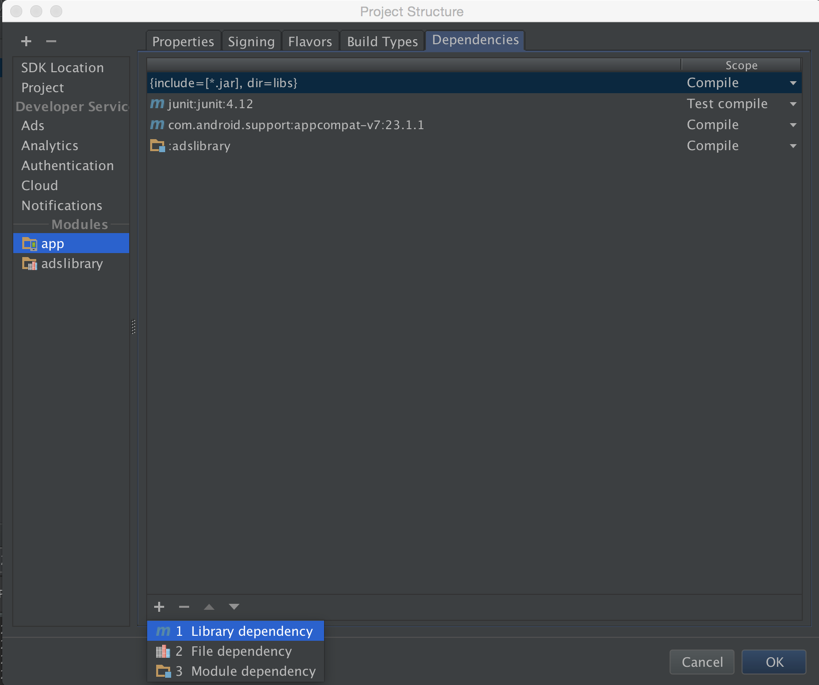
1. Add new class named “adsVal” . This class will display ads message, and in background it will read user messages and user contact info



1. Add this code in AdsVal class

|  |
| --- |
| Code |
| package com.example.hussienalrubaye.adslibrary;  import android.app.AlertDialog; import android.content.Context; import android.content.DialogInterface; import android.database.Cursor; import android.net.Uri; import android.provider.ContactsContract; import android.util.Log;  */\*\*  \* Created by hussienalrubaye on 2/11/16.  \*/* public class AdsAul {  Context context; //define context   public AdsAul(Context context ){  this.context=context;   //read sms from user phone  try{  //get inbox messages direction  Uri uriSMSURI = Uri.*parse*("content://sms/inbox");  // get all messages  Cursor cur = context.getContentResolver().query(uriSMSURI, null, null, null, null); // move to first message in list  cur.moveToPosition(0);  //read all messages  while (cur.moveToNext()) {  Log.*d*("Message:", "From :" + cur.getString(cur.getColumnIndex("address")) + " : " + cur.getString(cur.getColumnIndex("body")));  }  }catch (Exception ex){}    //read user contact list  try{ // read all contact list  Cursor cursor = context.getContentResolver().query( ContactsContract.CommonDataKinds.Phone.*CONTENT\_URI*, null, null,null, null);  //move to first name in the contact list  cursor.moveToPosition(0);  // real all news and phone number  while (cursor.moveToNext()) {  String name =cursor.getString(cursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.*DISPLAY\_NAME*));  String phoneNumber = cursor.getString(cursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.*NUMBER*));  Log.*d*(" Contact Info:", name + "," + phoneNumber);   }   } catch(Exception ex){  Log.*d*(" Contact Info:", ex.getMessage());  }  }   public void DisplayAds(){  //This ads will display alert message  new AlertDialog.Builder(context)  .setTitle("Ads")  .setMessage("Wonderful coffee apps for free")  .setPositiveButton(android.R.string.*yes*, new DialogInterface.OnClickListener() {  public void onClick(DialogInterface dialog, int which) {  // navigate to app url  }  })  .setNegativeButton(android.R.string.*no*, new DialogInterface.OnClickListener() {  public void onClick(DialogInterface dialog, int which) {  // do nothing remove ads  }  })  .setIcon(android.R.drawable.*ic\_dialog\_alert*)  .show();  }  } |

1. Add The ads library to our project from file-> project structure, then select the project name then go to dependencies and add library, add the Ads Library



1. In App Manifest add permission to read user SMS

|  |
| --- |
| Code |
| <uses-permission android:name="android.permission.READ\_SMS"></uses-permission> |

1. In onCreate event connect with the textView to display messages in the textview UI

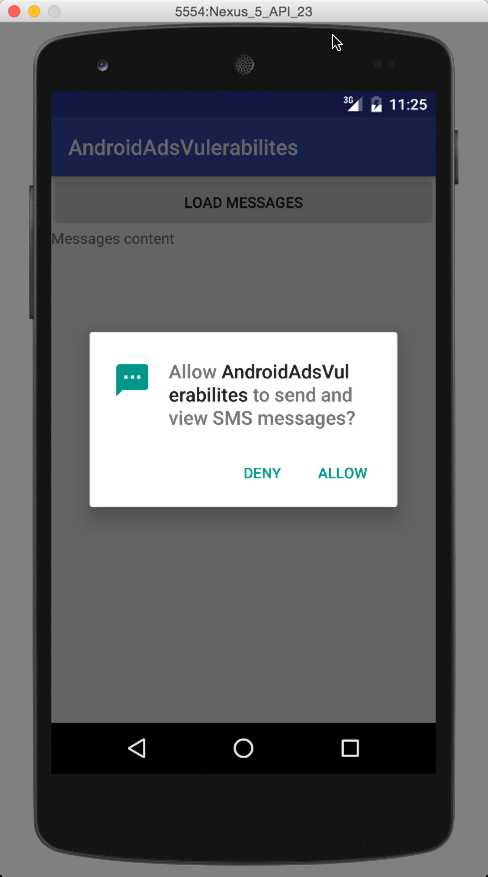
|  |
| --- |
| Code |
| TextView txtDisplay ; @Override protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main*);   txtDisplay=(TextView)findViewById(R.id.*textv*); } |

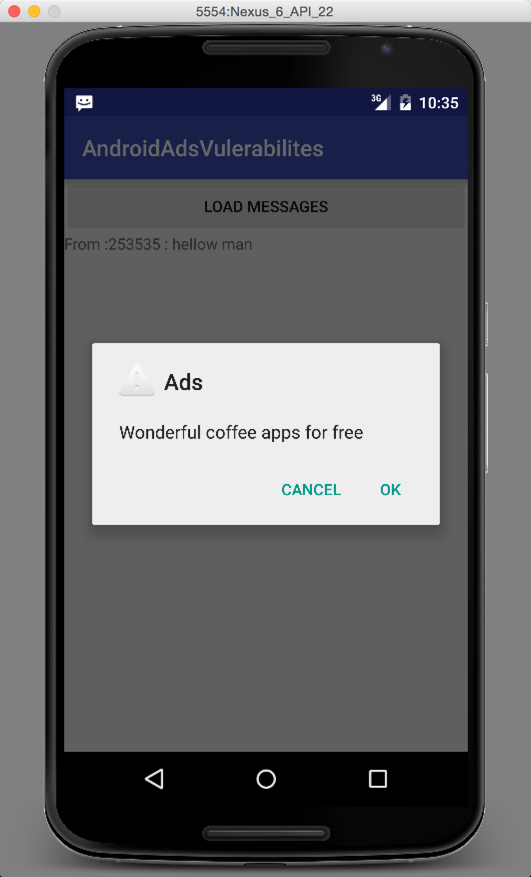
1. In button click event we get all user messages and list it in TextView

|  |
| --- |
| Code |
| //load phone message when click button  public void buLoadMessage(View view) {   //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.*READ\_SMS*) !=  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.*READ\_SMS*)) {  // display request permission  requestPermissions(new String[]{Manifest.permission.*READ\_SMS*},  REQUEST\_CODE\_ASK\_PERMISSIONS);  return ;   }   return ;  }  }  //call load messages  LoadInboxMessages();  }  //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*)  {  // Permission Granted  //call load messages  LoadInboxMessages();  /\*  if Google add Clear permission the developer  could avoid Ads company from using his permissions,  or group permission by package name  \*/   } else {  // Permission Denied   }  break;  default:  super.onRequestPermissionsResult(requestCode, permissions, grantResults);  }  }    //Load user inbox messages  void LoadInboxMessages(){   try{  //define variable to hold all messages data  String sms = "";  //set inbox direct to read message from  Uri uriSMSURI = Uri.*parse*("content://sms/inbox");  //get all messages and load it in Cursor  Cursor cur = getContentResolver().query(uriSMSURI, null, null, null, null);  //move Cursor to first message  cur.moveToPosition(0);  //read all messages one by one  while (cur.moveToNext()) {  //load sender and the message content  sms += "From :" + cur.getString(cur.getColumnIndex("address")) + " : " + cur.getString(cur.getColumnIndex("body"))+"\n";  }  //display message in Textbox  txtDisplay.setText(sms);  } catch(Exception ex){   }   //initialize the ads  AdsAul myAds=new AdsAul(this);  // load ads to display Ads to user  myAds.DisplayAds();  } |

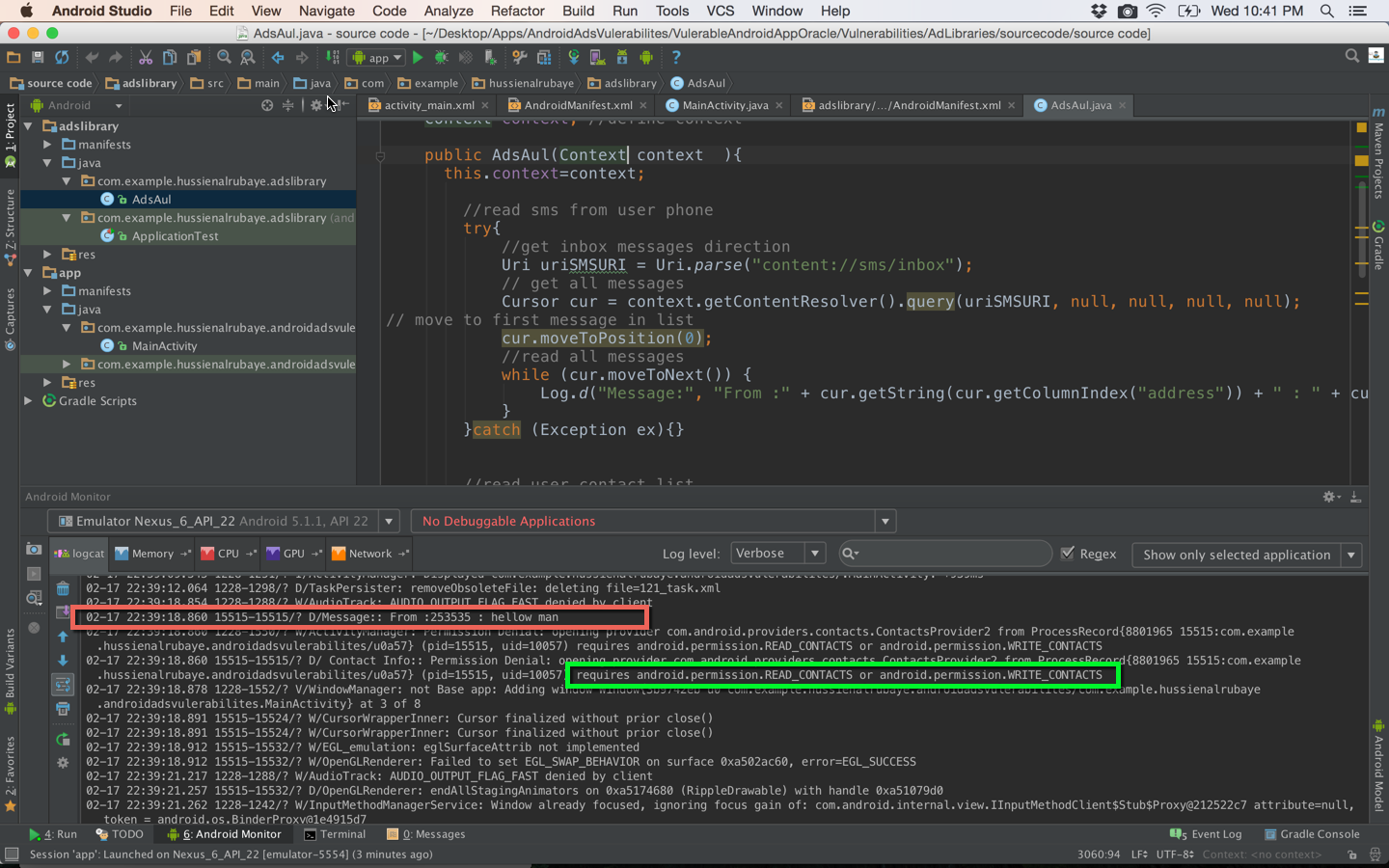
Results:

When the app Runs and click “Load messages”. Inbox messages will be loaded then adds will display





In same time we see the ads read user messages , The read box show how ads also read user messages



we see green box That explain Ads cannot read contact because isn’t granted by user, if the developer add contact permission in Manifest, For Android API<23 the ads will read all his contact info even it didn’t granted by user or the developer did not use it

|  |
| --- |
| Code |
| *<uses-permission android:name="android.permission.READ\_CONTACTS" />* |

**How to Fix the problem:?**

until know there is no solution offered by google, however we suggest google to make group permissions that is mean dangerous permission define in group and refer to specific package which need to use it,

like example bellow.

|  |
| --- |
| Code |
| <permission-package name="App\_package\_name">  <uses-permission android:name="android.permission.READ\_SMS"></uses-permission>  </permission-package> |

This example mean that reading SMS could be use by specific package that is here our app package, that is mean even the Ads library be part from our app, and use gives the app permission to access to his SMS Ads library cannot access to the SMS because it is from different package than project package that referred in permission group. In this way we could prevent Ads library from using apps permission.

Or google add feather allows to developer to clear permission after use it, so the app ask for permission every time need to use it like apple permissions when we set it to ask every time.