**Intent**

**Description:**

In Android, we can send data between apps quite easily. For example, when we want to share images on Facebook from a local app, we send the image to the Facebook app over the bundle. Then, the Facebook app reads the image data and posts it. Another example might be that when we want to call someone, we send the phone number to call over the bundle to the calling API.

Sending data between apps is especially important for companies with multiple apps that need to interact in some way.

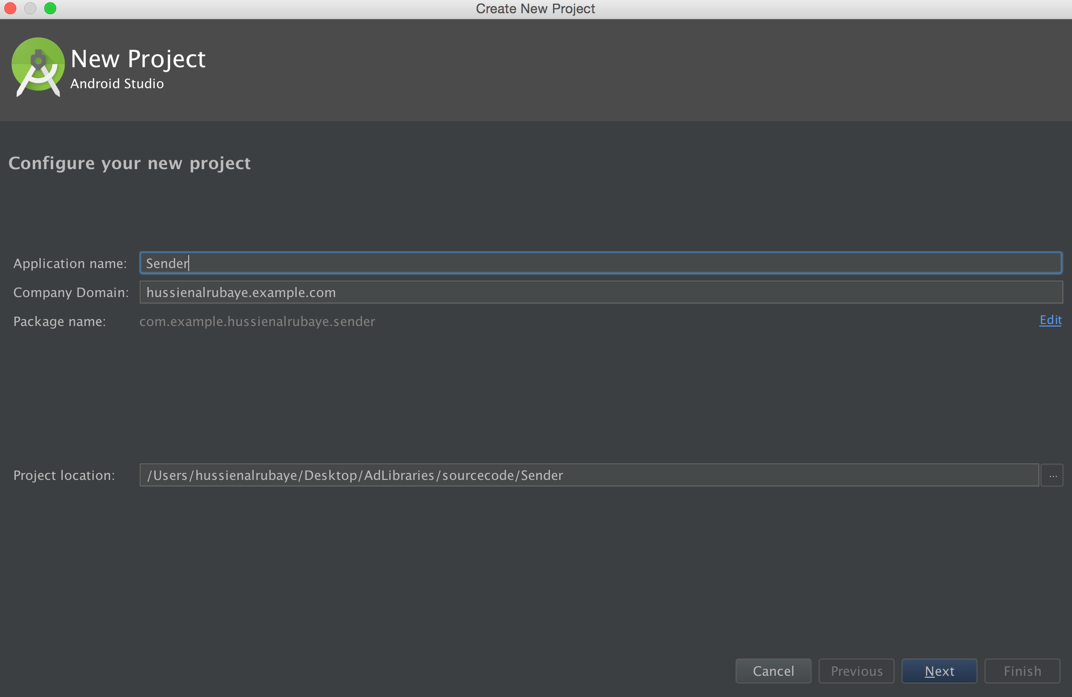
The problem is that sending data between apps is insecure in Android. For example, if we send data from app A to app B, any app C might have the same application ID of app B. Then, app C could receive the data that should be received by app B. We cannot send important data between applications without encryption.

We will demonstrate examples to show how to send data between two applications. Next, we will show how a (possibly) malicious app could read the data, and finally, we will explain how we could prevent against this from happening.

**Steps to build the sender app (app A):**

This app will send comment data to app B.

1. Create a new project with name “Sender”. Again, remember to save the package name. We will need that later.



1. Add some objects to the **activity\_main.xml**, found under “app/res/layout/”. Add a TextView, EditText, and a Button.

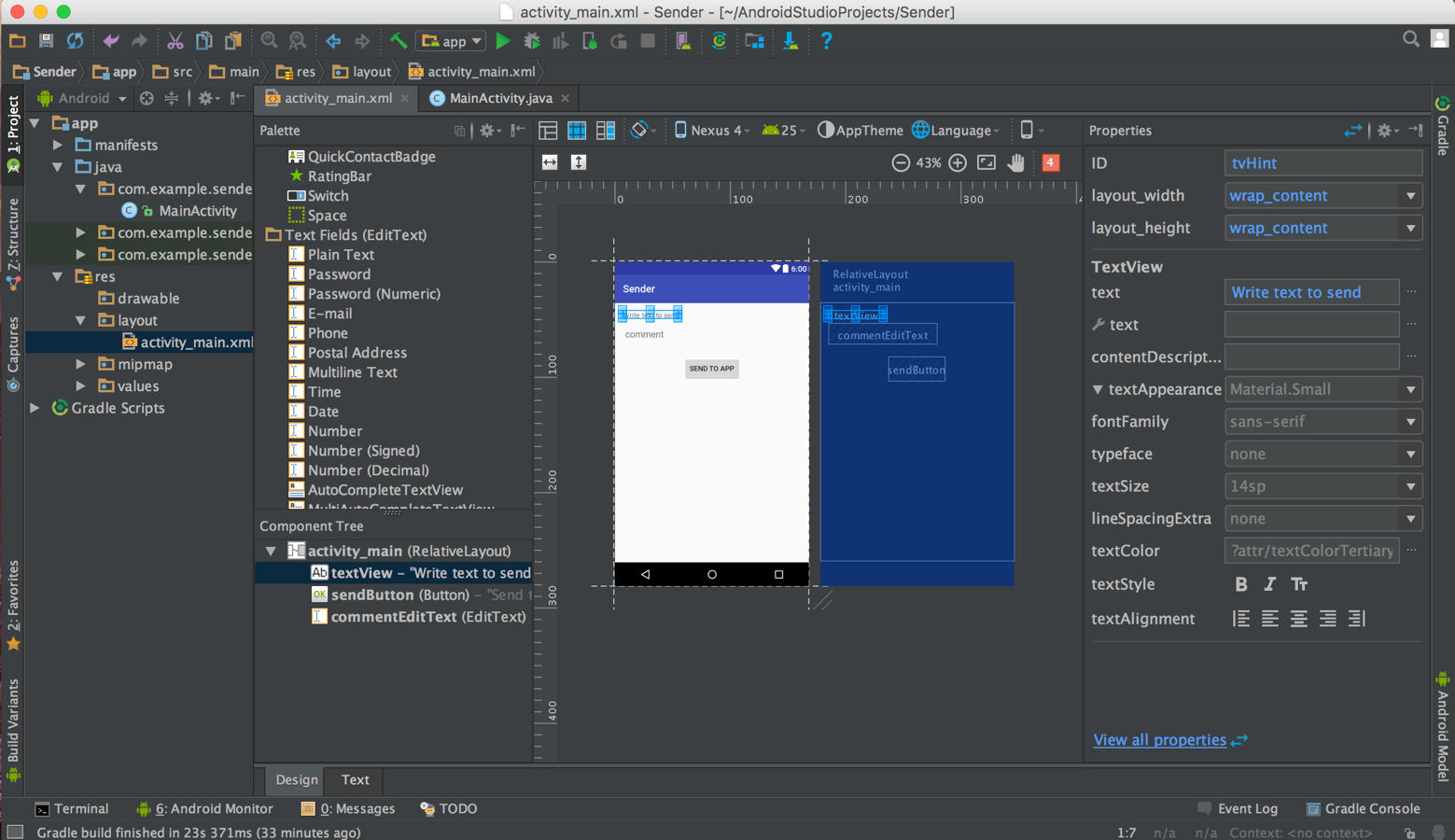
Once you have the objects in the device layout, click on the TextView and change its properties from right side panel to the following:

* ID: tvHint

The EditText should have these properties:

* ID: commentEditText
* Hint: comment
* Delete all text in the box next to the “Text” label

For the Button:

* ID: sendButton
* Text: Send to app
  + You may need to play around with this to avoid Android Studio autocompleting the word “app” to “@string/app\_name”

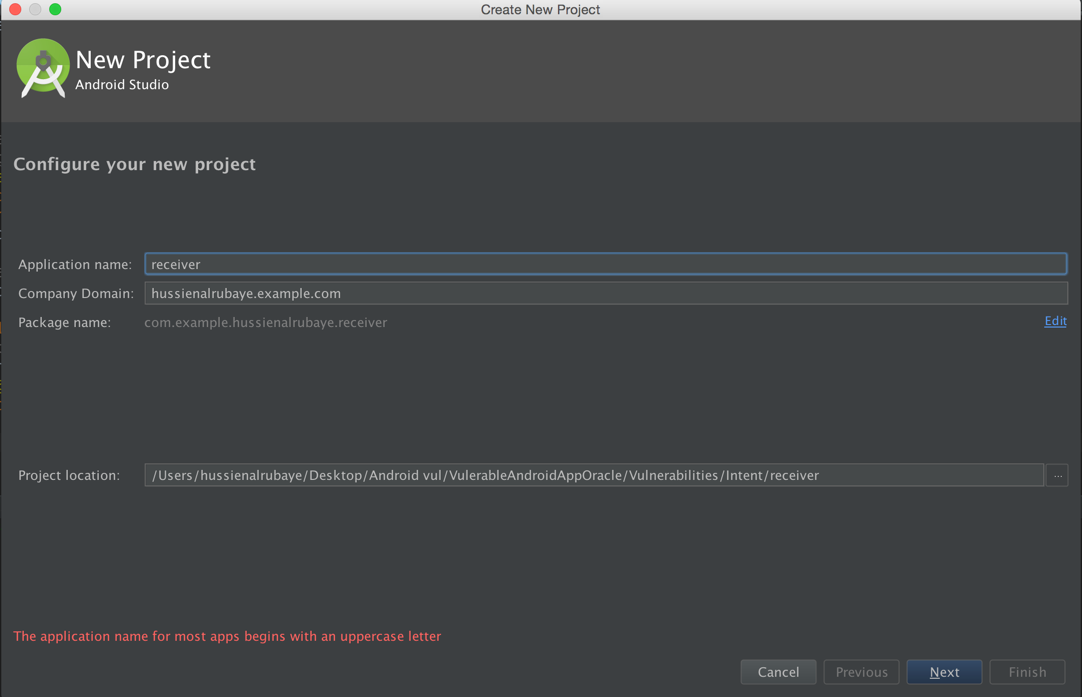
1. The code in **MainActivity.java**, found in “app/java/your\_package\_name\_here/”,will look like this:

|  |
| --- |
| **package** com.example.sender;  **import** android.content.Intent; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.view.View; **import** android.widget.Button; **import** android.widget.EditText;  **public class** MainActivity **extends** AppCompatActivity {  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   *// Define send button* Button sendButton = (Button)findViewById(R.id.***sendButton***);   *// init commentEditText* **final** EditText commentEditText = (EditText)findViewById(R.id.***commentEditText***);   *// button listen to click event* sendButton.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  *// set the package that we want to run* Intent intent = getPackageManager().getLaunchIntentForPackage(**"com.example.receiver"**);   *// put the data that we want to send over intent* intent.putExtra(**"Comment"**, commentEditText.getText().toString());   *// start another app* startActivity(intent);  }  });  } } |

**Steps to build the Receiver app (app B):**

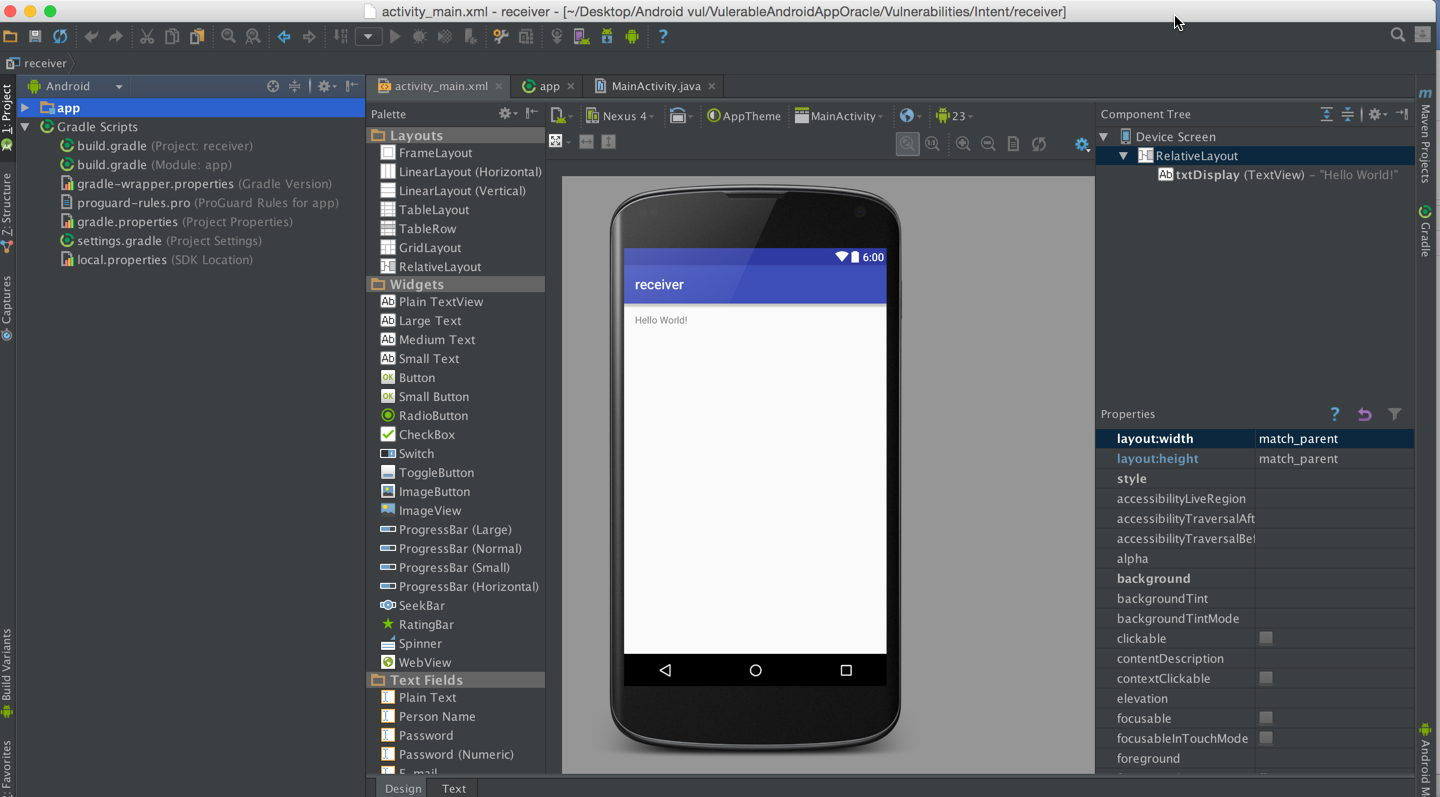
This app will receive the comment data and read it.

1. Create a new project with name “Receiver”. Save the package name, as usual.



1. Drag and drop a TextView from the “Palette” bar into the design view of **activity\_main.xml**, found in “app/res/layout/”. Edit the properties of the TextView in the Properties panel that pops up when you click on it to be:

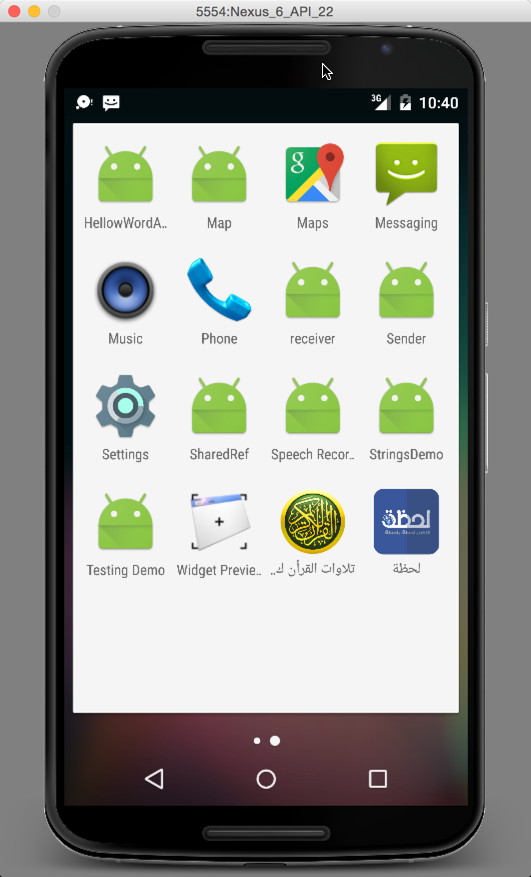
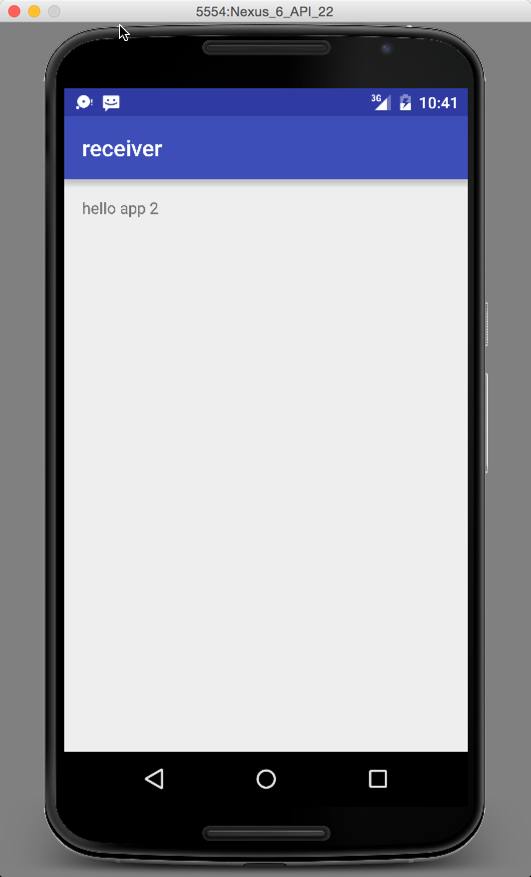
* ID: txtDisplay



1. Copy and paste the following code into **MainActivity.java**.

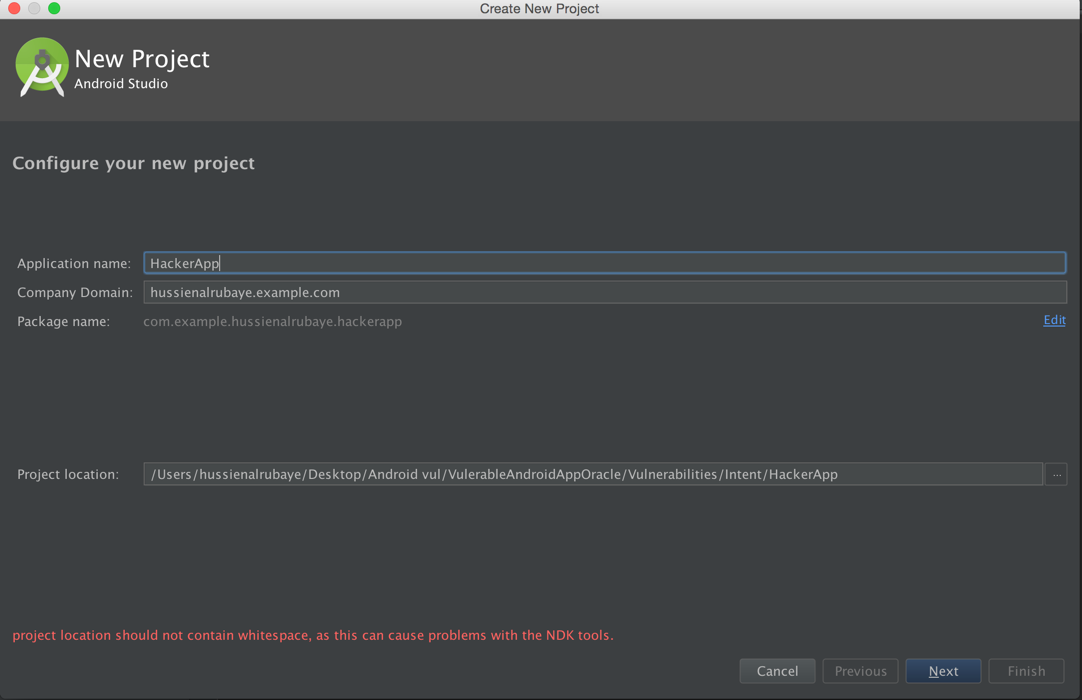
|  |
| --- |
| **package** com.example.receiver;  **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.widget.TextView;  **public class** MainActivity **extends** AppCompatActivity {  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   *// Define the display Text view* TextView txtview = (TextView)findViewById(R.id.***txtDisplay***);   *// get app the data sent on bundle* Bundle b = getIntent().getExtras();   *// display the key that have the data* txtview.setText(b.getString(**"Comment"**));  } } |

**Example of Run the apps and send comment “hi app 2” from App “A” to App “B”**

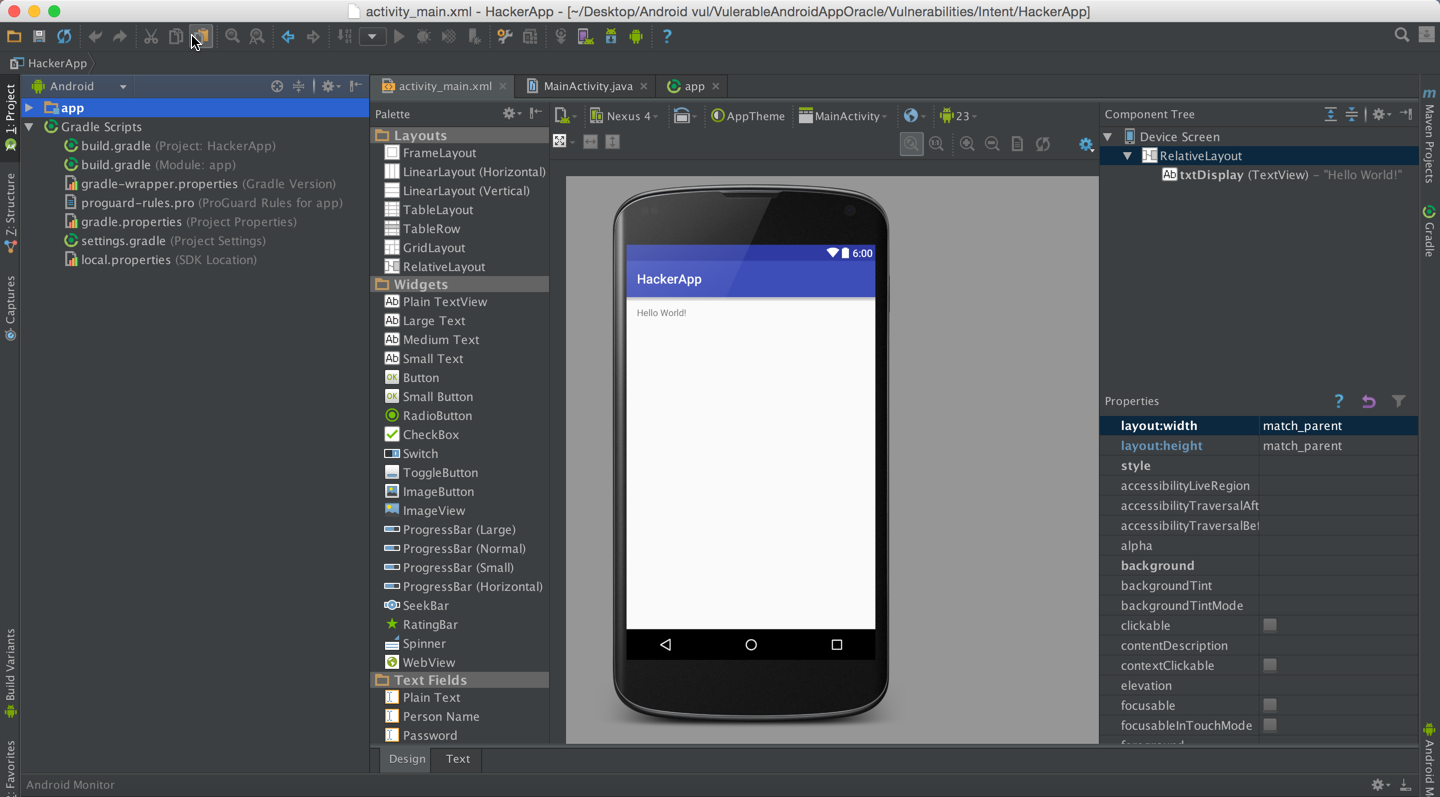
  

**Steps to build the hacker app:** This app will try to read the data that should be read only by APP “B”.

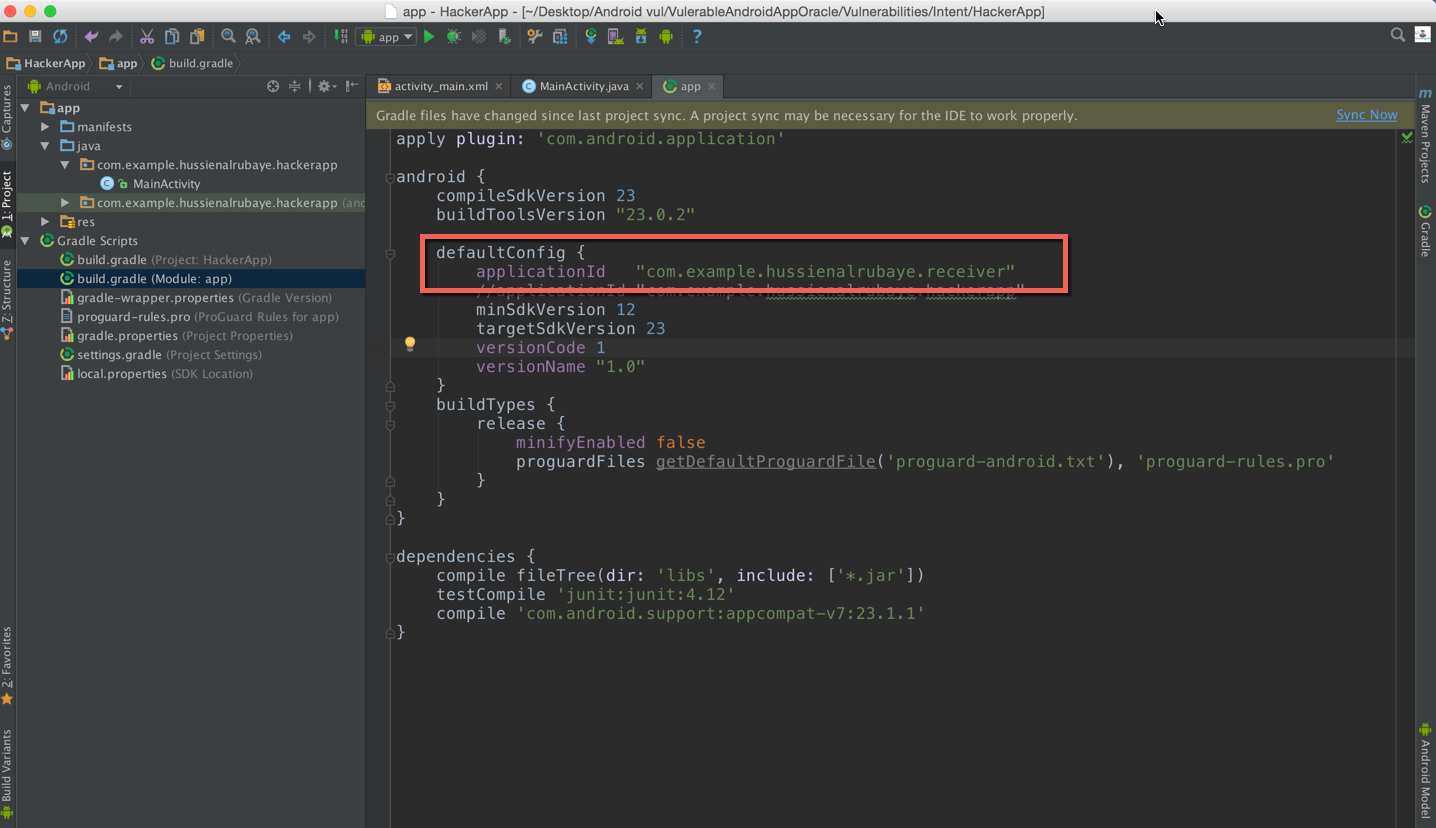
1. Open new project with name “HackerApp”, save the package name will will need next



1. add some objects ( TextView) and make the app like this, see the name of every tool in the right.



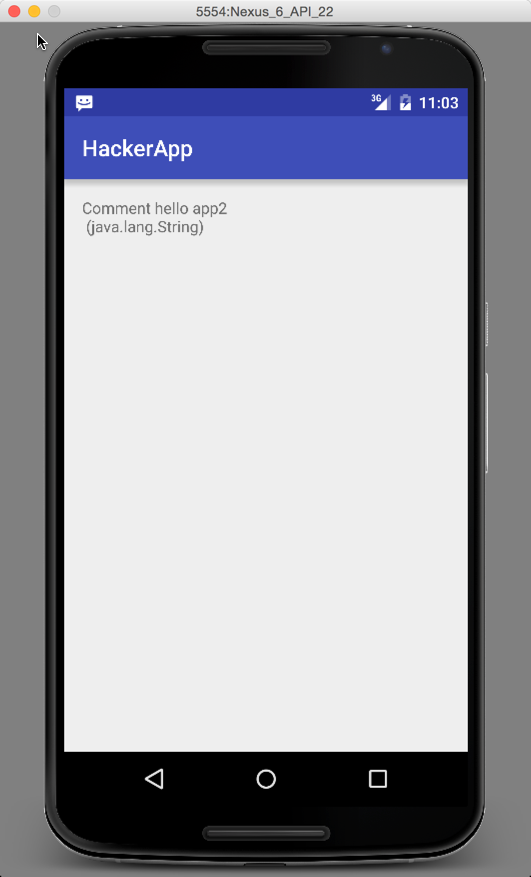
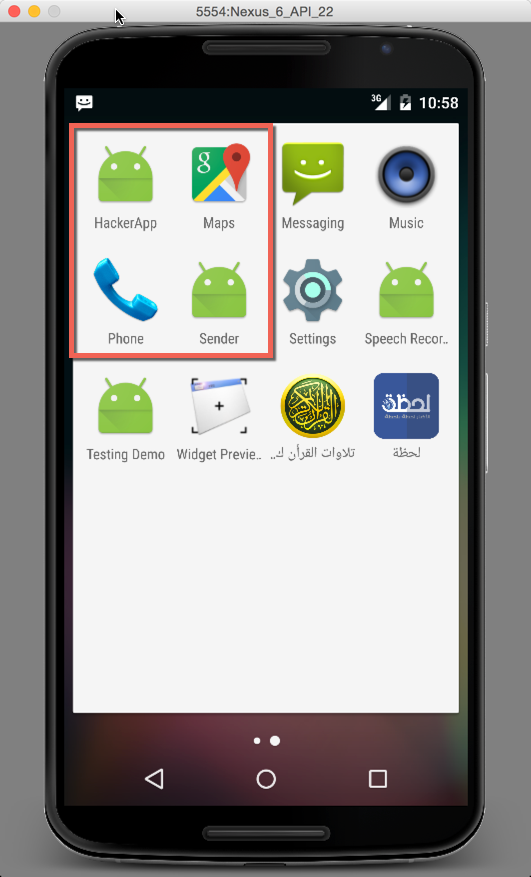
1. Go to the Gradle and change the Hacker application id to same the receiver app ID. So when it has the same receiver app ID, it could receive the data that should be receiver by app App “B”



1. The code will be like this code

|  |
| --- |
| Java |
| @Override protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main*);  // init the textView to dipslay data  TextView txtDisplay=(TextView)findViewById(R.id.*txtDisplay*);  String DataBundel="";  // get app the data sent on bundle  Bundle bundle=getIntent().getExtras();  //loop through all keys in the bundle  for (String key : bundle.keySet()) {  // get object by key( we define object became it may be text or image or whatever  Object value = bundle.get(key);  //get all keys  DataBundel+= String.*format*("%s %s (%s)", key, value.toString(), value.getClass().getName());  }  txtDisplay.setText(DataBundel); } |

**Example of Run the apps and send comment “hi app 2” from App “A” then Hacker app read the data**



**Fix This Problem**

To fix this problem use encryption and decryption theory. That is mean that in sender app we encrypt the data before send it over bundle. Then in receiver app we decrypt that bundle data so the hacker app even could access to the data, it could not understand it.

**see the example below that shows the secure data send between App “A” and App “B”. It shows that even Hacker app access to the data but it cannot understand it.**

