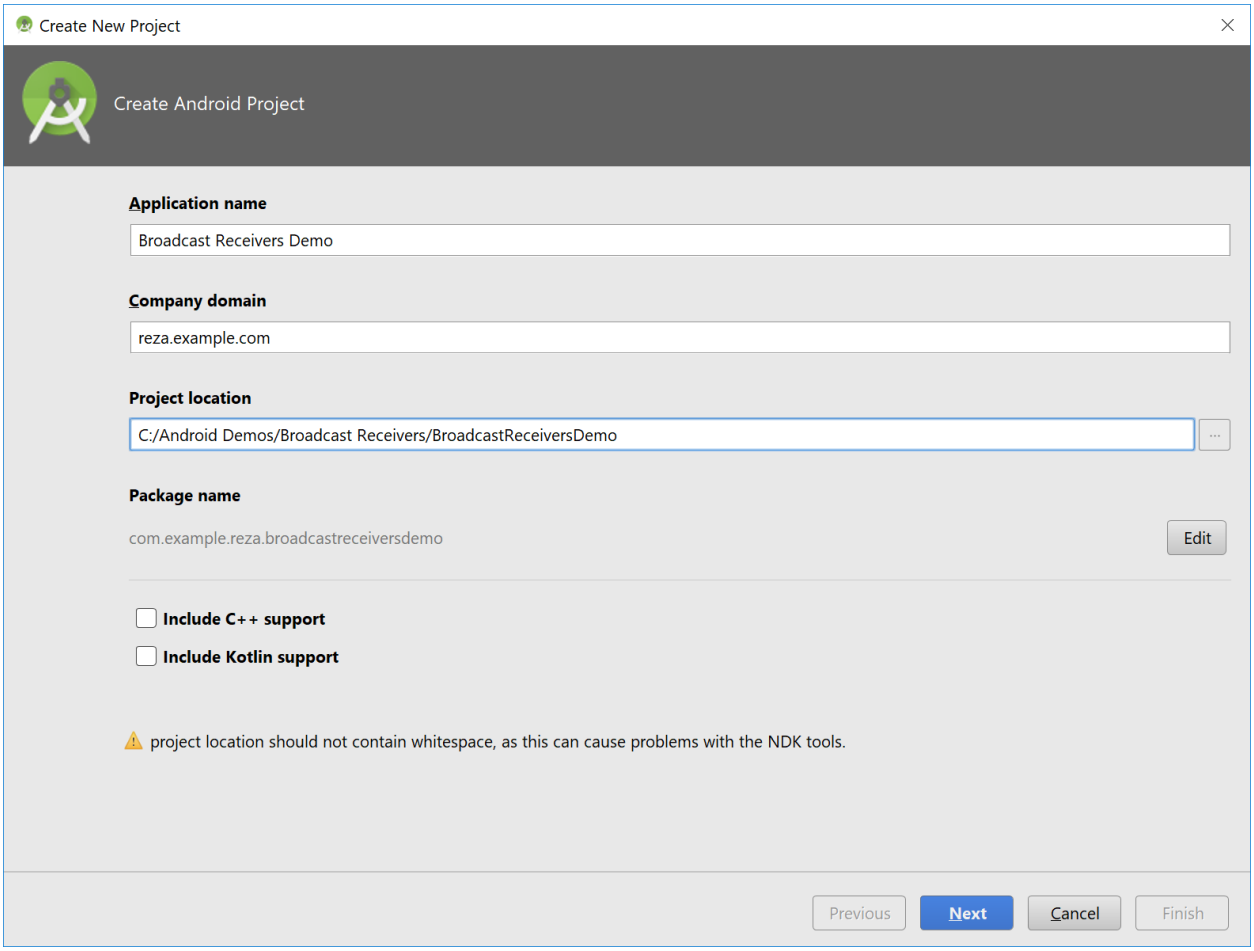
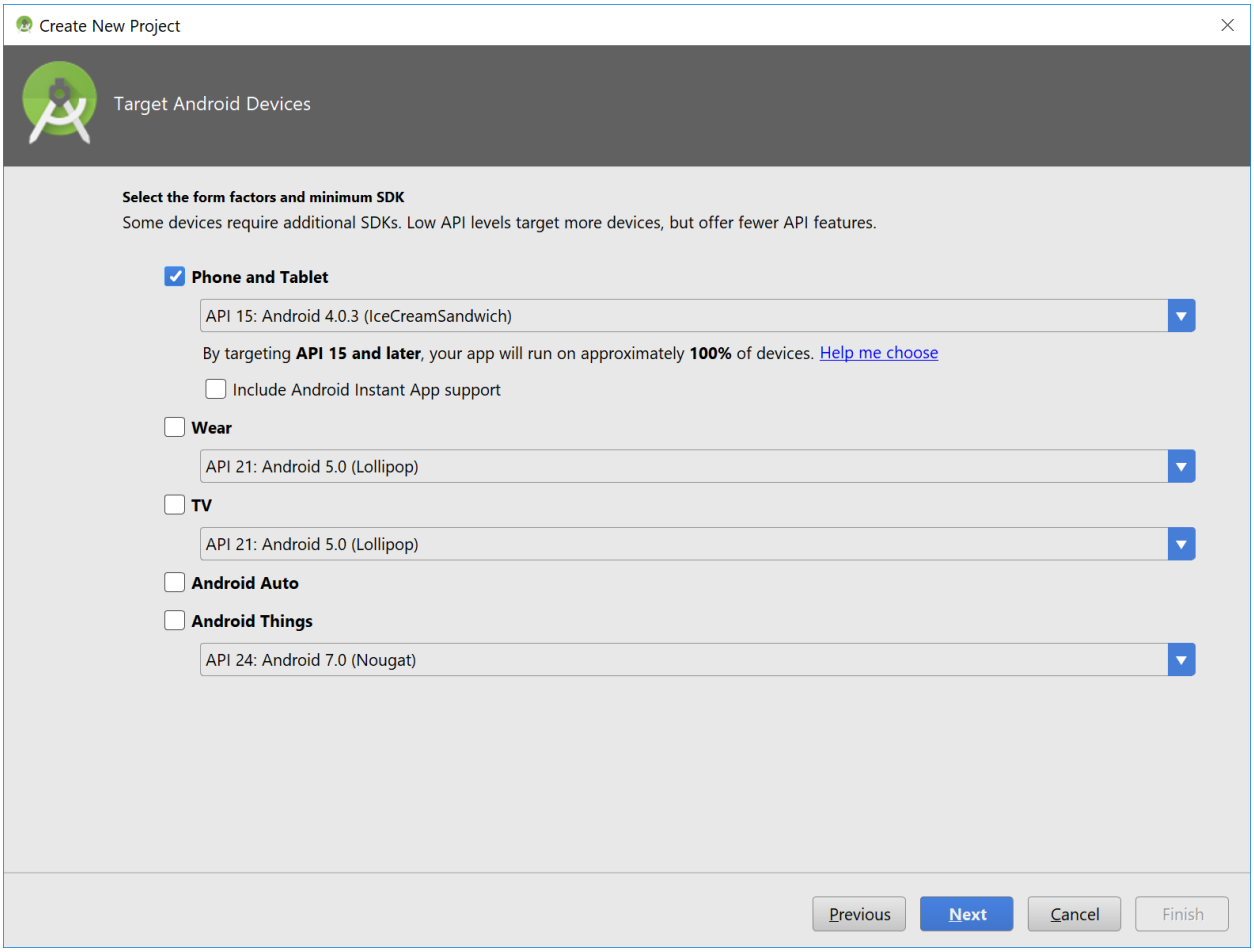
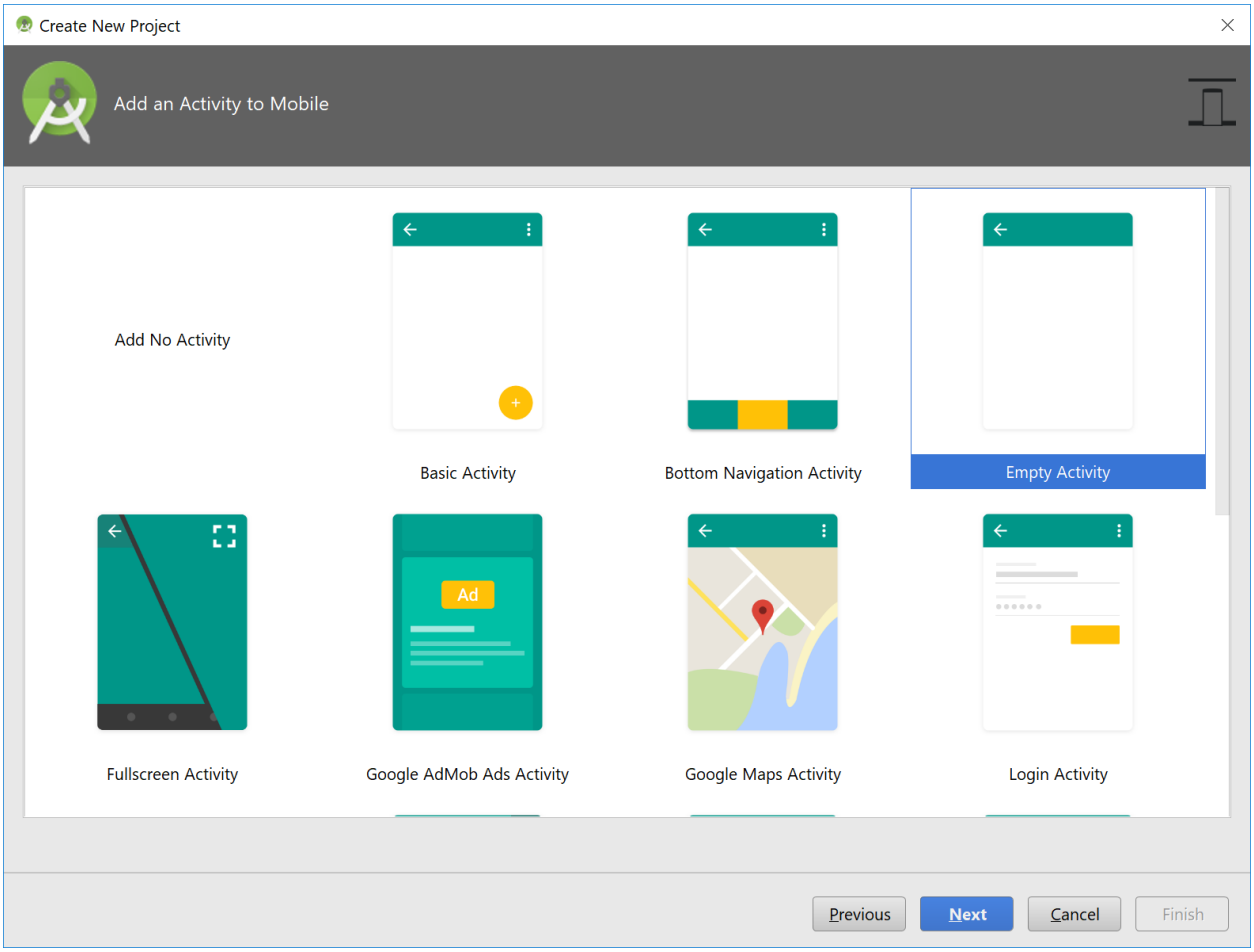
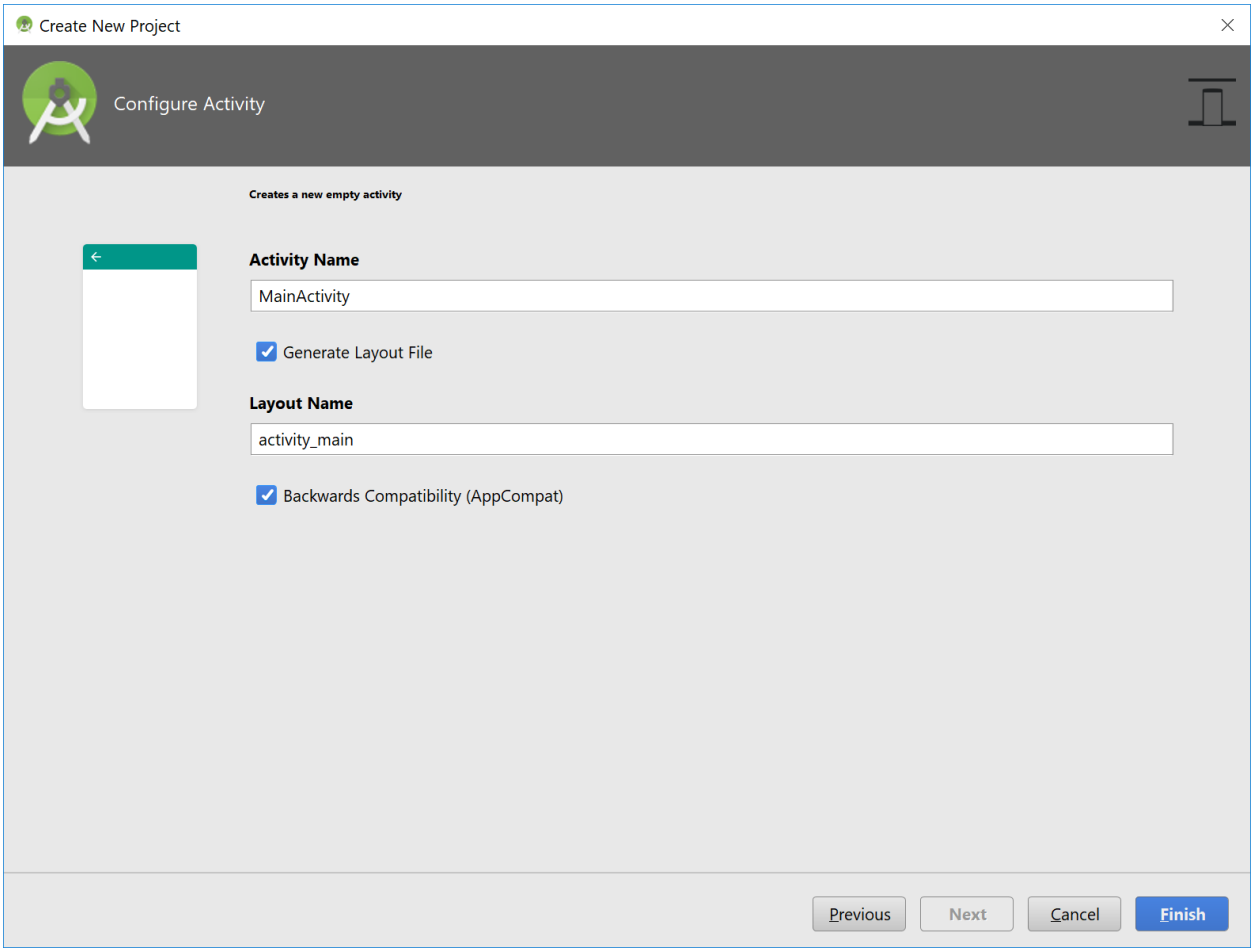
Broadcast Receivers

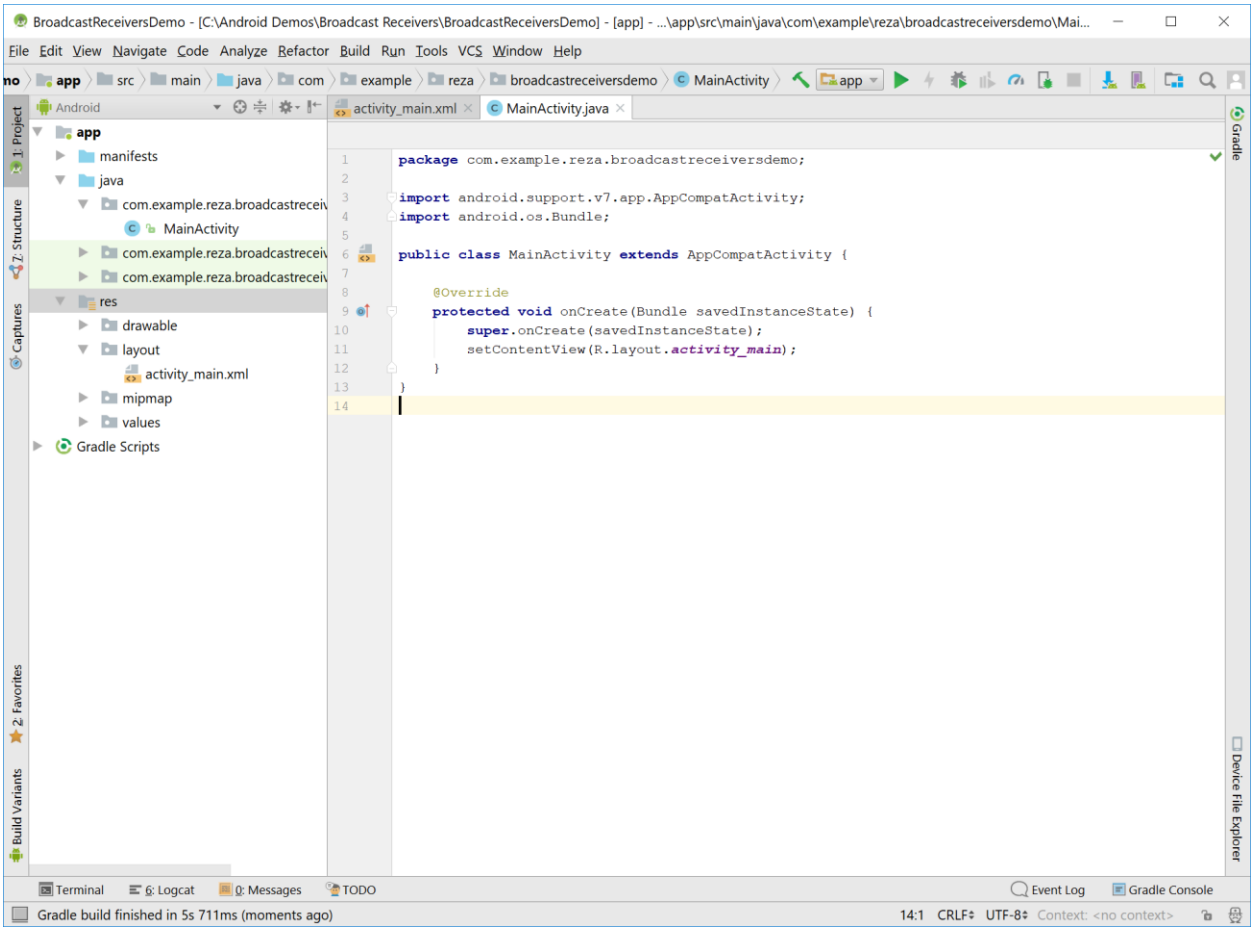
Create a new project as shown:

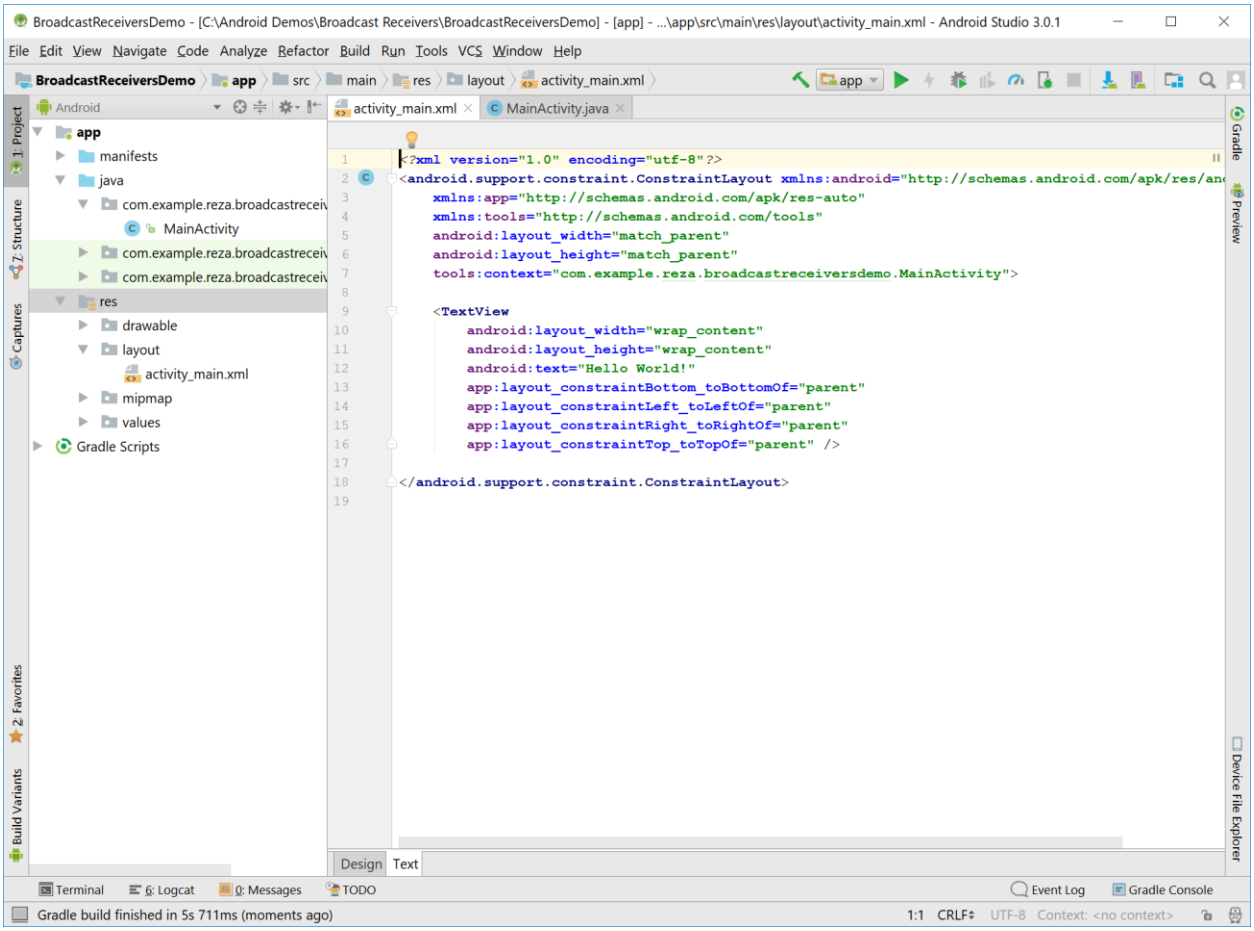


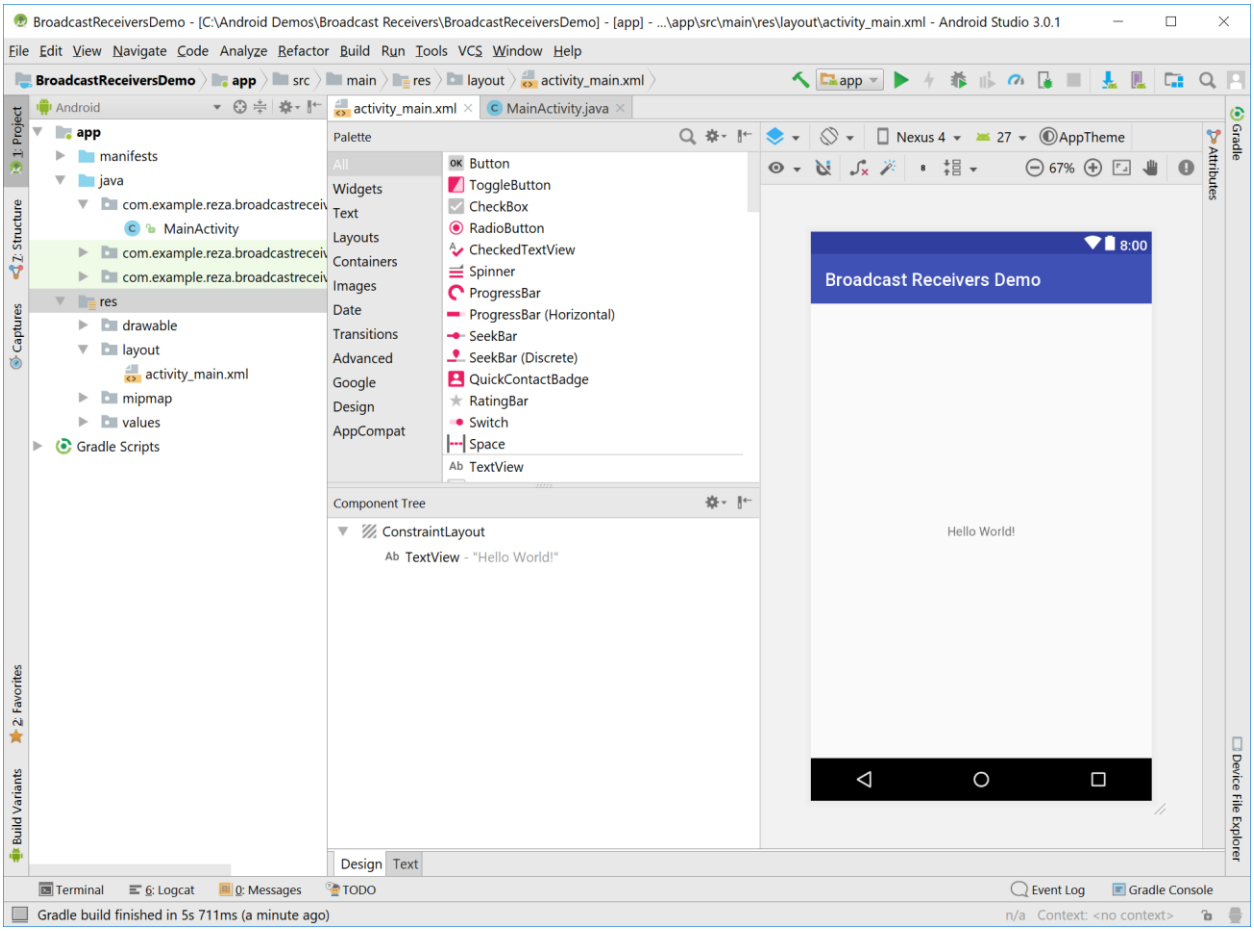

Create New Project
Target Android Devices
Select the form factors and minimum SDK
Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.
Phone and Tablet
API 15: Android 4.0.3 (IceCreamSandwich)
By targeting API 15 and later, your app will run on approximately 100% of devices. Help me choose
Include Android Instant App support
Wear
API 21: Android 5.0 (Lollipop)
TV
API 21: Android 5.0 (Lollipop)
Android Auto
Android Things
API 24: Android 7.0 (Nougat)
Previous
Next
Cancel
Finish


Create New Project
Add an Activity to Mobile
Add No Activity
Basic Activity
Bottom Navigation Activity
Empty Activity
Fullscreen Activity
Ad
Google AdMob Ads Activity
Google Maps Activity
Login Activity
Previous
Next
Cancel
Finish

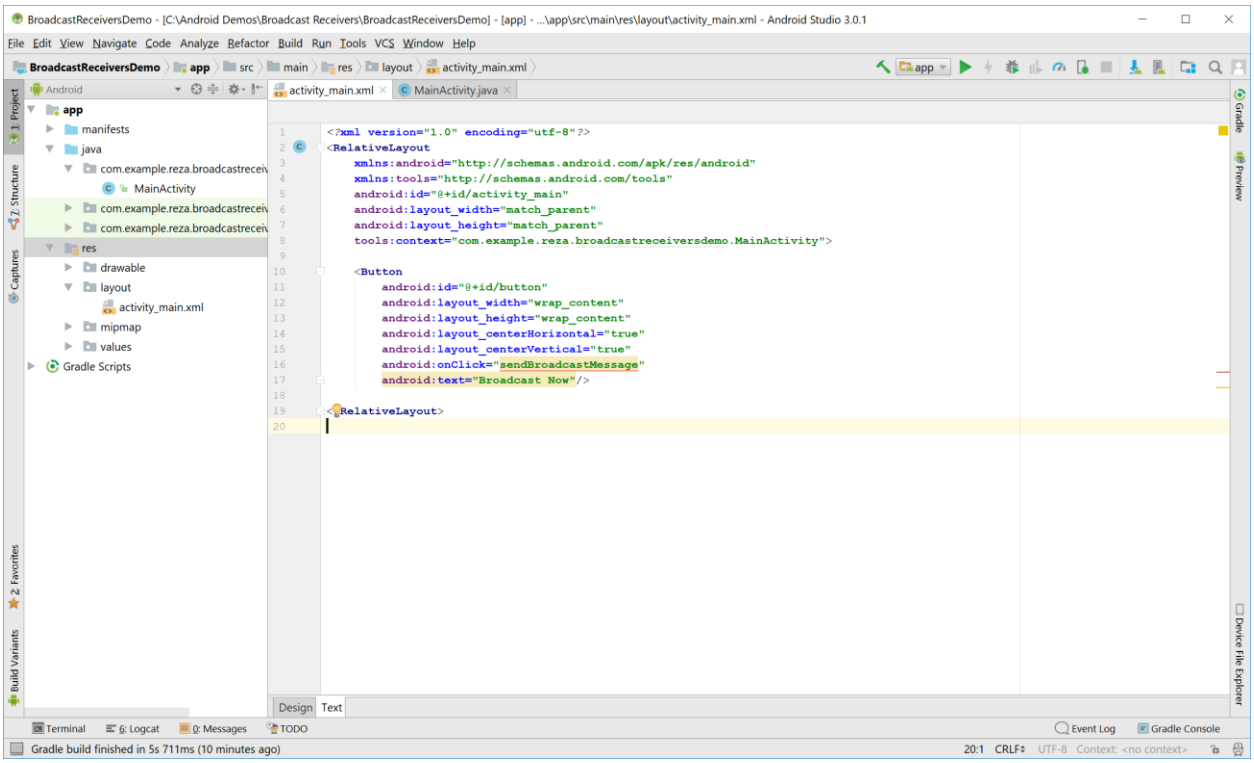

Create New Project
Configure Activity
Creates a new empty activity
Activity Name
MainActivity
Generate Layout File
Layout Name
activity_main
Backwards Compatibility (AppCompat)
Previous
Next
Cancel
Finish


BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\Mai...
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app src main java com example reza broadcastreceiversdemo MainActivity app
Android
app
manifests
java
com.example.reza.broadcastreceiv
MainActivity
com.example.reza.broadcastreceiv
com.example.reza.broadcastreceiv
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
activity_main.xml
MainActivity.java
package com.example.reza.broadcastreceiversdemo;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}
}
1: Project
7: Structure
Captures
2: Favorites
Build Variants
Gradle
Device File Explorer
Terminal
6: Logcat
0: Messages
TODO
Event Log
Gradle Console
Gradle build finished in 5s 711ms (moments ago)
14:1 CRLF UTF-8 Context: <no context>

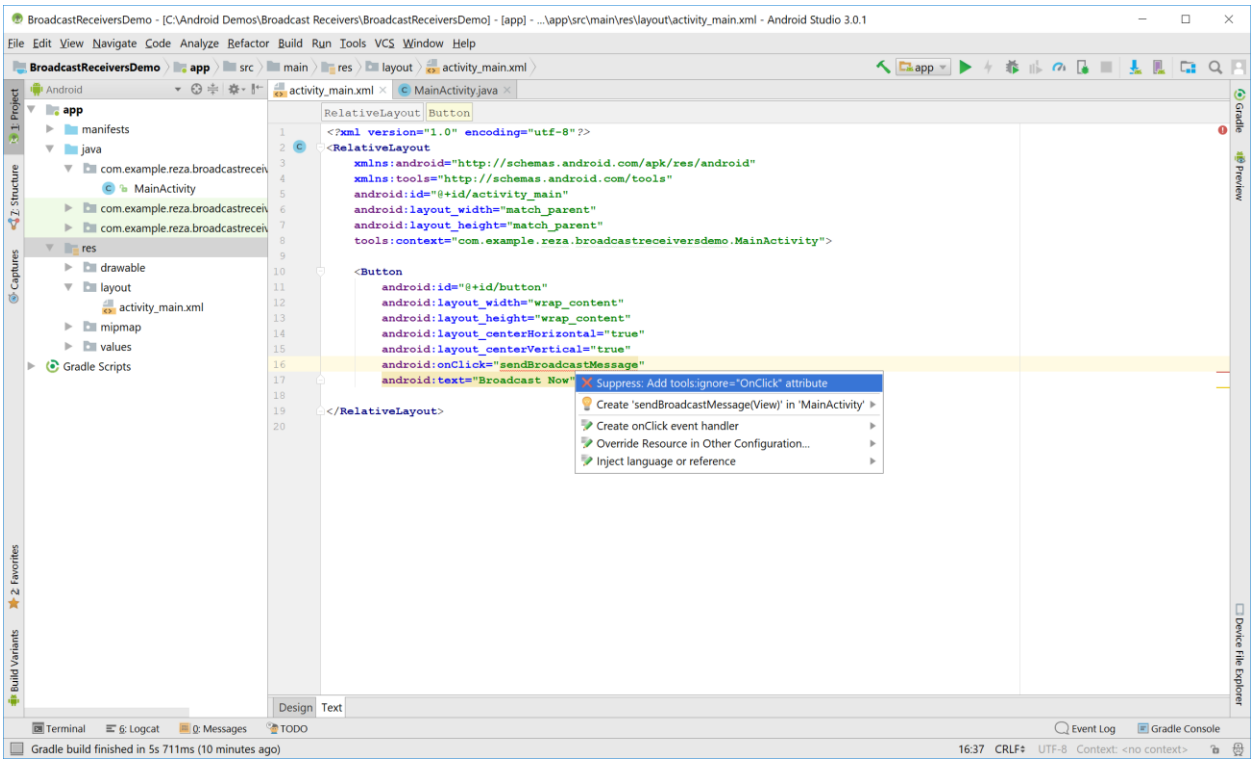

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\res\layout\activity_main.xml - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main res layout activity_main.xml
Android
app
manifests
java
com.example.reza.broadcastreceiv
MainActivity
com.example.reza.broadcastreceiv
com.example.reza.broadcastreceiv
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
activity_main.xml
MainActivity.java
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/an
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.reza.broadcastreceiversdemo.MainActivity">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World!"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
Design Text
Terminal 6: Logcat 0: Messages TODO
Event Log Gradle Console
Gradle build finished in 5s 711ms (moments ago)
1:1 CRLF UTF-8 Context: <no context>
1: Project 7: Structure Captures 2: Favorites Build Variants
Gradle Preview Device File Explorer

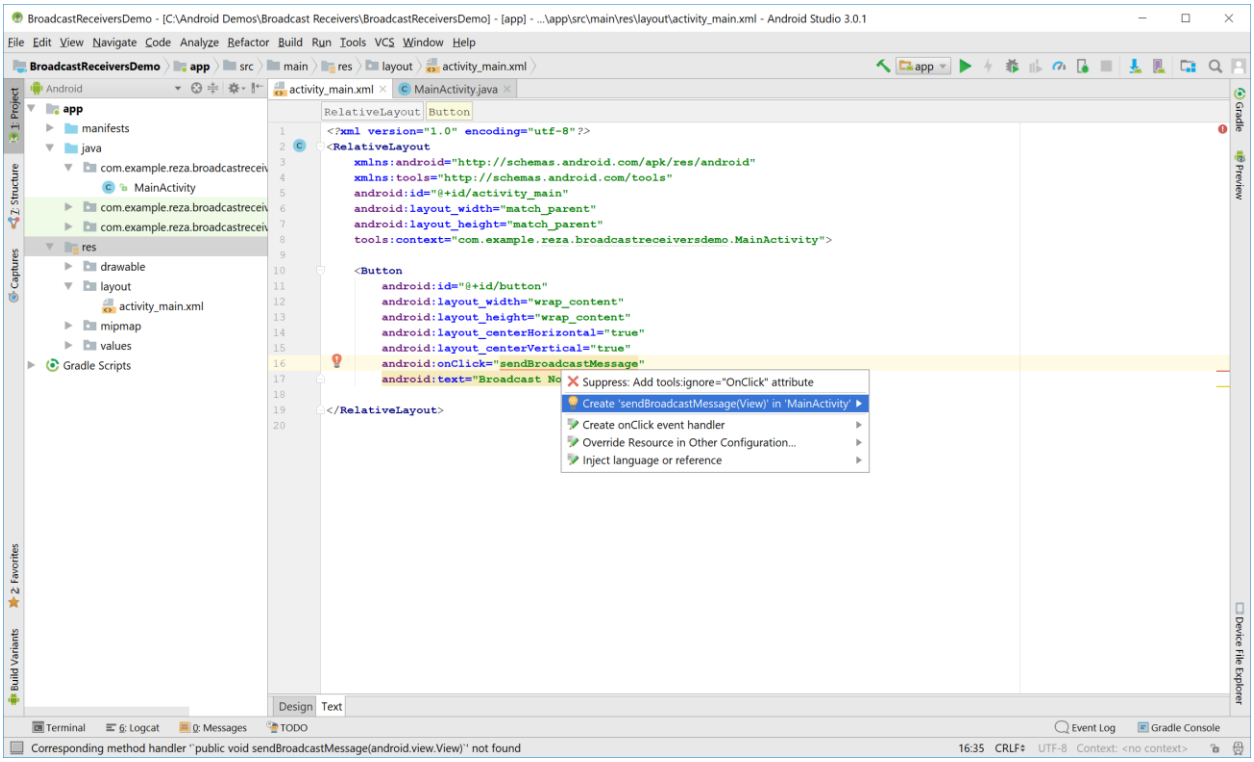

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\res\layout\activity_main.xml - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main res layout activity_main.xml
Android
app
manifests
java
com.example.reza.broadcastreceiv
MainActivity
com.example.reza.broadcastreceiv
com.example.reza.broadcastreceiv
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
activity_main.xml
MainActivity.java
Palette
All
Widgets
Text
Layouts
Containers
Images
Date
Transitions
Advanced
Google
Design
AppCompat
Button
ToggleButton
CheckBox
RadioButton
CheckedTextView
Spinner
ProgressBar
ProgressBar (Horizontal)
SeekBar
SeekBar (Discrete)
QuickContactBadge
RatingBar
Switch
Space
TextView
Component Tree
ConstraintLayout
TextView - "Hello World!"
Nexus 4
27
AppTheme
67%
8:00
Broadcast Receivers Demo
Hello World!
Design
Text
Terminal
6: Logcat
0: Messages
TODO
Event Log
Gradle Console
Gradle build finished in 5s 711ms (a minute ago)
n/a
Context: <no context>
1: Project
7: Structure
Captures
2: Favorites
Build Variants
Gradle
Attributes
Device File Explorer

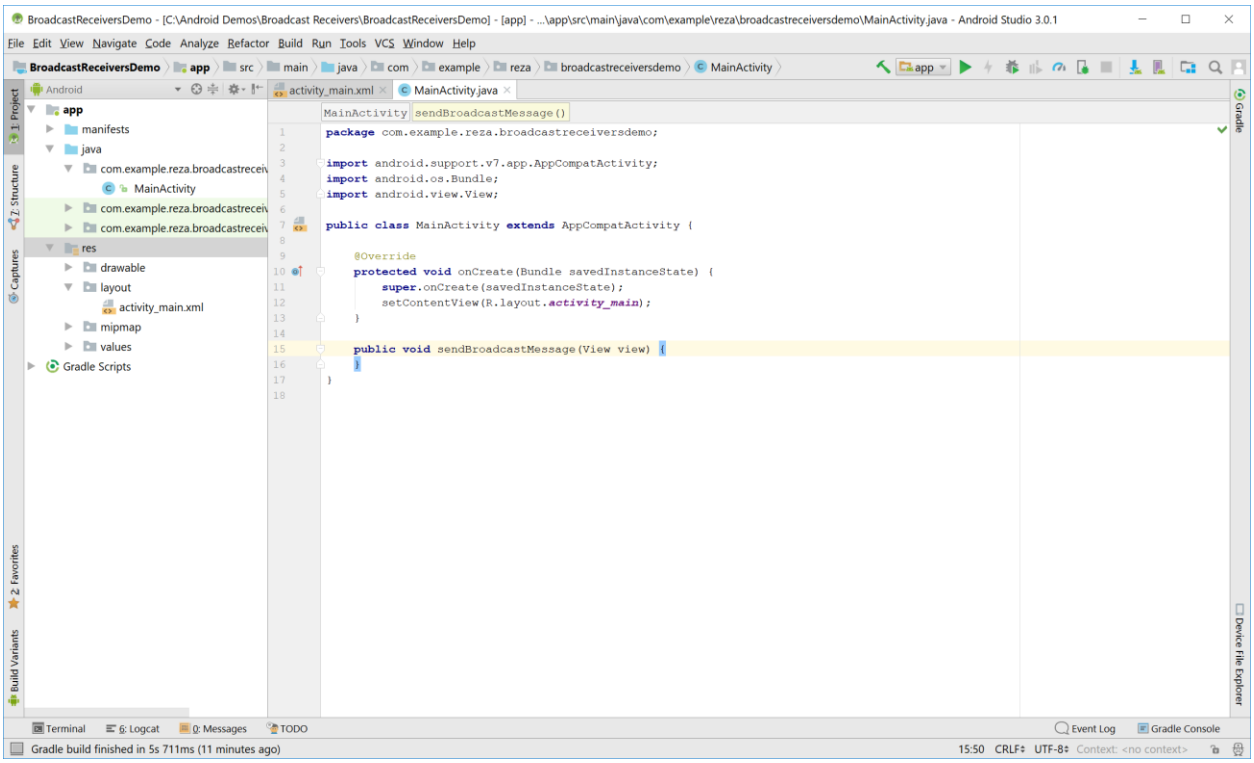
Change the activity_main.xml file as shown:



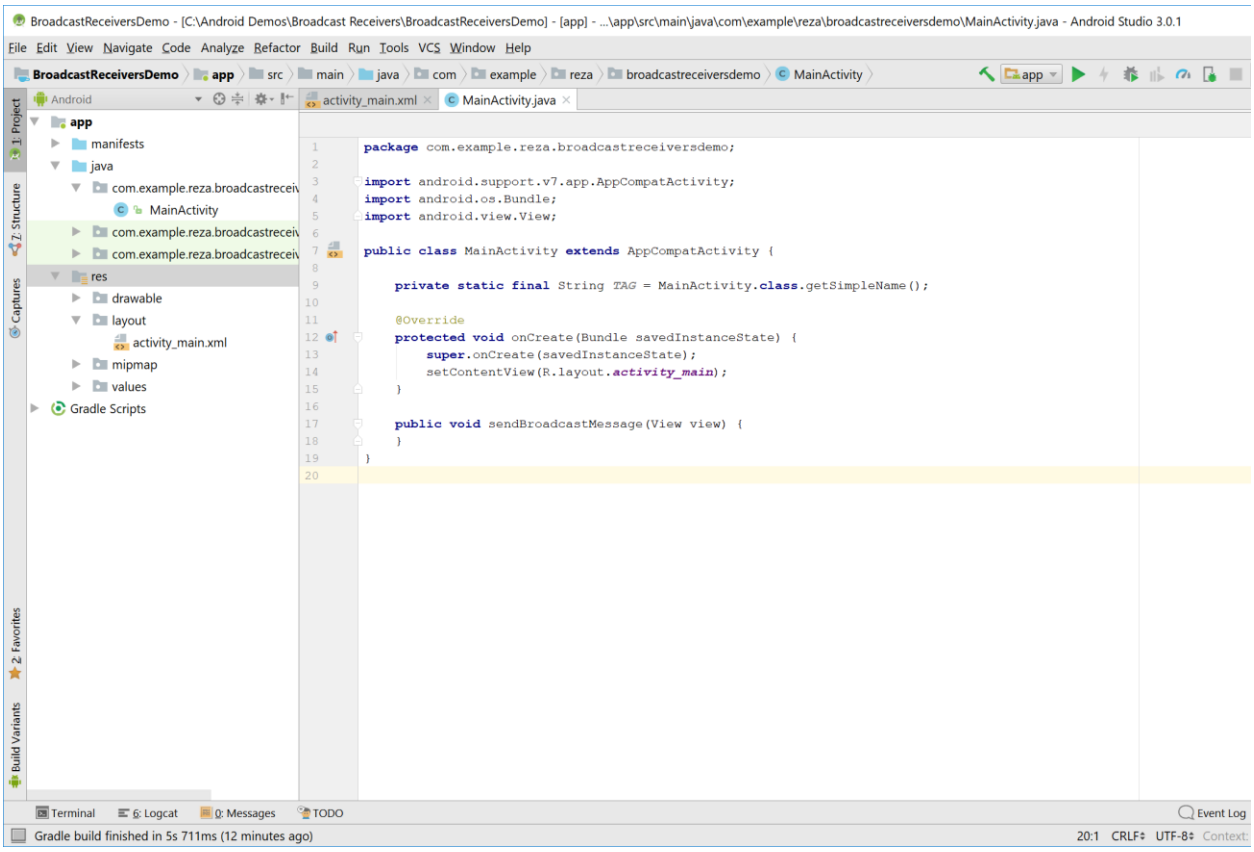
Press Alt + Enter:

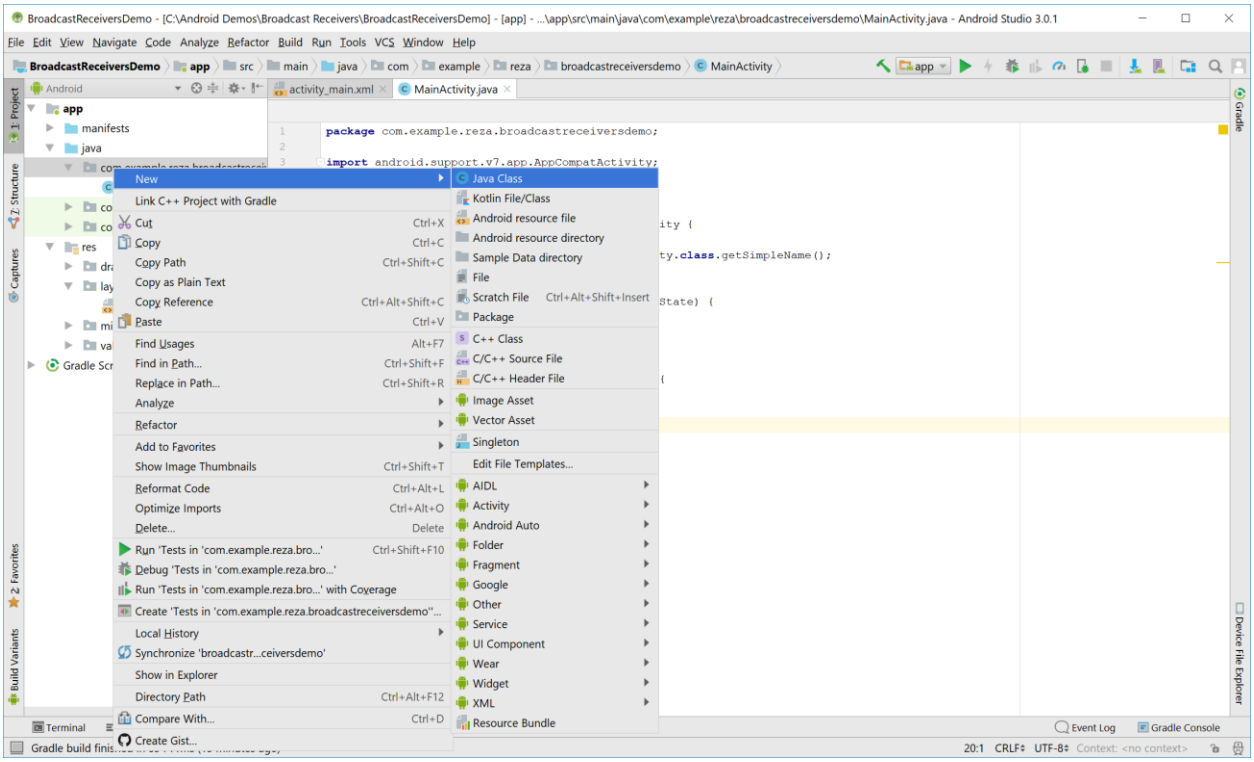


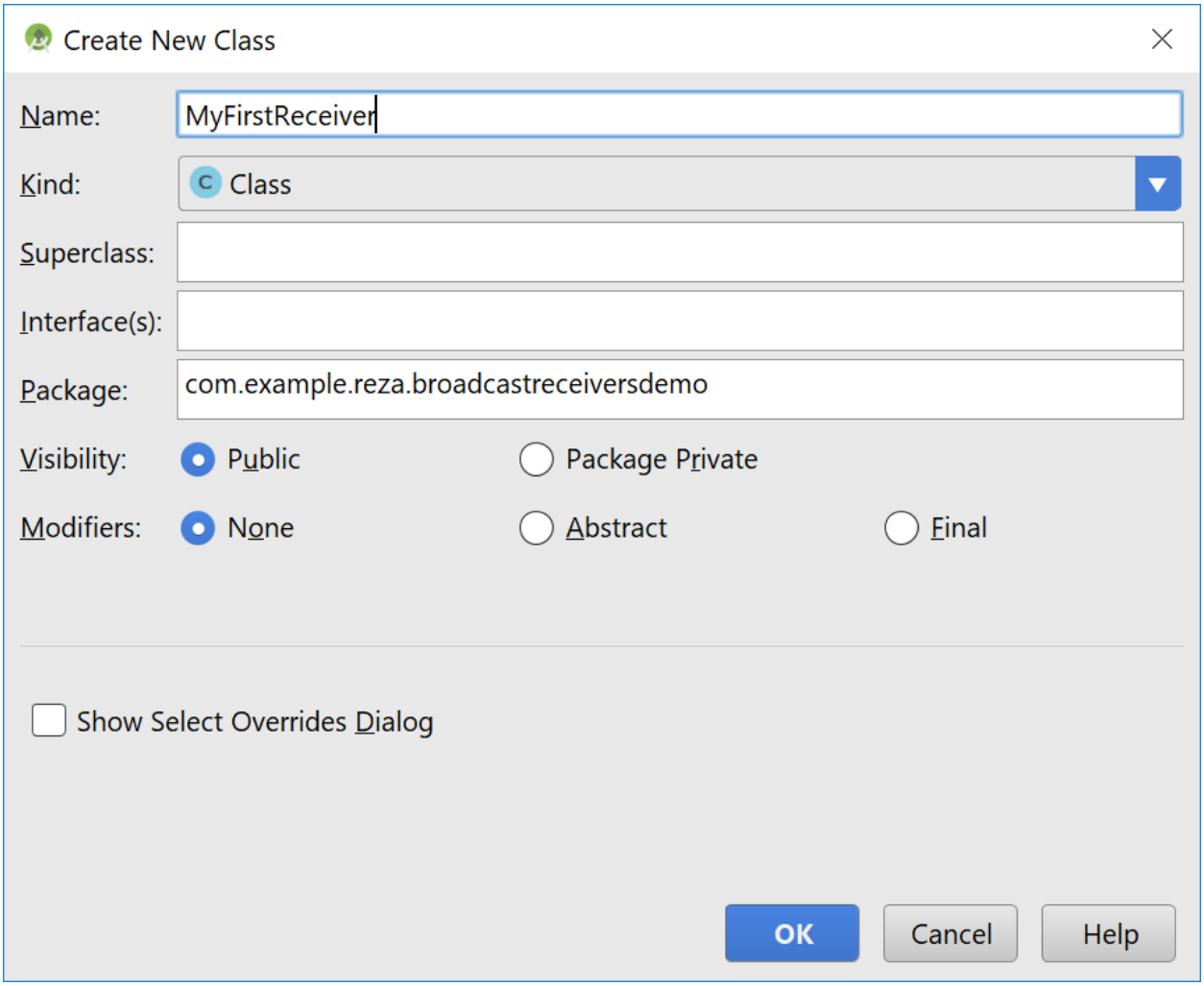

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\res\layout\activity_main.xml - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main res layout activity_main.xml
activity_main.xml MainActivity.java
RelativeLayout Button
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/activity_main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.reza.broadcastreceiversdemo.MainActivity">
<Button
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:layout_centerVertical="true"
android:onClick="sendBroadcastMessage"
android:text="Broadcast No
</RelativeLayout>
Suppress: Add tools:ignore="OnClick" attribute
Create 'sendBroadcastMessage(View)' in 'MainActivity'
Create onClick event handler
Override Resource in Other Configuration...
Inject language or reference
Design Text
Terminal 6: Logcat 0: Messages TODO
Event Log Gradle Console
Corresponding method handler ''public void sendBroadcastMessage(android.view.View)'' not found
16:35 CRLF UTF-8 Context: <no context>

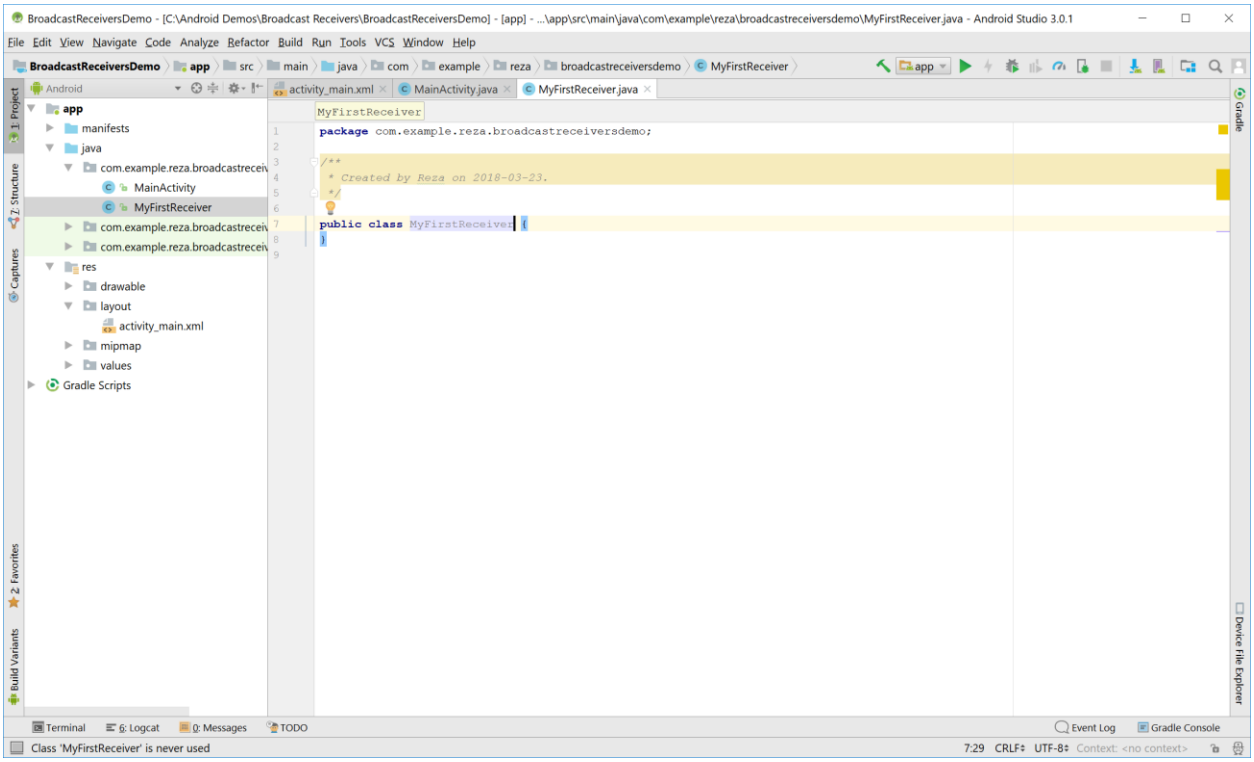

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\MainActivity.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main java com example reza broadcastreceiversdemo MainActivity
activity_main.xml MainActivity.java
MainActivity sendBroadcastMessage()
package com.example.reza.broadcastreceiversdemo;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}
public void sendBroadcastMessage(View view) {
}
}
Terminal 6: Logcat 0: Messages TODO
Event Log Gradle Console
Gradle build finished in 5s 711ms (11 minutes ago)
15:50 CRLF UTF-8 Context: <no context>

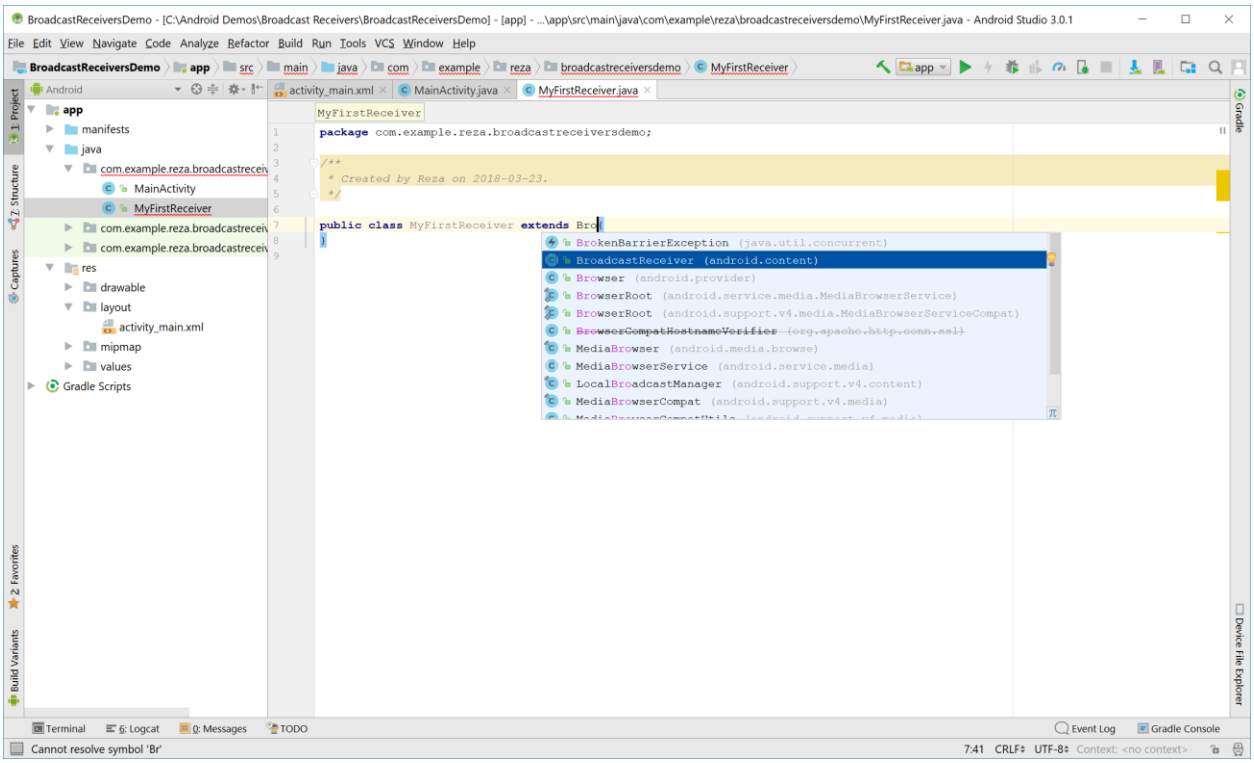
Add line 9 as shown:

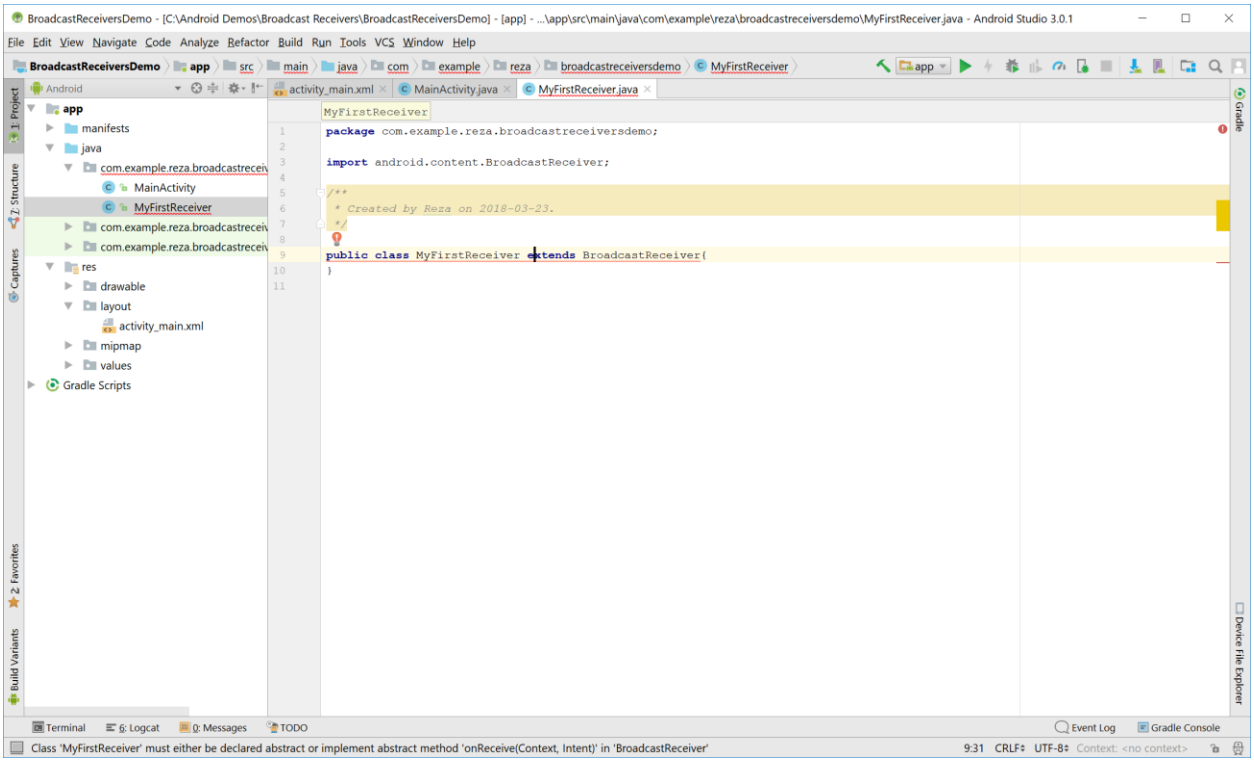




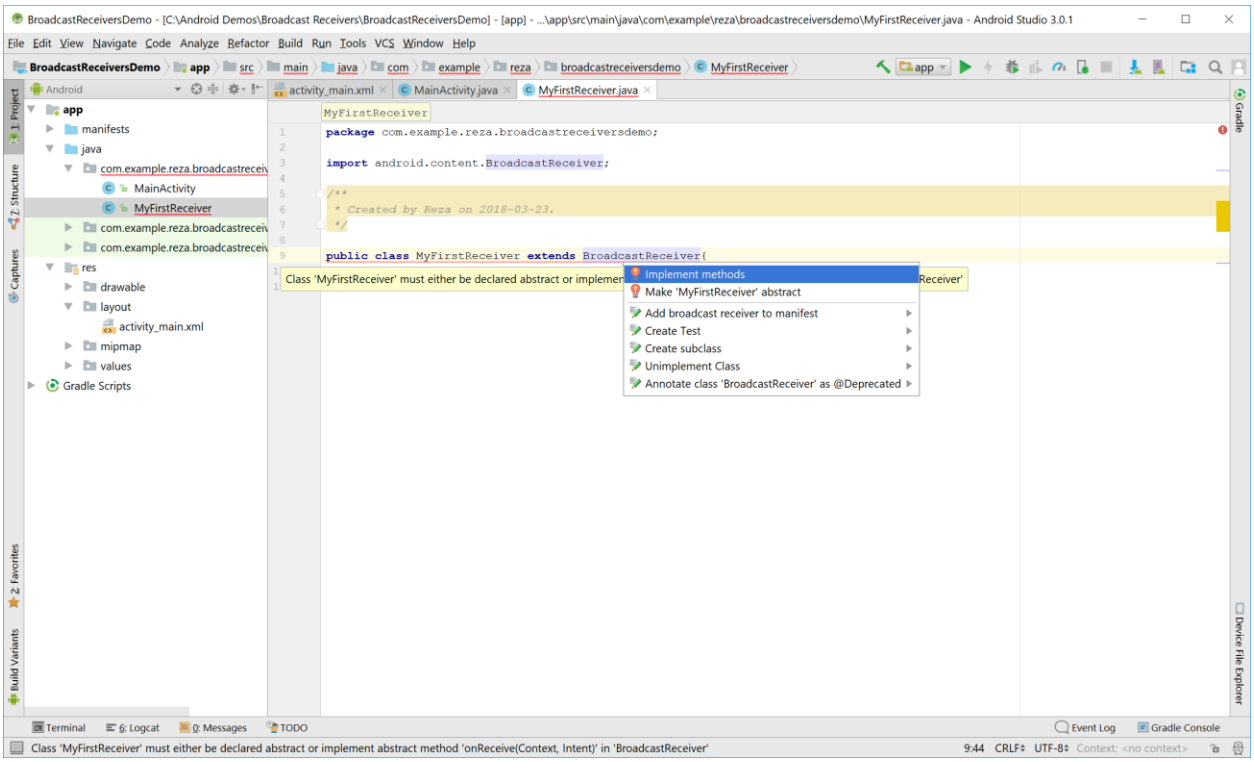

Create New Class
Name:
MyFirstReceiver
Kind:
Class
Superclass:
Interface(s):
Package:
com.example.reza.broadcastreceiversdemo
Visibility:
Public
Package Private
Modifiers:
None
Abstract
Final
Show Select Overrides Dialog
OK
Cancel
Help

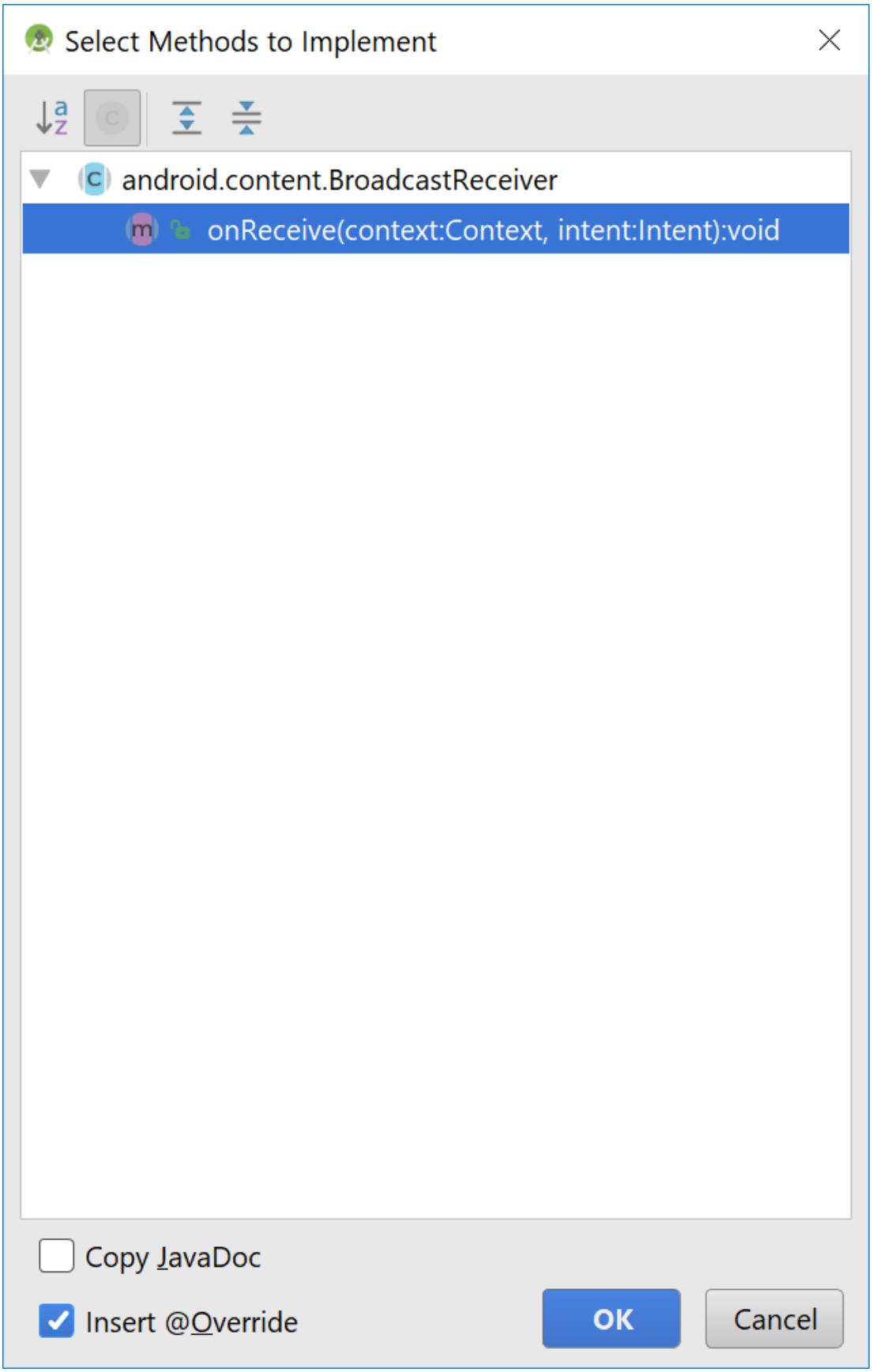

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\MyFirstReceiver.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MyFirstReceiver
package com.example.reza.broadcastreceiversdemo;
/**
* Created by Reza on 2018-03-23.
*/
public class MyFirstReceiver {
}
Class 'MyFirstReceiver' is never used

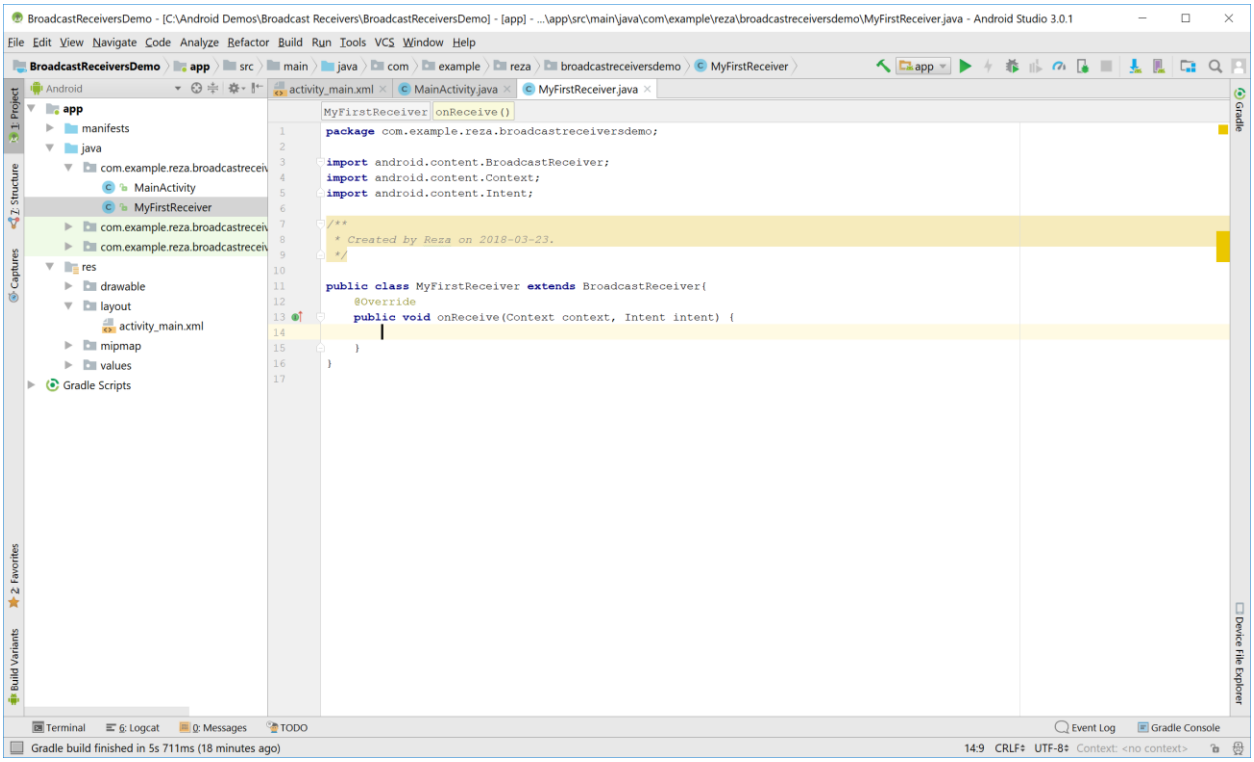

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\MyFirstReceiver.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MyFirstReceiver
package com.example.reza.broadcastreceiversdemo;
/**
* Created by Reza on 2018-03-23.
*/
public class MyFirstReceiver extends Bro
BrokenBarrierException (java.util.concurrent)
BroadcastReceiver (android.content)
Browser (android.provider)
BrowserRoot (android.service.media.MediaBrowserService)
BrowserRoot (android.support.v4.media.MediaBrowserServiceCompat)
MediaBrowser (android.media.browse)
MediaBrowserService (android.service.media)
LocalBroadcastManager (android.support.v4.content)
MediaBrowserCompat (android.support.v4.media)
Cannot resolve symbol 'Br'

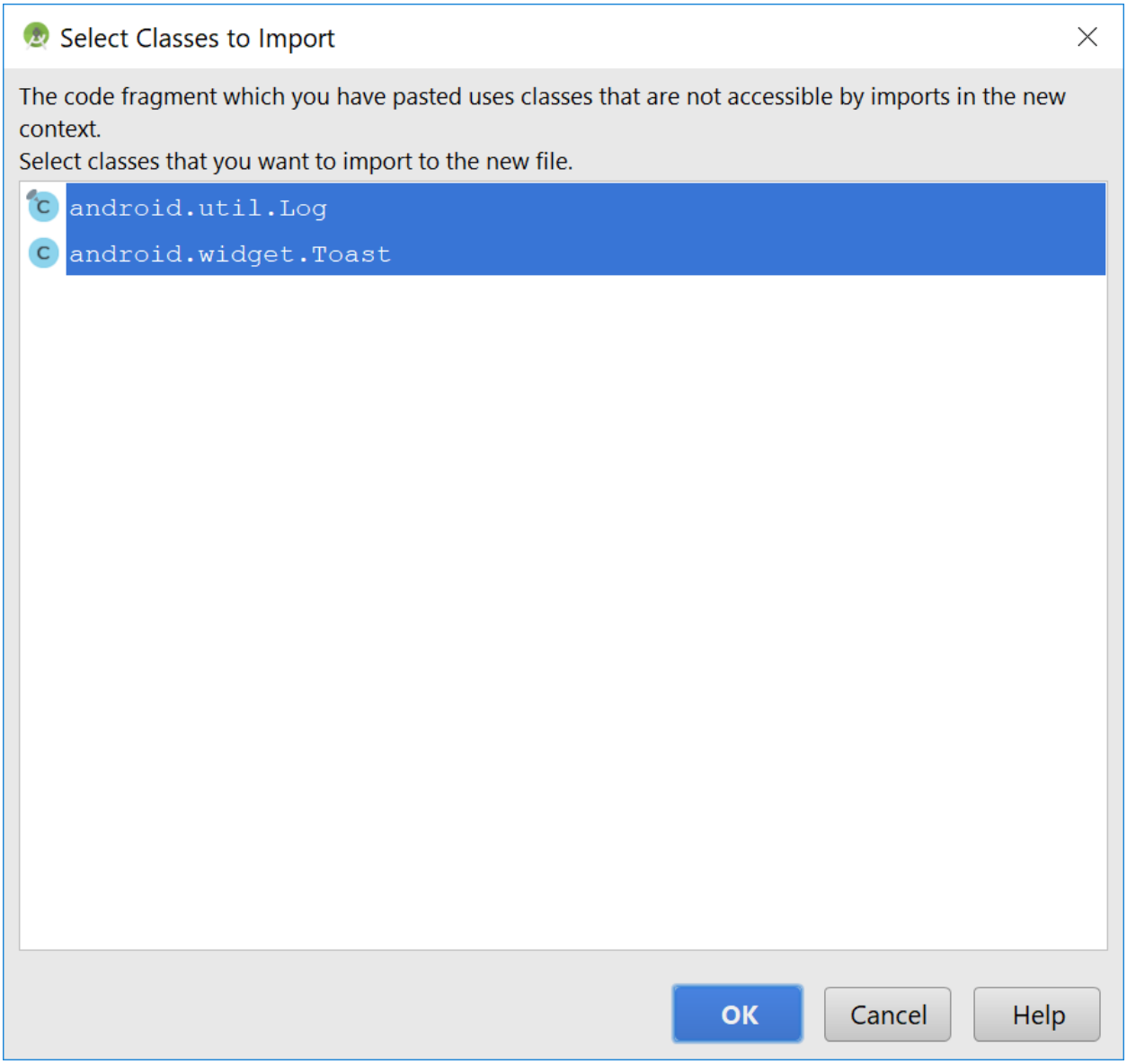


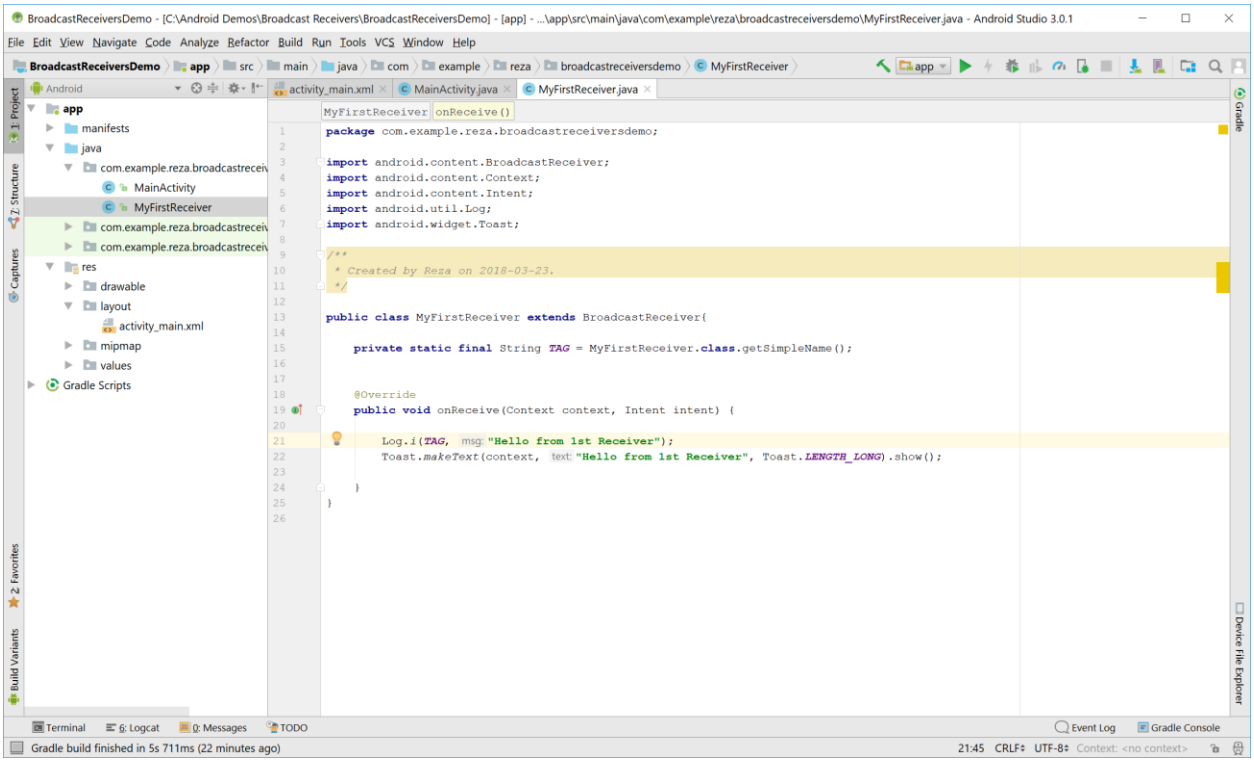
Press Alt + Enter:



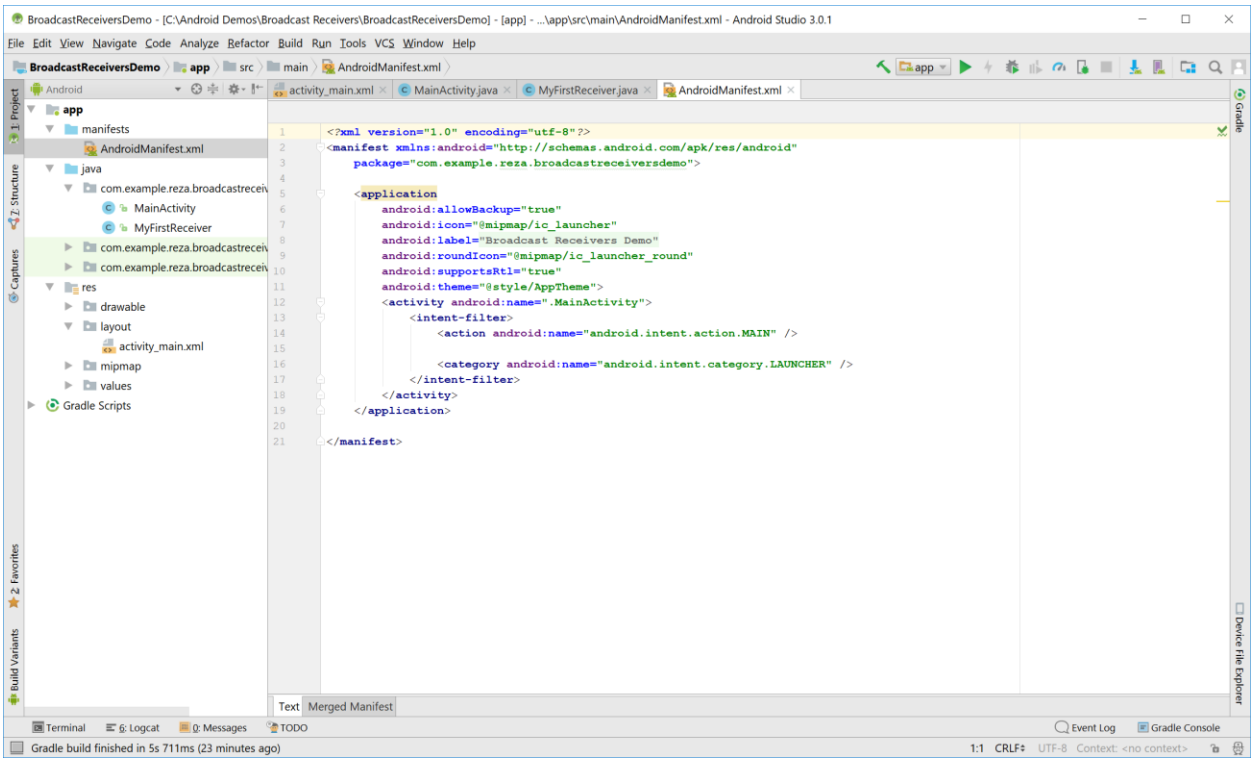

Select Methods to Implement
android.content.BroadcastReceiver
onReceive(context:Context, intent:Intent):void
Copy JavaDoc
Insert @Override
OK
Cancel


BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\MyFirstReceiver.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main java com example reza broadcastreceiversdemo MyFirstReceiver
activity_main.xml MainActivity.java MyFirstReceiver.java
MyFirstReceiver onReceive()
app
manifests
java
com.example.reza.broadcastreceiv
MainActivity
MyFirstReceiver
com.example.reza.broadcastreceiv
com.example.reza.broadcastreceiv
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
package com.example.reza.broadcastreceiversdemo;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
/**
* Created by Reza on 2018-03-23.
*/
public class MyFirstReceiver extends BroadcastReceiver{
@Override
public void onReceive(Context context, Intent intent) {
}
}
Terminal 6: Logcat 0: Messages TODO
Event Log Gradle Console
Gradle build finished in 5s 711ms (18 minutes ago)
14:9 CRLF UTF-8 Context: <no context>

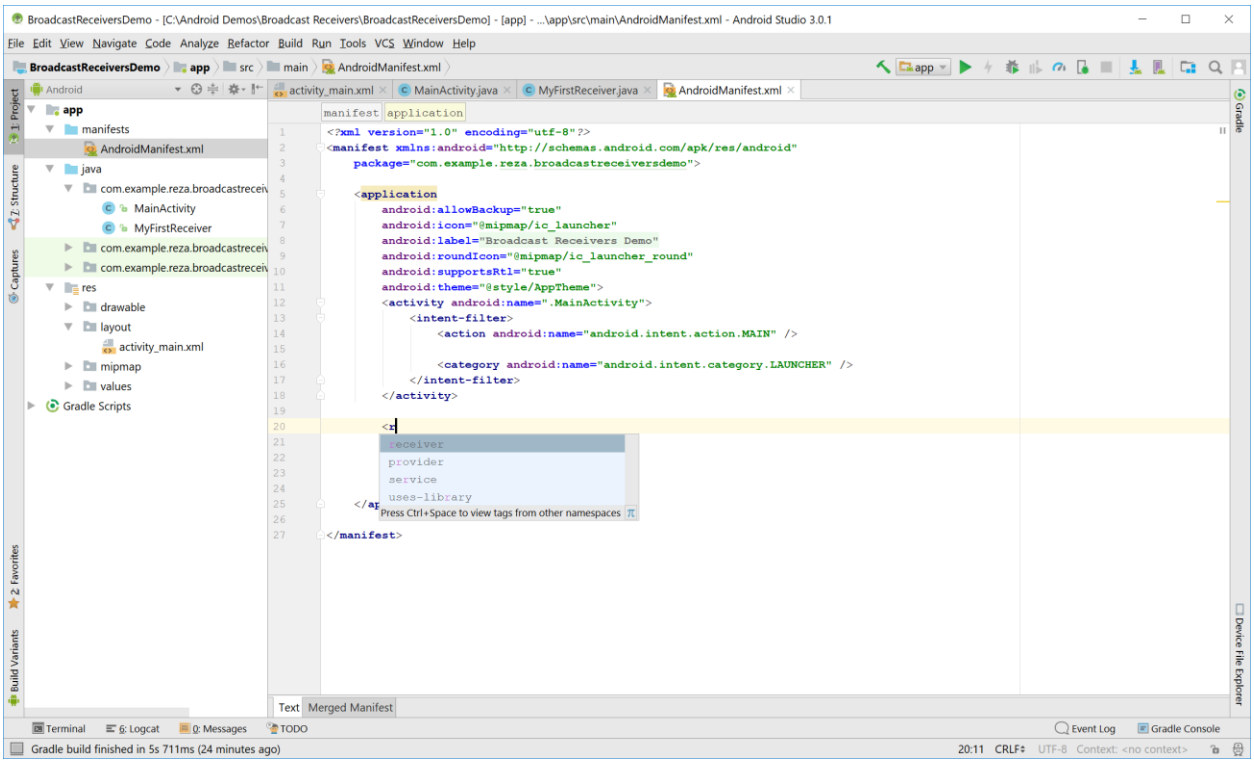

Select Classes to Import
The code fragment which you have pasted uses classes that are not accessible by imports in the new context.
Select classes that you want to import to the new file.
android.util.Log
android.widget.Toast
OK
Cancel
Help

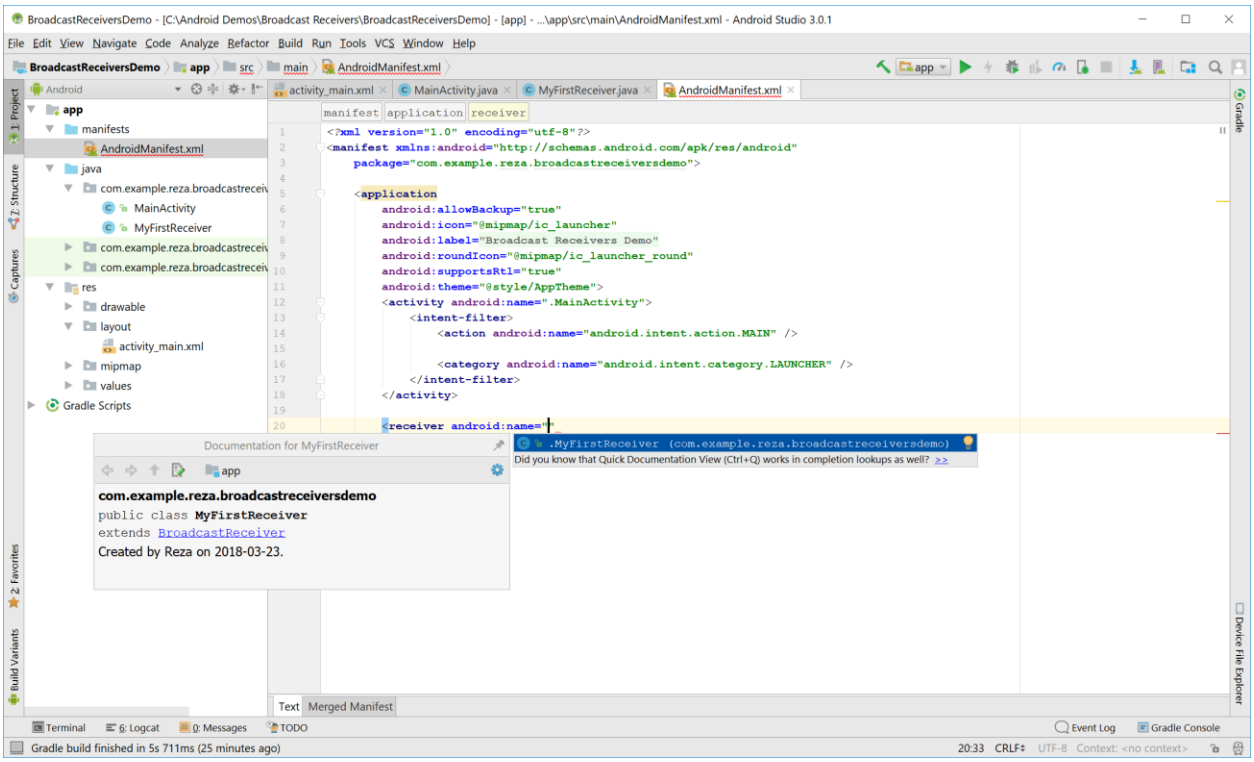


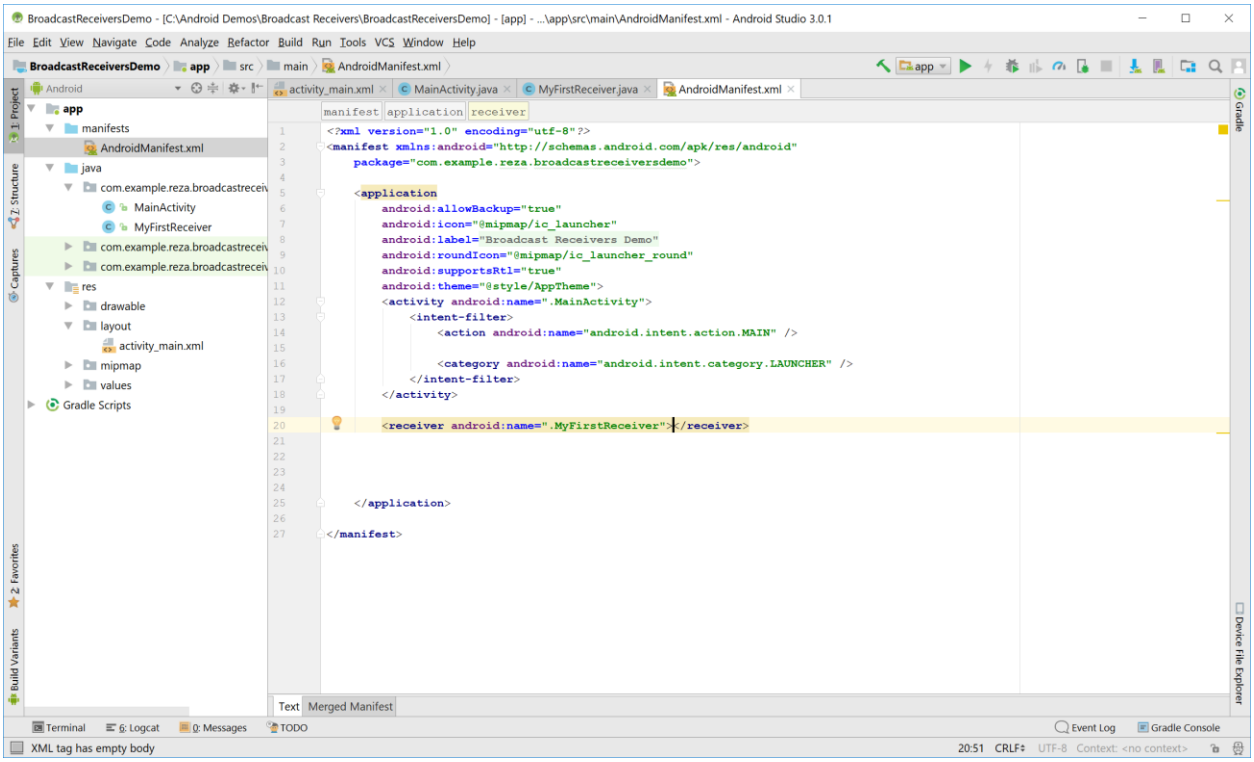
Your current manifest file:

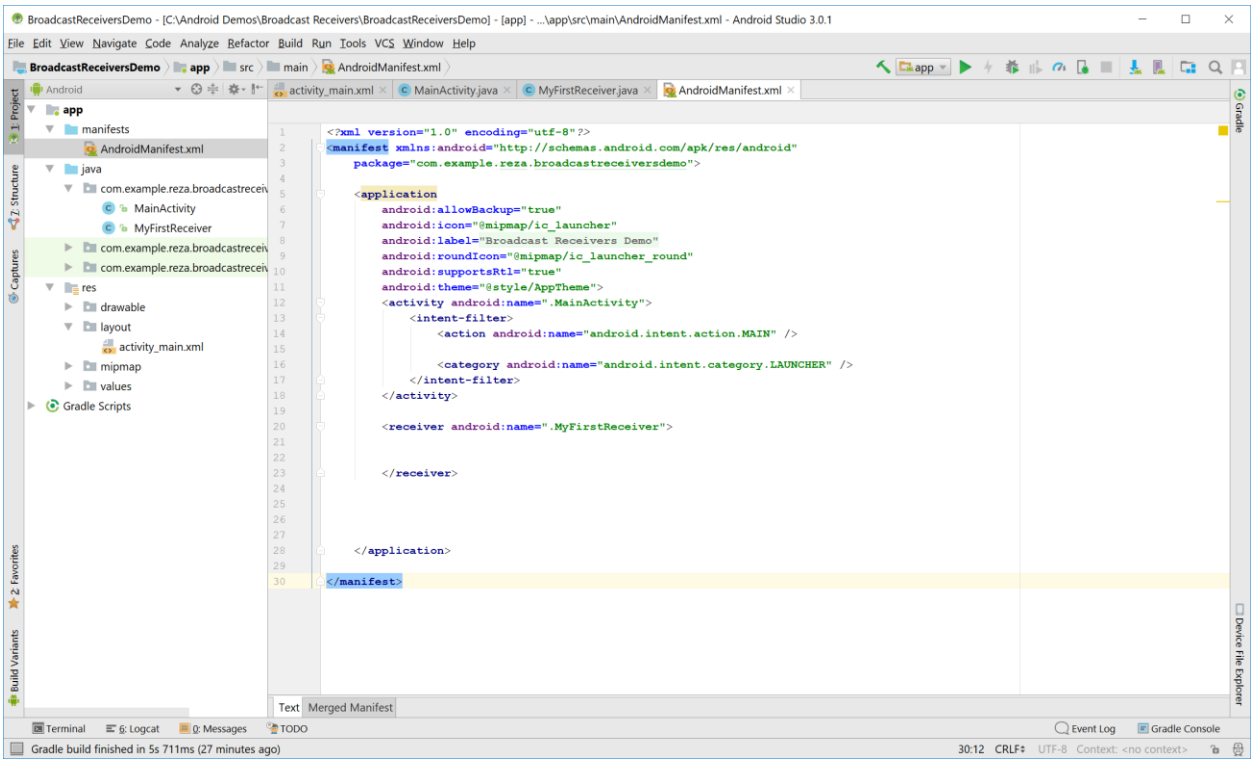


Modify the file as shown:

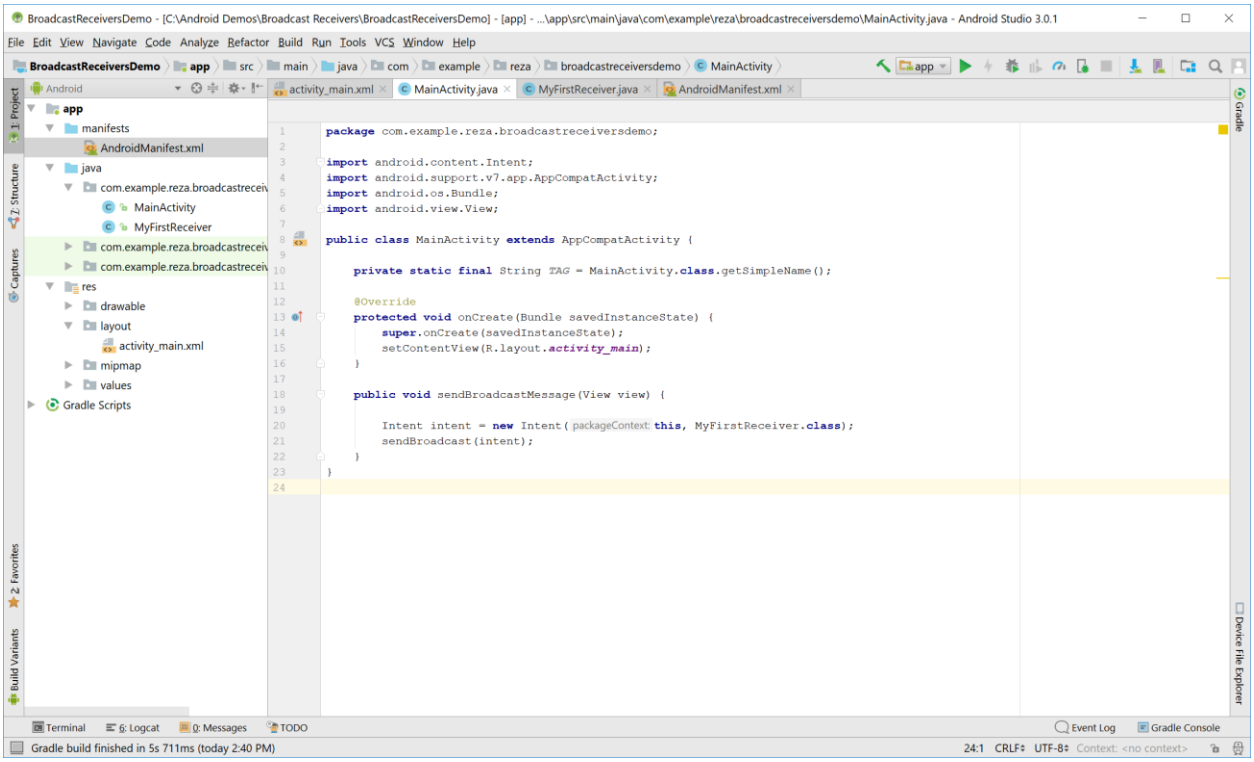




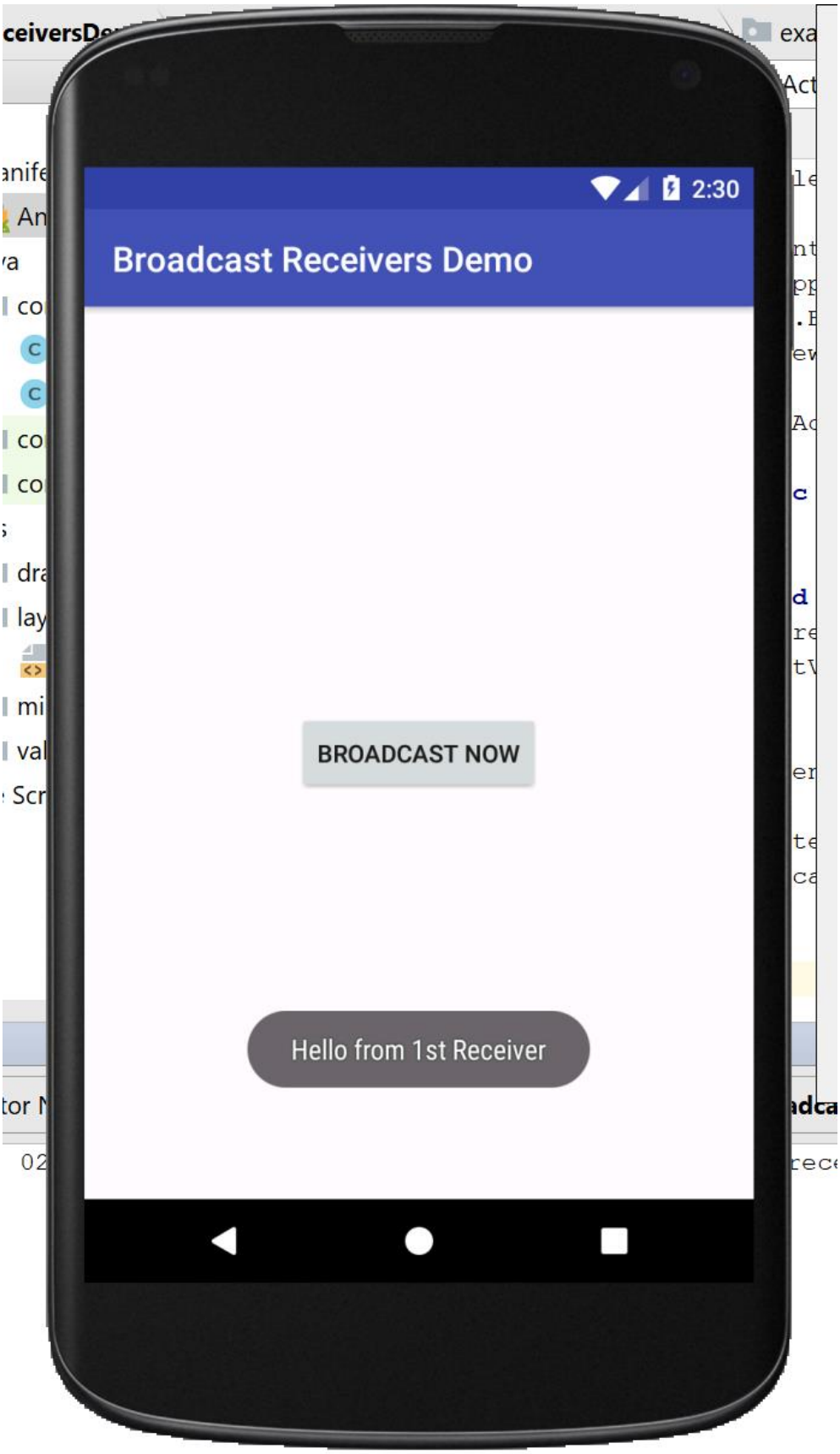

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\AndroidManifest.xml - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main AndroidManifest.xml
activity_main.xml MainActivity.java MyFirstReceiver.java AndroidManifest.xml
manifest application receiver
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.reza.broadcastreceiversdemo">
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="Broadcast Receivers Demo"
android:roundIcon="@mipmap/ic_launcher_round"
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>
<receiver android:name=".MyFirstReceiver"></receiver>
</application>
</manifest>
Text Merged Manifest
Terminal 6: Logcat 0: Messages TODO
Event Log Gradle Console
XML tag has empty body
20:51 CRLF UTF-8 Context: <no context>

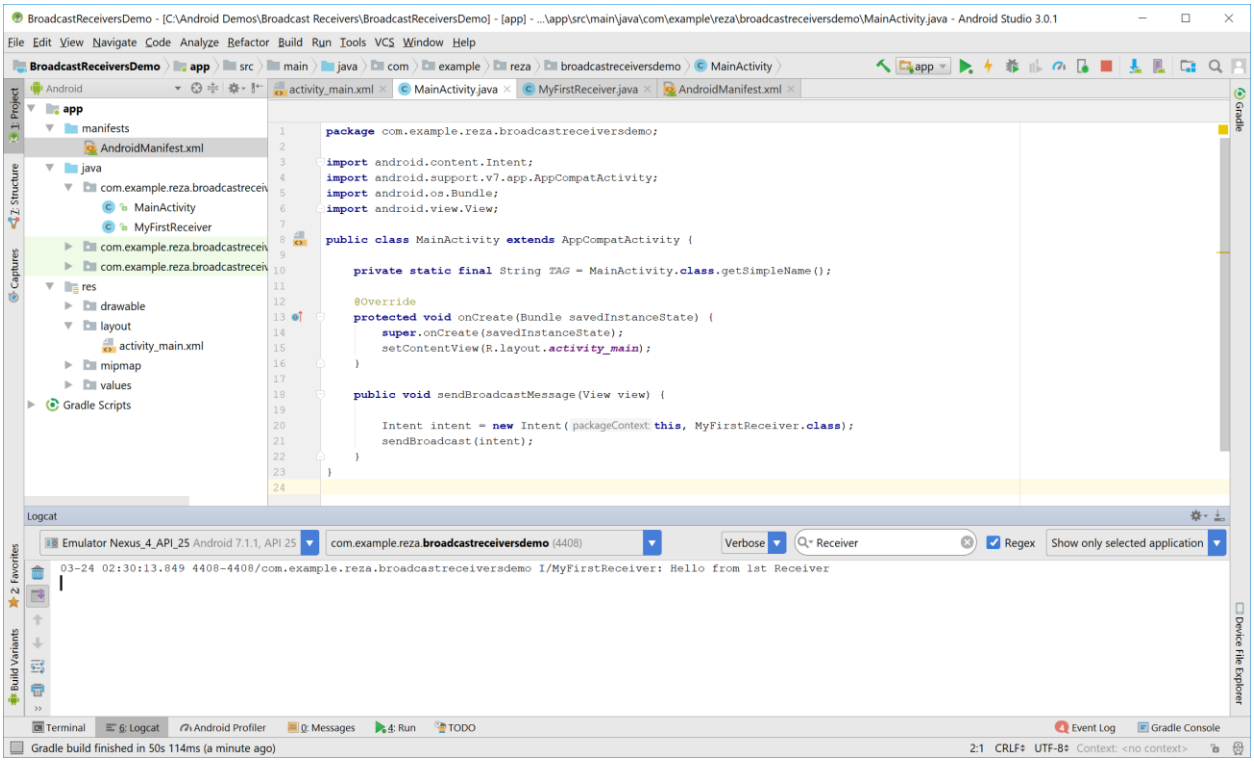

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\AndroidManifest.xml - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main AndroidManifest.xml
activity_main.xml MainActivity.java MyFirstReceiver.java AndroidManifest.xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.reza.broadcastreceiversdemo">
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="Broadcast Receivers Demo"
android:roundIcon="@mipmap/ic_launcher_round"
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>
<receiver android:name=".MyFirstReceiver">
</receiver>
</application>
</manifest>
Text Merged Manifest
Terminal 6: Logcat 0: Messages TODO
Event Log Gradle Console
Gradle build finished in 5s 711ms (27 minutes ago)
30:12 CRLF UTF-8 Context: <no context>

Now you need to write code to send a Broadcast from your activity to your receiver and to do that you need to write your code in sendBroadcastMessage() method in the main activity:



Now run the application:

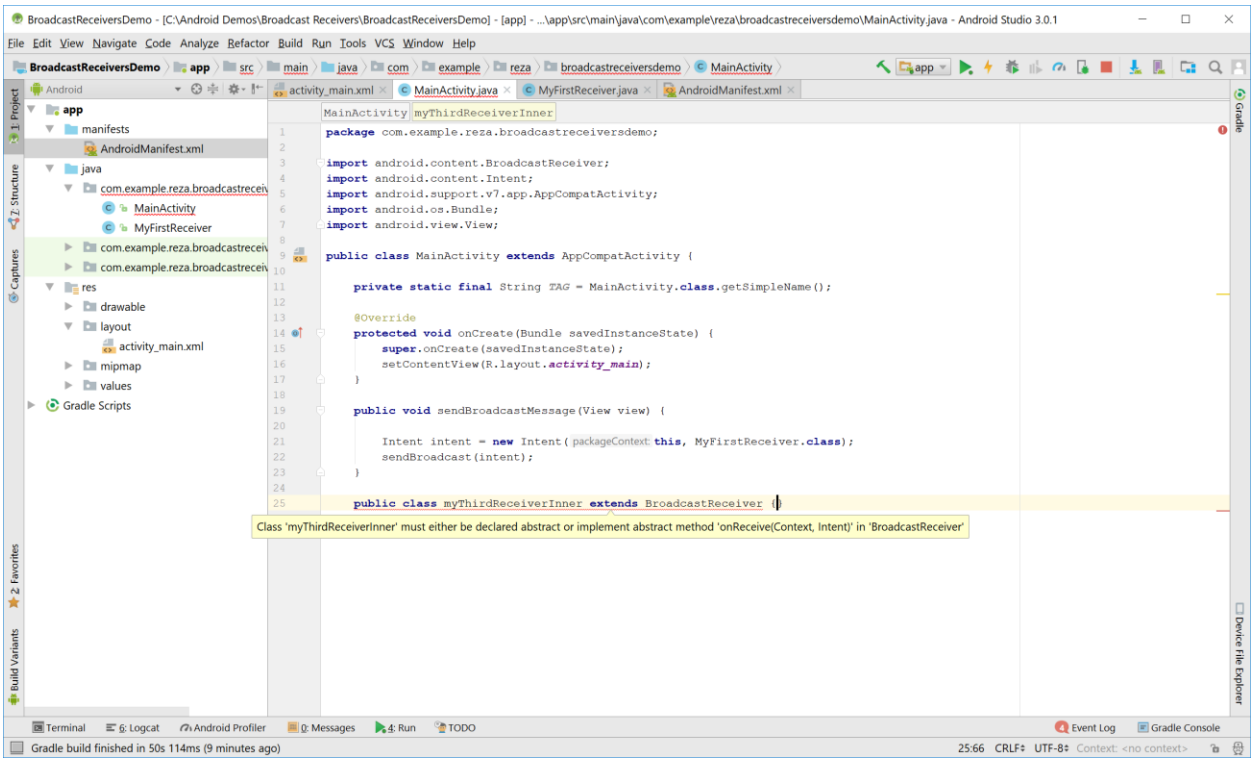

2:30
Broadcast Receivers Demo
BROADCAST NOW
Hello from 1st Receiver



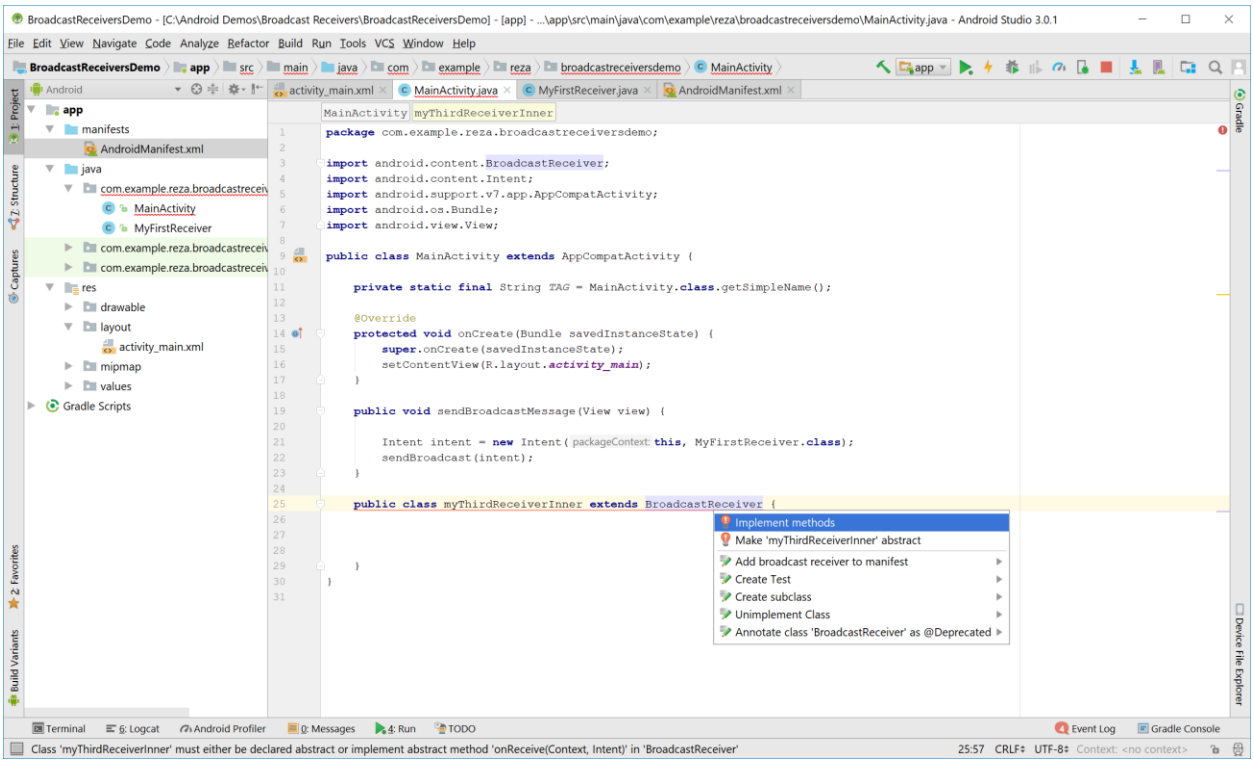
So everything is working.

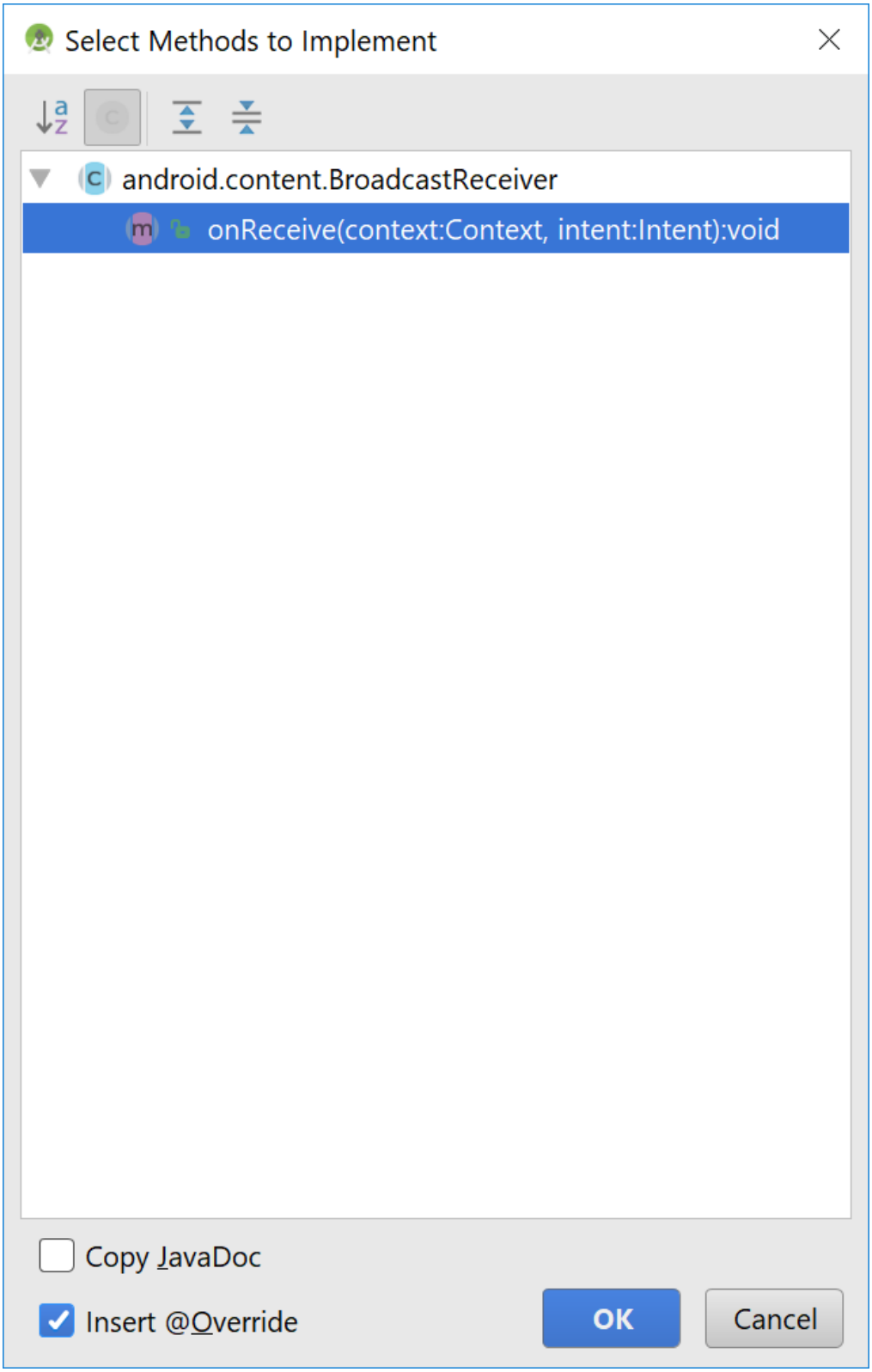
Declaring the Broadcast as an inner class

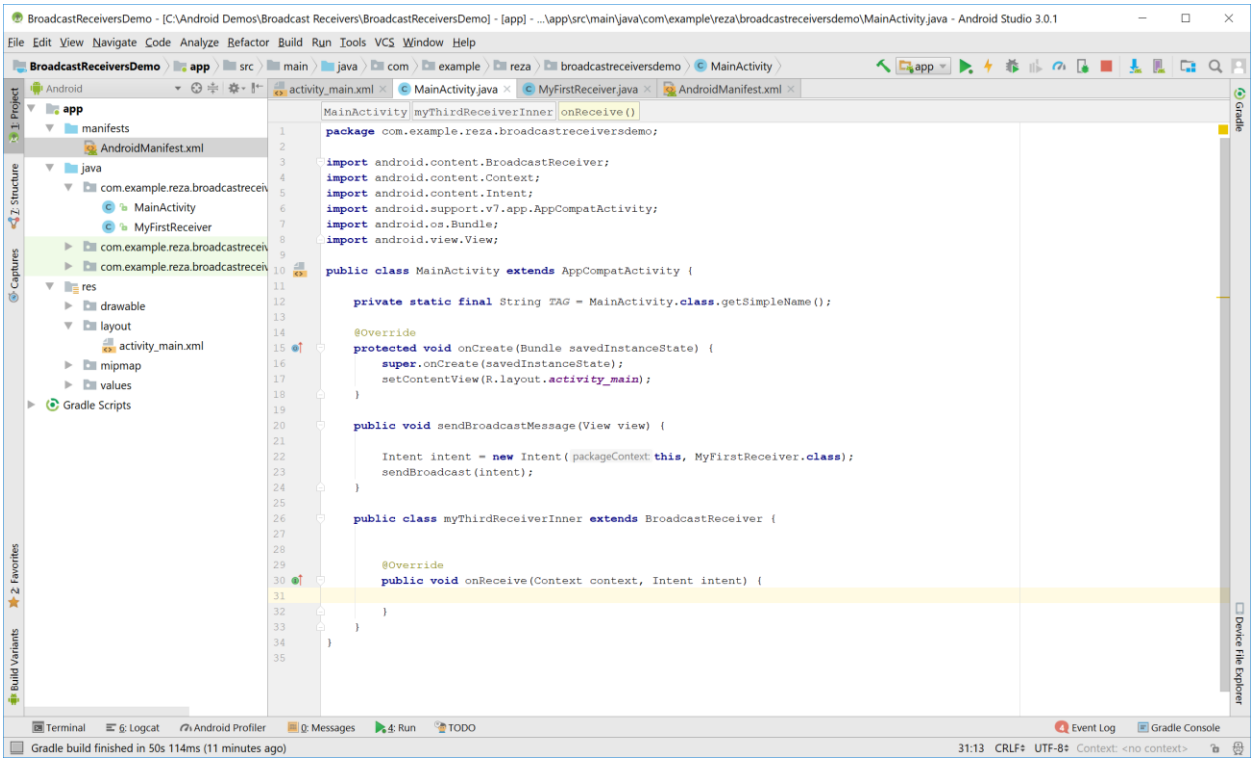
You can also declare your receiver inside the activity as shown below:



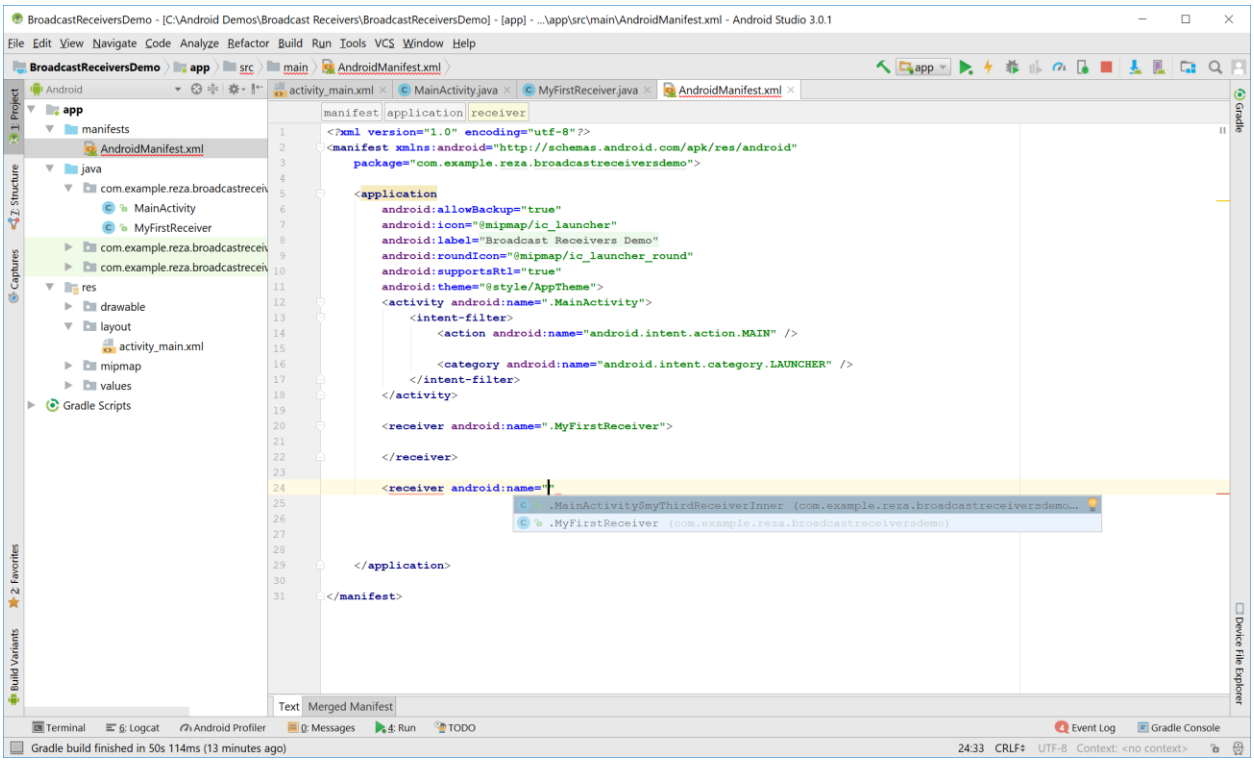
Press Alt + Enter:

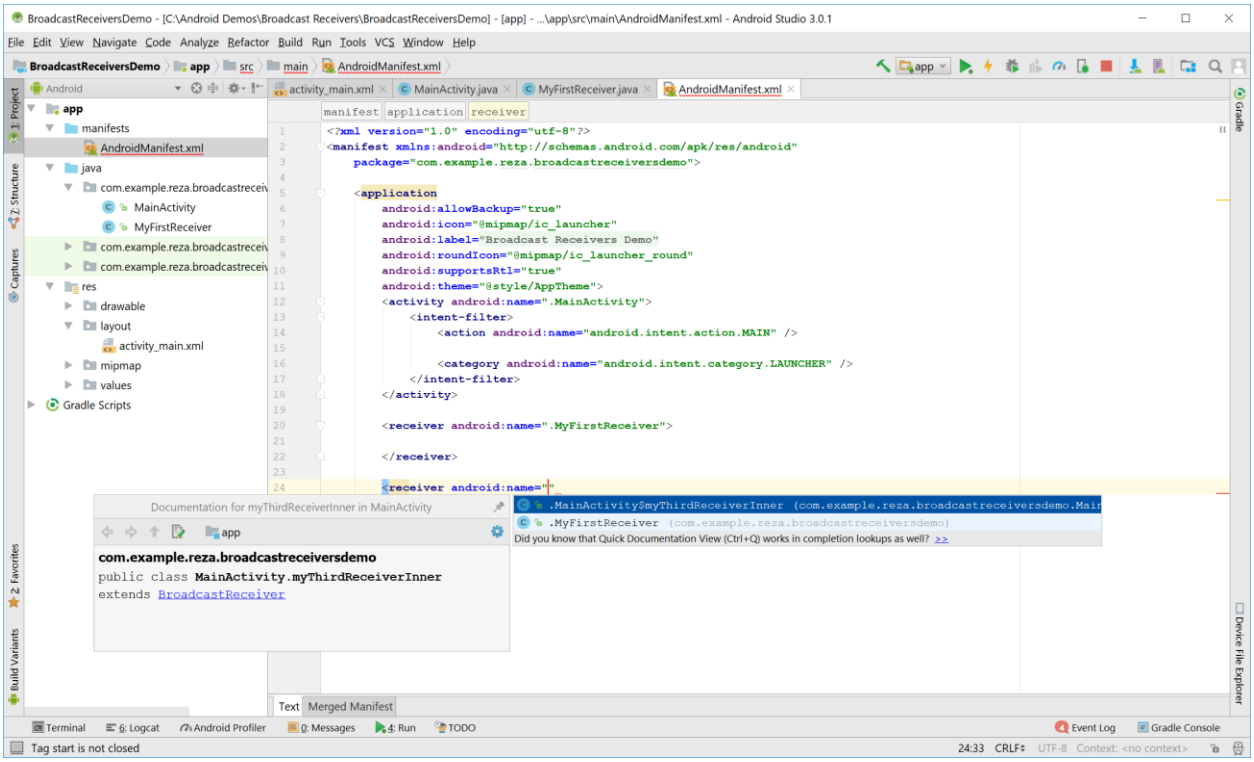



Select Methods to Implement
android.content.BroadcastReceiver
onReceive(context:Context, intent:Intent):void
Copy JavaDoc
Insert @Override
OK
Cancel

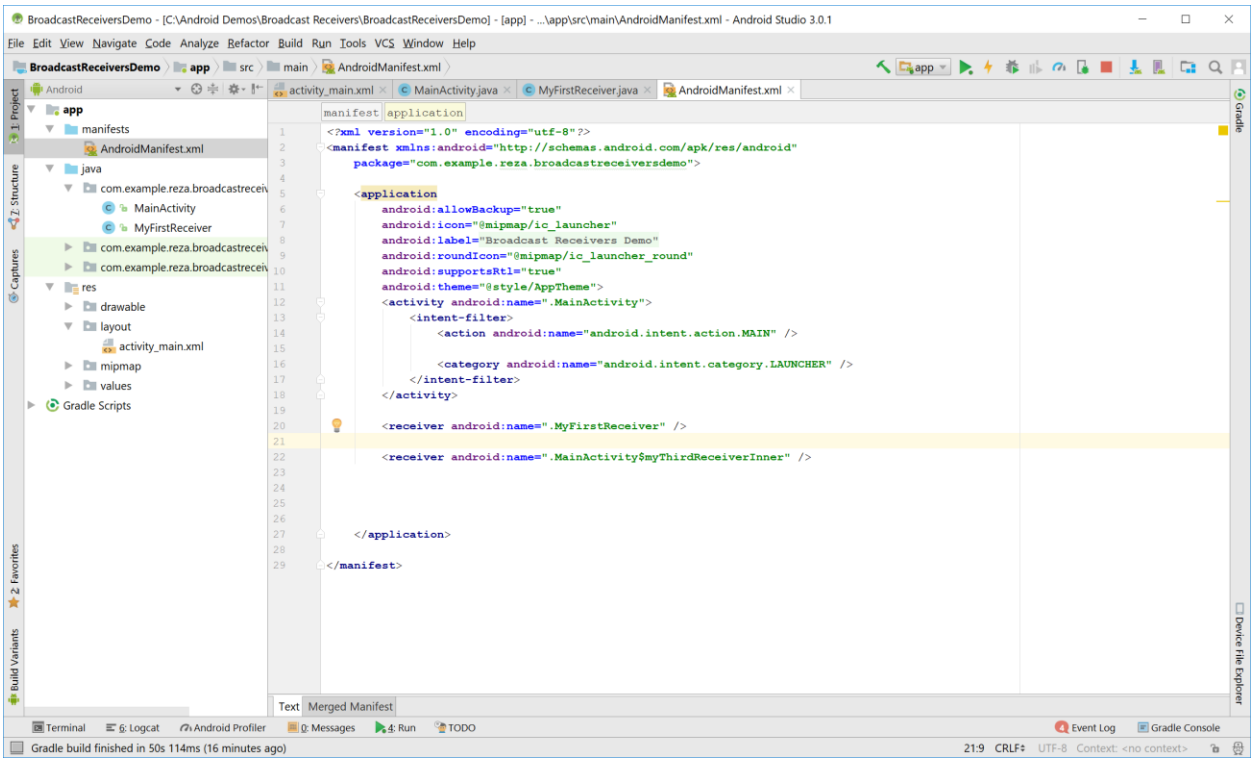


Now go to the manifest file and declare your new receiver as shown:

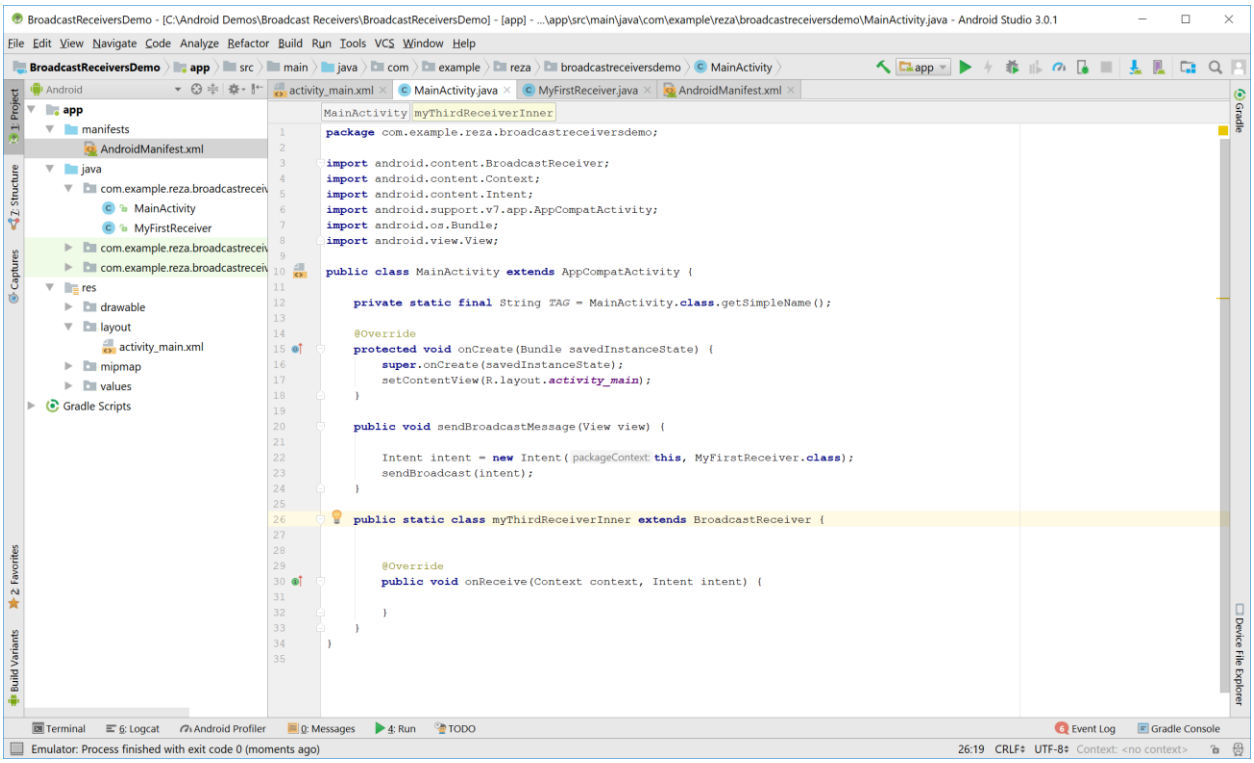



BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\AndroidManifest.xml - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main AndroidManifest.xml
activity_main.xml MainActivity.java MyFirstReceiver.java AndroidManifest.xml
manifest application receiver
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.reza.broadcastreceiversdemo">
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="Broadcast Receivers Demo"
android:roundIcon="@mipmap/ic_launcher_round"
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>
<receiver android:name=".MyFirstReceiver">
</receiver>
<receiver android:name="
.MainActivity$myThirdReceiverInner (com.example.reza.broadcastreceiversdemo.Mai
.MyFirstReceiver (com.example.reza.broadcastreceiversdemo)
Did you know that Quick Documentation View (Ctrl+Q) works in completion lookups as well? >>
Documentation for myThirdReceiverInner in MainActivity
app
com.example.reza.broadcastreceiversdemo
public class MainActivity.myThirdReceiverInner
extends BroadcastReceiver
Text Merged Manifest
Terminal 6: Logcat Android Profiler 0: Messages 4: Run TODO Event Log Gradle Console
Tag start is not closed
24:33 CRLF UTF-8 Context: <no context>

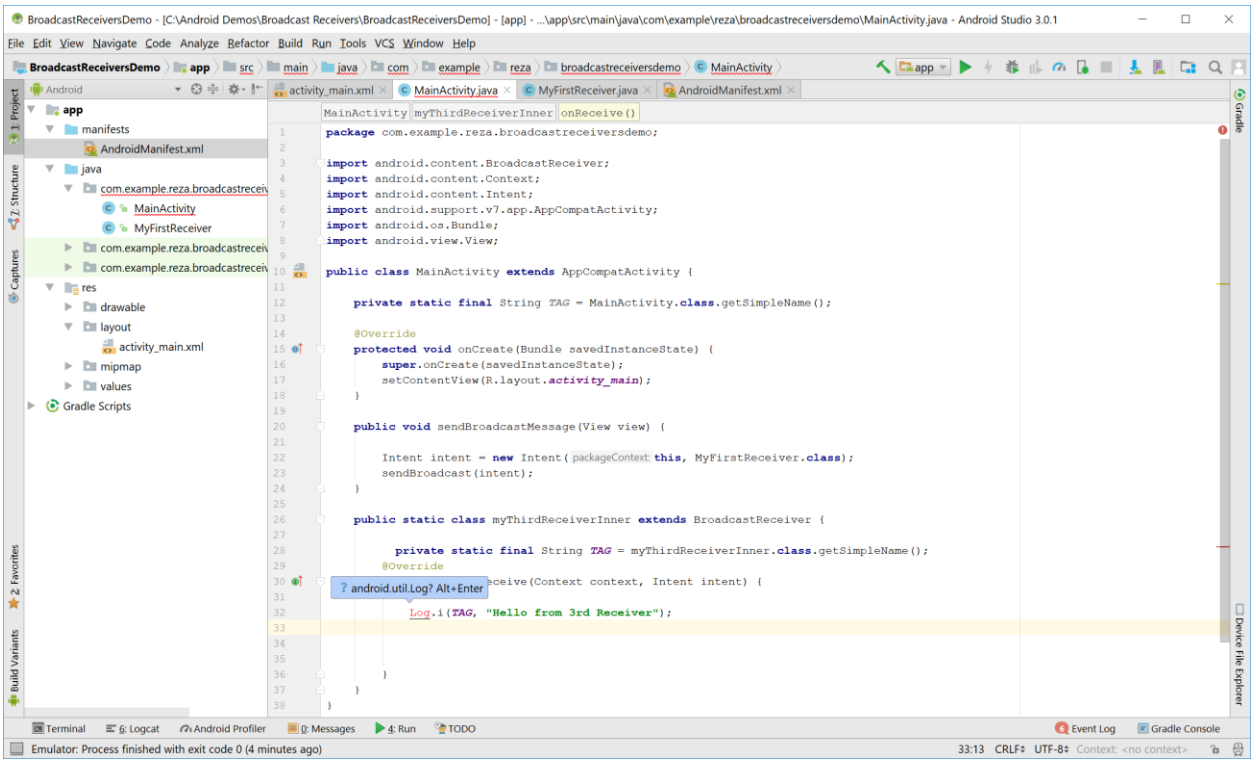
Notice the proper syntax to access the inner class inside the main activity (“.MainActivity$myThirdReceiverInner”) as shown. Note the way we are accessing the inner class, indicates that this class is static so we need to add the static keyword in our main activity.

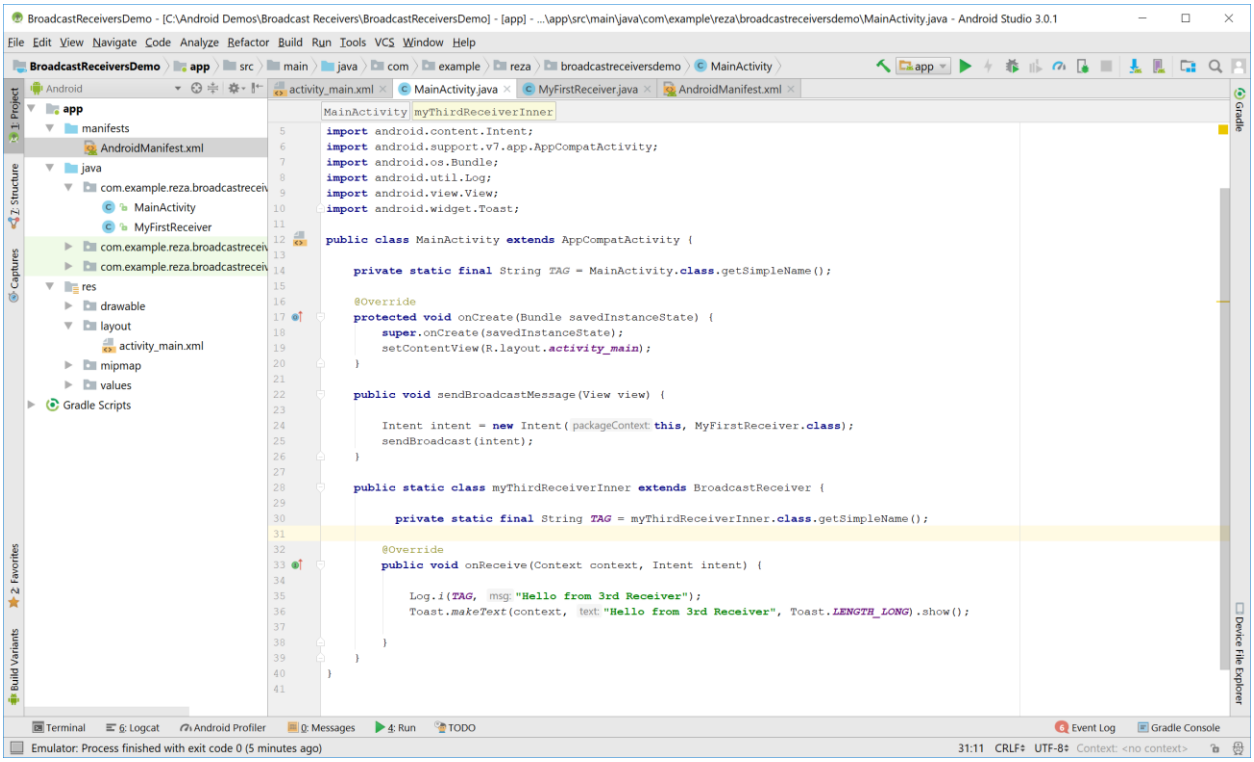


In the main activity use the static keyword for the inner class as shown:

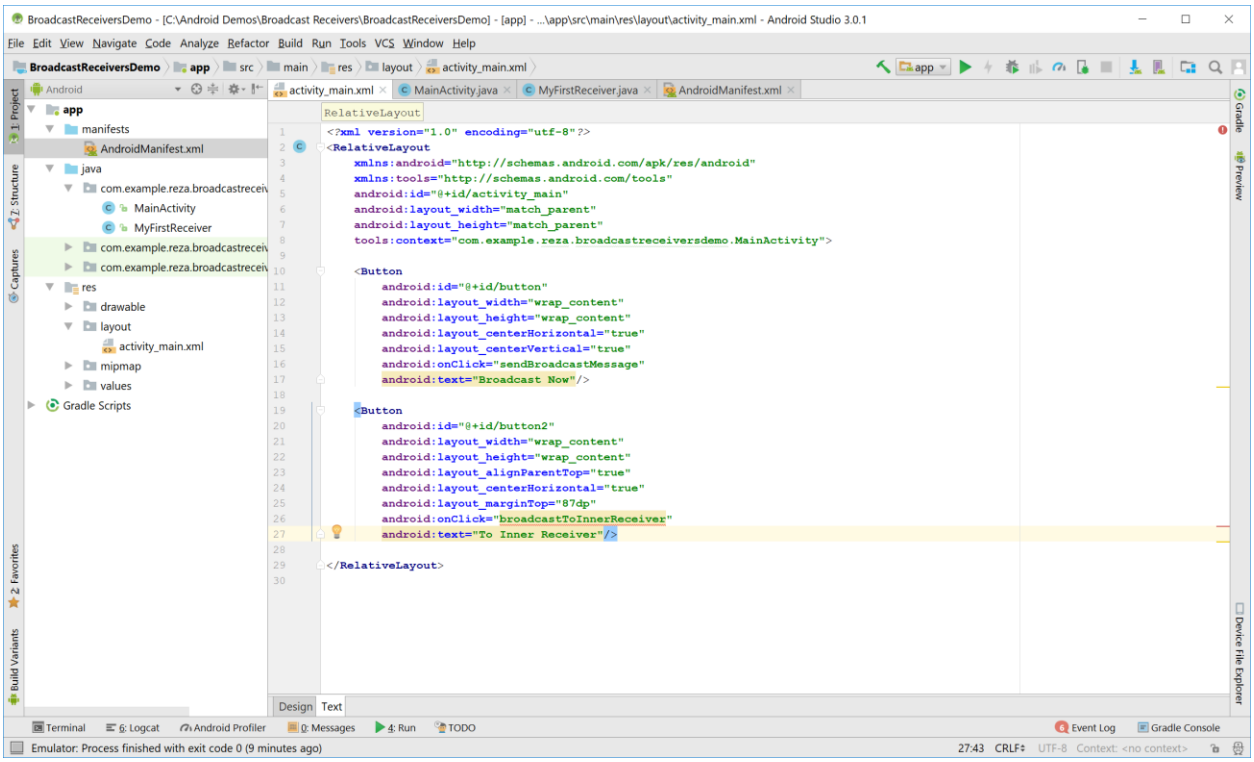


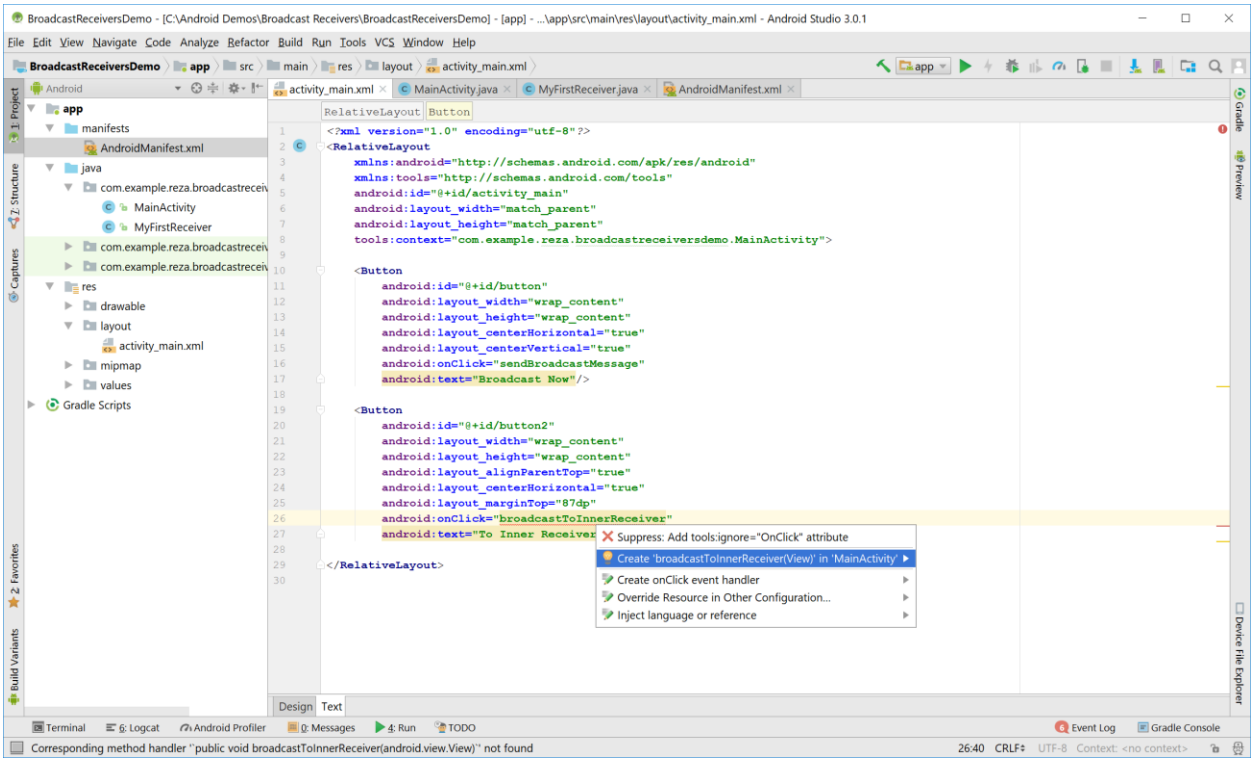
Now you need to implement the onReceive() method of this inner class as shown:

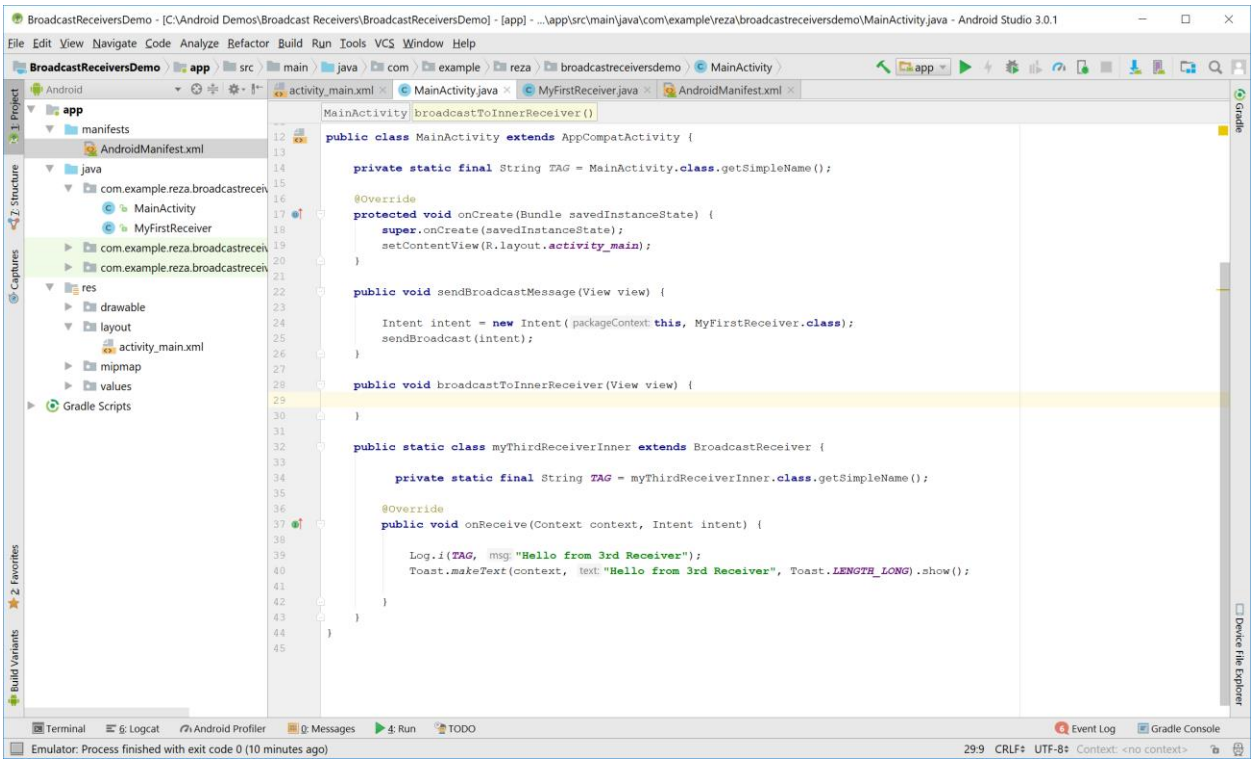




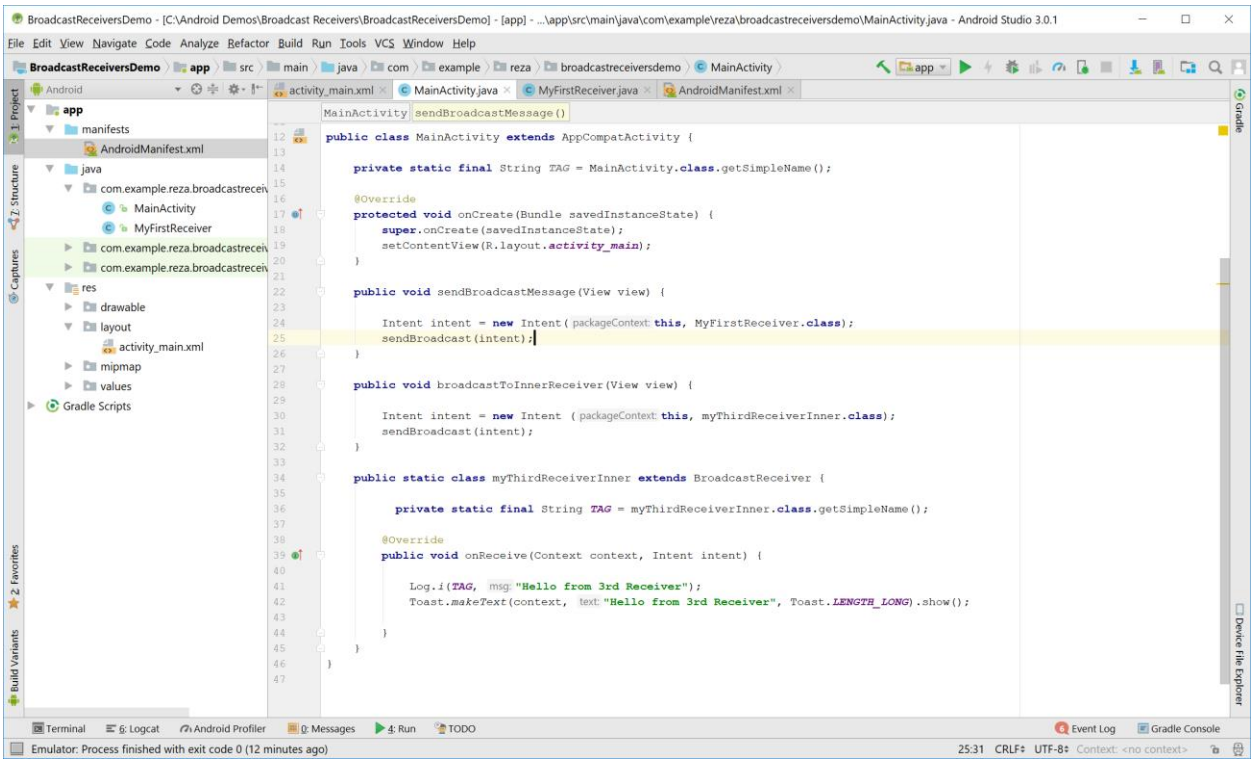
Also modify the activiy_main.xml and add a new button for our new receiver:

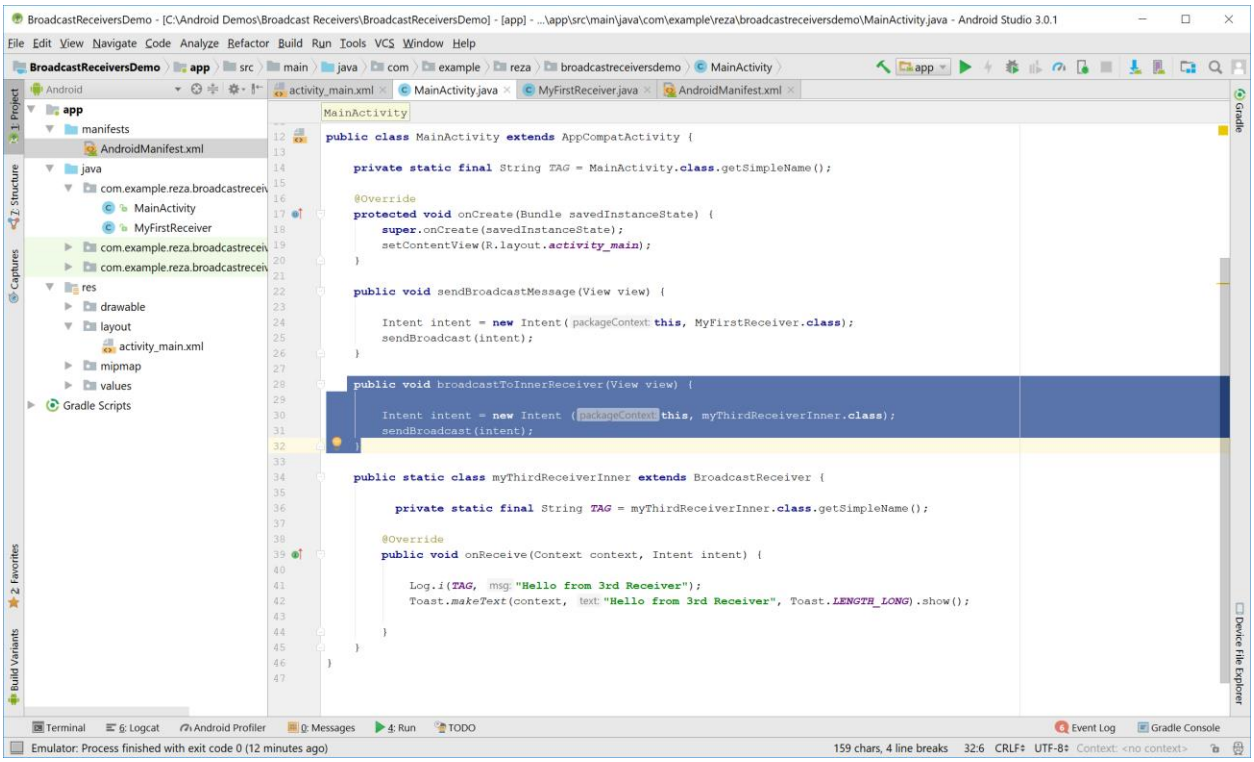



BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\res\layout\activity_main.xml - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
activity_main.xml
MainActivity.java
MyFirstReceiver.java
AndroidManifest.xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/activity_main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.reza.broadcastreceiversdemo.MainActivity">
<Button
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:layout_centerVertical="true"
android:onClick="sendBroadcastMessage"
android:text="Broadcast Now"/>
<Button
android:id="@+id/button2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
android:layout_marginTop="87dp"
android:onClick="broadcastToInnerReceiver"
android:text="To Inner Receiver
Suppress: Add tools:ignore="OnClick" attribute
Create 'broadcastToInnerReceiver(View)' in 'MainActivity'
Create onClick event handler
Override Resource in Other Configuration...
Inject language or reference
</RelativeLayout>
Corresponding method handler 'public void broadcastToInnerReceiver(android.view.View)' not found

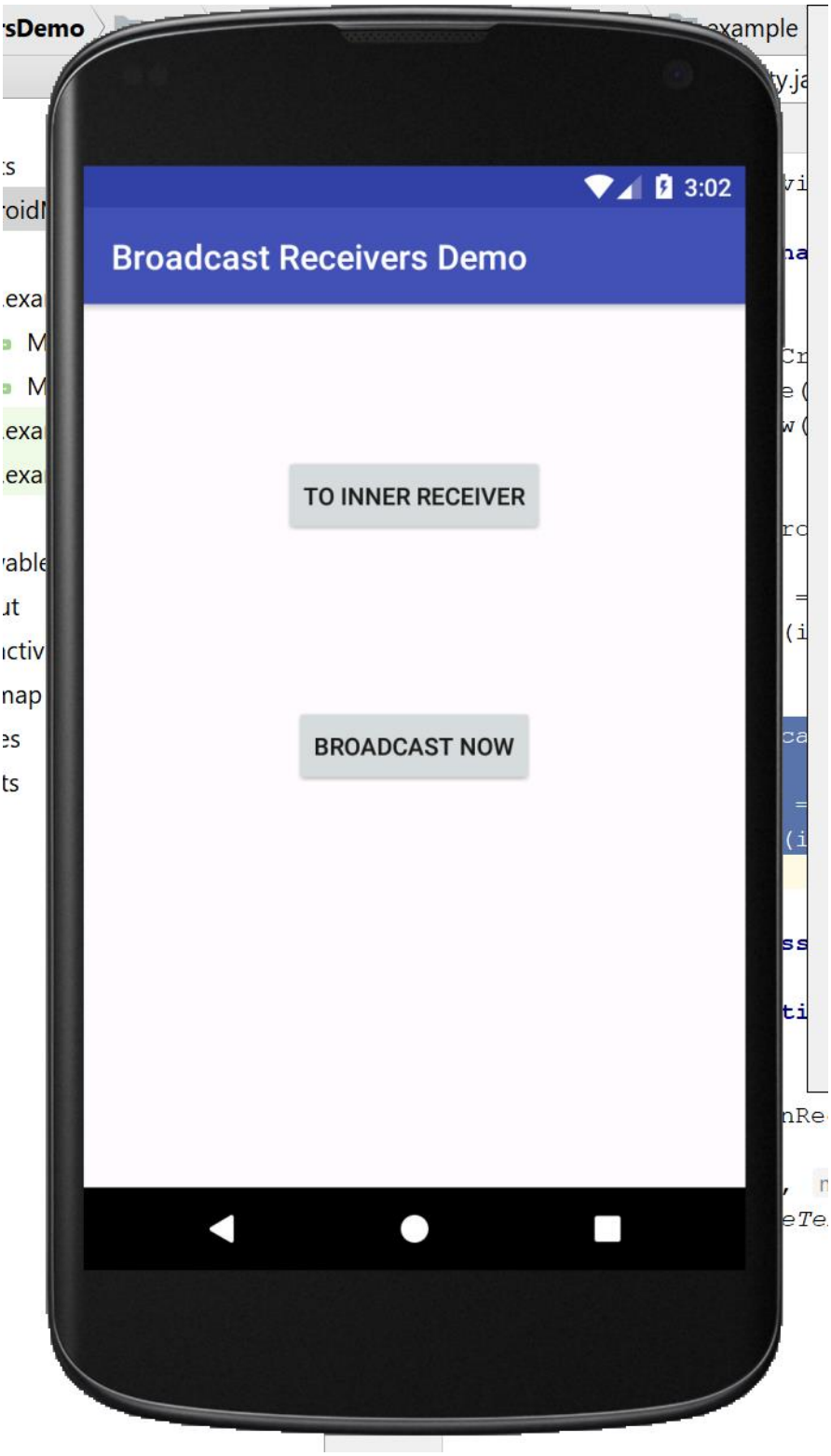

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\MainActivity.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
activity_main.xml
MainActivity.java
MyFirstReceiver.java
AndroidManifest.xml
public class MainActivity extends AppCompatActivity {
private static final String TAG = MainActivity.class.getSimpleName();
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}
public void sendBroadcastMessage(View view) {
Intent intent = new Intent(packageContext: this, MyFirstReceiver.class);
sendBroadcast(intent);
}
public void broadcastToInnerReceiver(View view) {
}
public static class myThirdReceiverInner extends BroadcastReceiver {
private static final String TAG = myThirdReceiverInner.class.getSimpleName();
@Override
public void onReceive(Context context, Intent intent) {
Log.i(TAG, msg: "Hello from 3rd Receiver");
Toast.makeText(context, text: "Hello from 3rd Receiver", Toast.LENGTH_LONG).show();
}
}
}
Emulator: Process finished with exit code 0 (10 minutes ago)

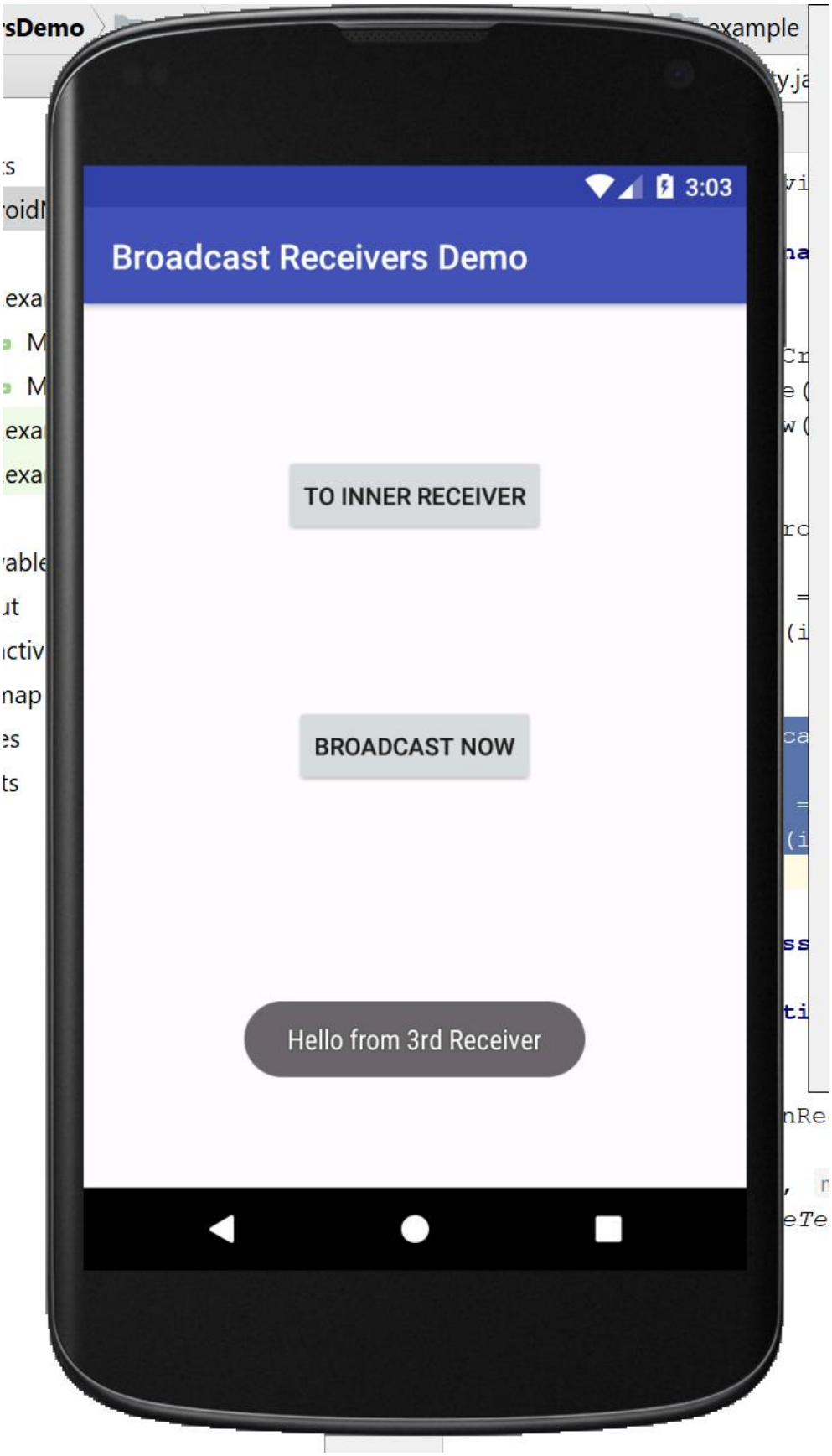
Now complete the code for broadcastToInnerReceiver() method as shown:

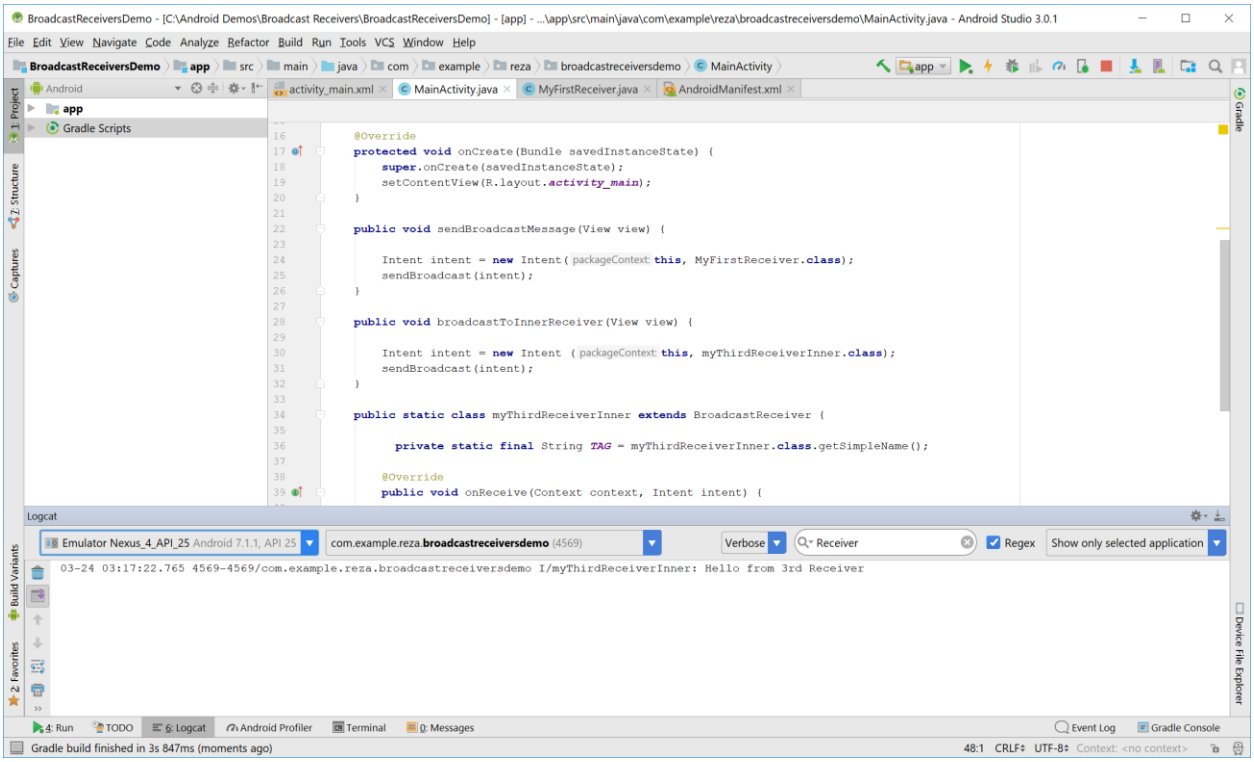


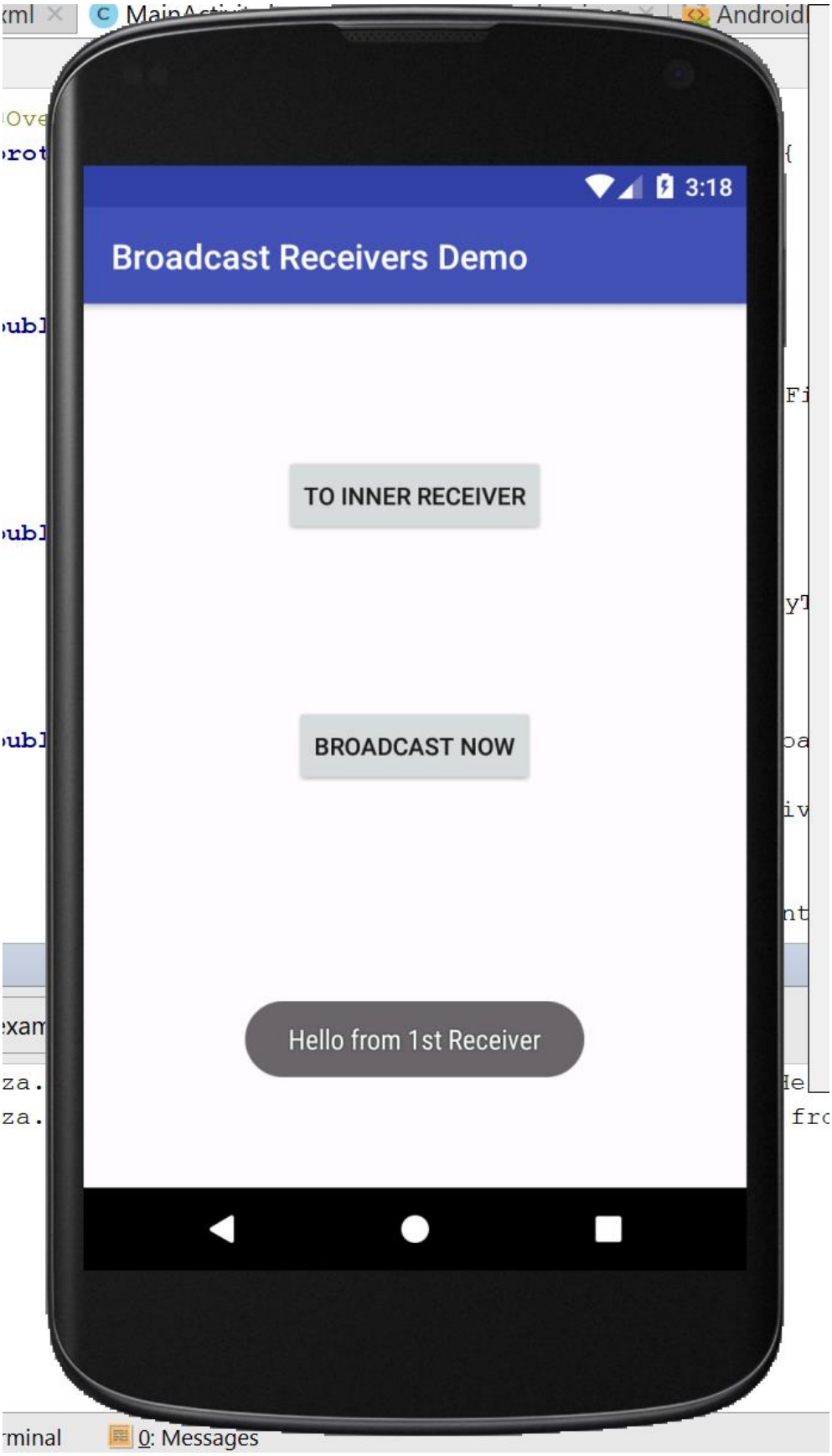


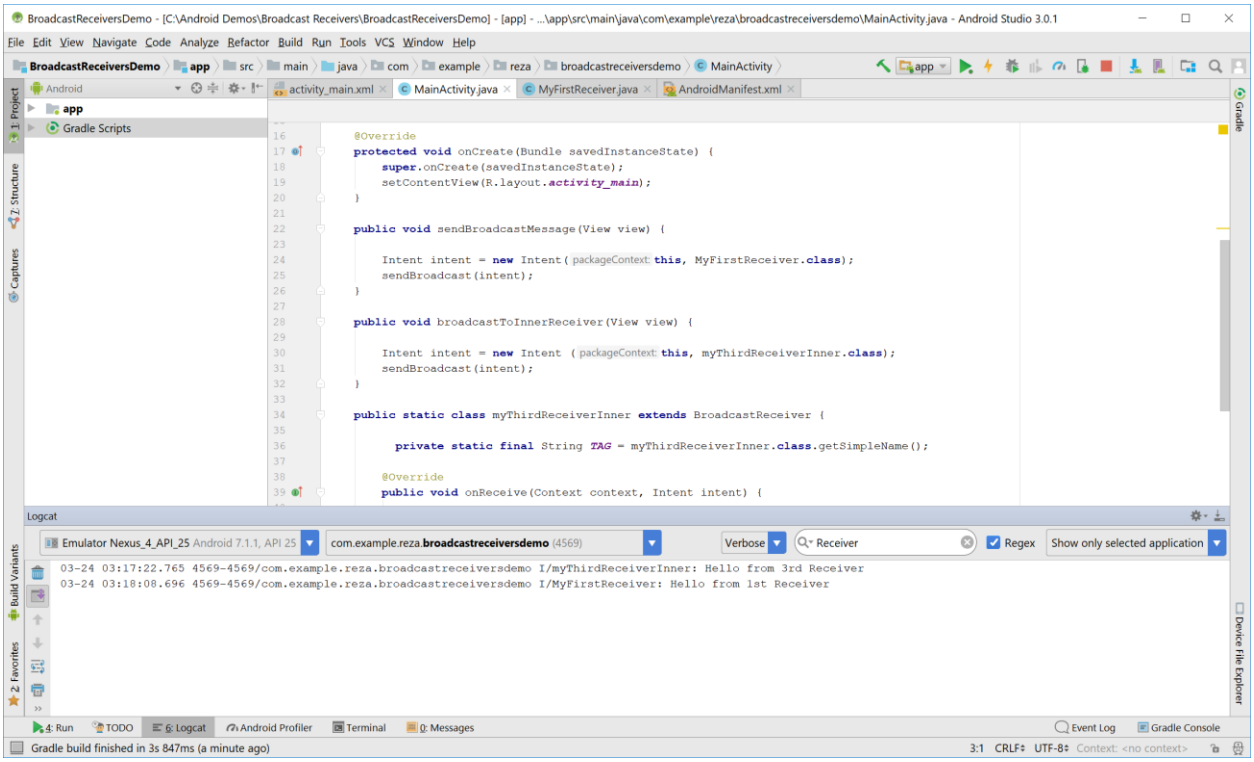
Now run the application to test your new receiver:




3:03
Broadcast Receivers Demo
TO INNER RECEIVER
BROADCAST NOW
Hello from 3rd Receiver

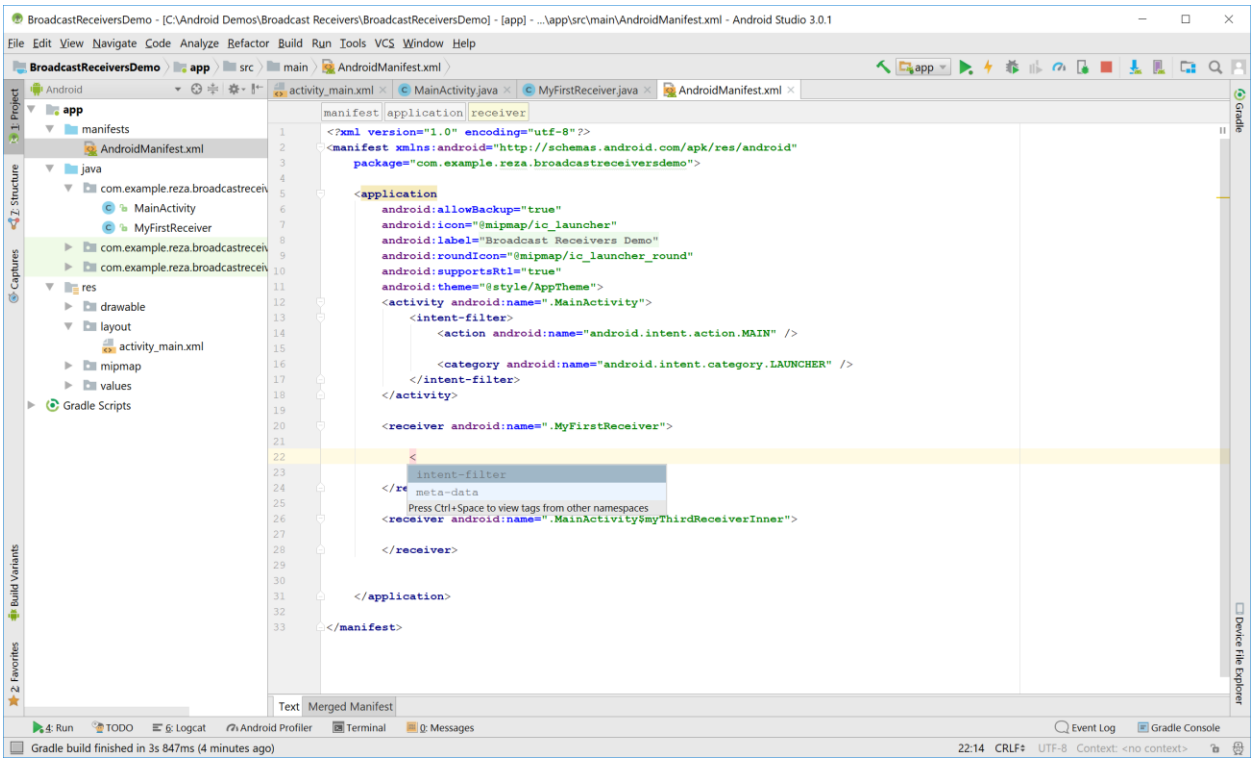

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\MainActivity.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
activity_main.xml
MainActivity.java
MyFirstReceiver.java
AndroidManifest.xml
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}
public void sendBroadcastMessage(View view) {
Intent intent = new Intent(packageContext: this, MyFirstReceiver.class);
sendBroadcast(intent);
}
public void broadcastToInnerReceiver(View view) {
Intent intent = new Intent (packageContext: this, myThirdReceiverInner.class);
sendBroadcast(intent);
}
public static class myThirdReceiverInner extends BroadcastReceiver {
private static final String TAG = myThirdReceiverInner.class.getSimpleName();
@Override
public void onReceive(Context context, Intent intent) {
Logcat
Emulator Nexus_4_API_25 Android 7.1.1, API 25
com.example.reza.broadcastreceiversdemo (4569)
Verbose
Receiver
Regex
Show only selected application
03-24 03:17:22.765 4569-4569/com.example.reza.broadcastreceiversdemo I/myThirdReceiverInner: Hello from 3rd Receiver
Gradle build finished in 3s 847ms (moments ago)

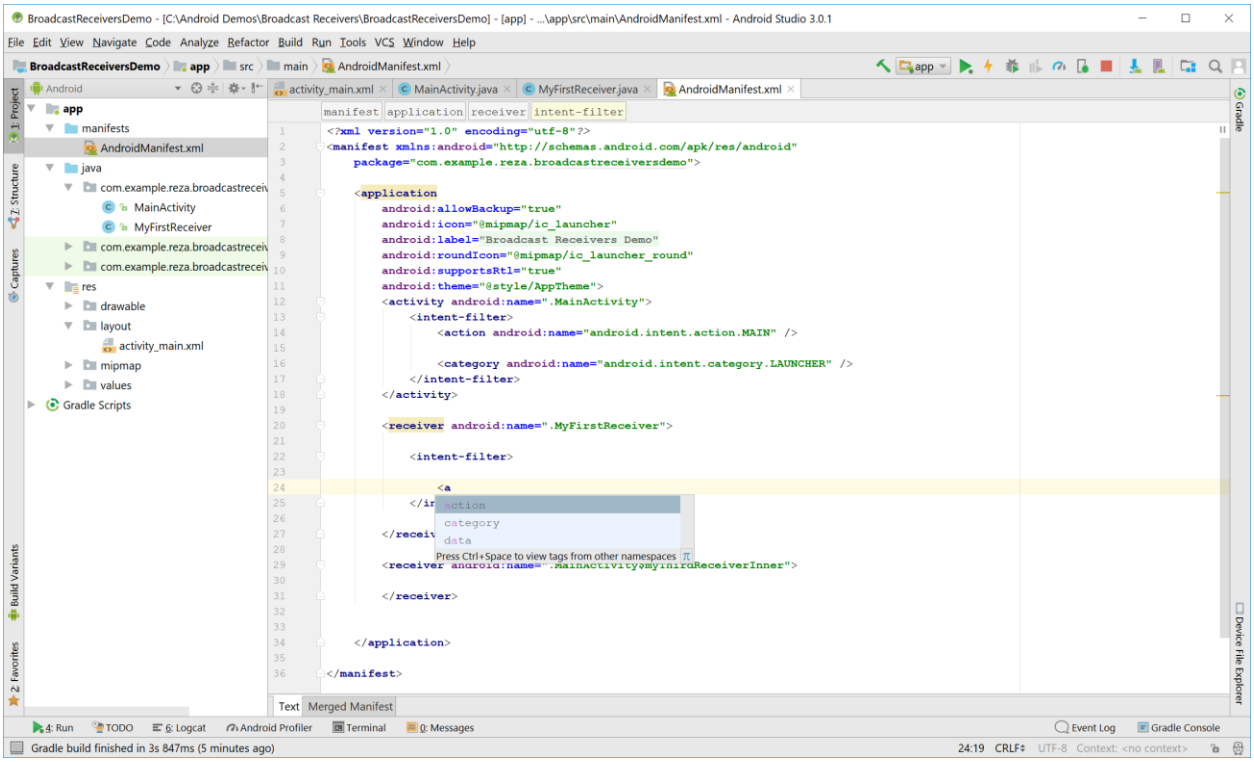

3:18
Broadcast Receivers Demo
TO INNER RECEIVER
BROADCAST NOW
Hello from 1st Receiver

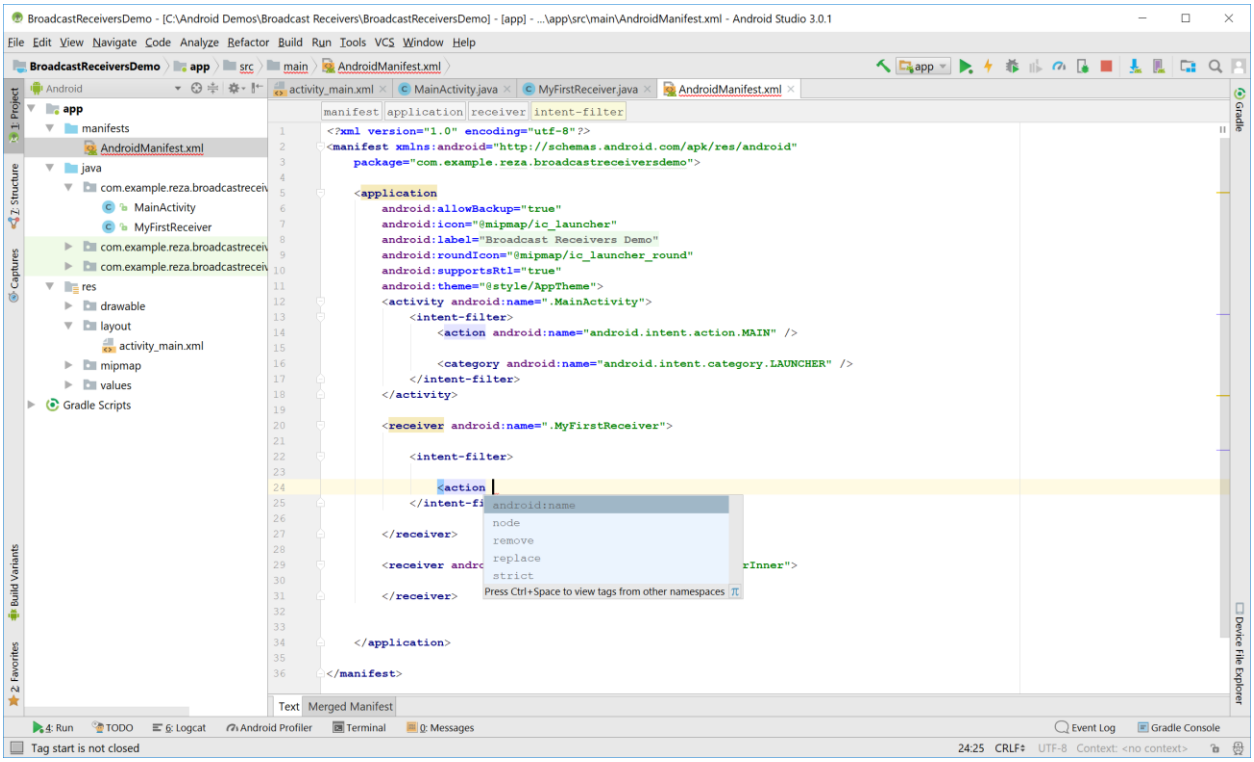

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\MainActivity.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
activity_main.xml
MainActivity.java
MyFirstReceiver.java
AndroidManifest.xml
app
Gradle Scripts
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}
public void sendBroadcastMessage(View view) {
Intent intent = new Intent(packageContext: this, MyFirstReceiver.class);
sendBroadcast(intent);
}
public void broadcastToInnerReceiver(View view) {
Intent intent = new Intent (packageContext: this, myThirdReceiverInner.class);
sendBroadcast(intent);
}
public static class myThirdReceiverInner extends BroadcastReceiver {
private static final String TAG = myThirdReceiverInner.class.getSimpleName();
@Override
public void onReceive(Context context, Intent intent) {
Logcat
Emulator Nexus_4_API_25 Android 7.1.1, API 25
com.example.reza.broadcastreceiversdemo (4569)
Verbose
Receiver
Regex
Show only selected application
03-24 03:17:22.765 4569-4569/com.example.reza.broadcastreceiversdemo I/myThirdReceiverInner: Hello from 3rd Receiver
03-24 03:18:08.696 4569-4569/com.example.reza.broadcastreceiversdemo I/MyFirstReceiver: Hello from 1st Receiver
4: Run
TODO
6: Logcat
Android Profiler
Terminal
0: Messages
Event Log
Gradle Console
Gradle build finished in 3s 847ms (a minute ago)

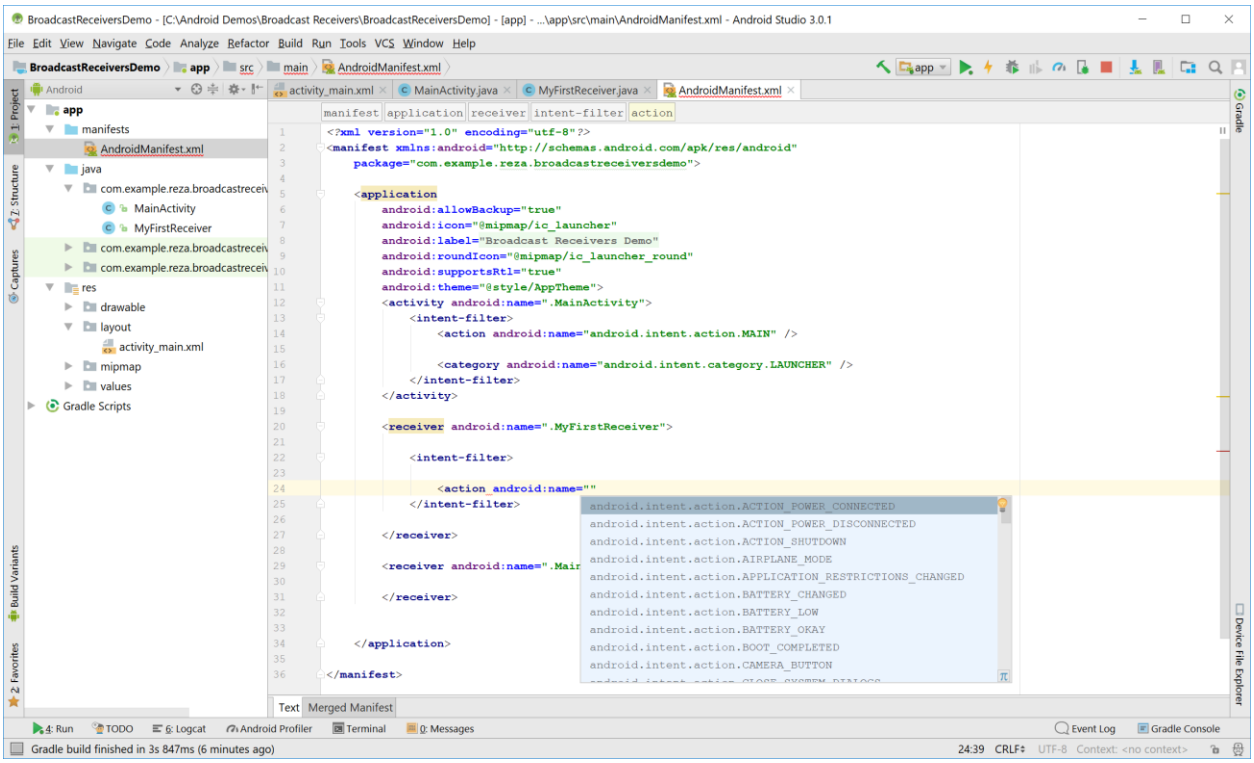
Using Intent Filter with Custom Action

Modify your manifest file and define your intent filter as shown:

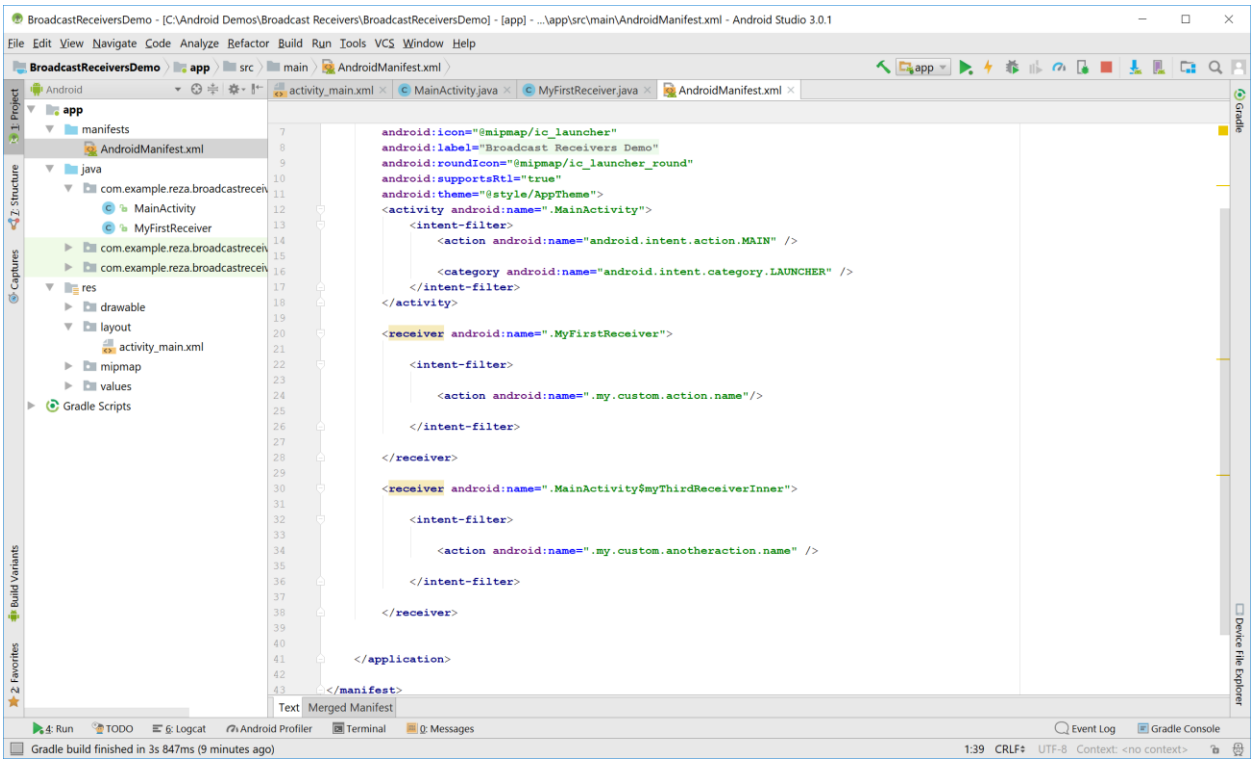


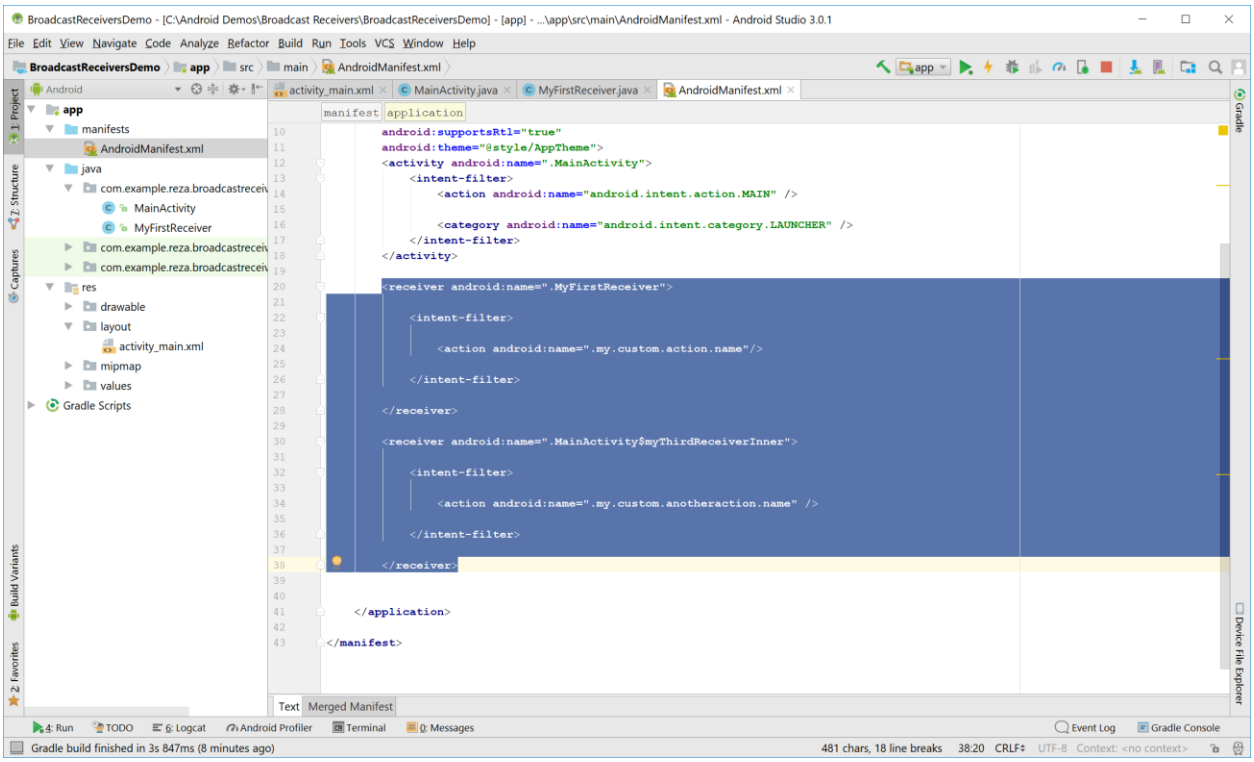



BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\AndroidManifest.xml - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main AndroidManifest.xml
activity_main.xml MainActivity.java MyFirstReceiver.java AndroidManifest.xml
manifest application receiver intent-filter
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.reza.broadcastreceiversdemo">
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="Broadcast Receivers Demo"
android:roundIcon="@mipmap/ic_launcher_round"
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>
<receiver android:name=".MyFirstReceiver">
<intent-filter>
<action
android:name
node
remove
replace
strict
Press Ctrl+Space to view tags from other namespaces
</receiver>
</application>
</manifest>
Text Merged Manifest
Tag start is not closed
24:25 CRLF UTF-8 Context: <no context>

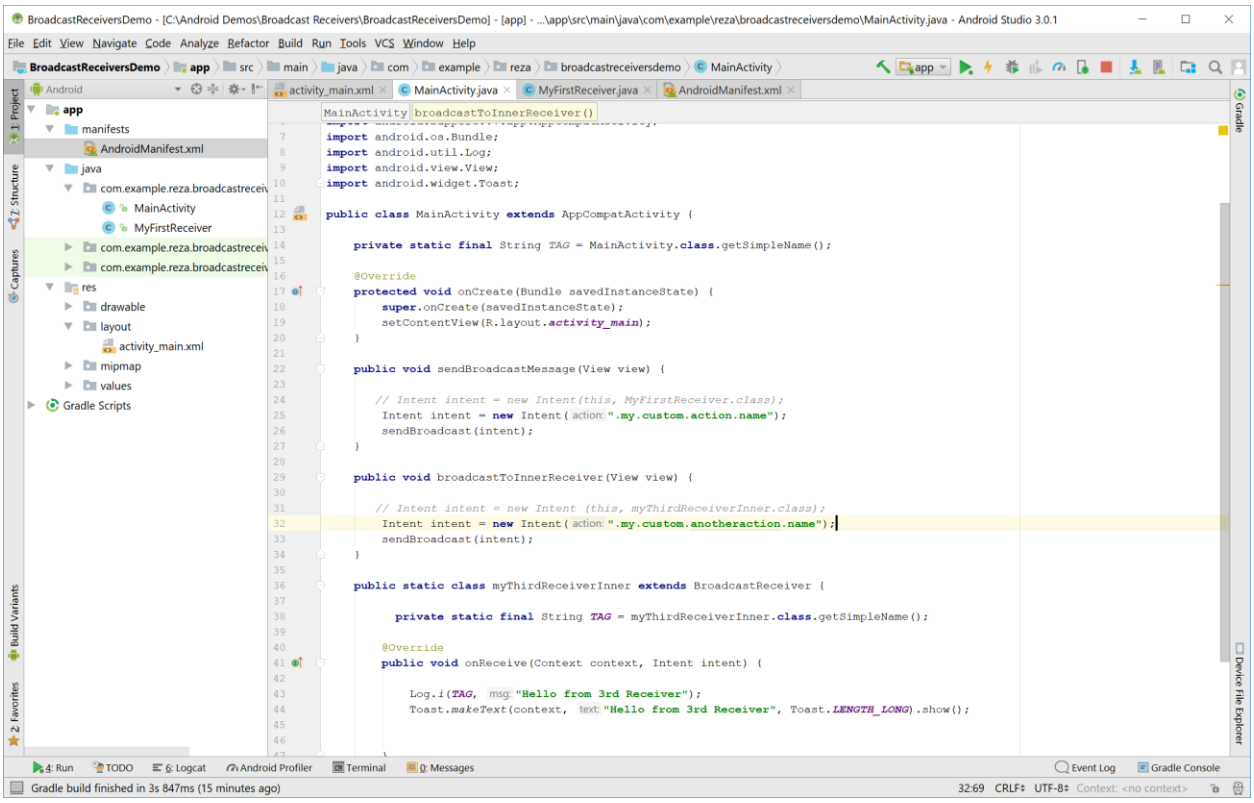

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\AndroidManifest.xml - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main AndroidManifest.xml
activity_main.xml MainActivity.java MyFirstReceiver.java AndroidManifest.xml
manifest application receiver intent-filter action
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.reza.broadcastreceiversdemo">
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="Broadcast Receivers Demo"
android:roundIcon="@mipmap/ic_launcher_round"
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>
<receiver android:name=".MyFirstReceiver">
<intent-filter>
<action android:name=""
android.intent.action.ACTION_POWER_CONNECTED
android.intent.action.ACTION_POWER_DISCONNECTED
android.intent.action.ACTION_SHUTDOWN
android.intent.action.AIRPLANE_MODE
android.intent.action.APPLICATION_RESTRICTIONS_CHANGED
android.intent.action.BATTERY_CHANGED
android.intent.action.BATTERY_LOW
android.intent.action.BATTERY_OKAY
android.intent.action.BOOT_COMPLETED
android.intent.action.CAMERA_BUTTON
</application>
</manifest>
Text Merged Manifest
Gradle build finished in 3s 847ms (6 minutes ago)
24:39 CRLF UTF-8 Context: <no context>

Here is your completed manifest file:



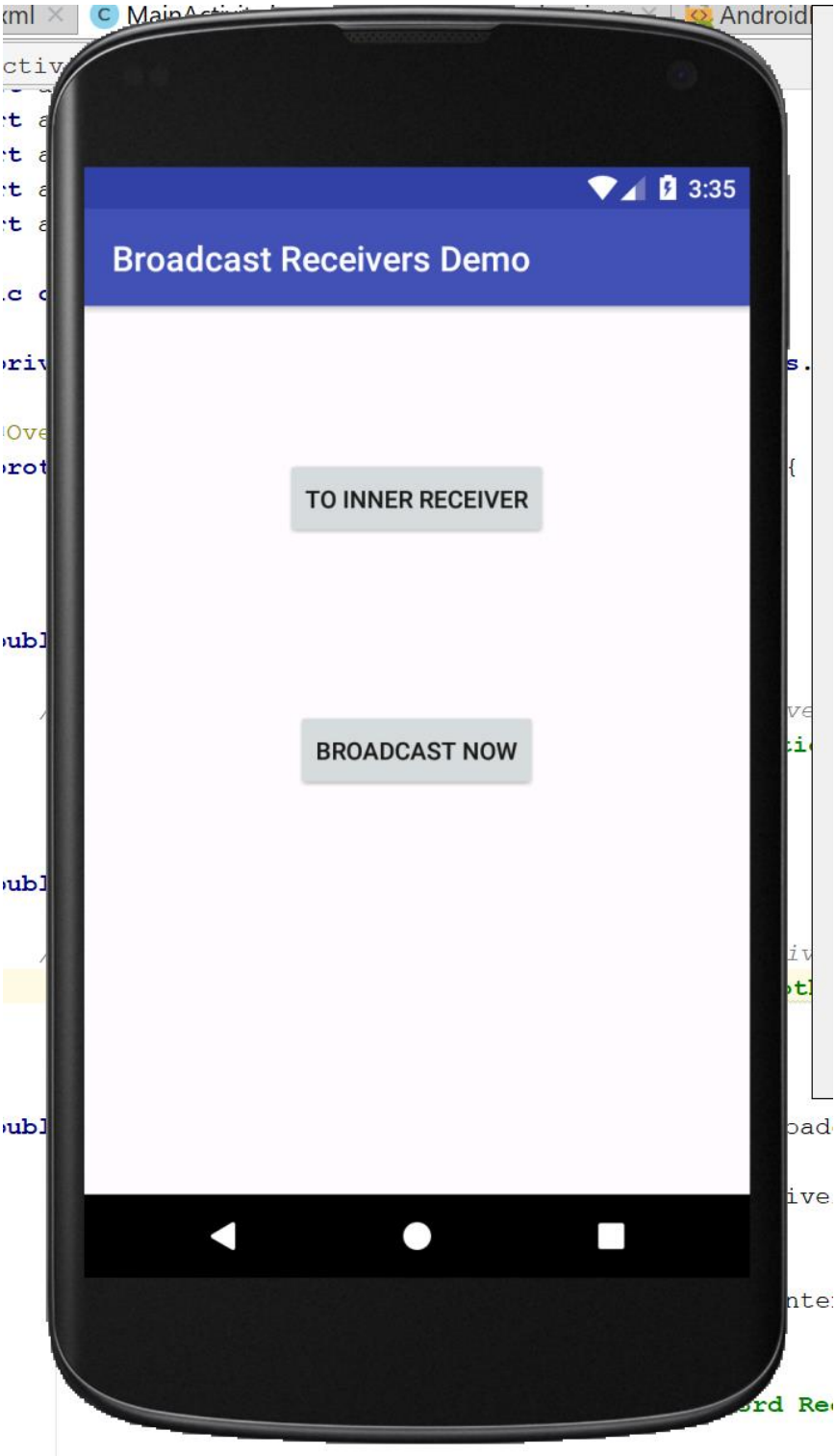


With defining an intent-filter and an action tag, you can send a broadcast implicitly versus sending a broadcast explicitly. In our code before we sent a broadcast explicitly to our first receiver or to our third receiver. Now we can send a broadcast implicitly as shown. Modify the main activity java file as shown:

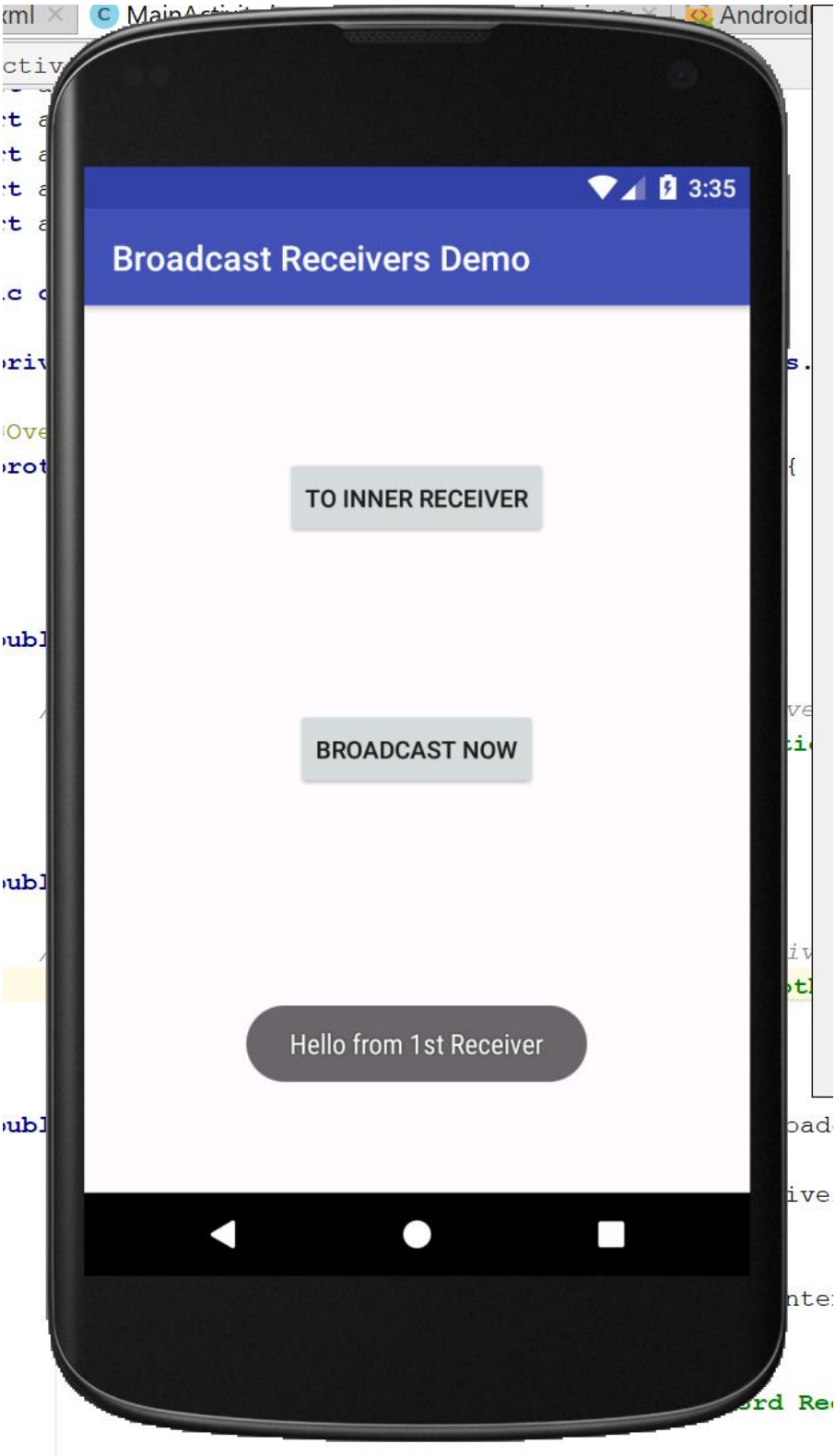


In our code we have commented out lines, 24 and 31 and added lines 25 and 32 which create our intents based on the action strings we specified in our manifest file.

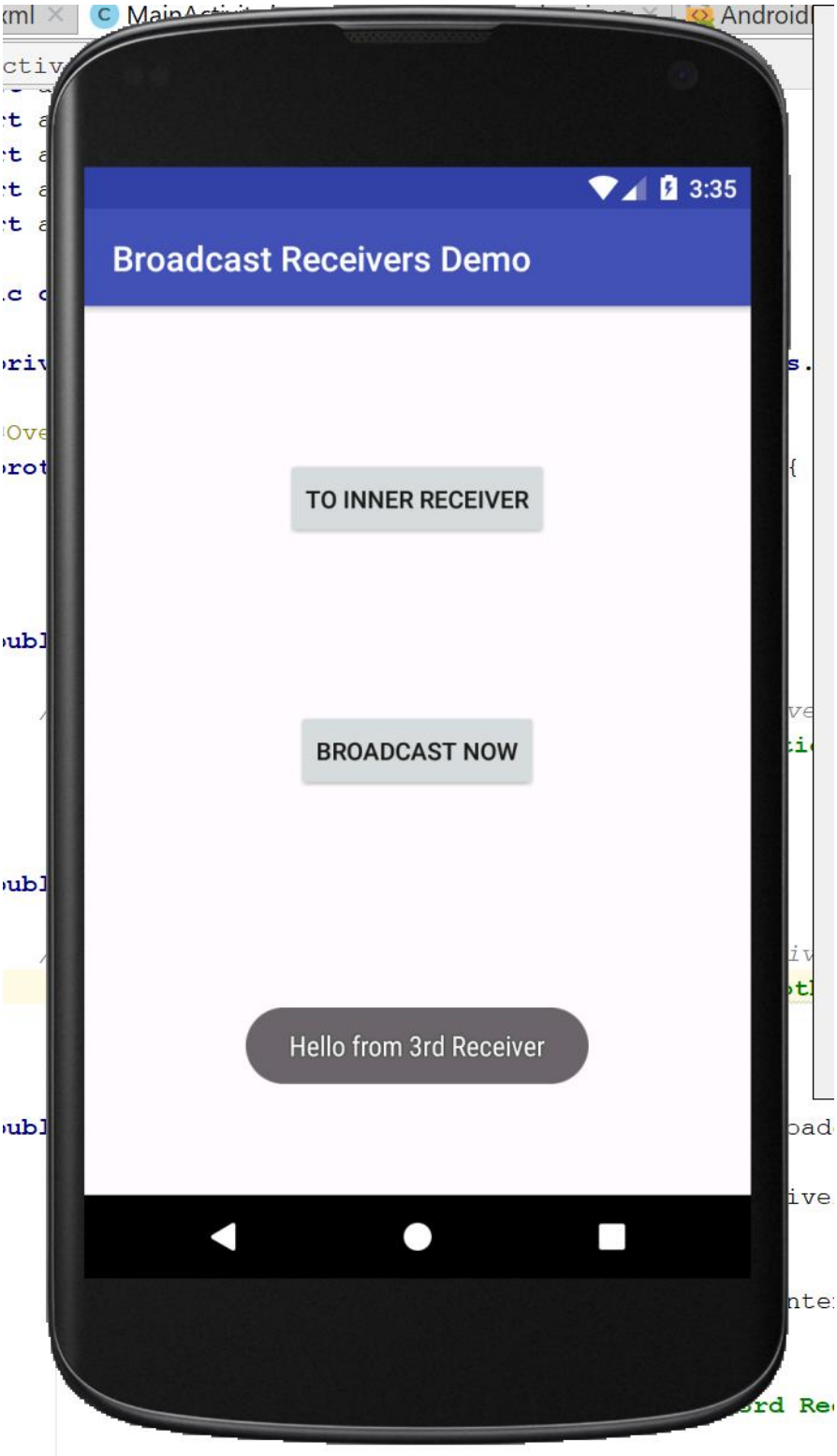
Now run the application:



Click “Broadcast Now” button:



Click “To Inner Receiver” button:



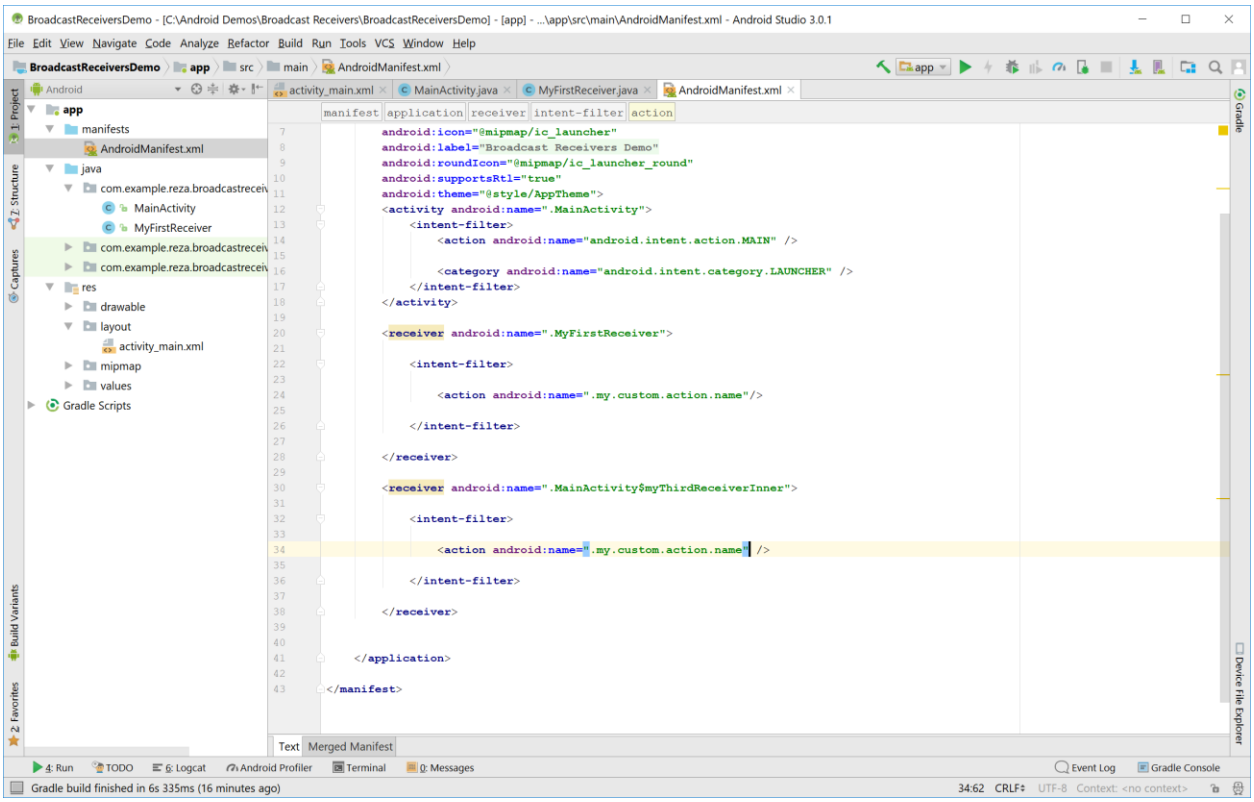
So the application works as before and delivers the broadcast to those Broadcast Receivers that are registered to those actions. So, for “my.custom.action.name” we have “MyFirstReceiver” registered and for “my.custom.anotheraction.name” we have our “myThirdReceiverInner” registered inside our manifest file.

Note that you can define as many receivers as you want inside the manifest file and then you should define the action for each of those receivers.

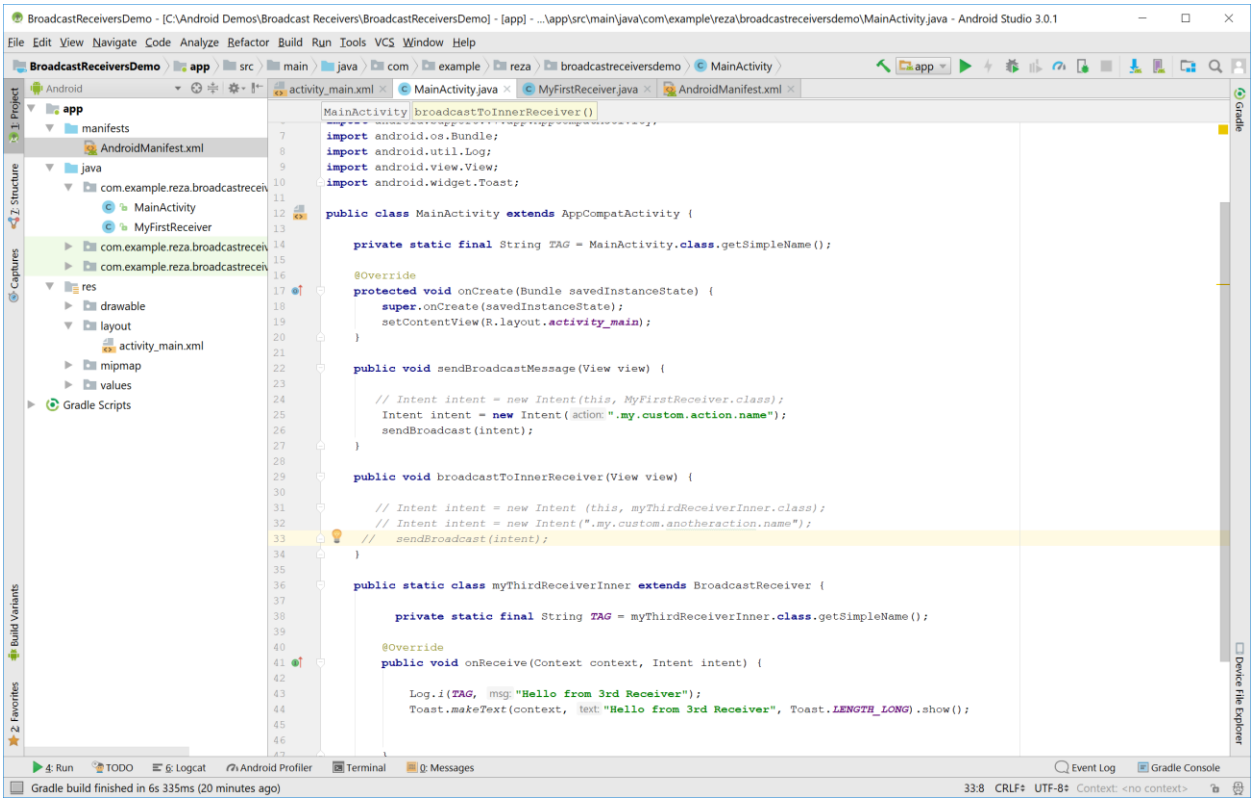
You can also have multiple receivers with the same action name which is the next topic.

Multiple Receivers with the Same Action Name

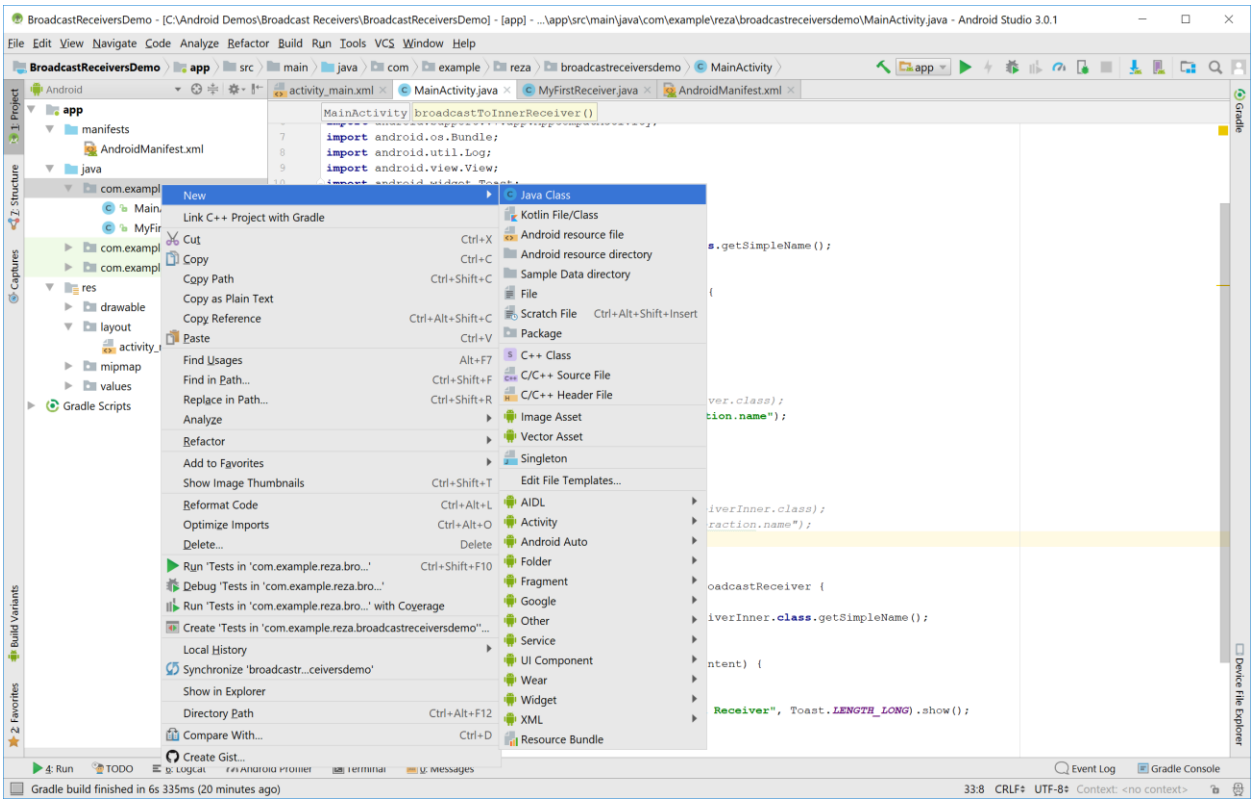
Modify the manifest file and choose the same action for both receivers as shown:

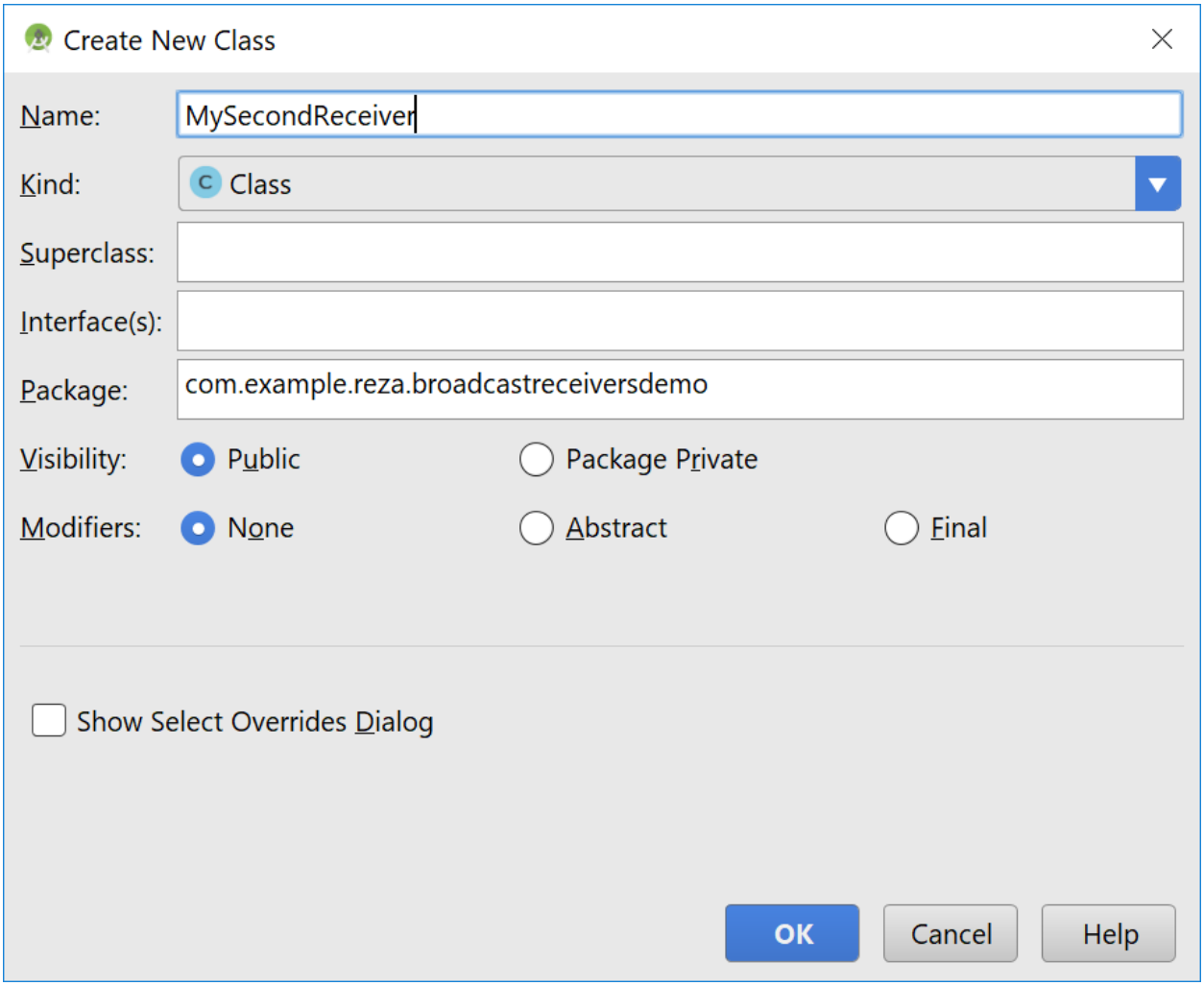


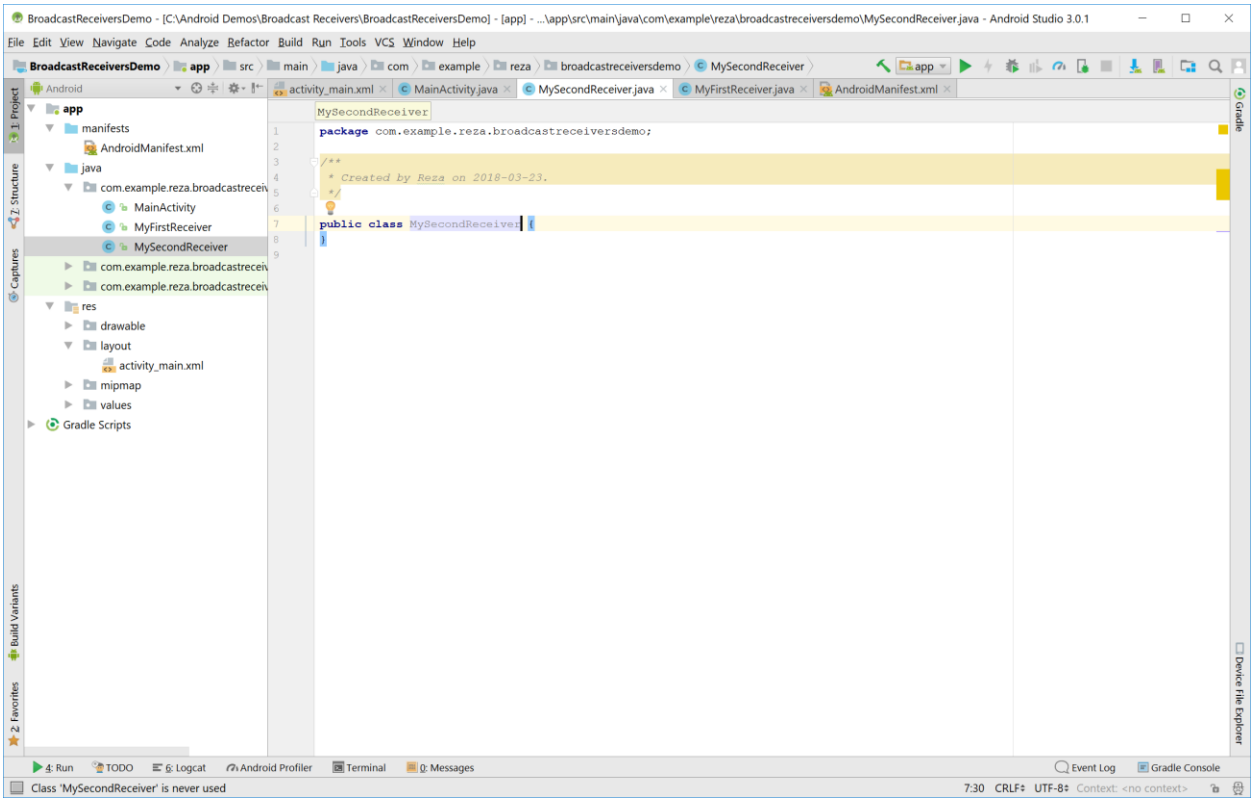
Now go back inside the main activity and modify the code as shown (I have commented out lines 32 and 33, since our sendBroadcastMessage() method defined in line 22 is sufficient now):

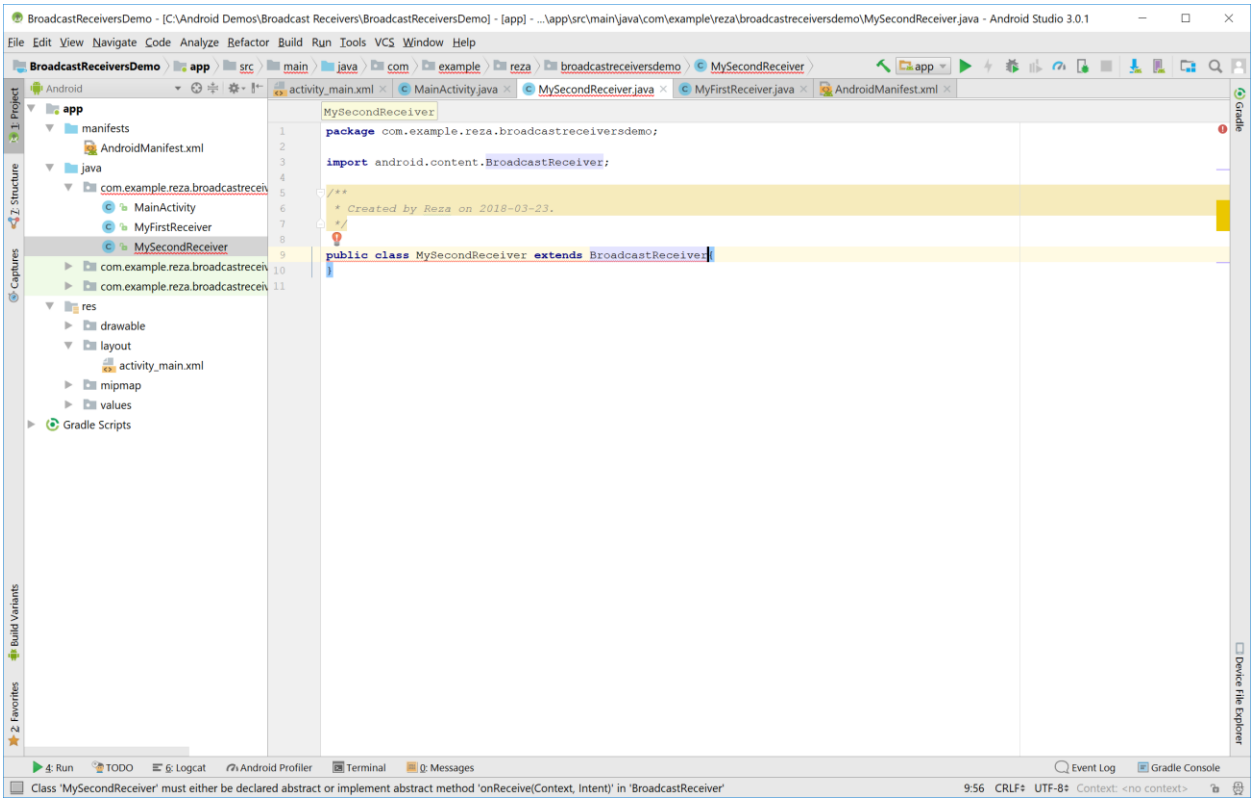


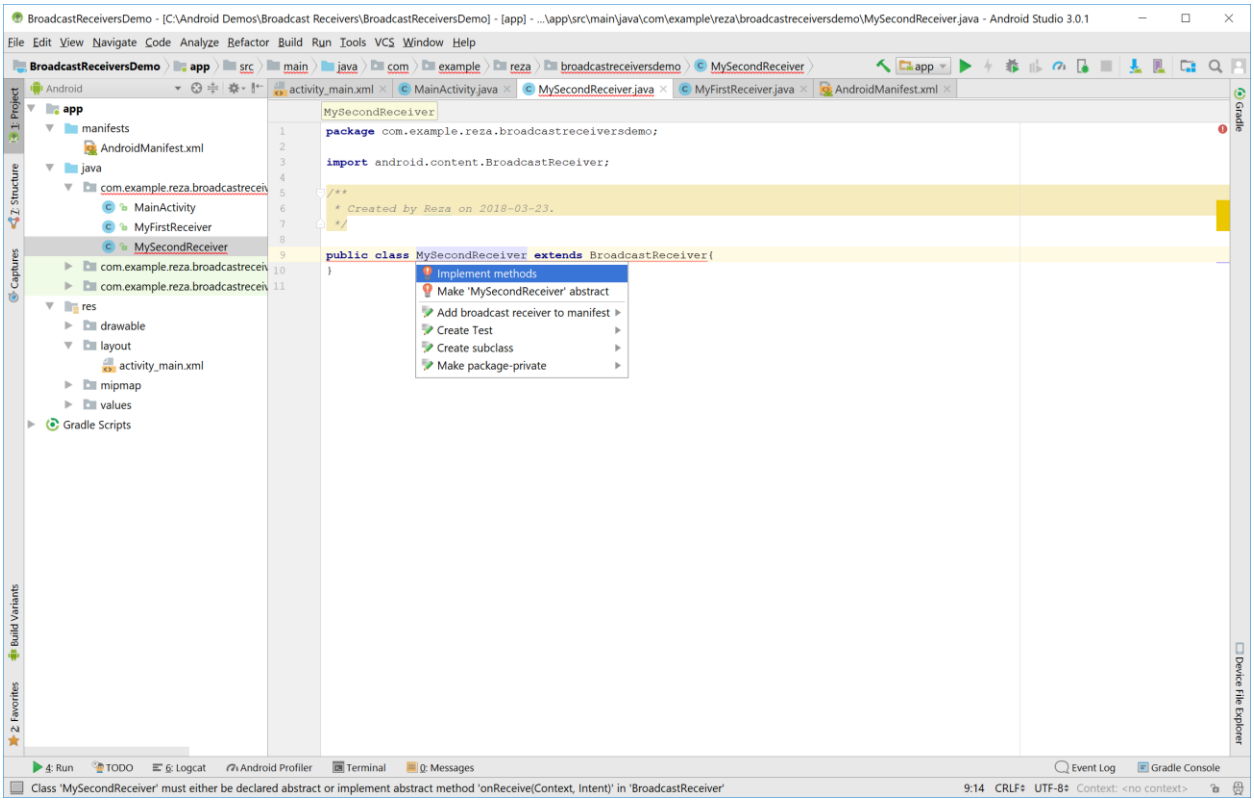
Now create your second receiver as shown:

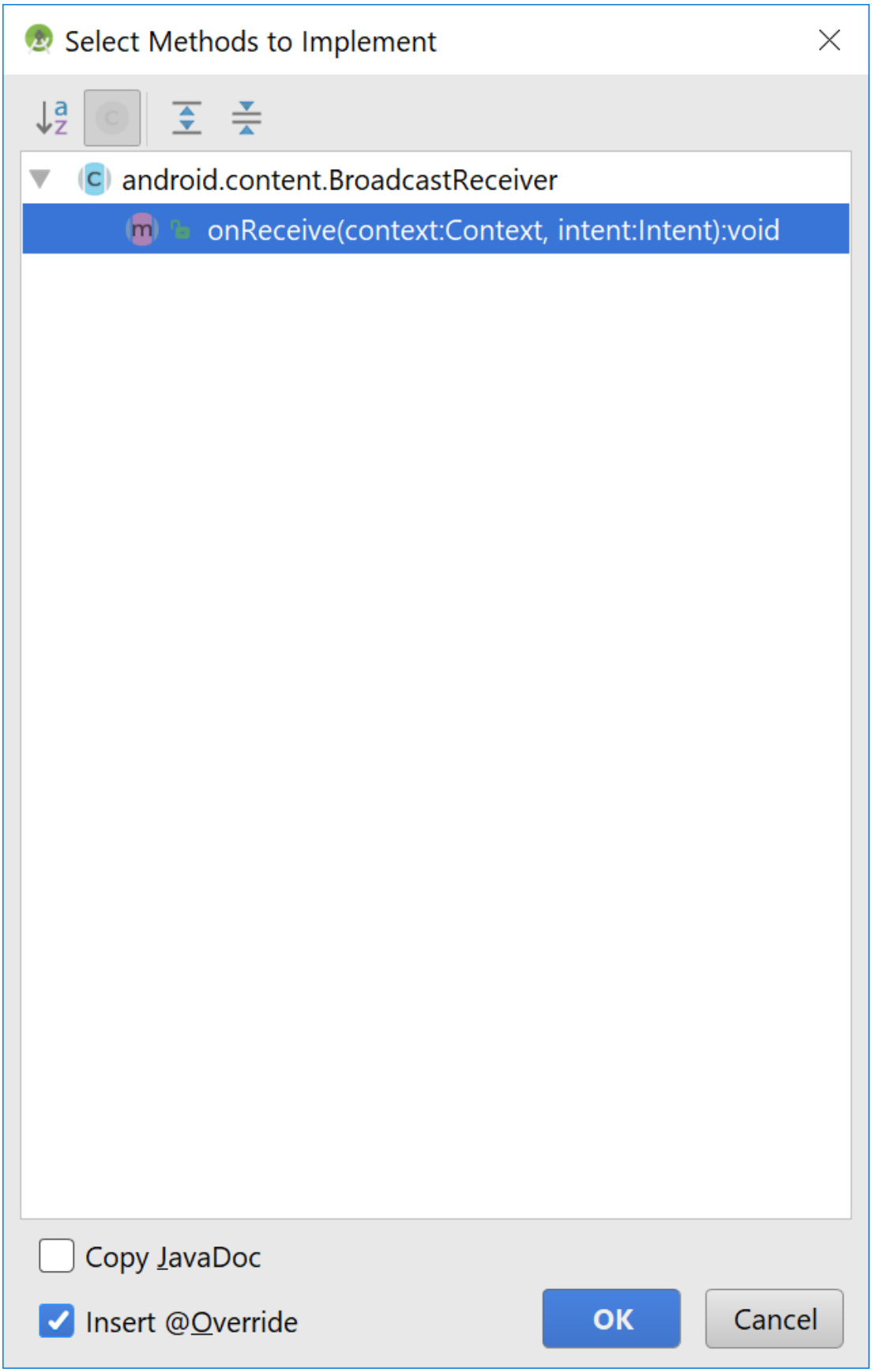


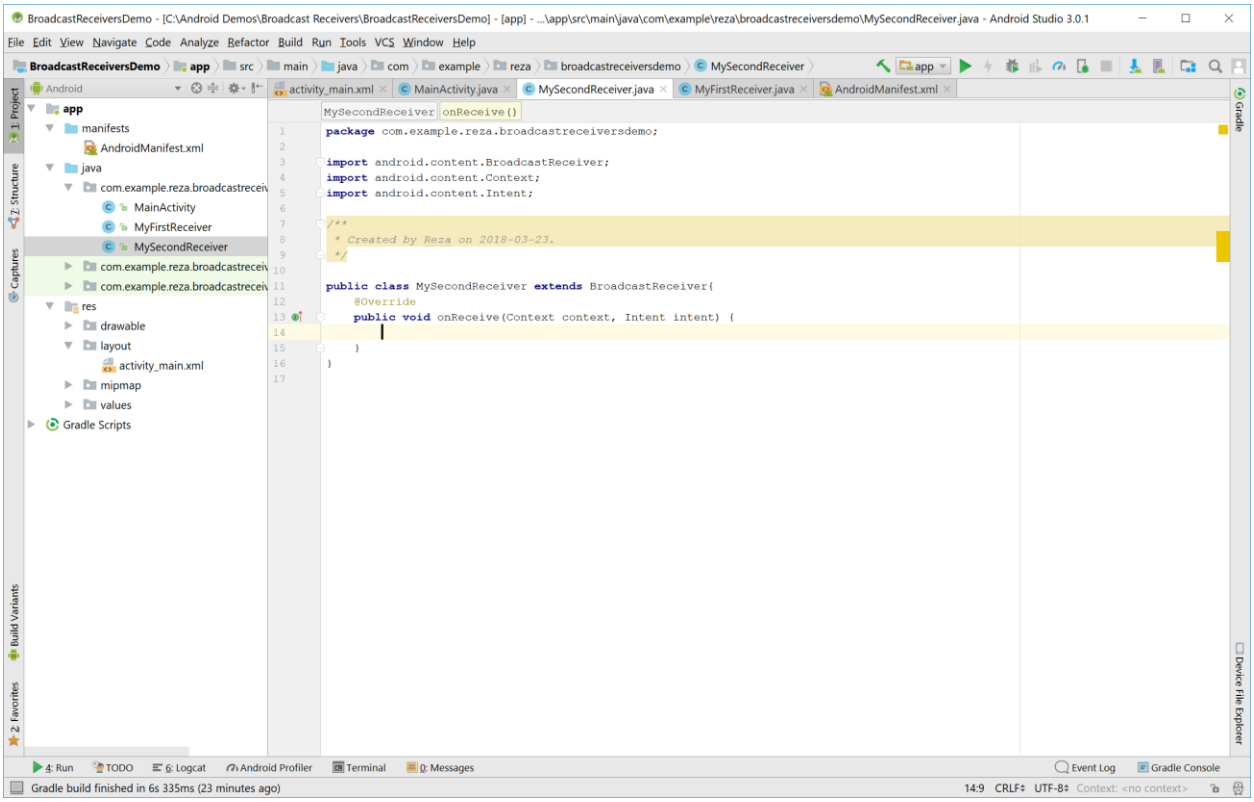

Create New Class
Name:
MySecondReceiver
Kind:
Class
Superclass:
Interface(s):
Package:
com.example.reza.broadcastreceiversdemo
Visibility:
Public
Package Private
Modifiers:
None
Abstract
Final
Show Select Overrides Dialog
OK
Cancel
Help


BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\MySecondReceiver.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main java com example reza broadcastreceiversdemo MySecondReceiver
activity_main.xml MainActivity.java MySecondReceiver.java MyFirstReceiver.java AndroidManifest.xml
MySecondReceiver
1 package com.example.reza.broadcastreceiversdemo;
3 /**
4 * Created by Reza on 2018-03-23.
5 */
7 public class MySecondReceiver {
8 }
app
manifests
AndroidManifest.xml
java
com.example.reza.broadcastreceiv
MainActivity
MyFirstReceiver
MySecondReceiver
com.example.reza.broadcastreceiv
com.example.reza.broadcastreceiv
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
4: Run TODO 6: Logcat Android Profiler Terminal 0: Messages Event Log Gradle Console
Class 'MySecondReceiver' is never used
7:30 CRLF UTF-8 Context: <no context>

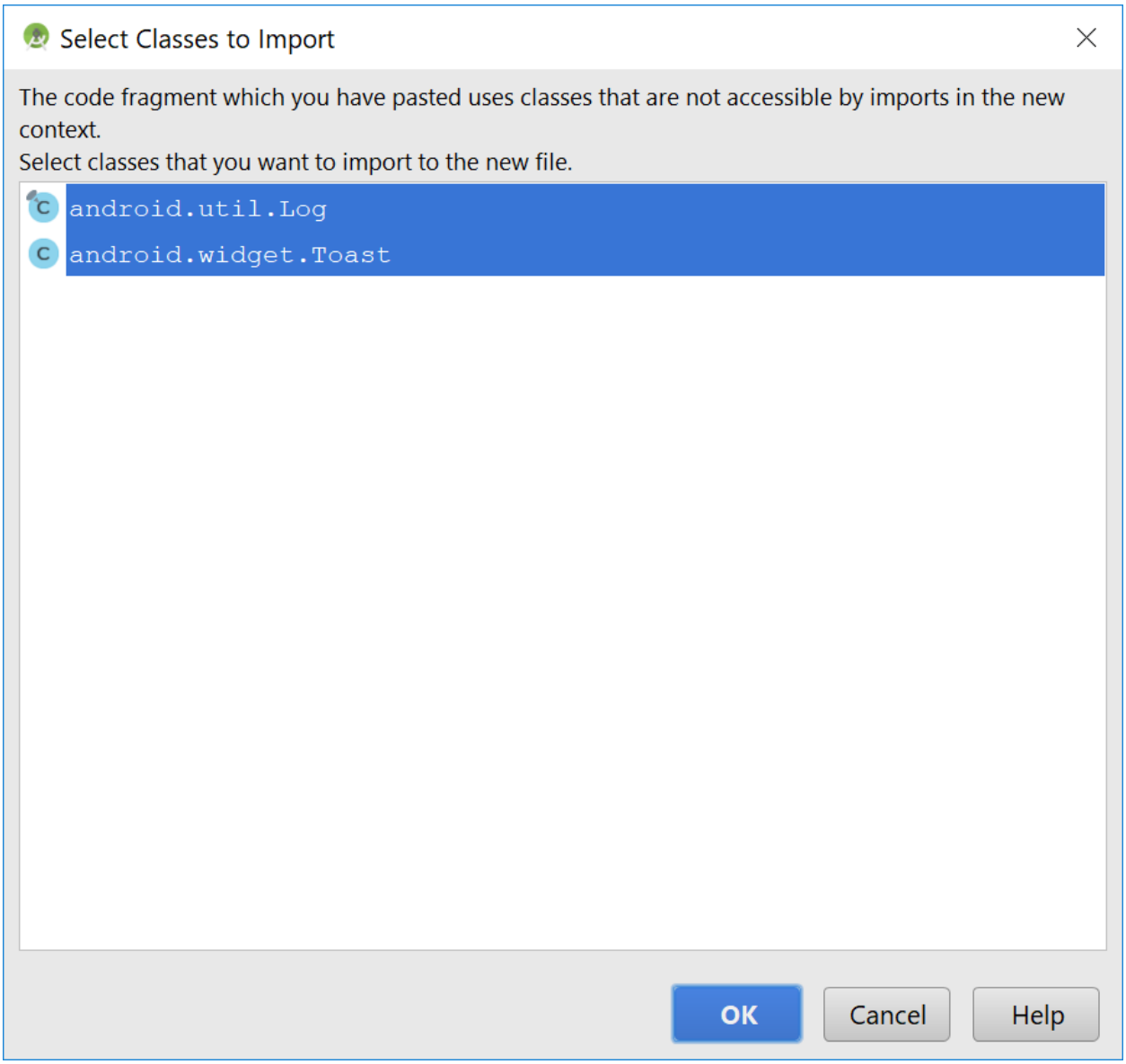

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\MySecondReceiver.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main java com example reza broadcastreceiversdemo MySecondReceiver
activity_main.xml MainActivity.java MySecondReceiver.java MyFirstReceiver.java AndroidManifest.xml
MySecondReceiver
1 package com.example.reza.broadcastreceiversdemo;
3 import android.content.BroadcastReceiver;
5 /**
6 * Created by Reza on 2018-03-23.
7 */
9 public class MySecondReceiver extends BroadcastReceiver{
10 }
app
manifests
AndroidManifest.xml
java
com.example.reza.broadcastreceiv
MainActivity
MyFirstReceiver
MySecondReceiver
com.example.reza.broadcastreceiv
com.example.reza.broadcastreceiv
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
4: Run TODO 6: Logcat Android Profiler Terminal 0: Messages Event Log Gradle Console
Class 'MySecondReceiver' must either be declared abstract or implement abstract method 'onReceive(Context, Intent)' in 'BroadcastReceiver'
9:56 CRLF UTF-8 Context: <no context>

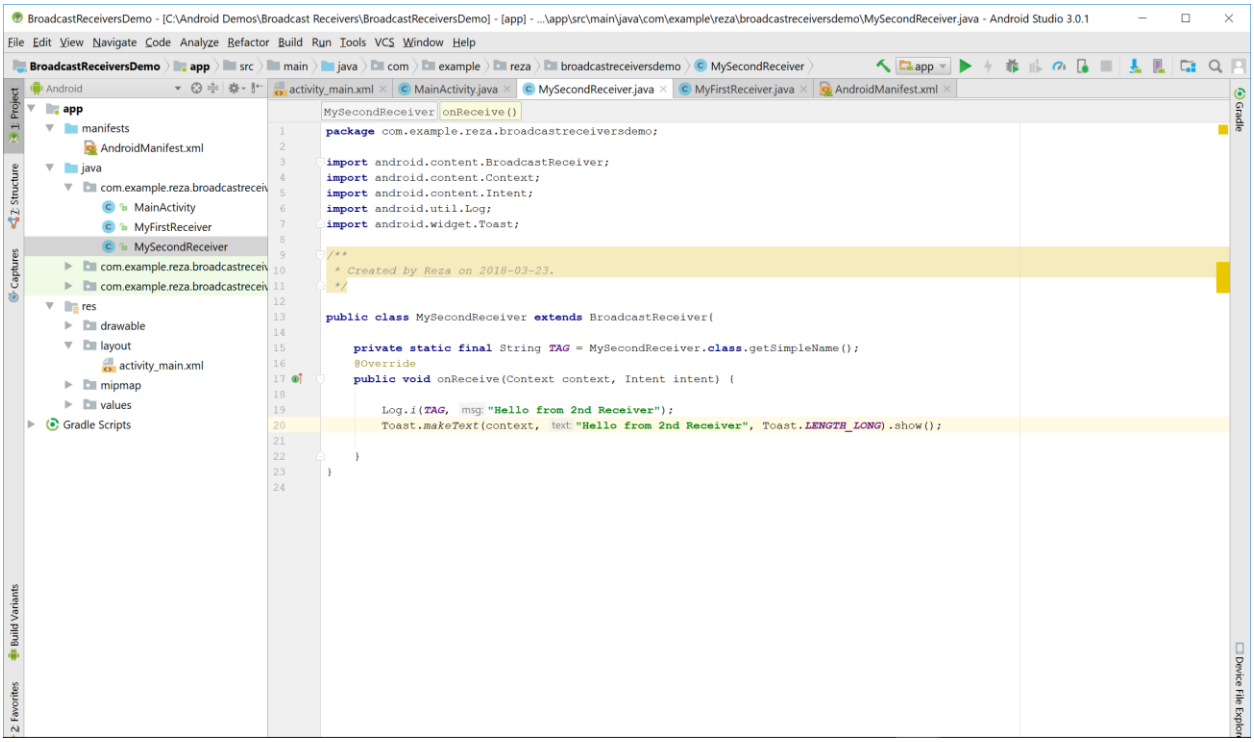

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\MySecondReceiver.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main java com example reza broadcastreceiversdemo MySecondReceiver
activity_main.xml MainActivity.java MySecondReceiver.java MyFirstReceiver.java AndroidManifest.xml
MySecondReceiver
package com.example.reza.broadcastreceiversdemo;
import android.content.BroadcastReceiver;
/**
* Created by Reza on 2018-03-23.
*/
public class MySecondReceiver extends BroadcastReceiver{
}
Implement methods
Make 'MySecondReceiver' abstract
Add broadcast receiver to manifest
Create Test
Create subclass
Make package-private
app
manifests
AndroidManifest.xml
java
com.example.reza.broadcastreceiv
MainActivity
MyFirstReceiver
MySecondReceiver
com.example.reza.broadcastreceiv
com.example.reza.broadcastreceiv
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
4: Run TODO 6: Logcat Android Profiler Terminal 0: Messages Event Log Gradle Console
Class 'MySecondReceiver' must either be declared abstract or implement abstract method 'onReceive(Context, Intent)' in 'BroadcastReceiver'
9:14 CRLF UTF-8 Context: <no context>


Select Methods to Implement
android.content.BroadcastReceiver
onReceive(context:Context, intent:Intent):void
Copy JavaDoc
Insert @Override
OK
Cancel

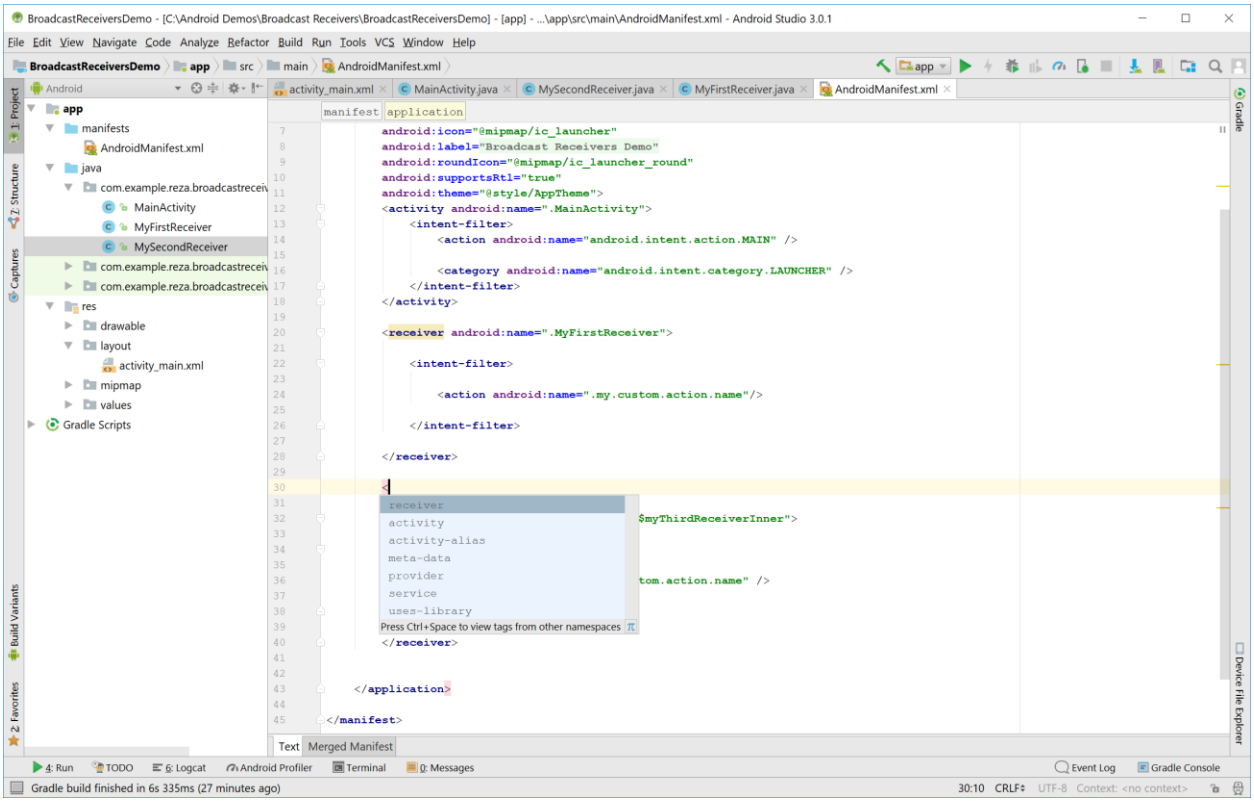

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\MySecondReceiver.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main java com example reza broadcastreceiversdemo MySecondReceiver
activity_main.xml MainActivity.java MySecondReceiver.java MyFirstReceiver.java AndroidManifest.xml
MySecondReceiver onReceive()
app
manifests
AndroidManifest.xml
java
com.example.reza.broadcastreceiv
MainActivity
MyFirstReceiver
MySecondReceiver
com.example.reza.broadcastreceiv
com.example.reza.broadcastreceiv
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
1 package com.example.reza.broadcastreceiversdemo;
2
3 import android.content.BroadcastReceiver;
4 import android.content.Context;
5 import android.content.Intent;
6
7 /**
8 * Created by Reza on 2018-03-23.
9 */
10
11 public class MySecondReceiver extends BroadcastReceiver{
12 @Override
13 public void onReceive(Context context, Intent intent) {
14
15 }
16 }
17
4: Run TODO 6: Logcat Android Profiler Terminal 0: Messages Event Log Gradle Console
Gradle build finished in 6s 335ms (23 minutes ago)
14:9 CRLF UTF-8 Context: <no context>

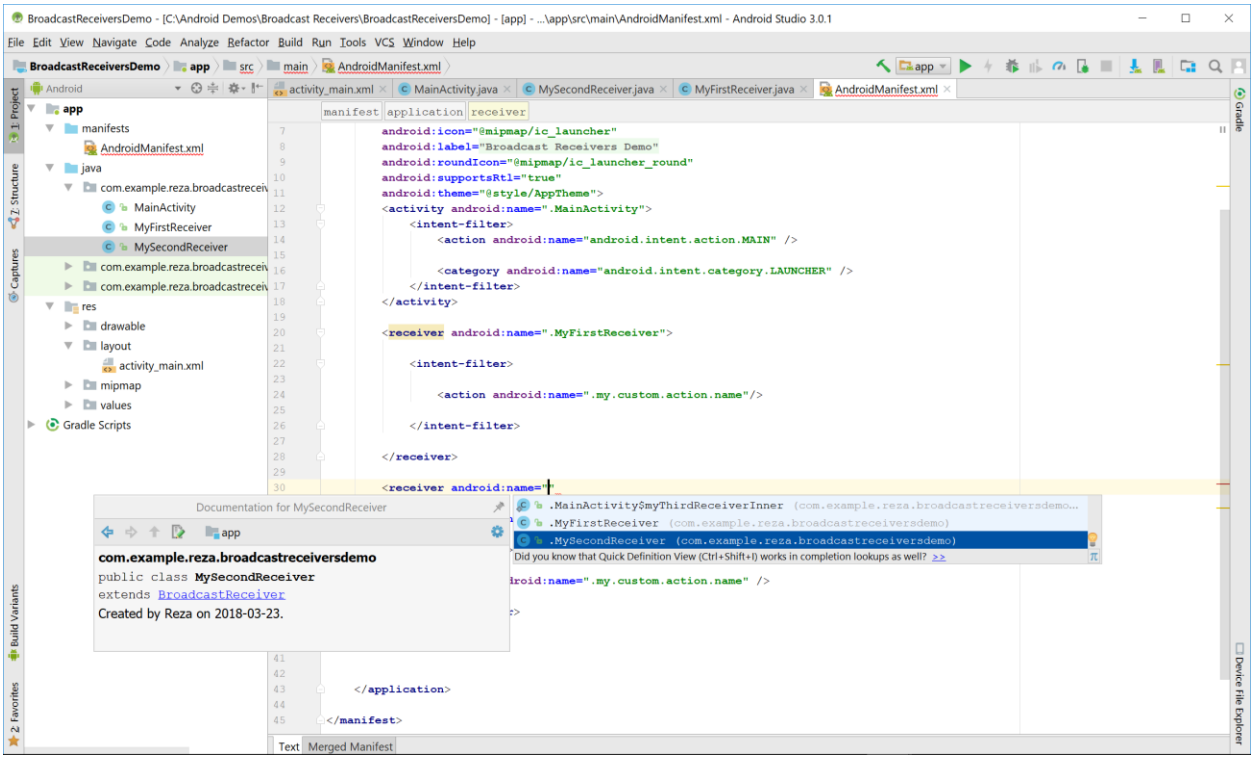
Now add the string tag and the toast message as you did for the previous receivers as shown:

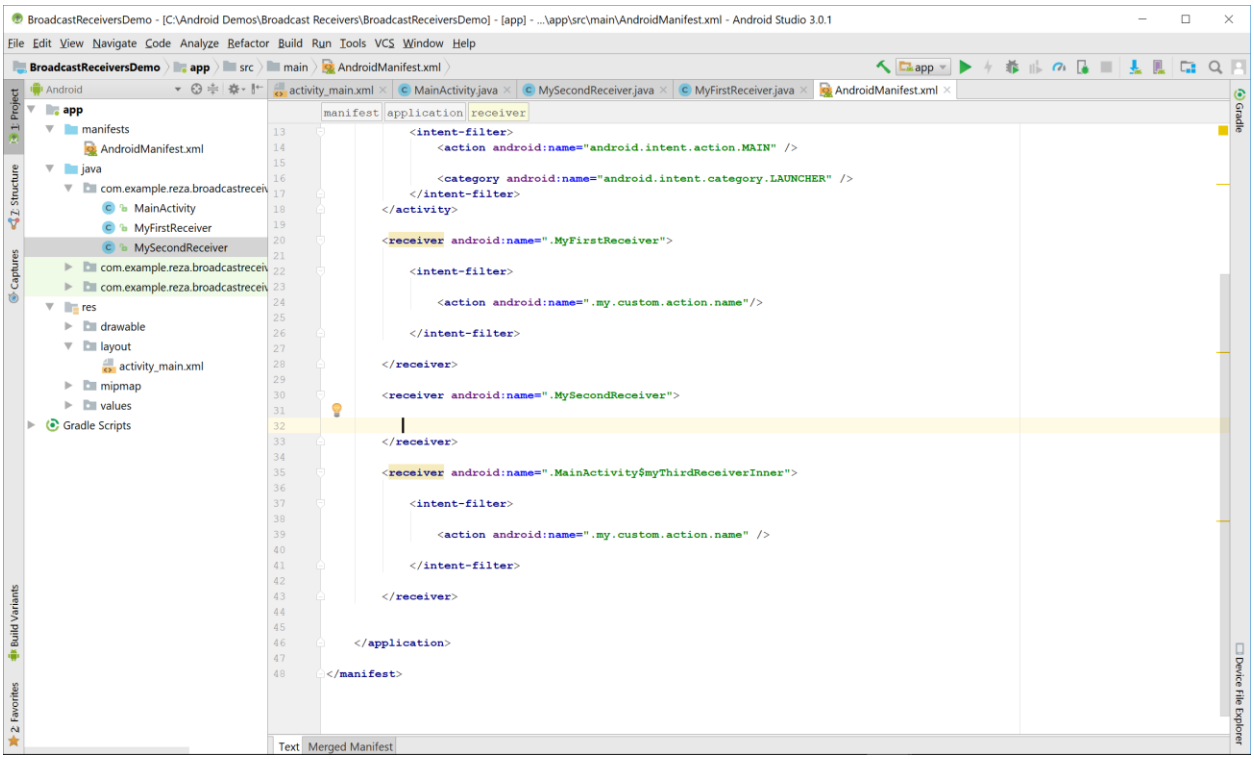




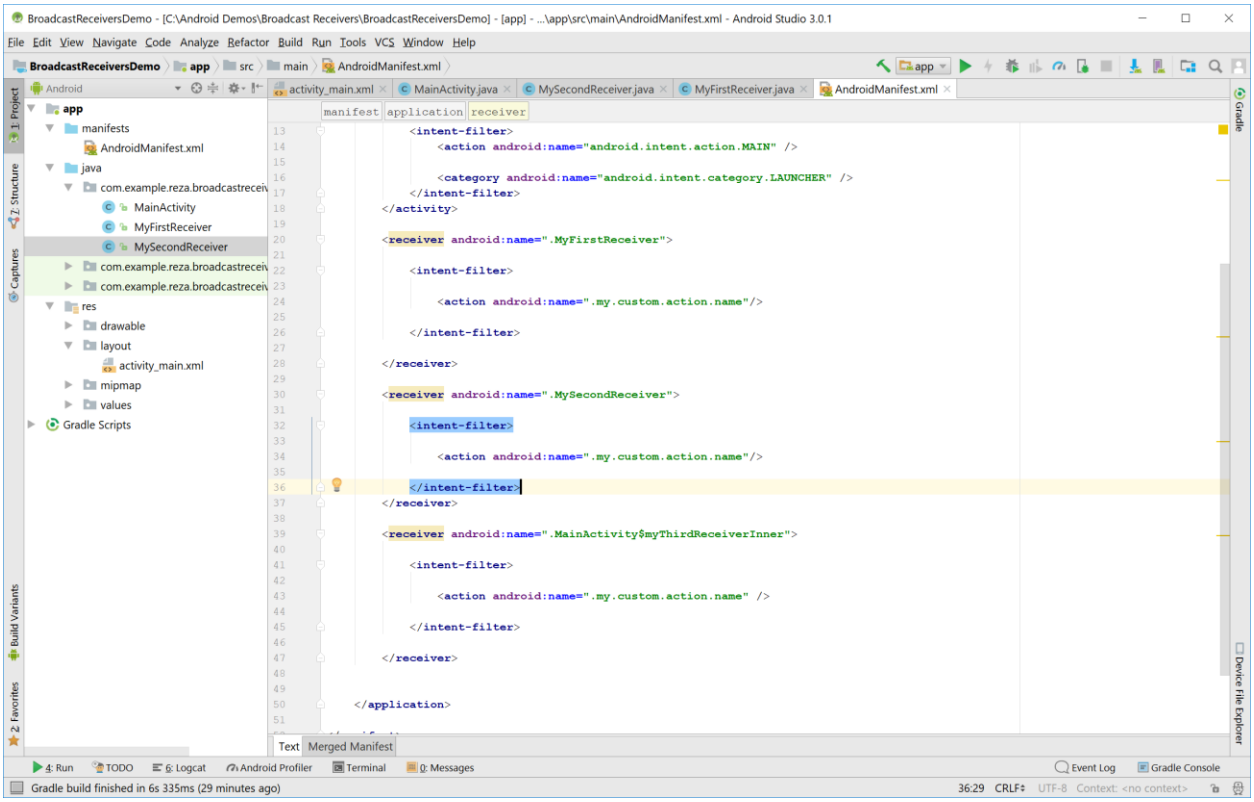
Now declare your receiver inside android manifest file as you have done it before as shown:




BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\AndroidManifest.xml - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main AndroidManifest.xml
activity_main.xml MainActivity.java MySecondReceiver.java MyFirstReceiver.java AndroidManifest.xml
manifest application receiver
android:icon="@mipmap/ic_launcher"
android:label="Broadcast Receivers Demo"
android:roundIcon="@mipmap/ic_launcher_round"
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>
<receiver android:name=".MyFirstReceiver">
<intent-filter>
<action android:name=".my.custom.action.name"/>
</intent-filter>
</receiver>
<receiver android:name="
Documentation for MySecondReceiver
com.example.reza.broadcastreceiversdemo
public class MySecondReceiver
extends BroadcastReceiver
Created by Reza on 2018-03-23.
.MainActivity$myThirdReceiverInner (com.example.reza.broadcastreceiversdemo...
.MyFirstReceiver (com.example.reza.broadcastreceiversdemo)
.MySecondReceiver (com.example.reza.broadcastreceiversdemo)
Did you know that Quick Definition View (Ctrl+Shift+I) works in completion lookups as well? >>
</application>
</manifest>
Text Merged Manifest

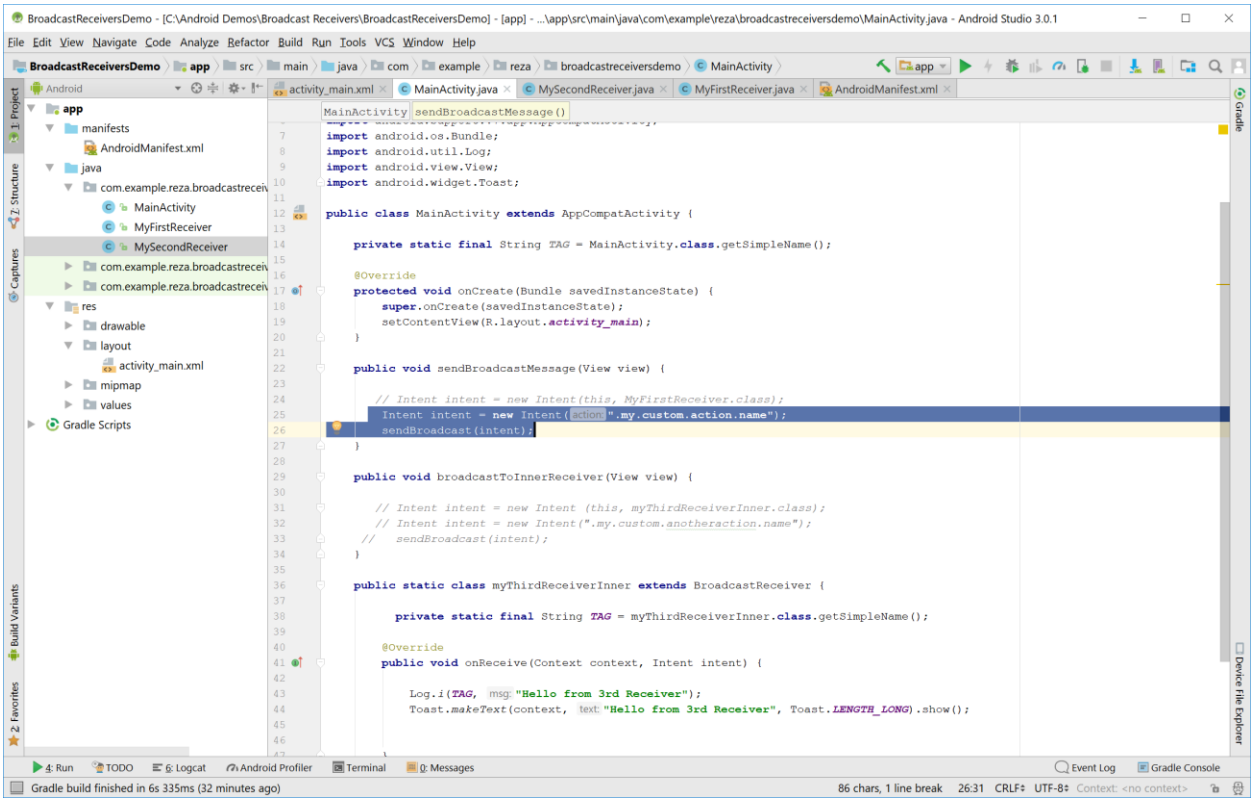

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\AndroidManifest.xml - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main AndroidManifest.xml
activity_main.xml MainActivity.java MySecondReceiver.java MyFirstReceiver.java AndroidManifest.xml
manifest application receiver
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<receiver android:name=".MyFirstReceiver">
<intent-filter>
<action android:name=".my.custom.action.name"/>
</intent-filter>
</receiver>
<receiver android:name=".MySecondReceiver">
</receiver>
<receiver android:name=".MainActivity$myThirdReceiverInner">
<intent-filter>
<action android:name=".my.custom.action.name" />
</intent-filter>
</receiver>
</application>
</manifest>
Text Merged Manifest

We are going to use the exact intent-filter and action as the other two receivers as shown so just copy the code from the other receiver and paste it:

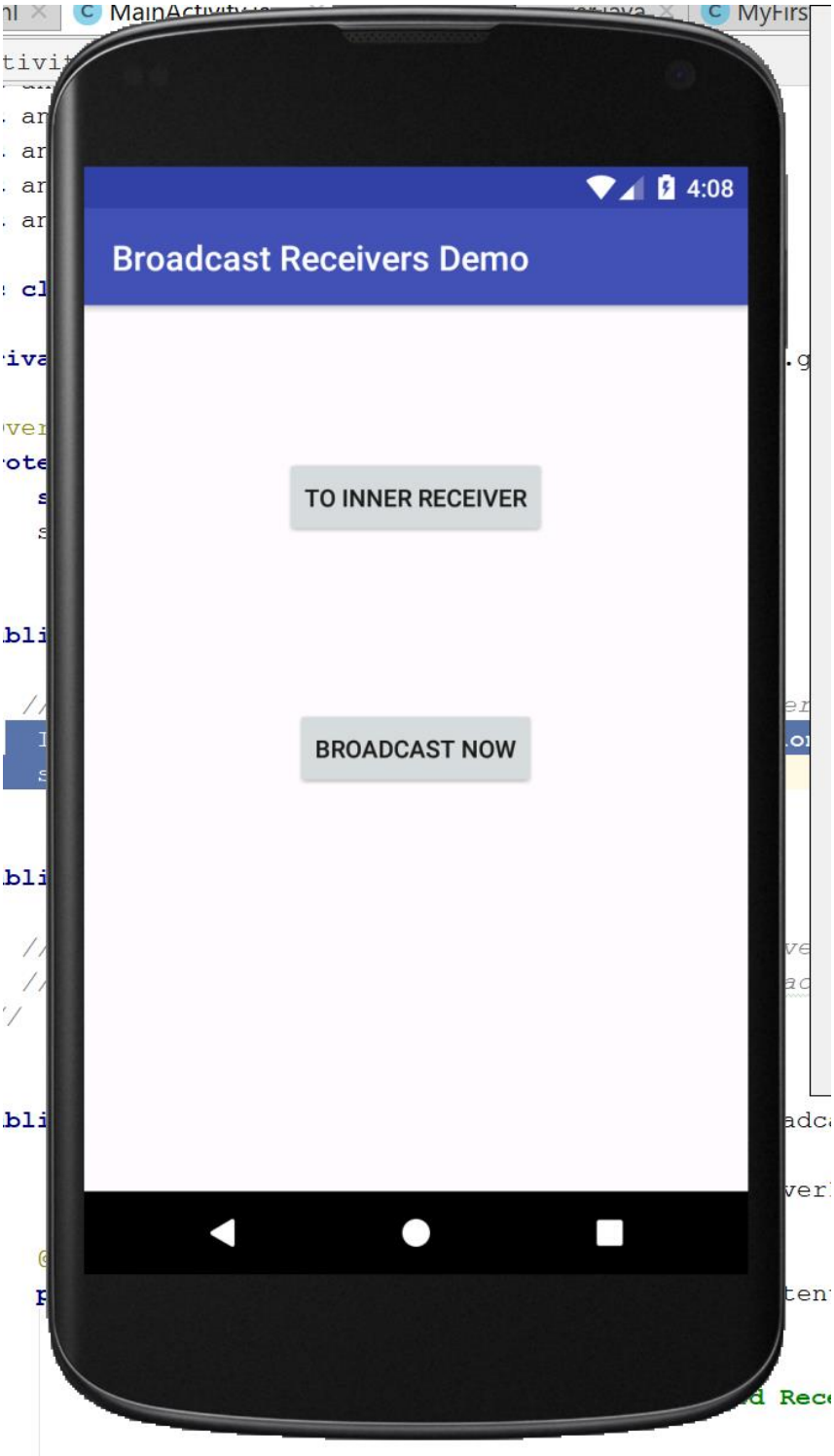


Now you have three receivers and all of them have been registered with ".my.custom.action.name" action.

Now when the sendBroadcastMessage() method in the main activity is called and the two highlighted statements in it are executed, then Android operating system sends the broadcast to all of these three receivers:

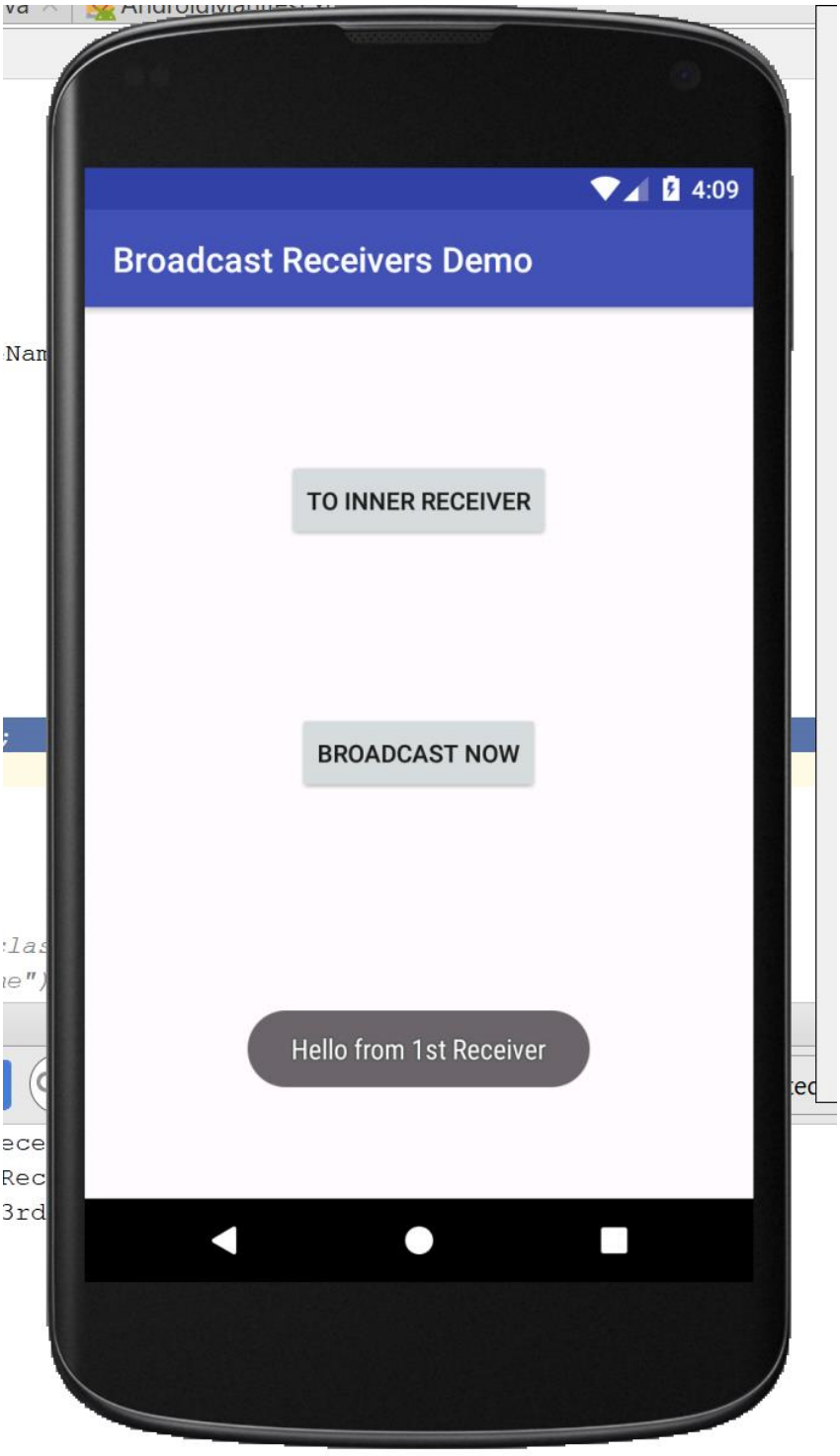


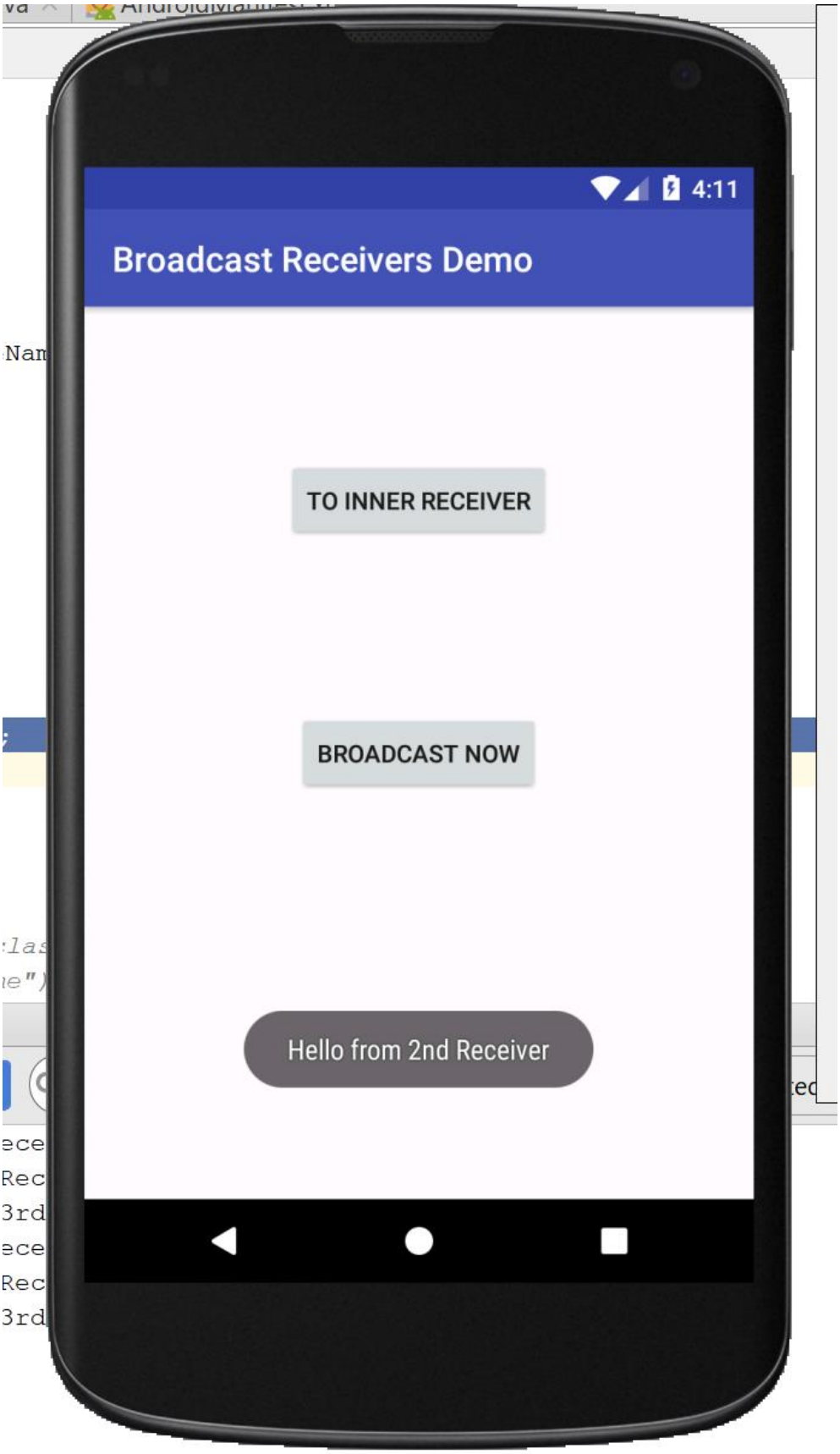
Now run the app and test to see which receiver is called first:

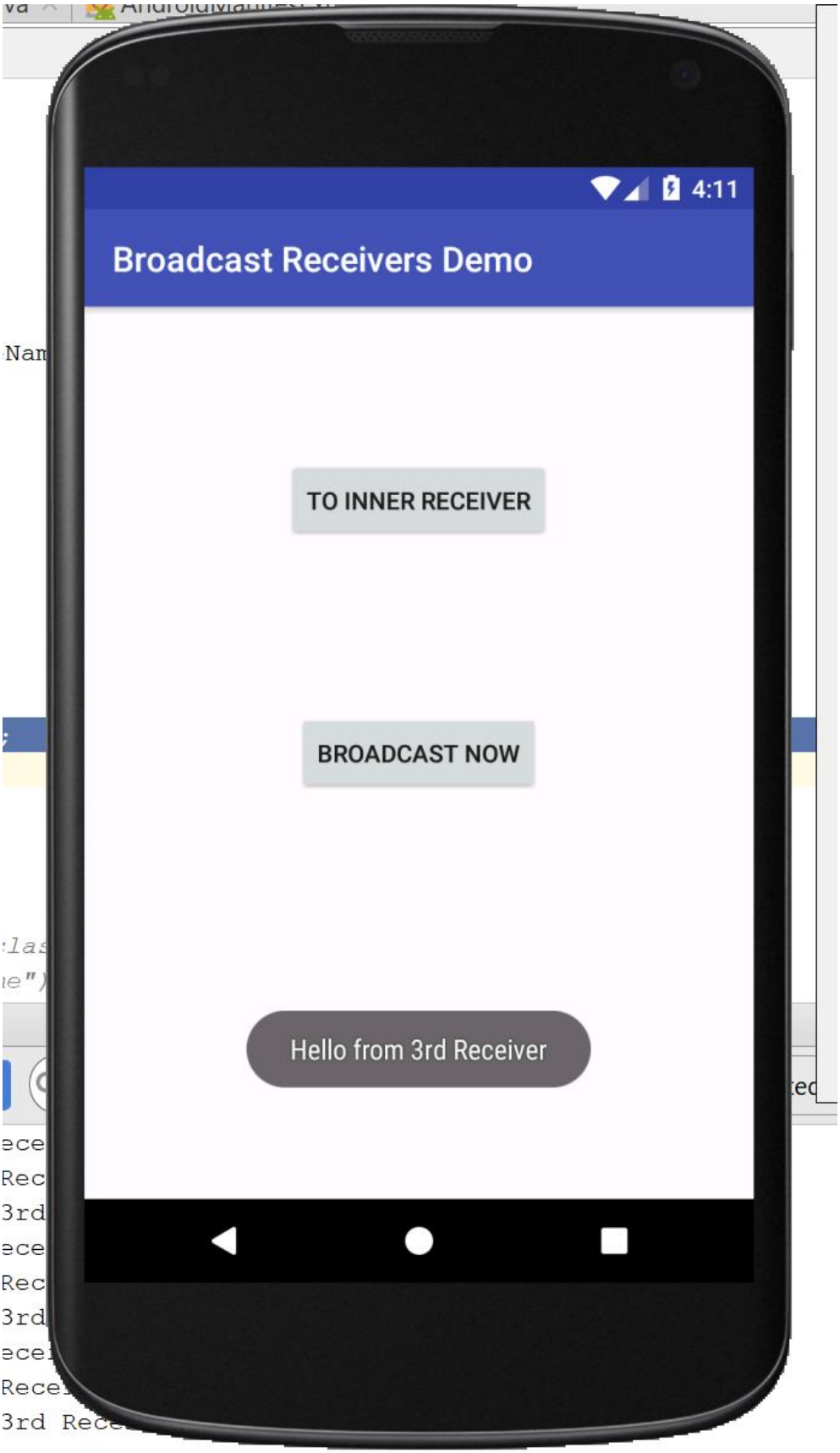


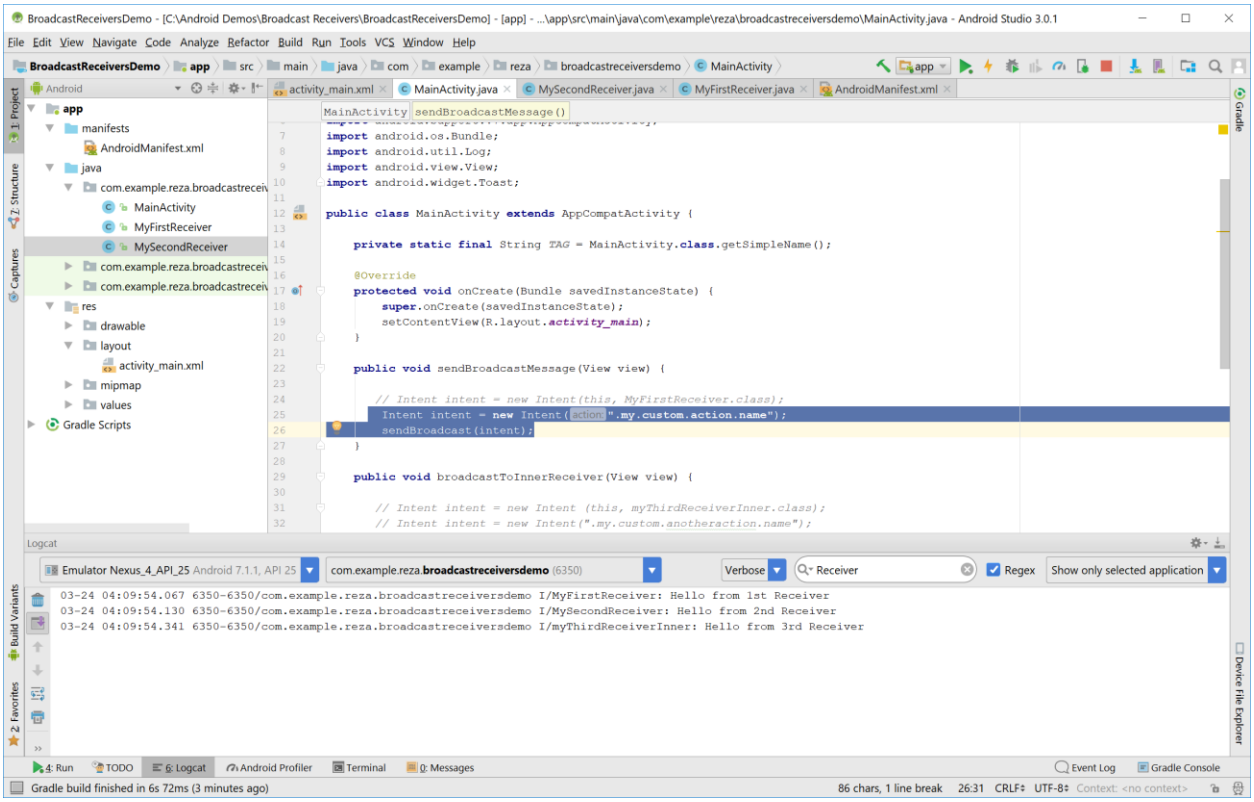
Note that now with clicking the “Broadcast Now” button, we will be sending the broadcast to all of our three receivers so what is the order that these guys are going to receive the broadcast?

Let’s try it. Click the “Broadcast Now” button:

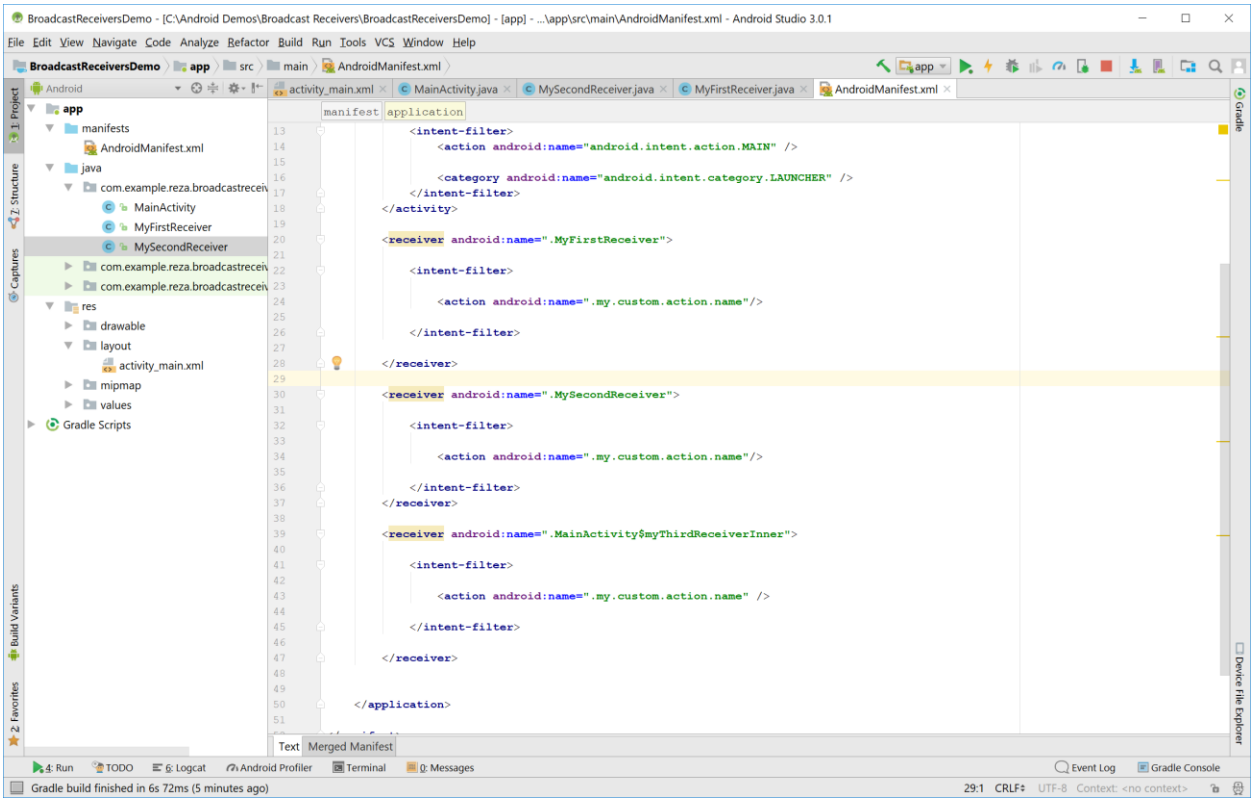



4:11
Broadcast Receivers Demo
TO INNER RECEIVER
BROADCAST NOW
Hello from 2nd Receiver

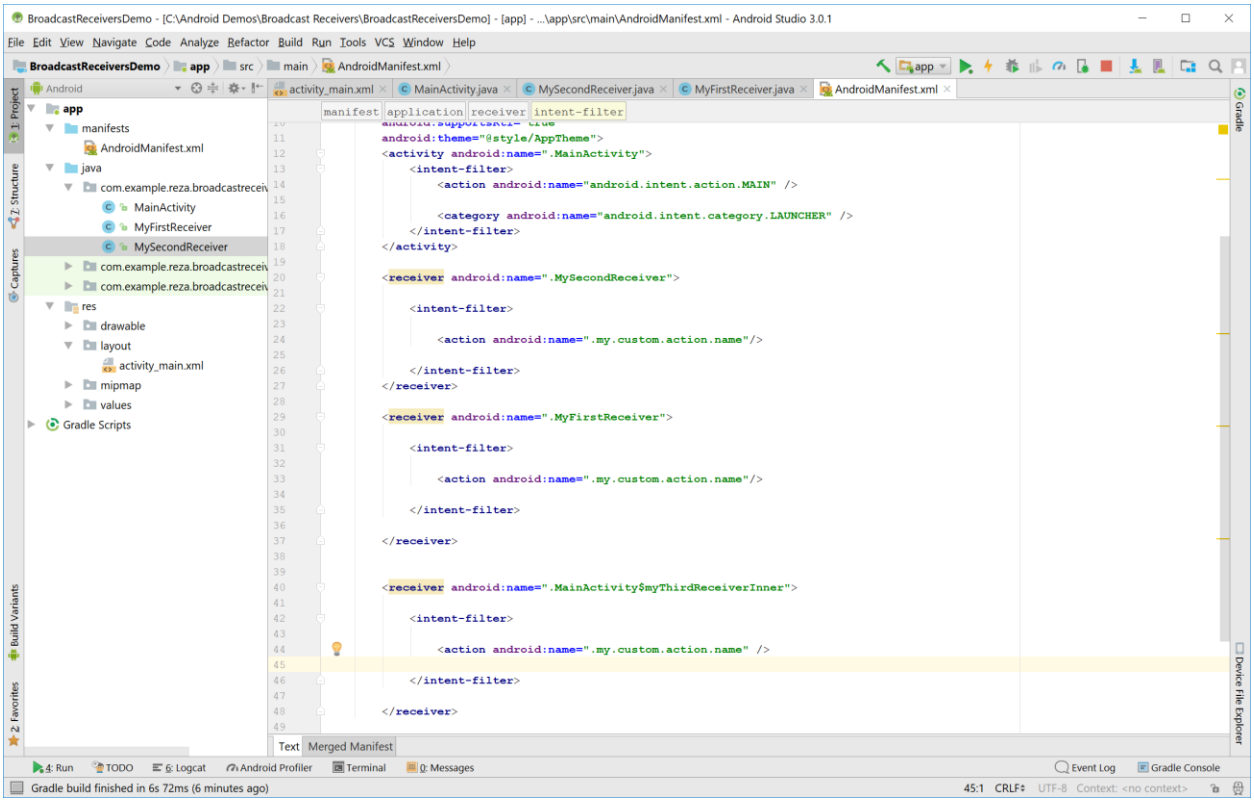

4:11
Broadcast Receivers Demo
TO INNER RECEIVER
BROADCAST NOW
Hello from 3rd Receiver


BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\MainActivity.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
activity_main.xml
MainActivity.java
MySecondReceiver.java
MyFirstReceiver.java
AndroidManifest.xml
MainActivity sendBroadcastMessage()
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
private static final String TAG = MainActivity.class.getSimpleName();
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}
public void sendBroadcastMessage(View view) {
// Intent intent = new Intent(this, MyFirstReceiver.class);
Intent intent = new Intent("my.custom.action.name");
sendBroadcast(intent);
}
public void broadcastToInnerReceiver(View view) {
// Intent intent = new Intent (this, myThirdReceiverInner.class);
// Intent intent = new Intent(".my.custom.anotheraction.name");
Logcat
Emulator Nexus_4_API_25 Android 7.1.1, API 25
com.example.reza.broadcastreceiversdemo (6350)
Verbose
Receiver
Regex
Show only selected application
03-24 04:09:54.067 6350-6350/com.example.reza.broadcastreceiversdemo I/MyFirstReceiver: Hello from 1st Receiver
03-24 04:09:54.130 6350-6350/com.example.reza.broadcastreceiversdemo I/MySecondReceiver: Hello from 2nd Receiver
03-24 04:09:54.341 6350-6350/com.example.reza.broadcastreceiversdemo I/myThirdReceiverInner: Hello from 3rd Receiver
4: Run
TODO
6: Logcat
Android Profiler
Terminal
0: Messages
Event Log
Gradle Console
Gradle build finished in 6s 72ms (3 minutes ago)
86 chars, 1 line break
26:31
CRLF
UTF-8
Context: <no context>

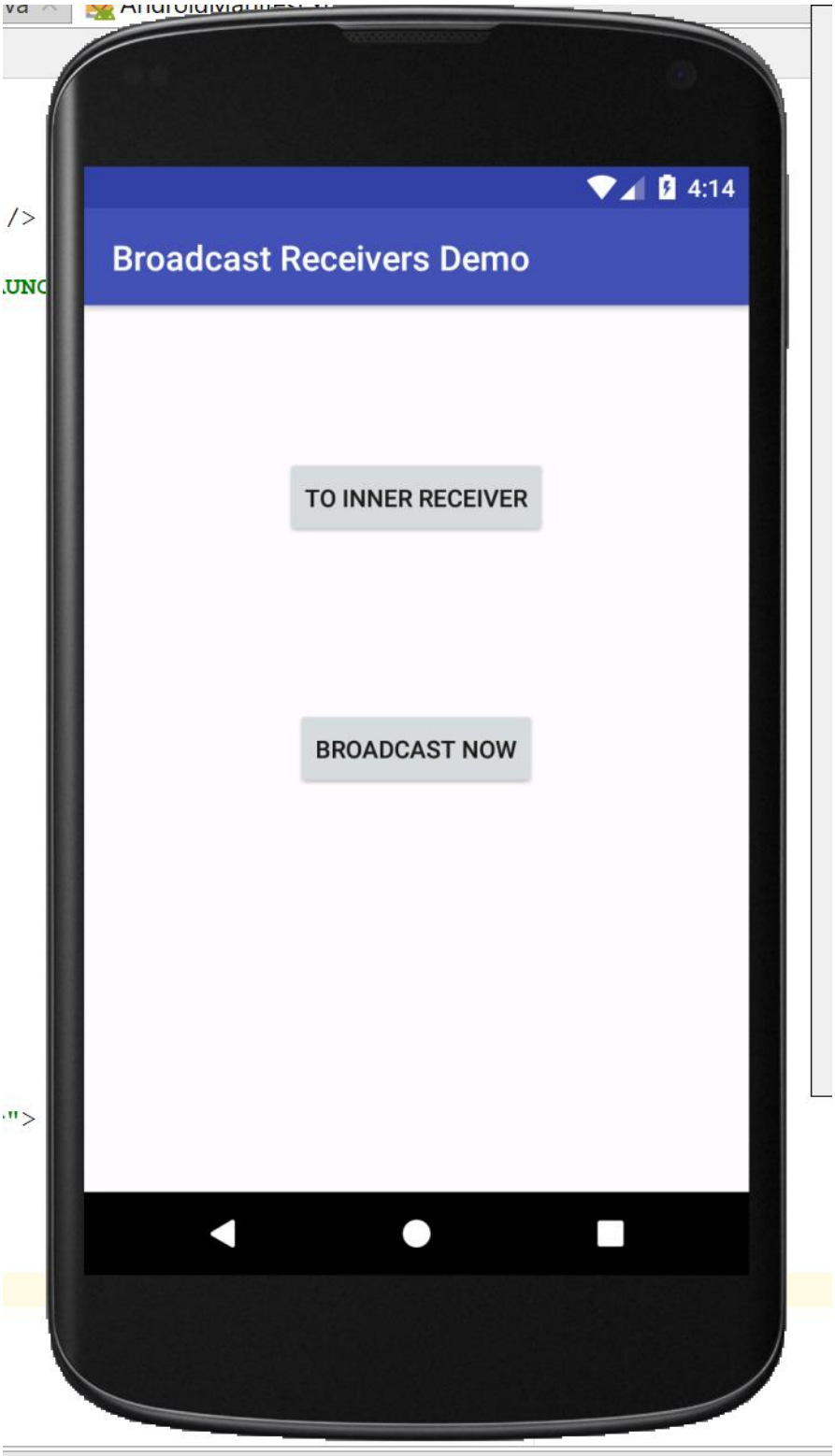
So our receivers received the broadcast in the same order as they were defined in the manifest file as shown:



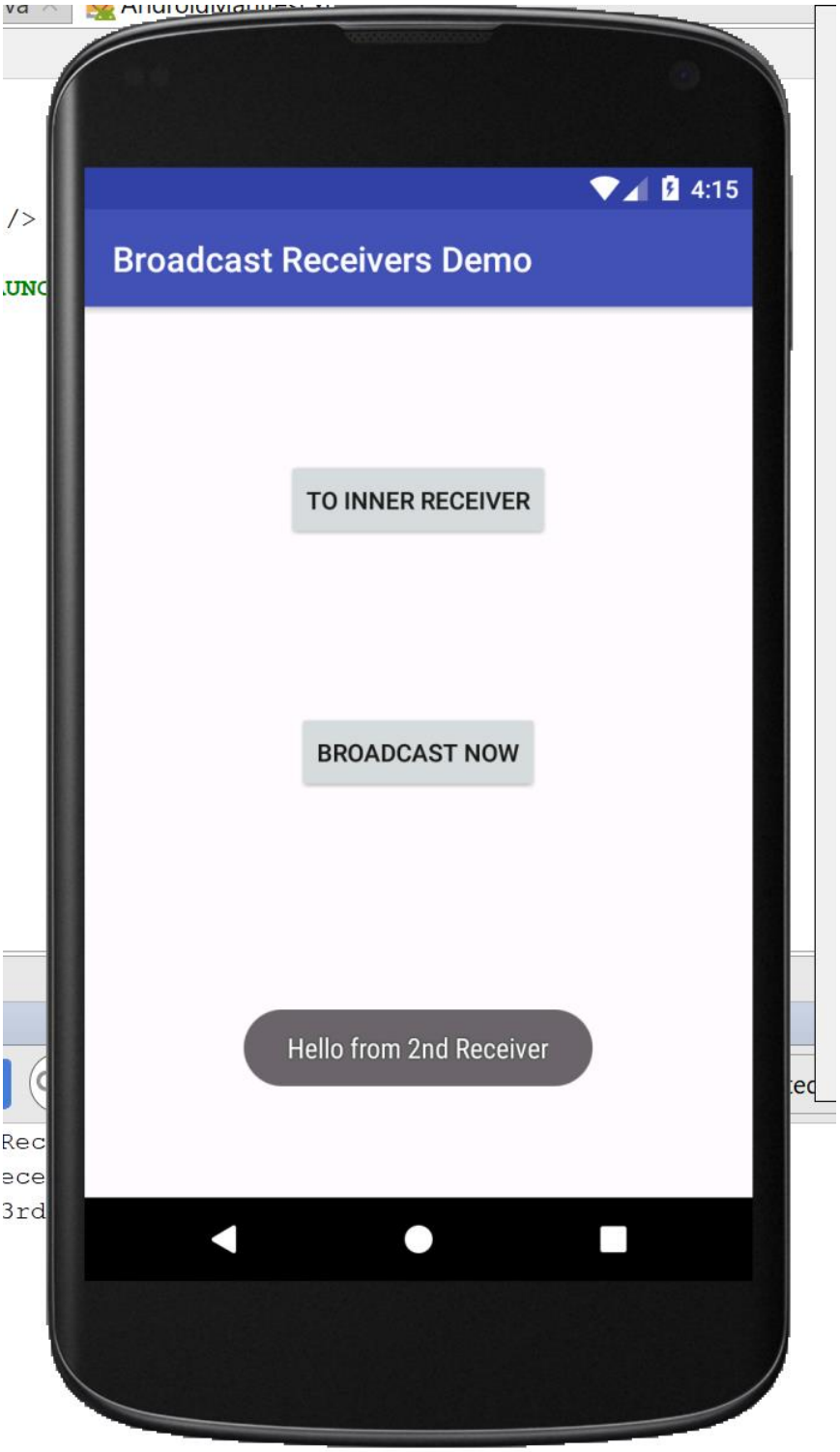
Let’s change the order. Cut the code for the second receiver and put it before the first receiver as shown:

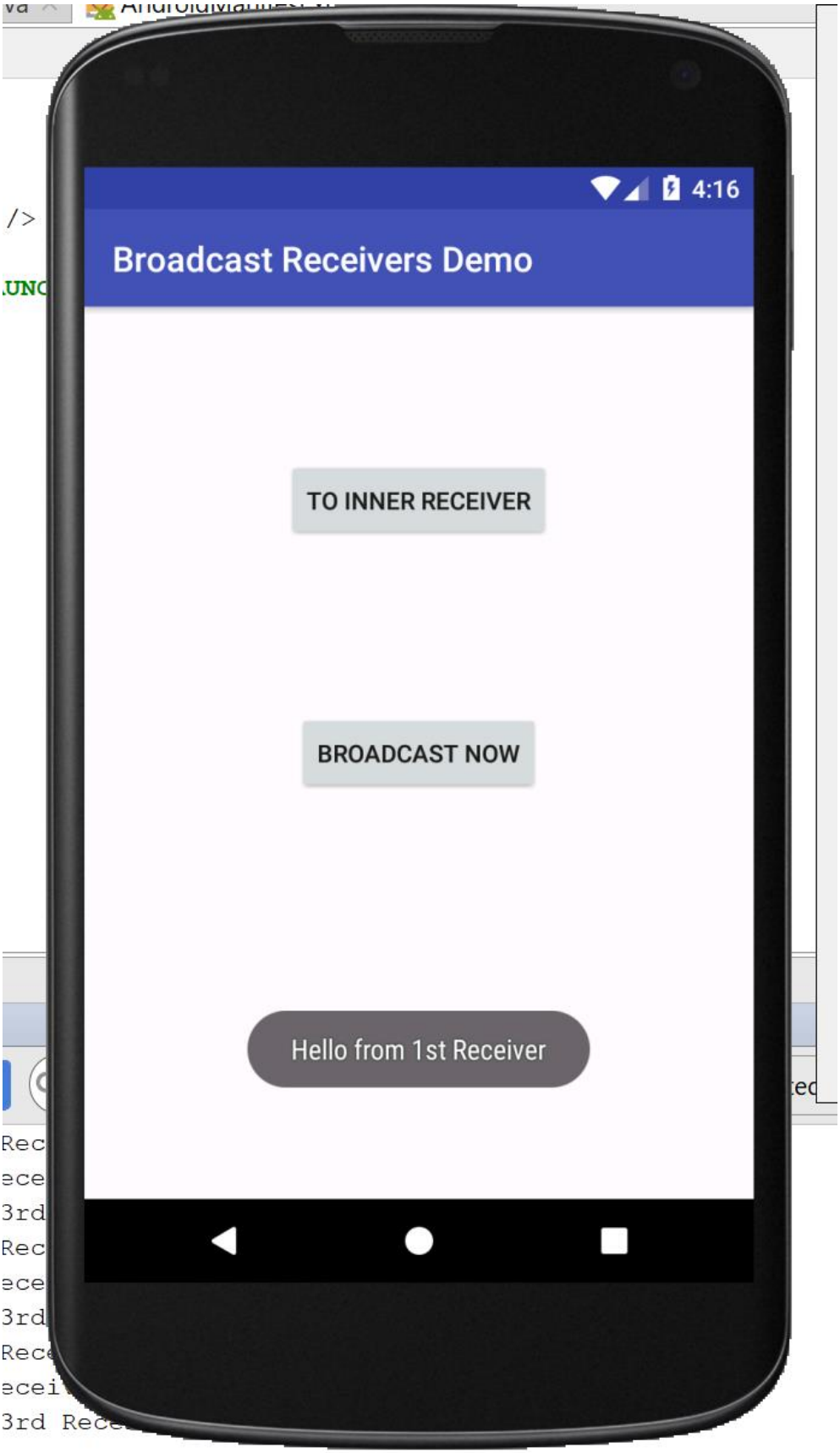


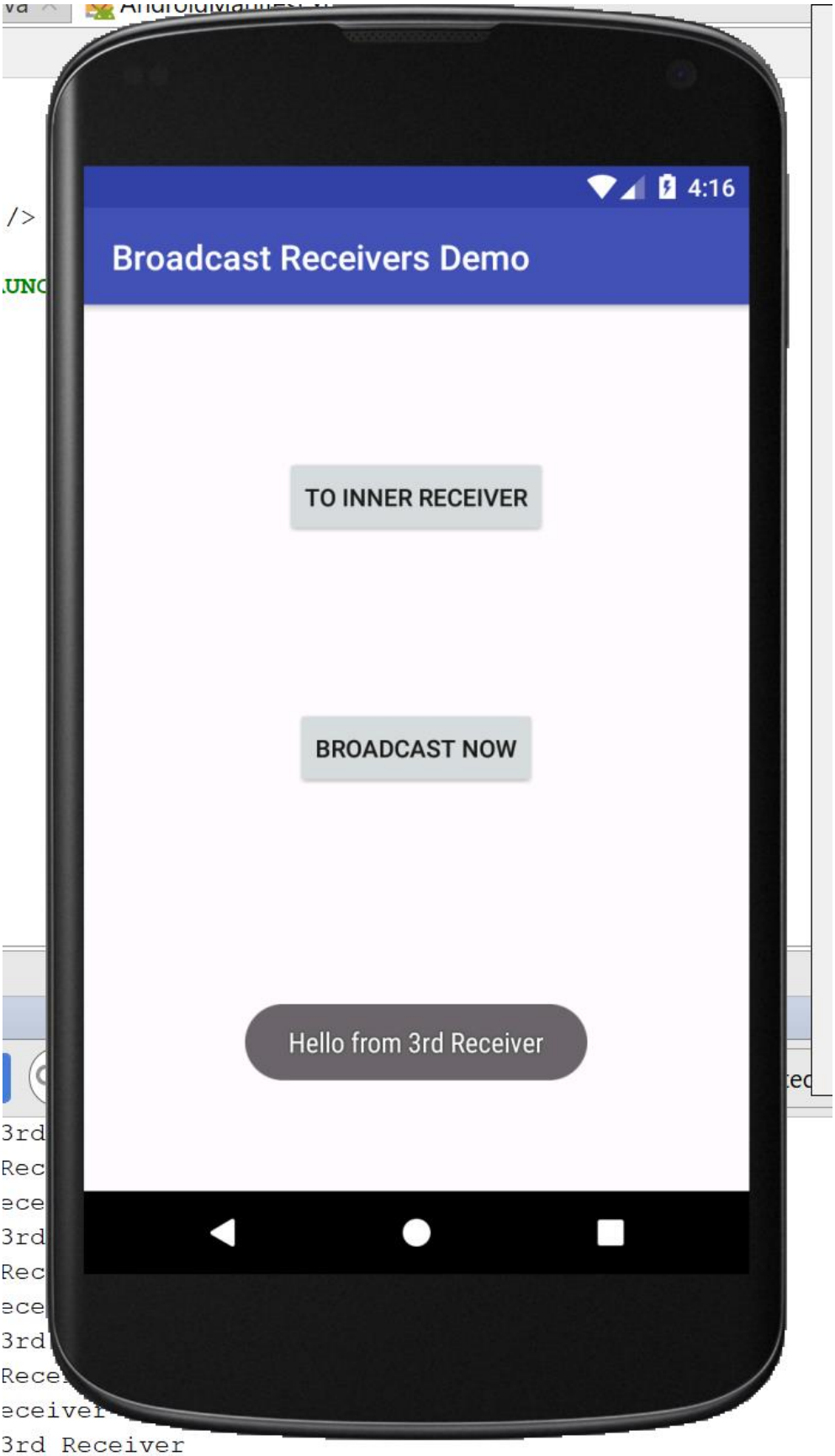
Now let’s run the app:

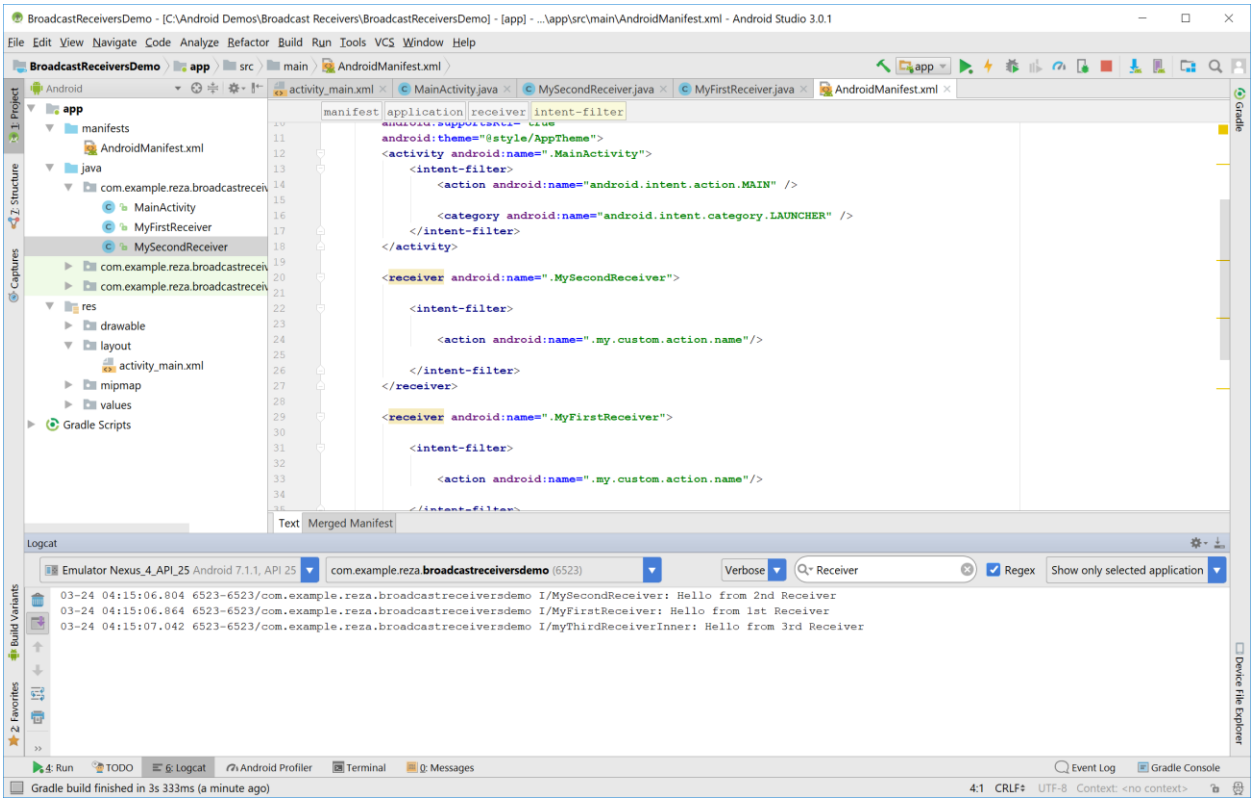


Click the “Broadcast Now” button:



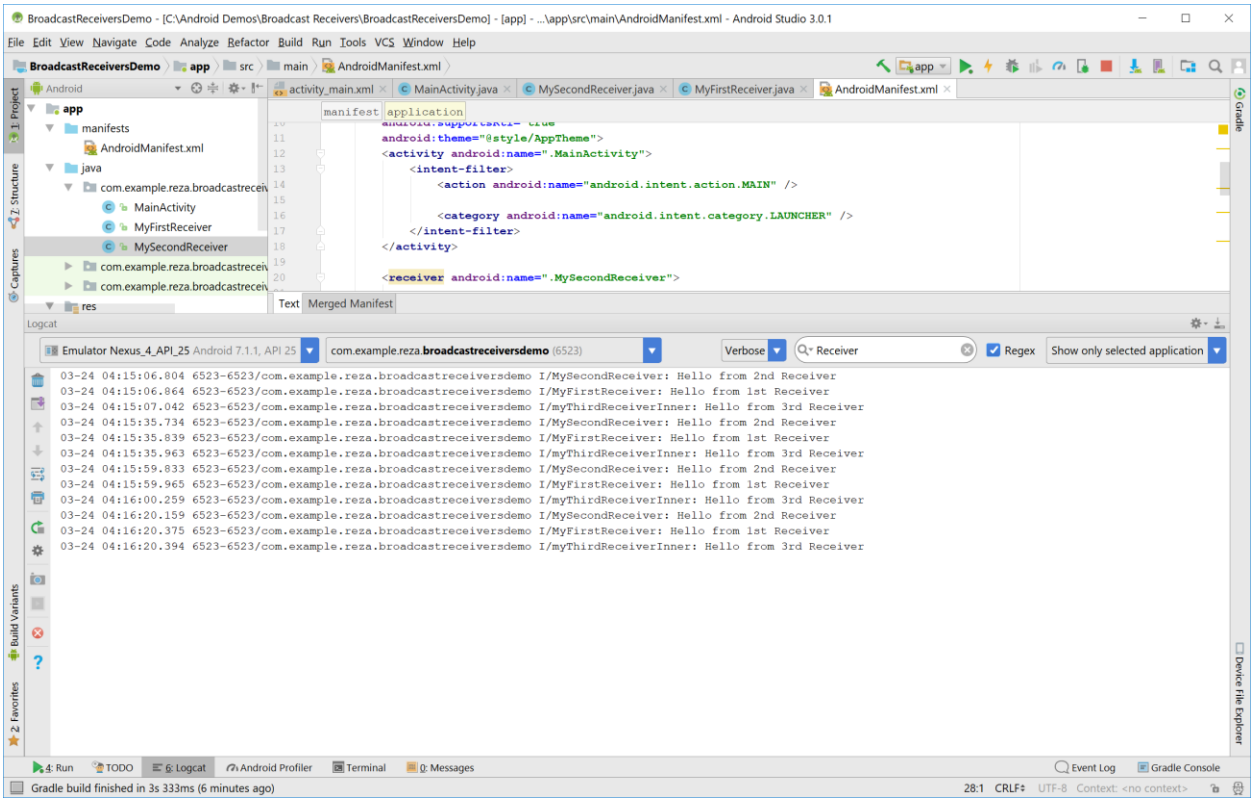

4:16
Broadcast Receivers Demo
TO INNER RECEIVER
BROADCAST NOW
Hello from 1st Receiver


4:16
Broadcast Receivers Demo
TO INNER RECEIVER
BROADCAST NOW
Hello from 3rd Receiver


BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\AndroidManifest.xml - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main AndroidManifest.xml
activity_main.xml MainActivity.java MySecondReceiver.java MyFirstReceiver.java AndroidManifest.xml
manifest application receiver intent-filter
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>
<receiver android:name=".MySecondReceiver">
<intent-filter>
<action android:name=".my.custom.action.name"/>
</intent-filter>
</receiver>
<receiver android:name=".MyFirstReceiver">
<intent-filter>
<action android:name=".my.custom.action.name"/>
Text Merged Manifest
Logcat
Emulator Nexus_4_API_25 Android 7.1.1, API 25
com.example.reza.broadcastreceiversdemo (6523)
Verbose
Receiver
Regex
Show only selected application
03-24 04:15:06.804 6523-6523/com.example.reza.broadcastreceiversdemo I/MySecondReceiver: Hello from 2nd Receiver
03-24 04:15:06.864 6523-6523/com.example.reza.broadcastreceiversdemo I/MyFirstReceiver: Hello from 1st Receiver
03-24 04:15:07.042 6523-6523/com.example.reza.broadcastreceiversdemo I/myThirdReceiverInner: Hello from 3rd Receiver
4: Run TODO 6: Logcat Android Profiler Terminal 0: Messages Event Log Gradle Console
Gradle build finished in 3s 333ms (a minute ago)
4:1 CRLF UTF-8 Context: <no context>

So, our three receivers received the broadcast in the exact order that they are declared in the manifest file.

And this order is preserved and is always the same no matter how many times you send the broadcast by clicking the “Broadcast Now” button as the Logcat window shows:

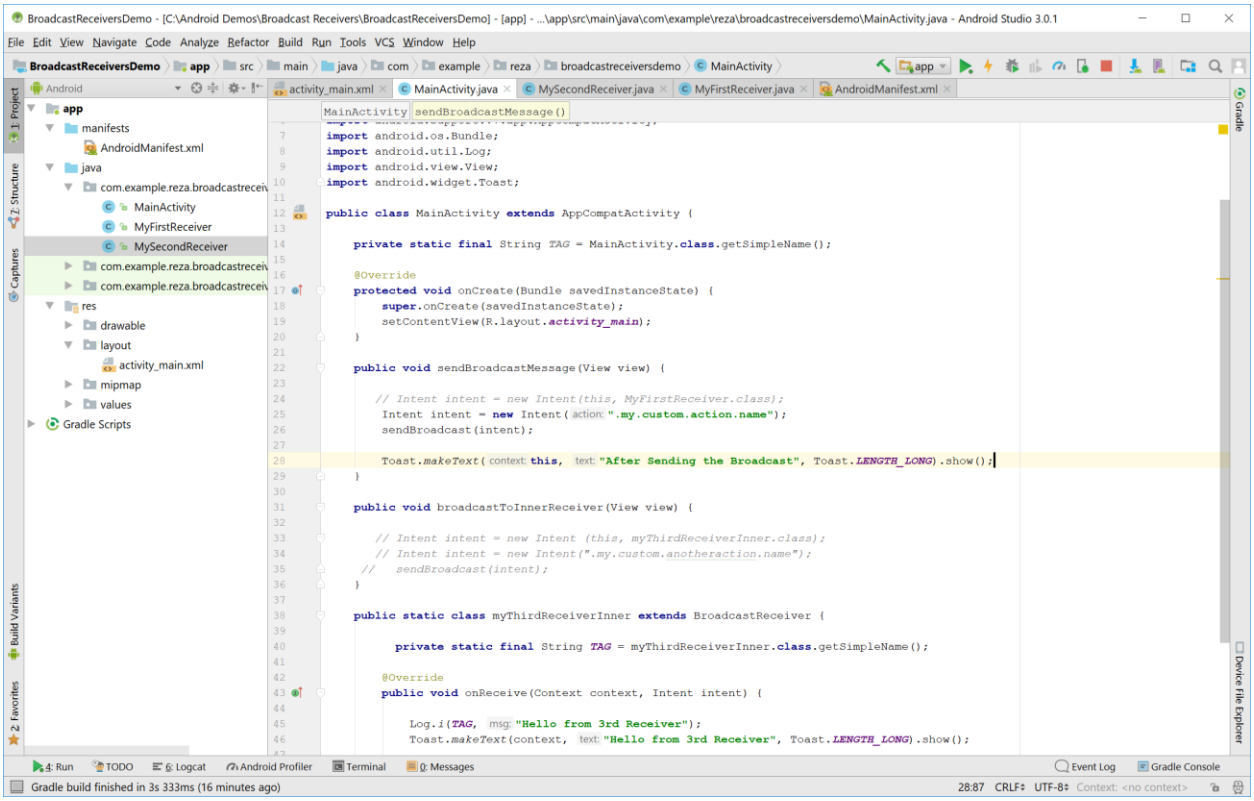


Note that all these receivers have the same action name and that’s the reason that the sequence in which they are declared inside the Androidmanifest.xml file determines the sequence in which the broadcast will be delivered to these receivers.

Broadcasts are sent Asynchronously

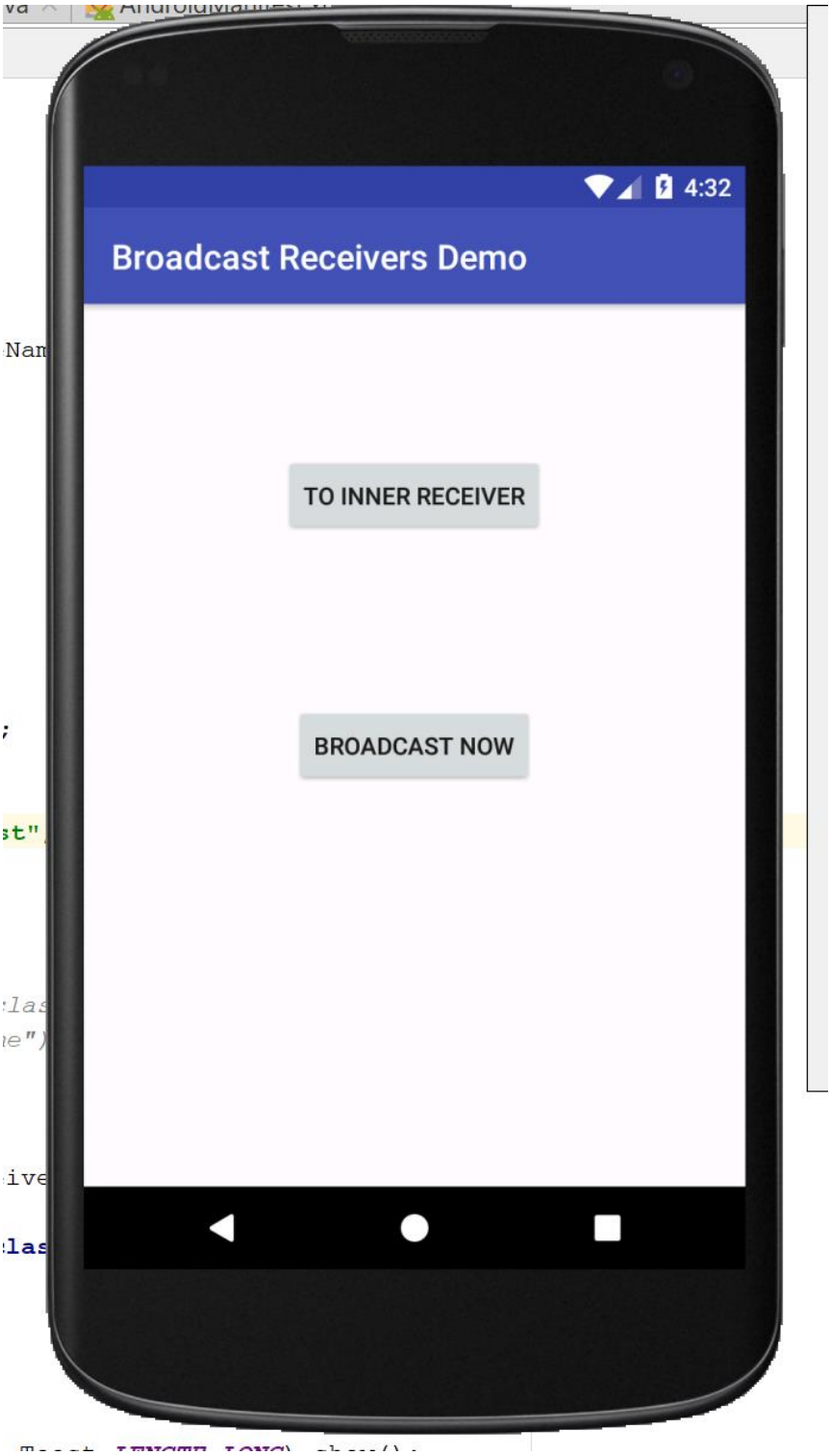
Anytime a broadcast is sent, it is sent asynchronously, another word in the background. To see this in action you can add a toast statement in the code for sendBroadcastMessage() method right after the sendBroadcast(intent); method and you will see that this toast message appears before any receivers receive the broadcast.

Modify the main activity code as shown (add line 28):



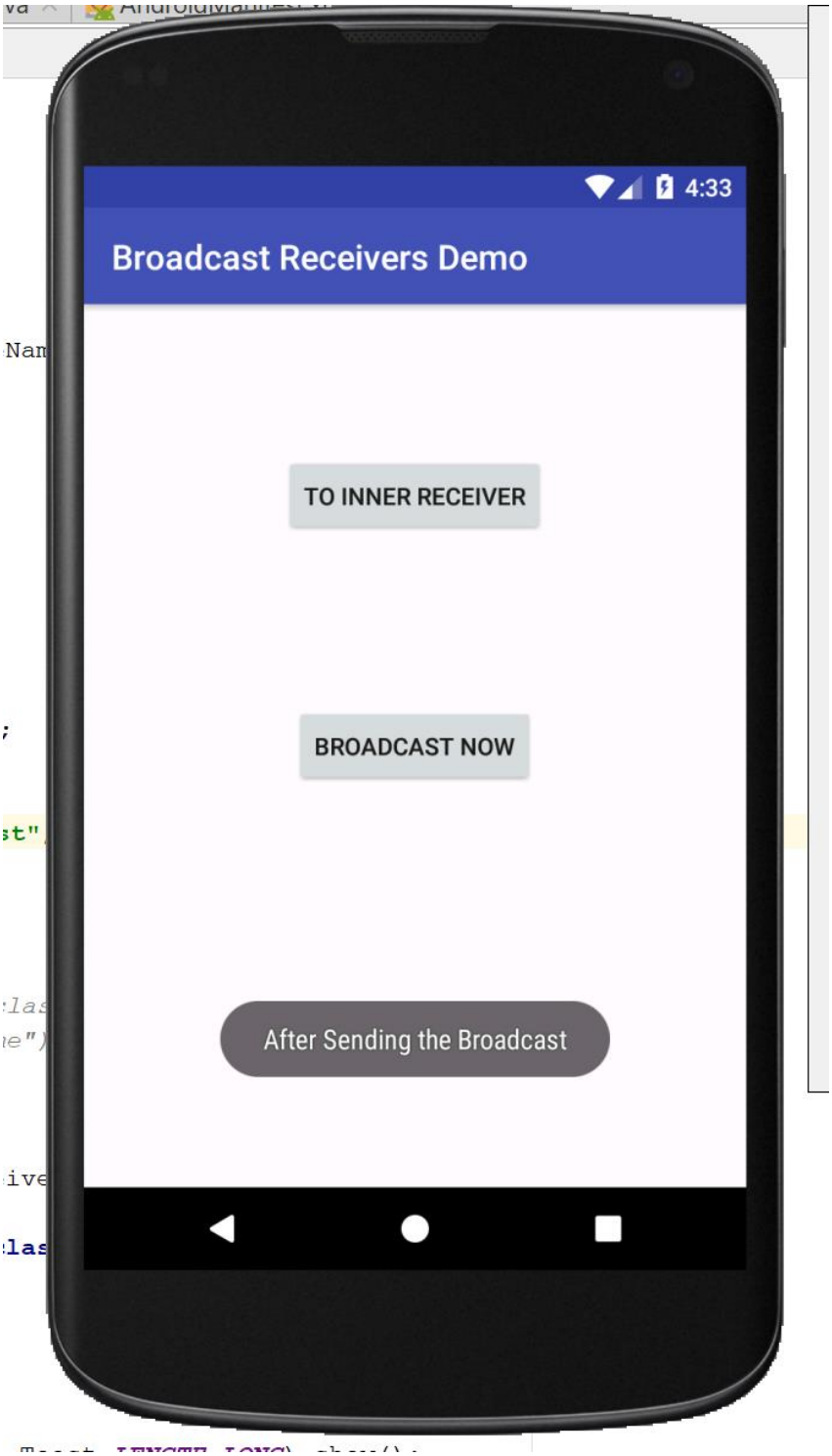
Now line 28 which is going to display a toast will be executed even before line 26 has finished since the program doesn't wait for line 26 to finish hence proving that sending broadcasts is done asynchronously.

Run the app:

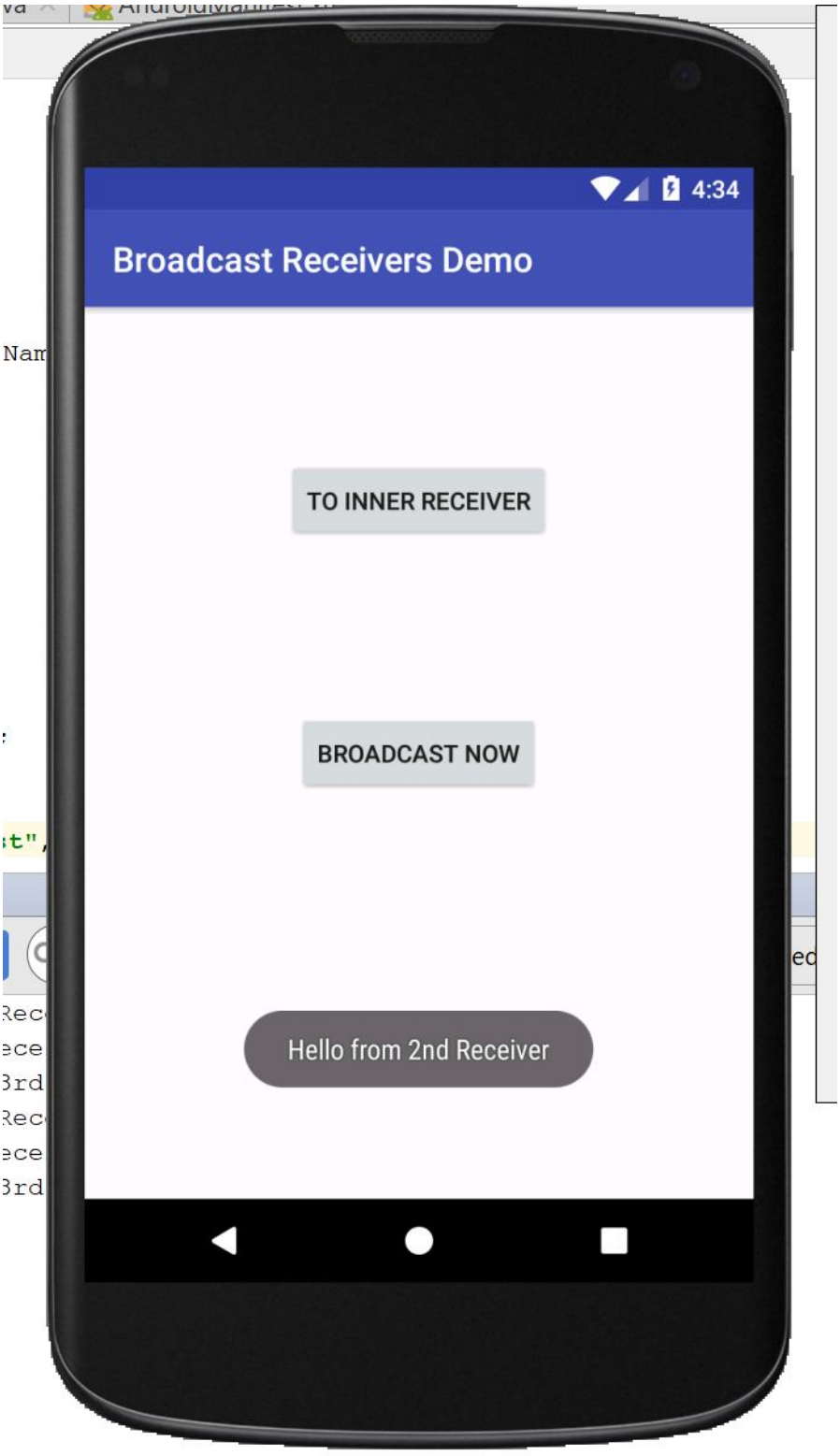

4:32
Broadcast Receivers Demo
TO INNER RECEIVER
BROADCAST NOW

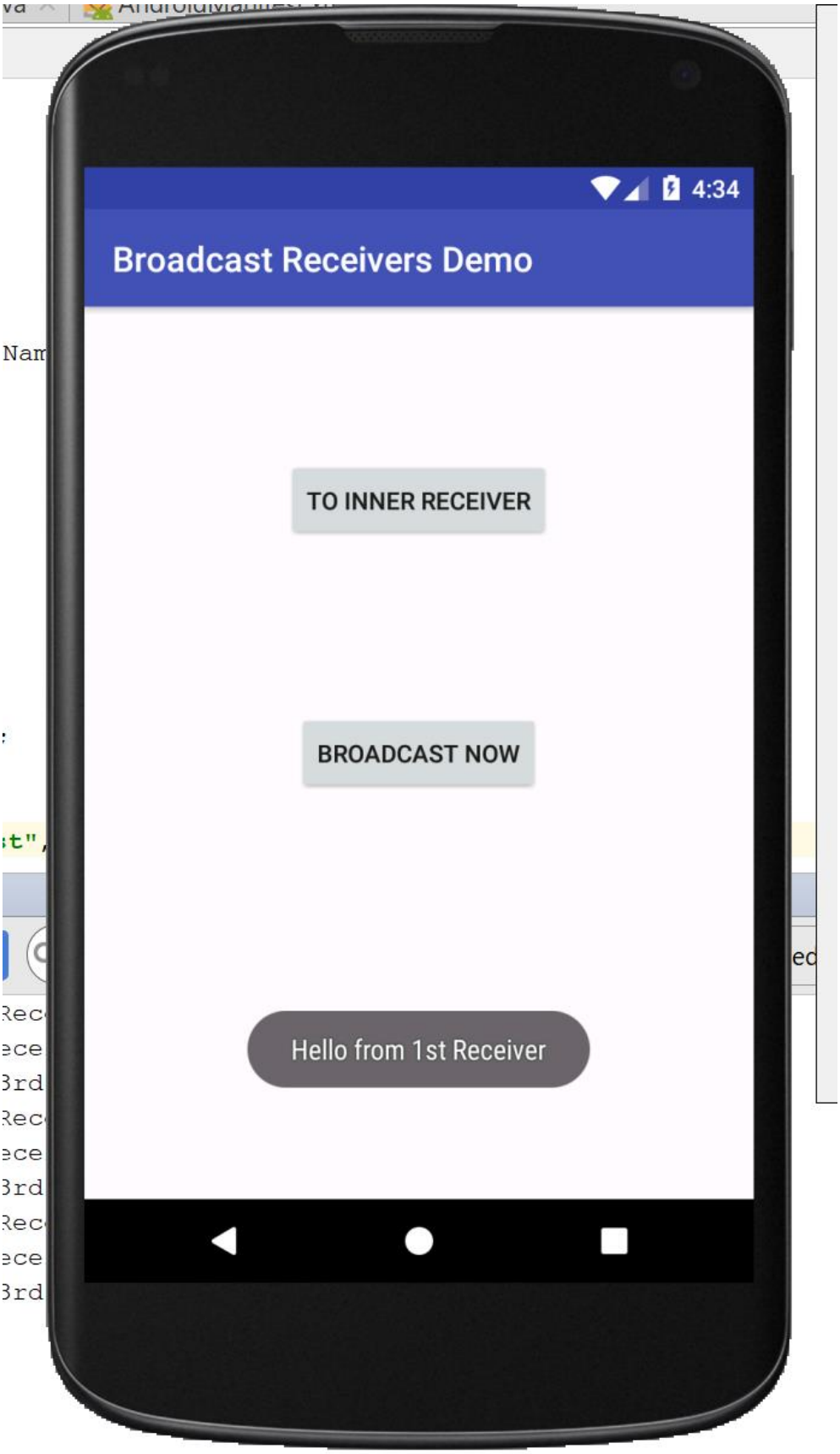
Click the “Broadcast Now” button and here are the toast messages in the order they will be displayed:

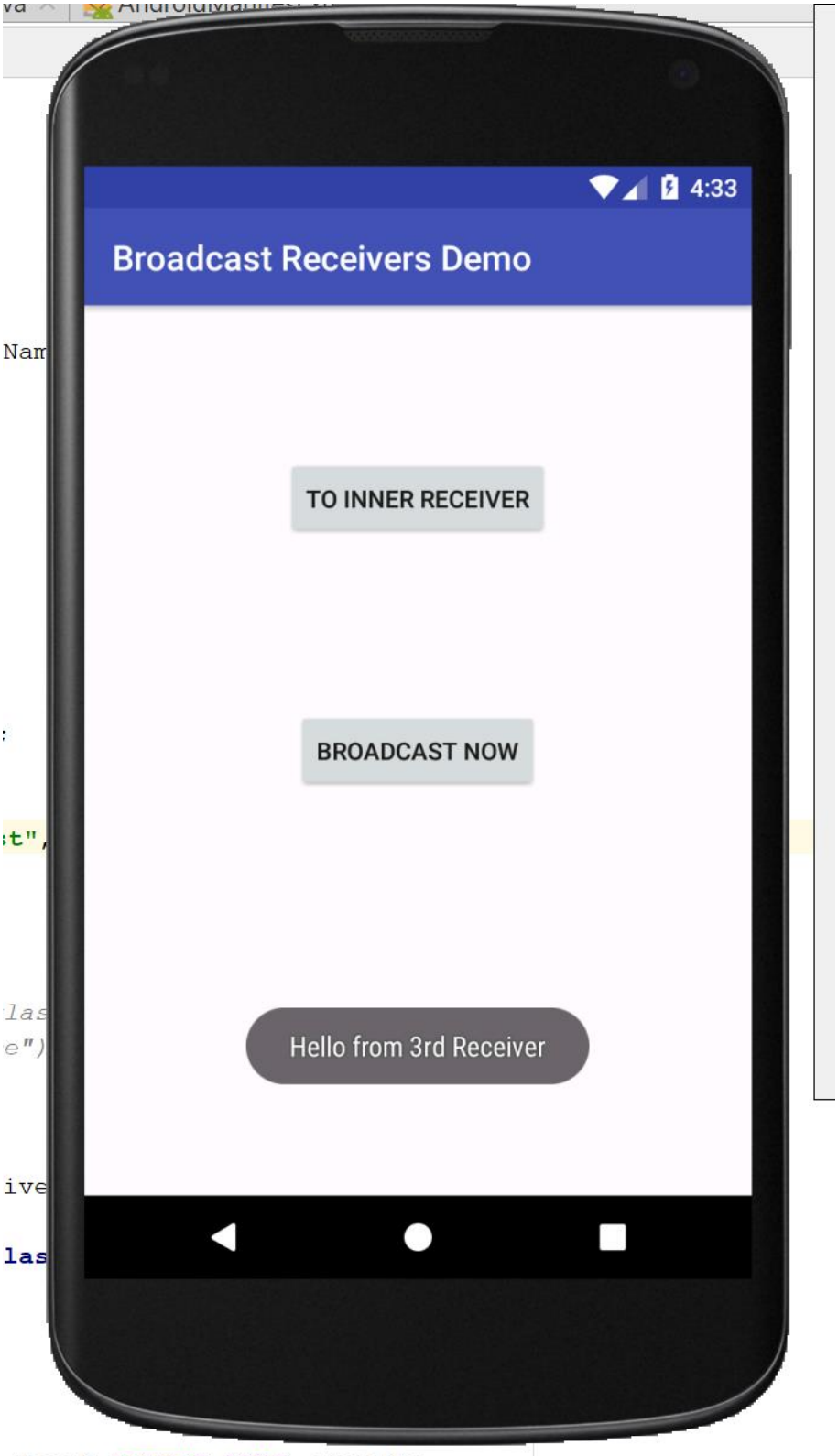
The very first one says” After Sending the Broadcasts” when no broadcast has even been sent:



And here come the broadcasts once they have been received in the order they have been received:




4:34
Broadcast Receivers Demo
TO INNER RECEIVER
BROADCAST NOW
Hello from 1st Receiver


4:33
Broadcast Receivers Demo
TO INNER RECEIVER
BROADCAST NOW
Hello from 3rd Receiver

The fact that the toast statement got executed right after calling the sendBroadcast(intent); and before returning from this method call, proves that broadcasts are sent asynchronously.

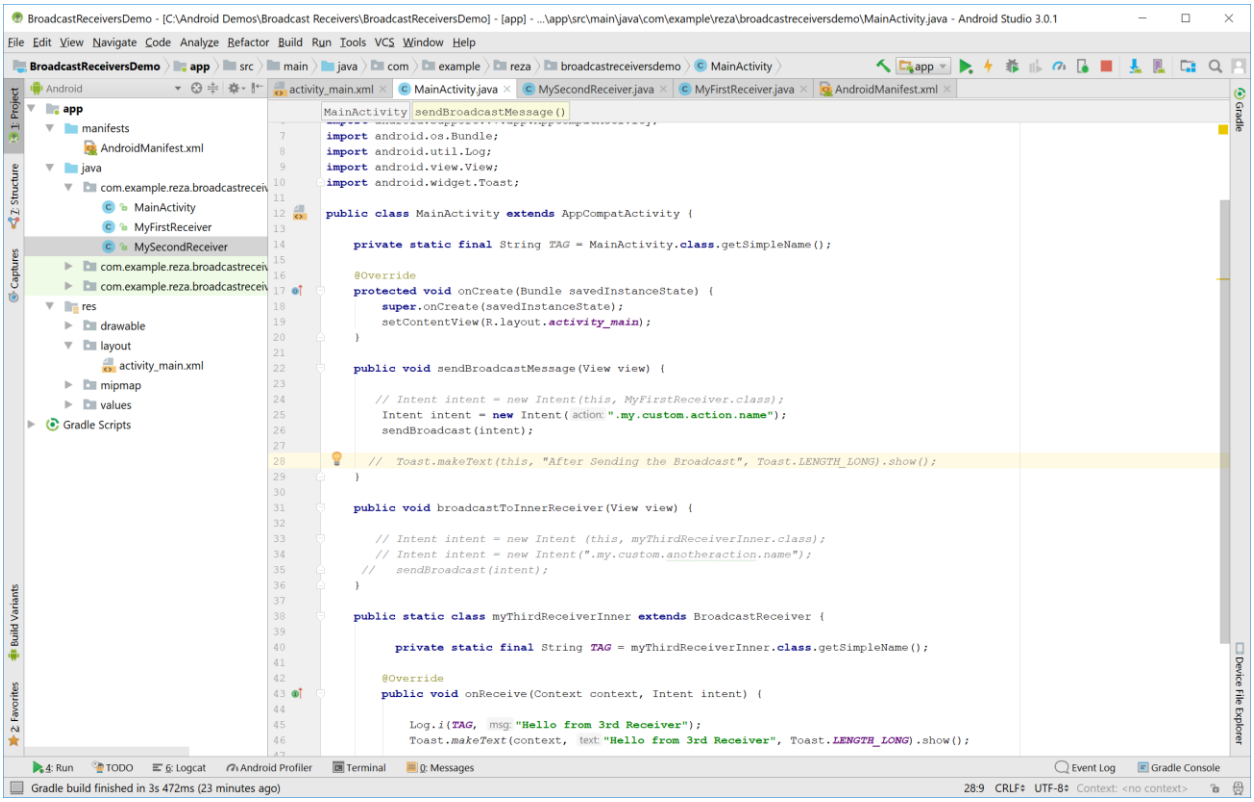
This is an important property of Broadcast Receivers.

Another important property is that the Broadcast Receivers are executed in the main thread which means they can not contain a long running task another word you can not write code to perform a long running task inside the Broadcast Receiver for example you cannot download a file inside the onReceive() method of a Broadcast Receiver.

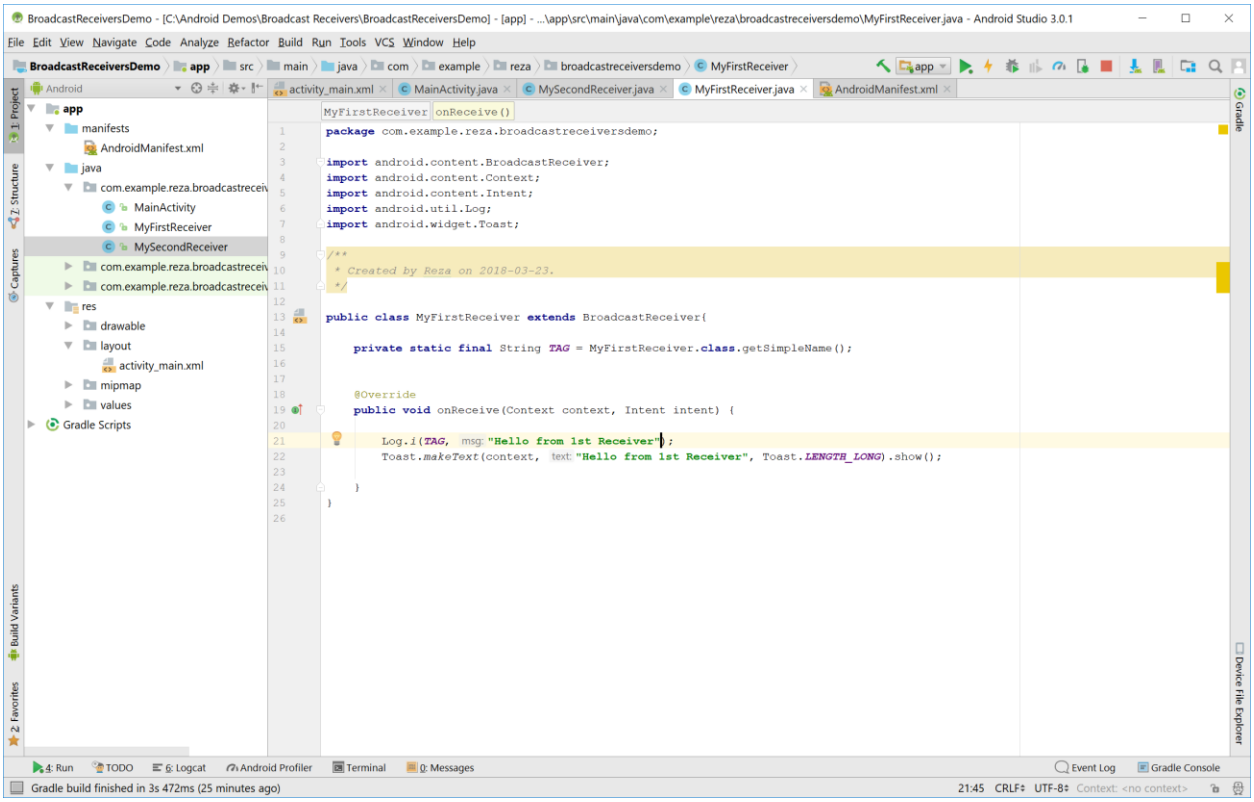
If you do perform a long running task inside the Broadcast Receiver, then the system will display an ANR (Application Not Responding) dialog and your app will crash. The Android documentation mentions a “Read Timeout of 10 Seconds”, another word if you perform some operation that will take some time around 10 seconds or more, then your app will probably crash.

So, what is the solution to this problem? You should use services for long duration tasks.

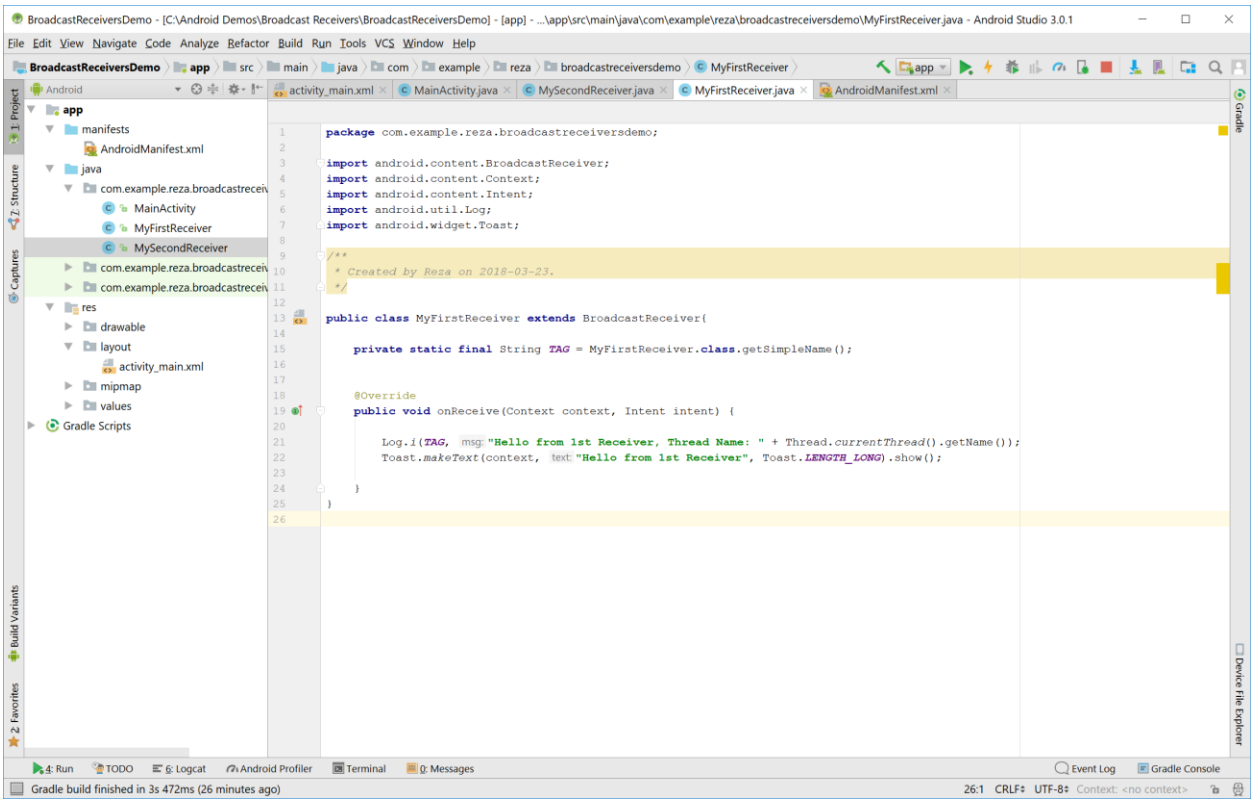
Now it is easy to check out if our Broadcast Receiver works in the main thread. To do so first comment out line 28 that we added in the previous step as shown:



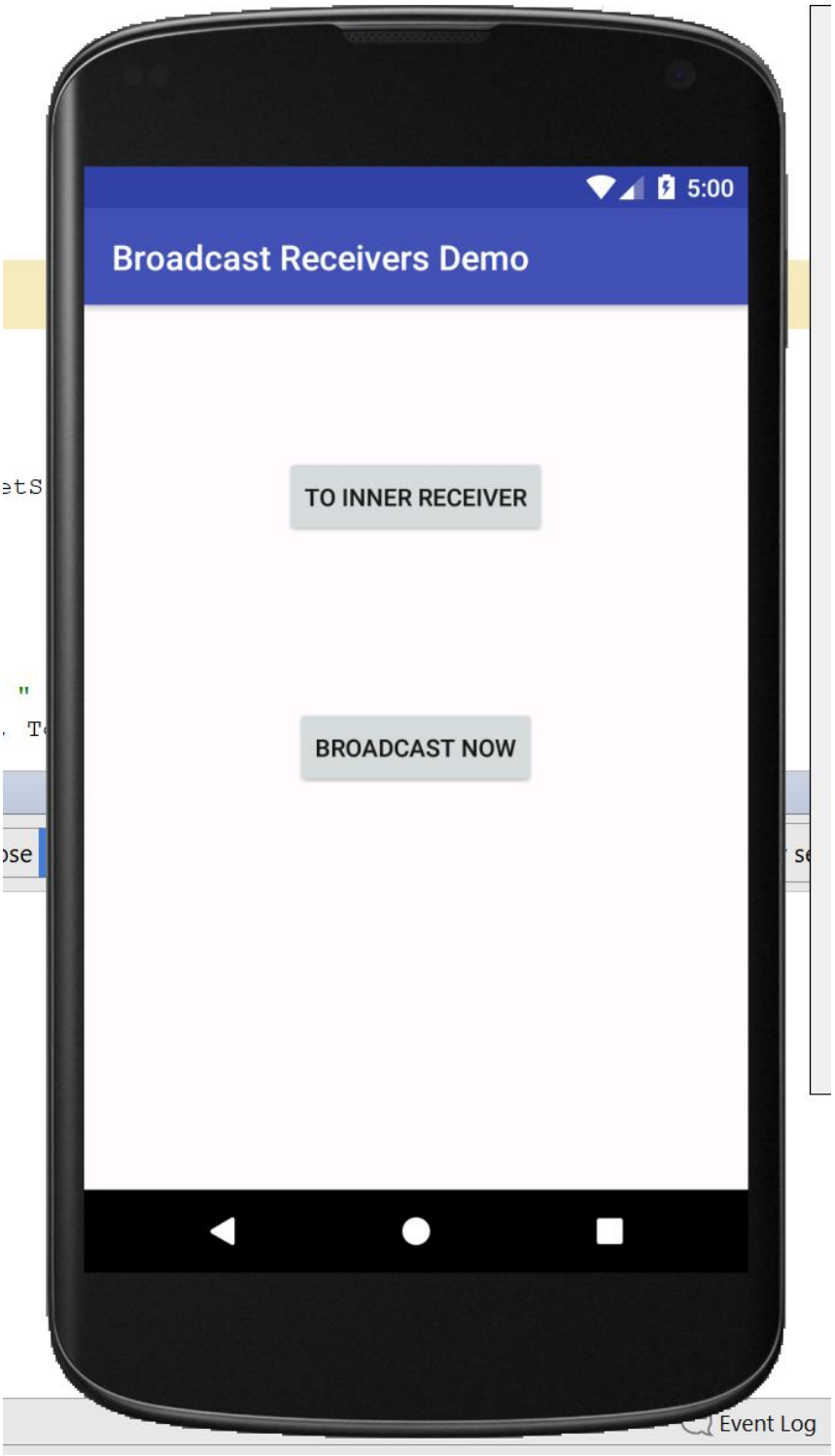
Now go to the MyFirstReceiver.java file and modify it as shown. This is how it is currently looks like:

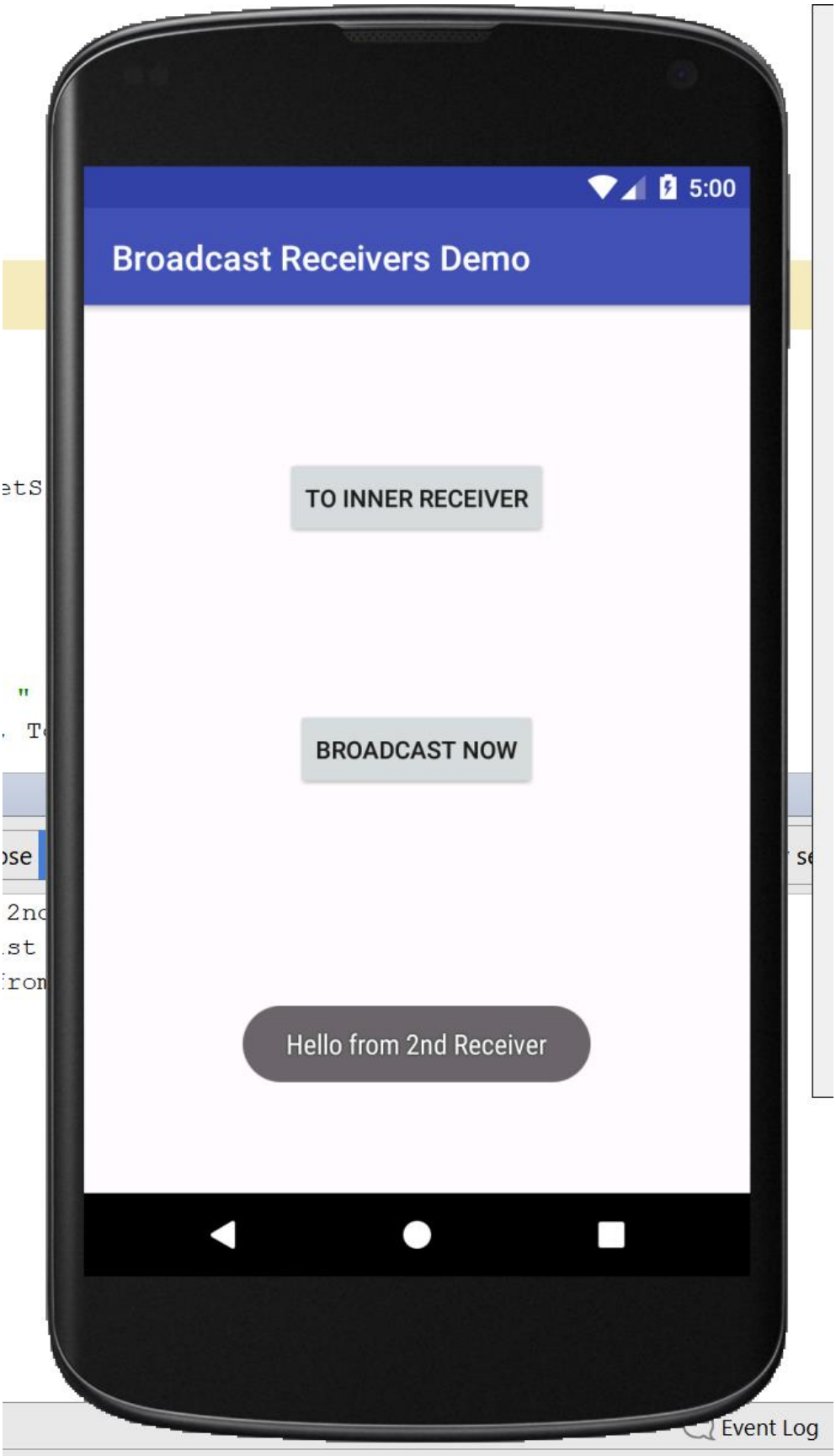


Modify it with modifying line 21 as shown:

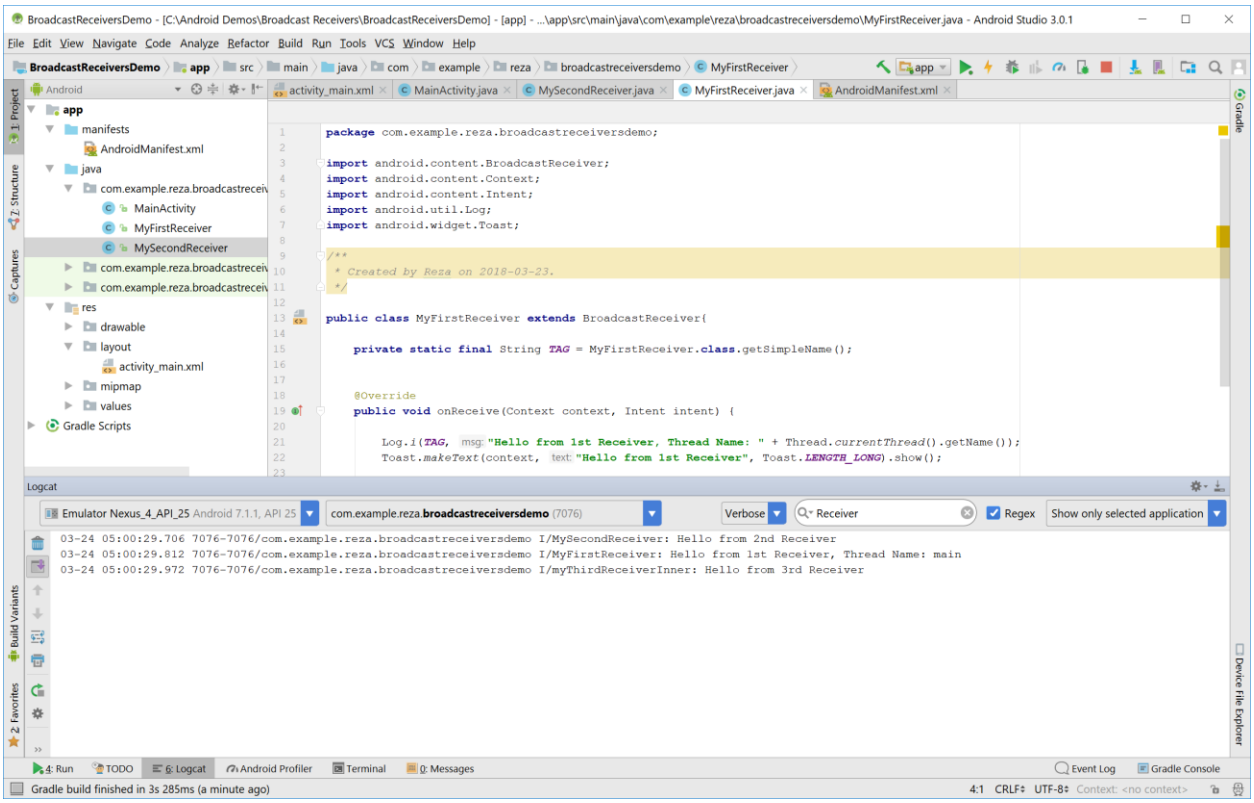


Now run the app and click on “Broadcast Now” button:



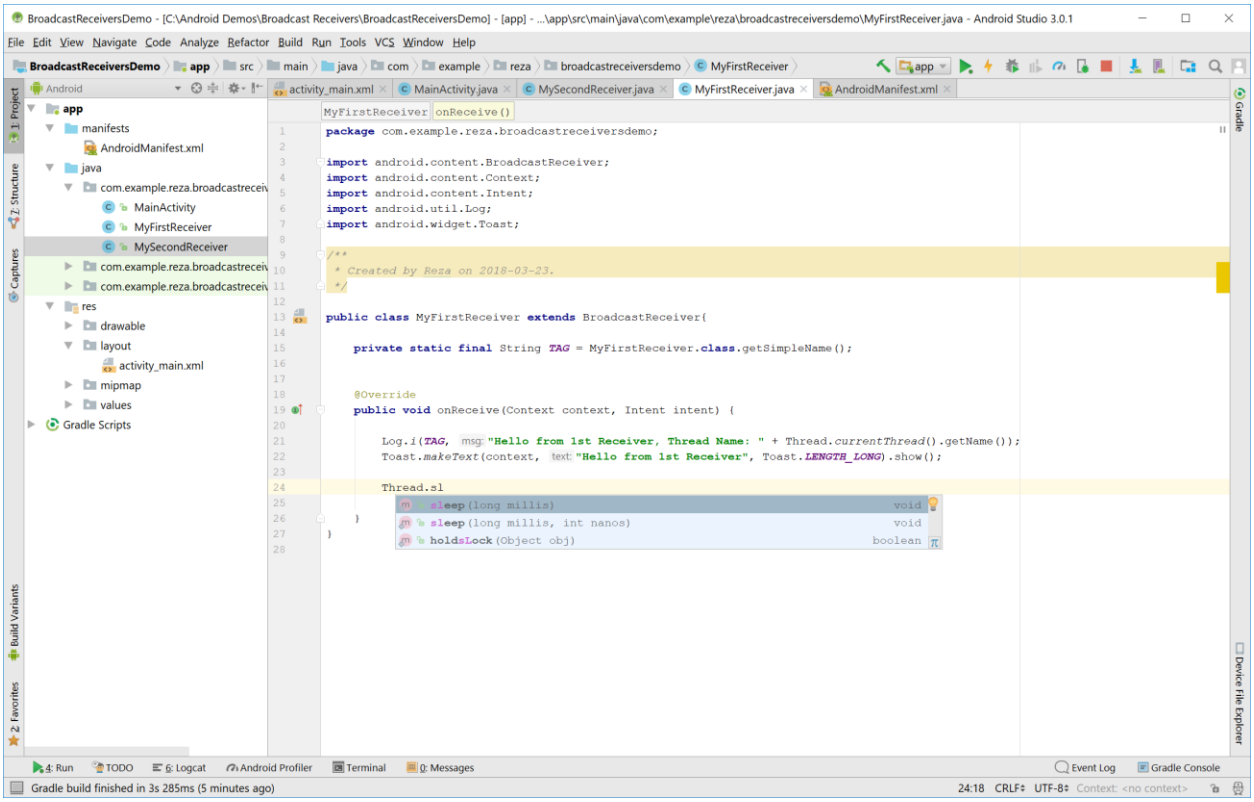

5:00
Broadcast Receivers Demo
TO INNER RECEIVER
BROADCAST NOW
Hello from 2nd Receiver
Event Log

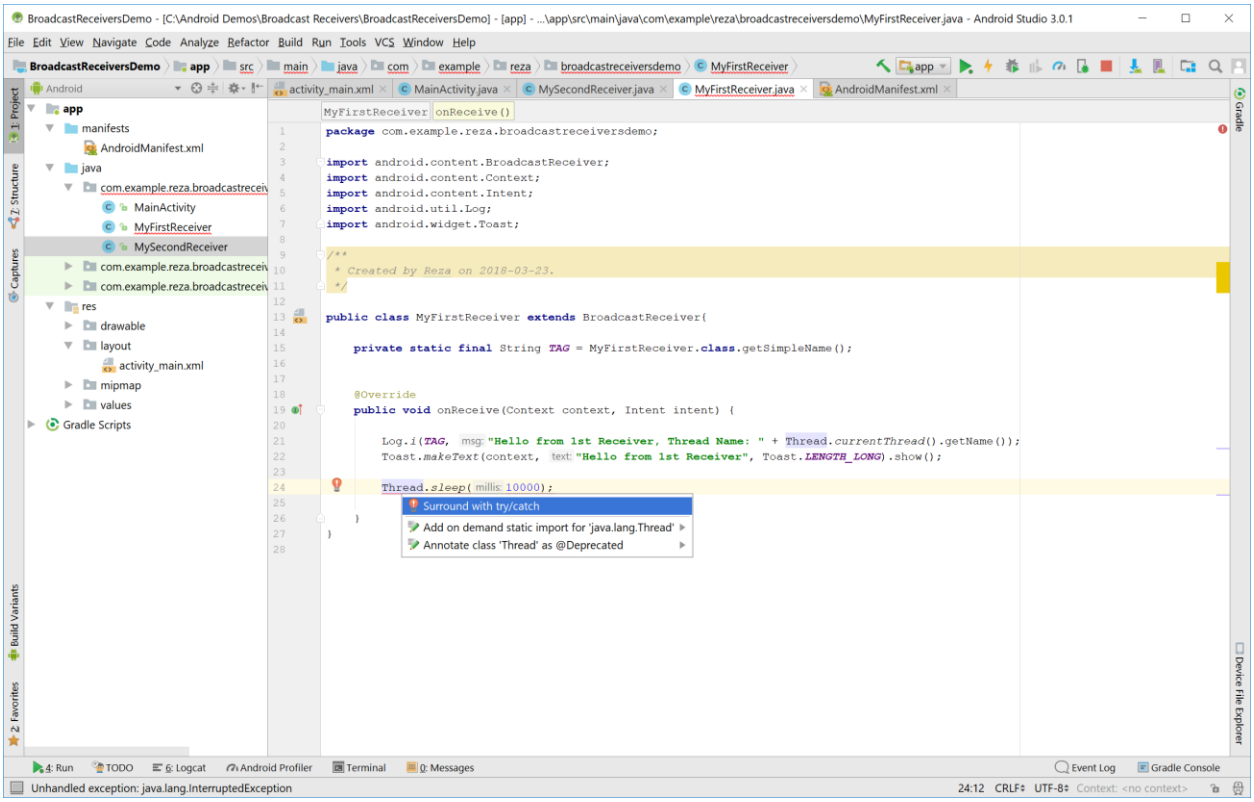
Now in the screen capture below you see that the thread name is “main”.

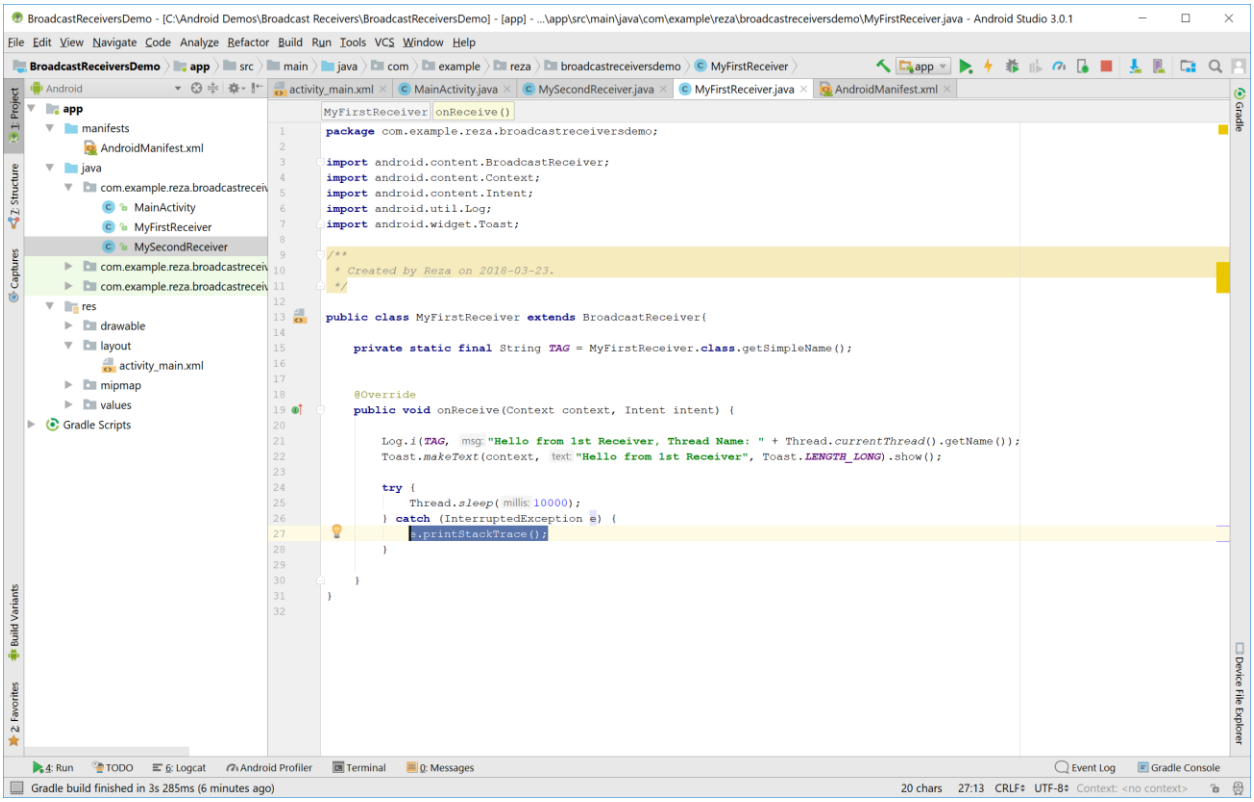


Now if you perform some heavy operation inside the onReceive() method, then your application will display an ANR dialog and it will crash and during that process your app user interface will be blocked.

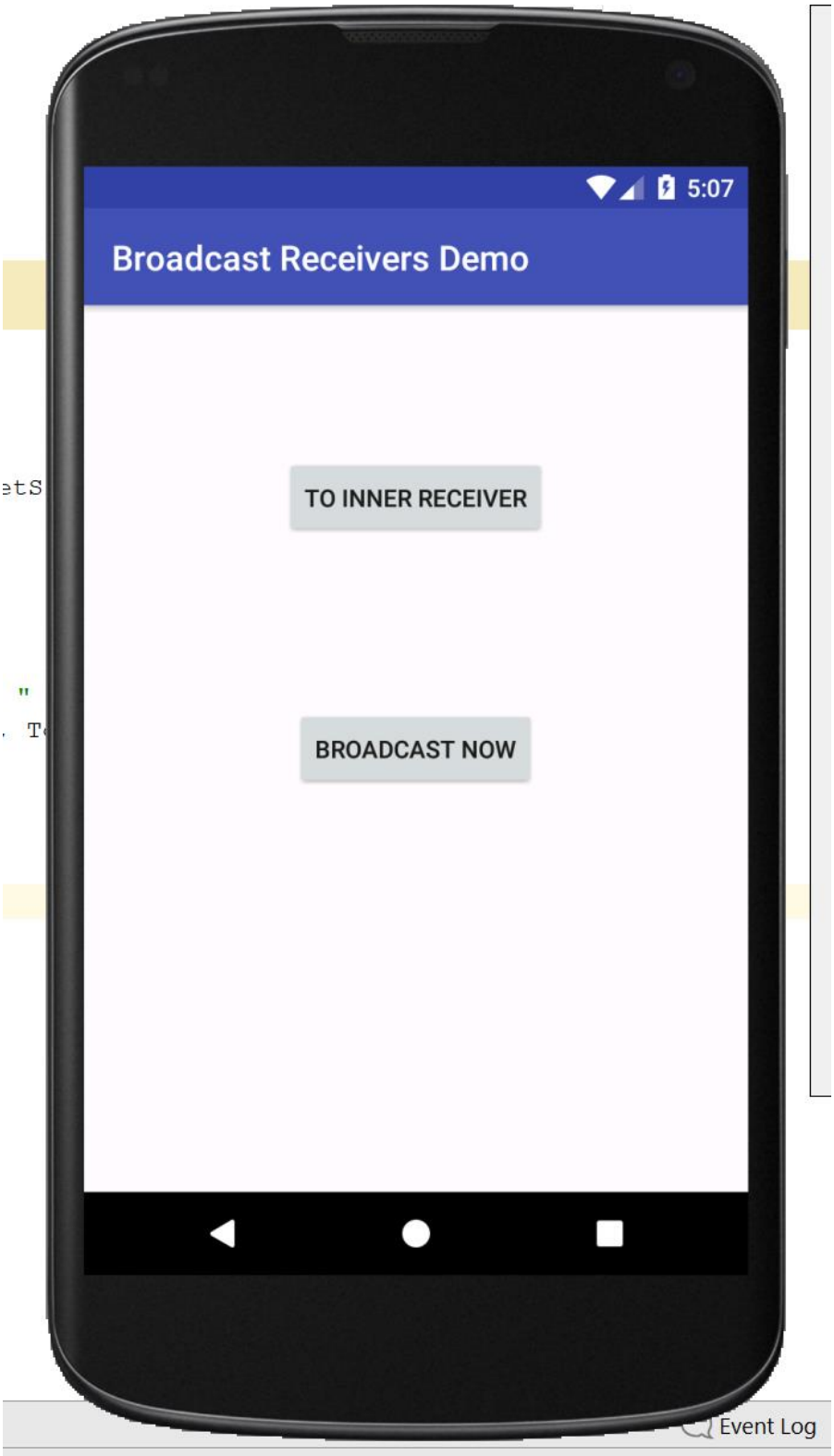
To demonstrate let's modify the onReceive() method in "MyFirstReceiver.java" file as shown:



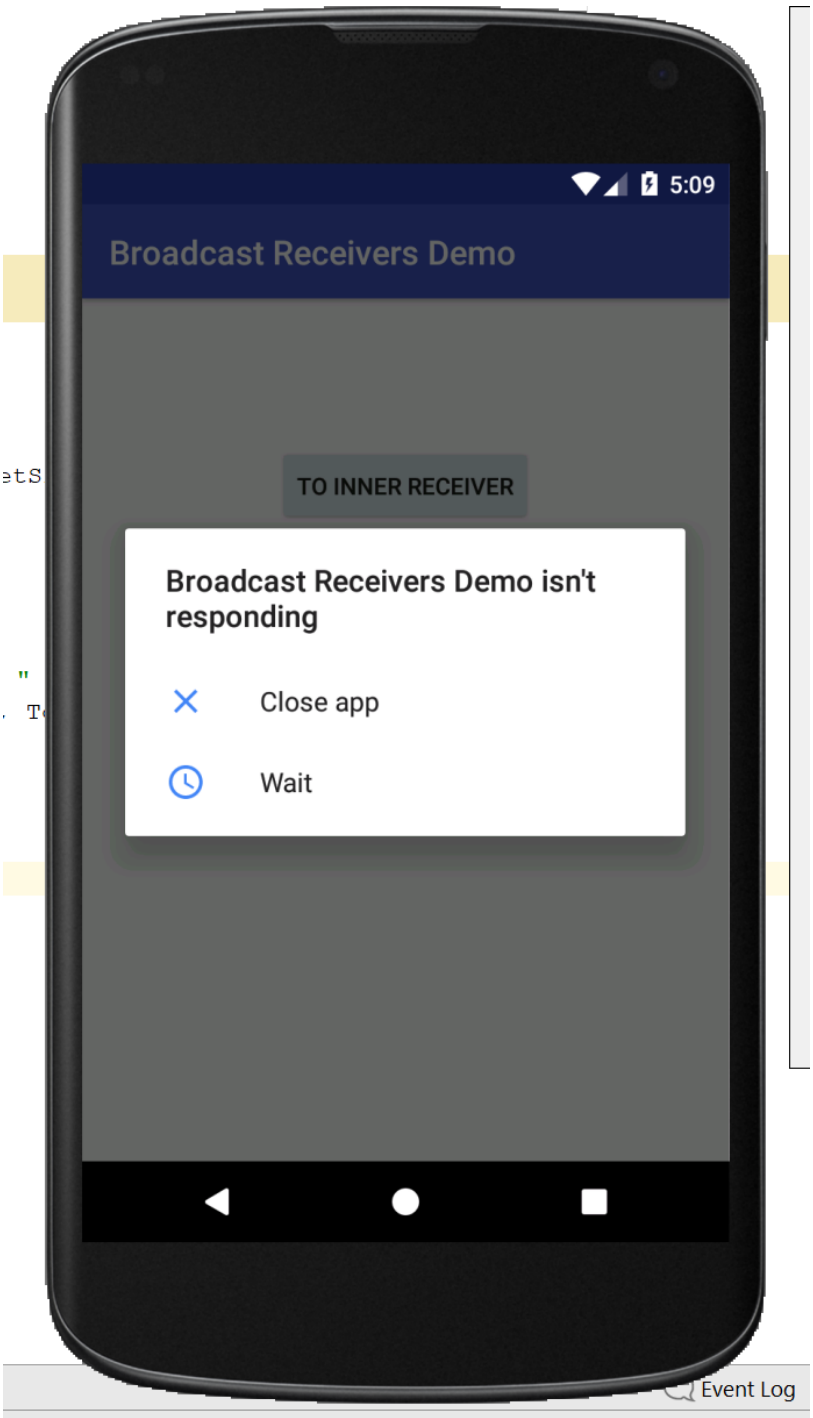

BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\MyFirstReceiver.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main java com example reza broadcastreceiversdemo MyFirstReceiver
activity_main.xml MainActivity.java MySecondReceiver.java MyFirstReceiver.java AndroidManifest.xml
MyFirstReceiver onReceive()
package com.example.reza.broadcastreceiversdemo;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.util.Log;
import android.widget.Toast;
/**
* Created by Reza on 2018-03-23.
*/
public class MyFirstReceiver extends BroadcastReceiver{
private static final String TAG = MyFirstReceiver.class.getSimpleName();
@Override
public void onReceive(Context context, Intent intent) {
Log.i(TAG, msg: "Hello from 1st Receiver, Thread Name: " + Thread.currentThread().getName());
Toast.makeText(context, text: "Hello from 1st Receiver", Toast.LENGTH_LONG).show();
Thread.sleep(millis: 10000);
Surround with try/catch
Add on demand static import for 'java.lang.Thread'
Annotate class 'Thread' as @Deprecated
Unhandled exception: java.lang.InterruptedException
24:12 CRLF UTF-8 Context: <no context>


BroadcastReceiversDemo - [C:\Android Demos\Broadcast Receivers\BroadcastReceiversDemo] - [app] - ...\app\src\main\java\com\example\reza\broadcastreceiversdemo\MyFirstReceiver.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BroadcastReceiversDemo app src main java com example reza broadcastreceiversdemo MyFirstReceiver
activity_main.xml MainActivity.java MySecondReceiver.java MyFirstReceiver.java AndroidManifest.xml
MyFirstReceiver onReceive()
package com.example.reza.broadcastreceiversdemo;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.util.Log;
import android.widget.Toast;
/**
* Created by Reza on 2018-03-23.
*/
public class MyFirstReceiver extends BroadcastReceiver{
private static final String TAG = MyFirstReceiver.class.getSimpleName();
@Override
public void onReceive(Context context, Intent intent) {
Log.i(TAG, msg: "Hello from 1st Receiver, Thread Name: " + Thread.currentThread().getName());
Toast.makeText(context, text: "Hello from 1st Receiver", Toast.LENGTH_LONG).show();
try {
Thread.sleep(millis: 10000);
} catch (InterruptedException e) {
e.printStackTrace();
}
}
}
Gradle build finished in 3s 285ms (6 minutes ago)
20 chars 27:13 CRLF UTF-8 Context: <no context>

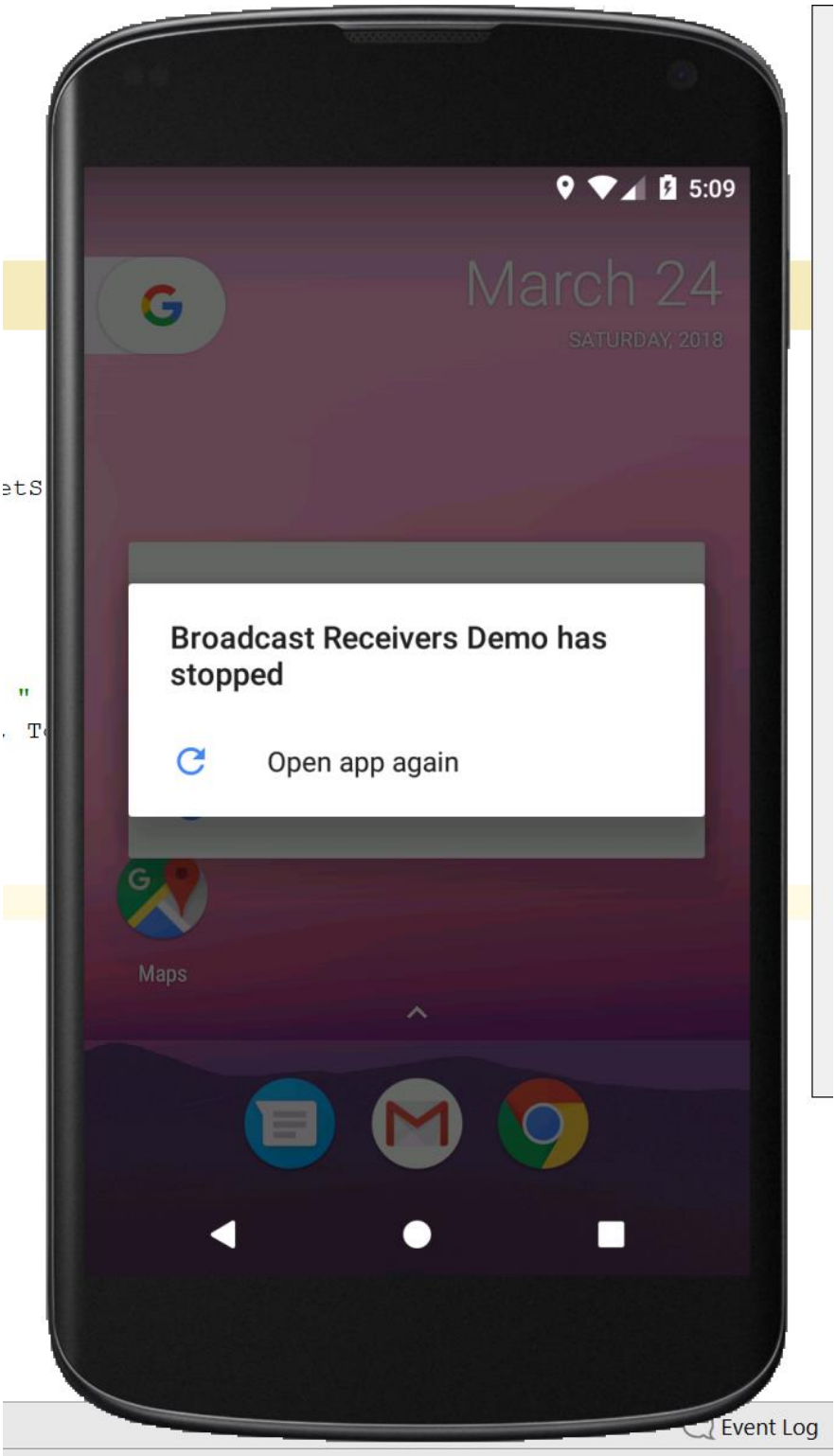
Now run the app:



Click “Broadcast Now” button and then click “To Inner Receiver” button a few times. You will see that the “To Inner Receiver” button is not responsive and after a few seconds you will get the following message from Android operating system:

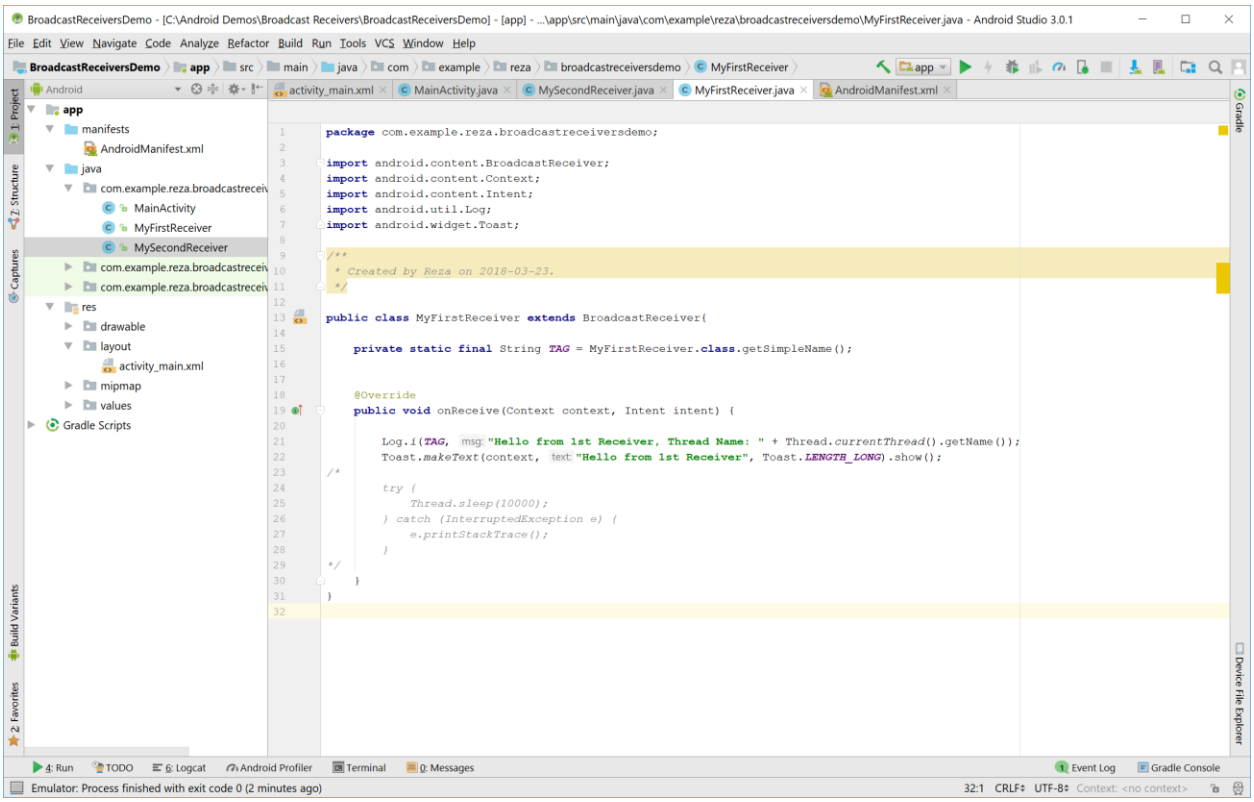


And then you will get the following:

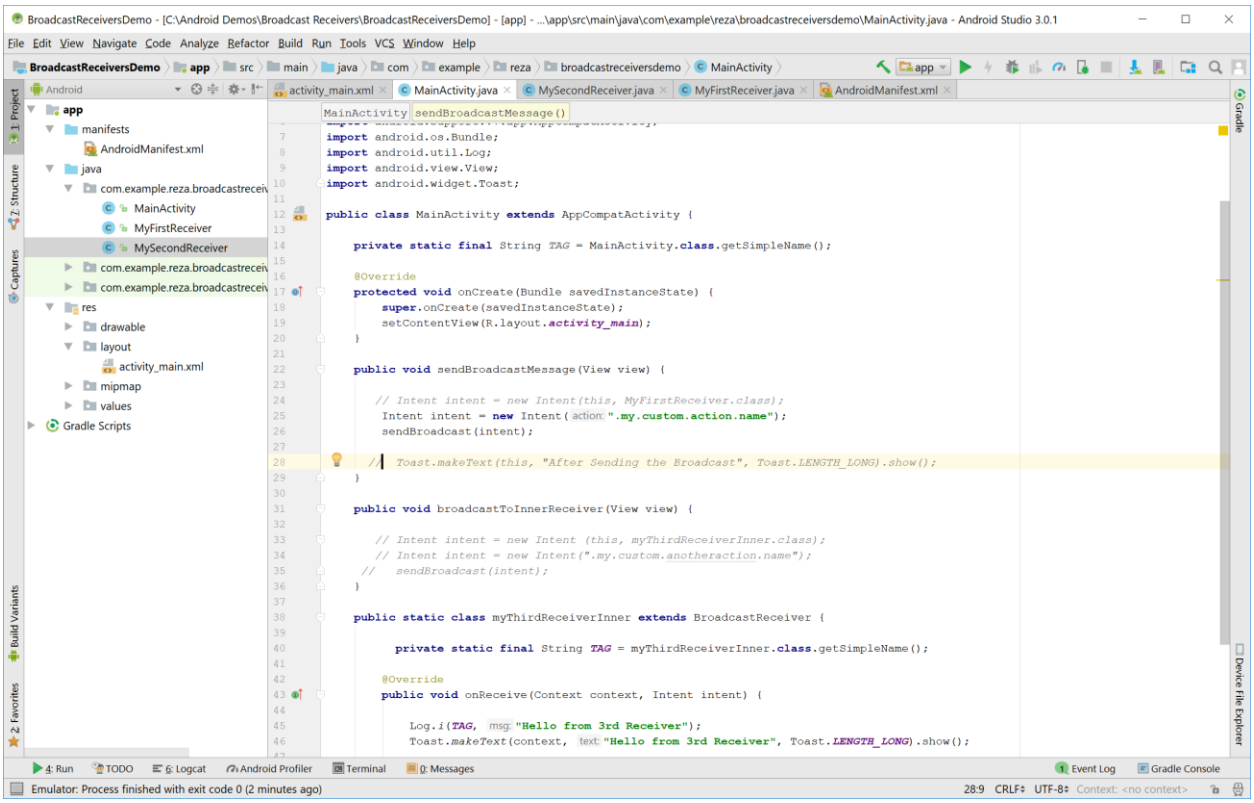


How to Pass Data from the Activity to a Receiver:

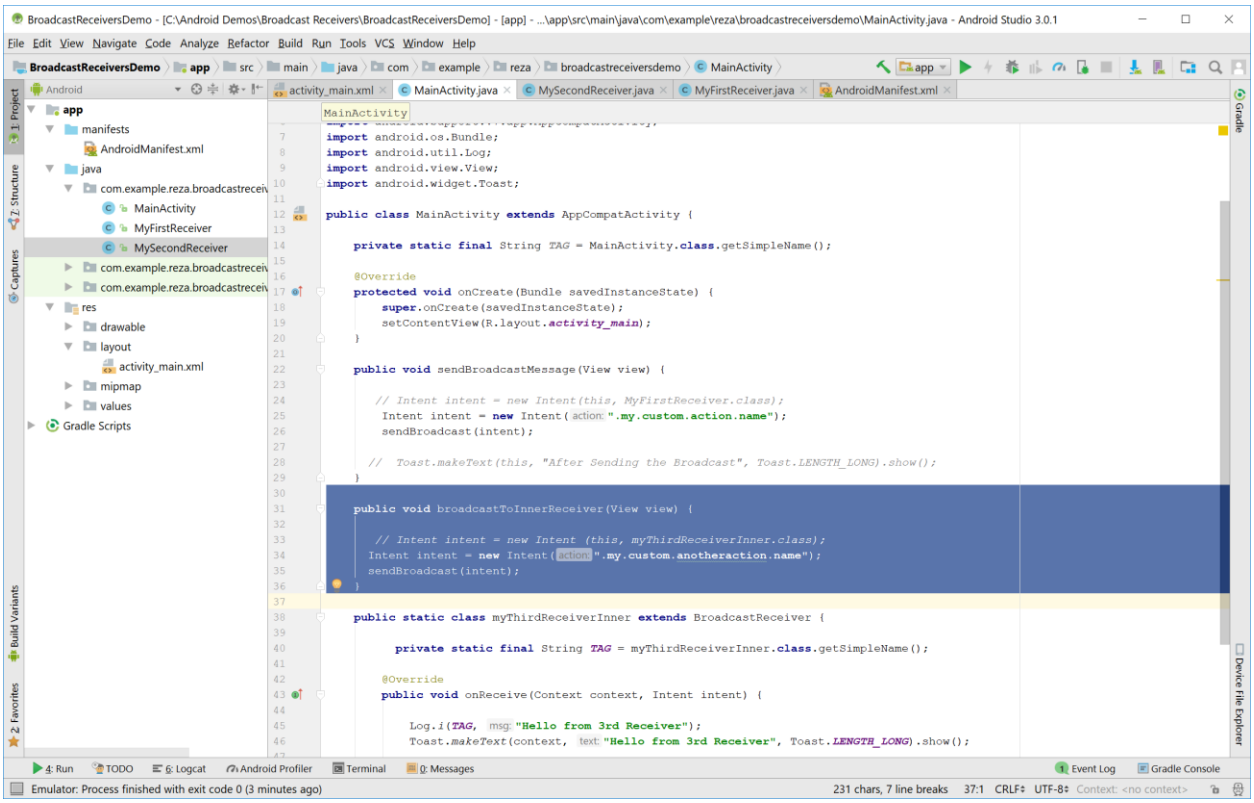
First comment out the code (the long running task) we added to illustrate the previous concept in your “MyFirstReceiver.java” file as shown:



Then go to MainActivity.java. It is currently looks like as follows:

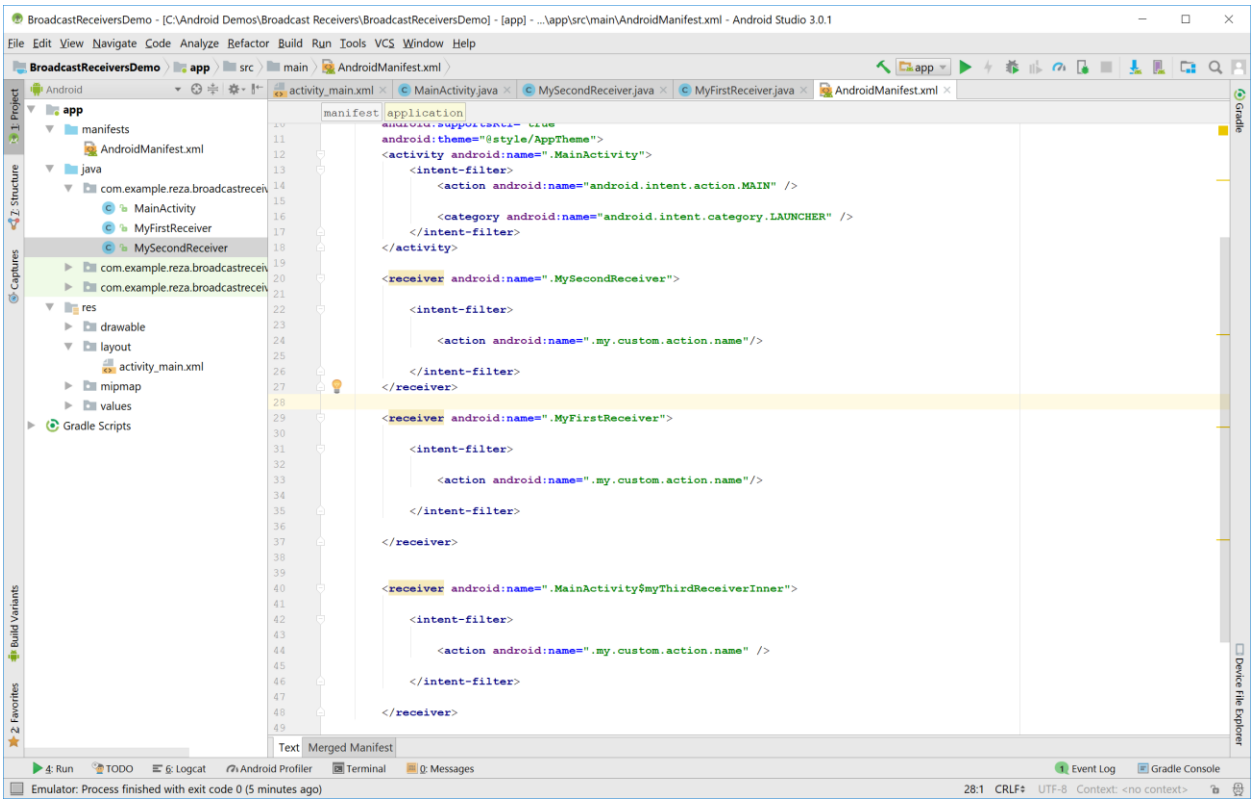


Uncomment lines 34 and 35 as shown:

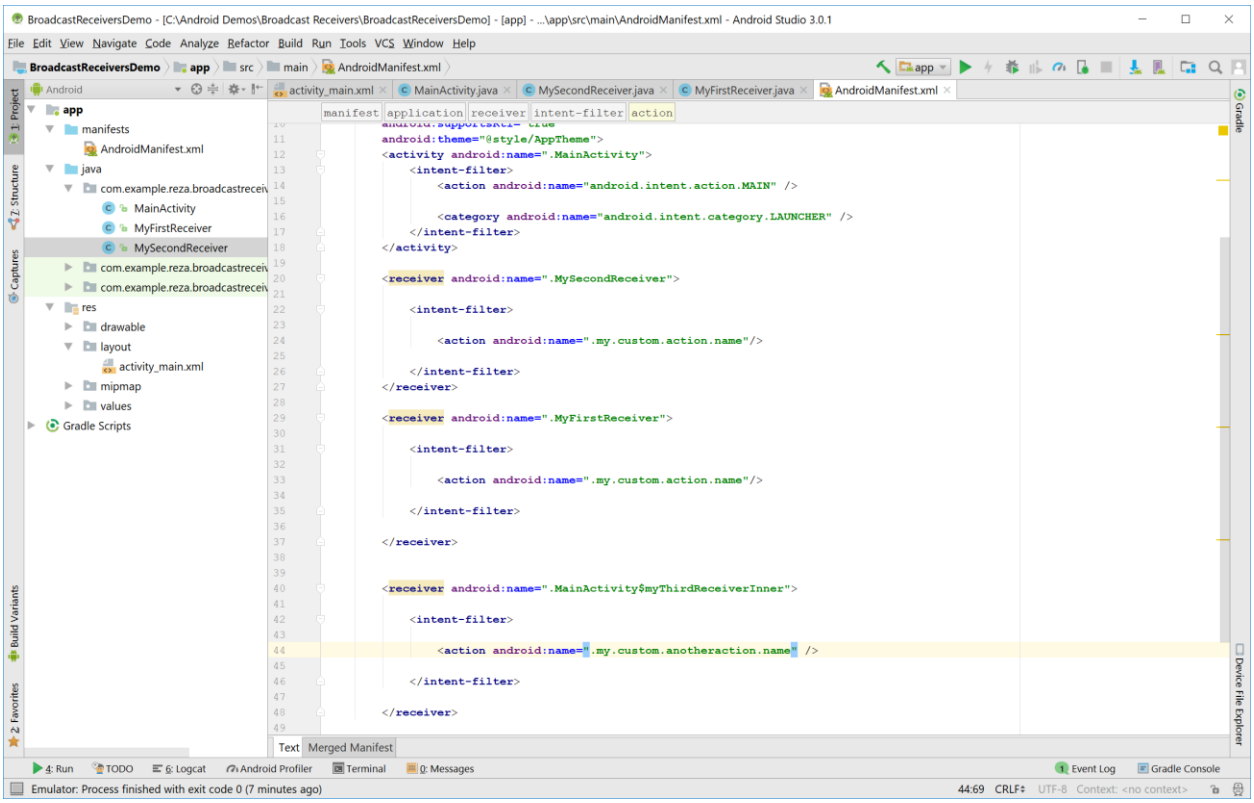


Now copy the string ".my.custom.anotheraction.name" from line 34 and copy that as the action for your third receiver in the manifest file as shown:

This is how our manifest file currently looks like:

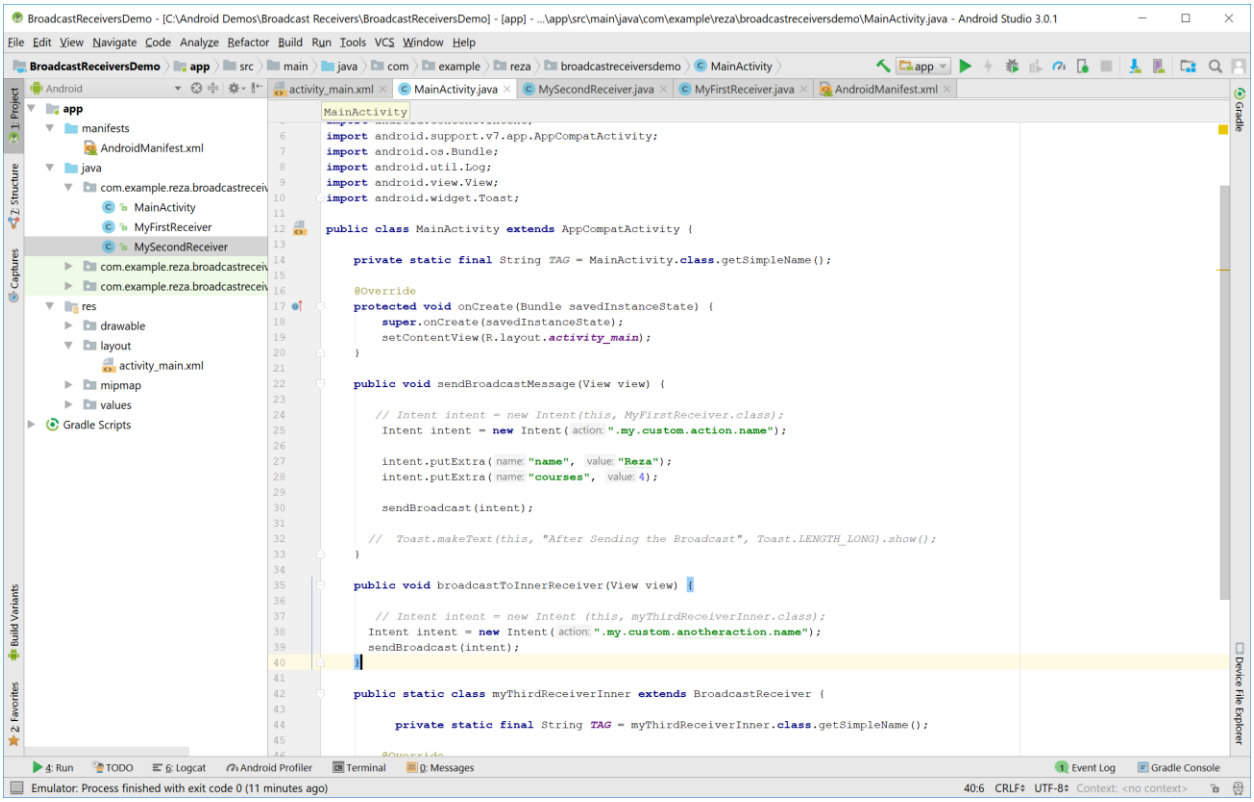


Replace the action in line 44 as shown:

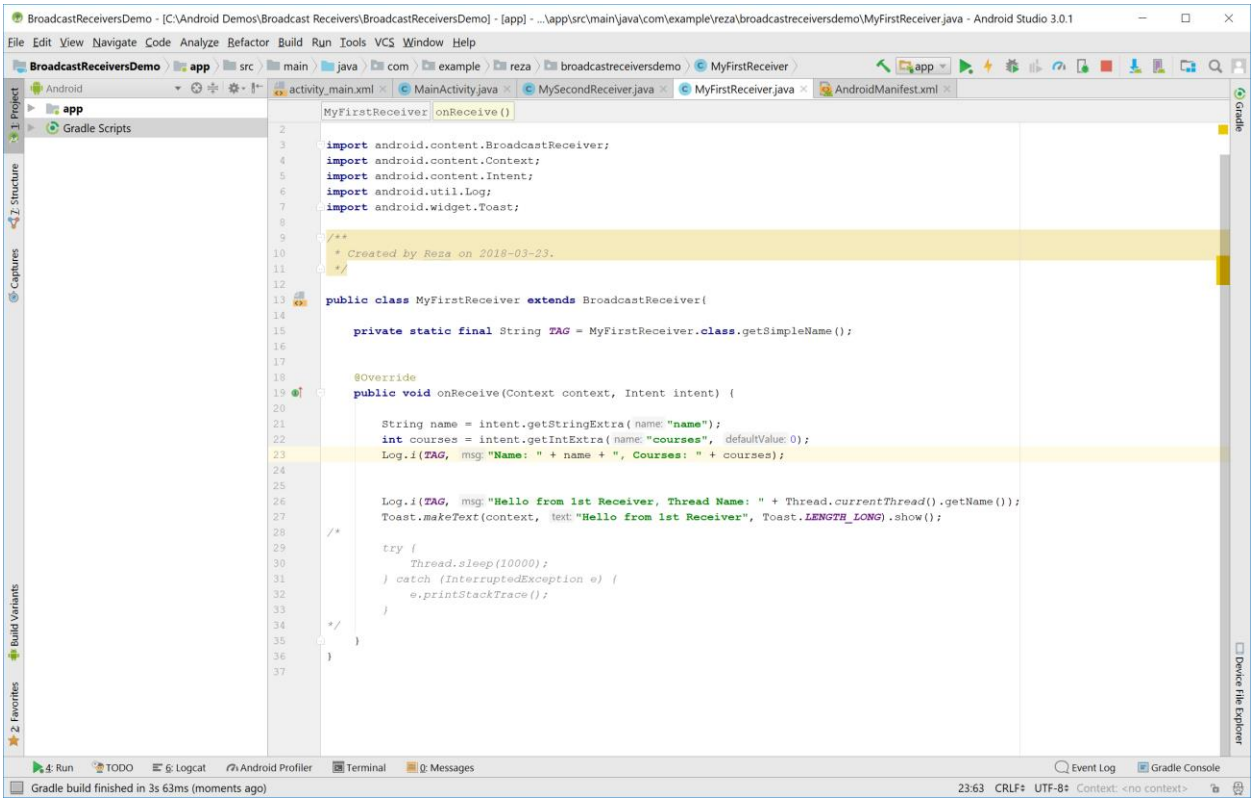


For the purpose of our demonstration in this part, we will be using our first and third receivers.

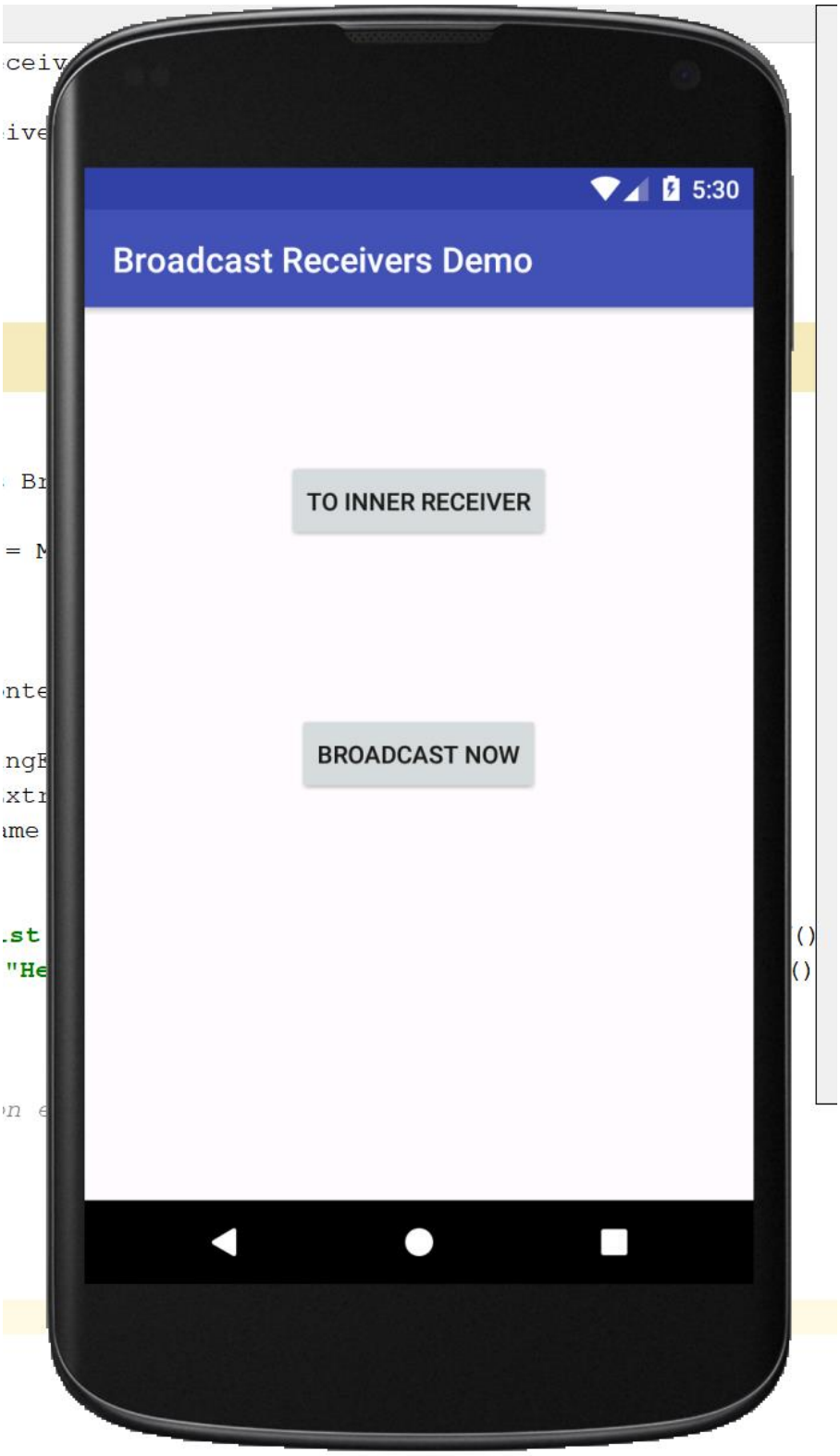
Now in the MainActivity.java let's modify our sendBroadcastMessage() method and send some data as shown:



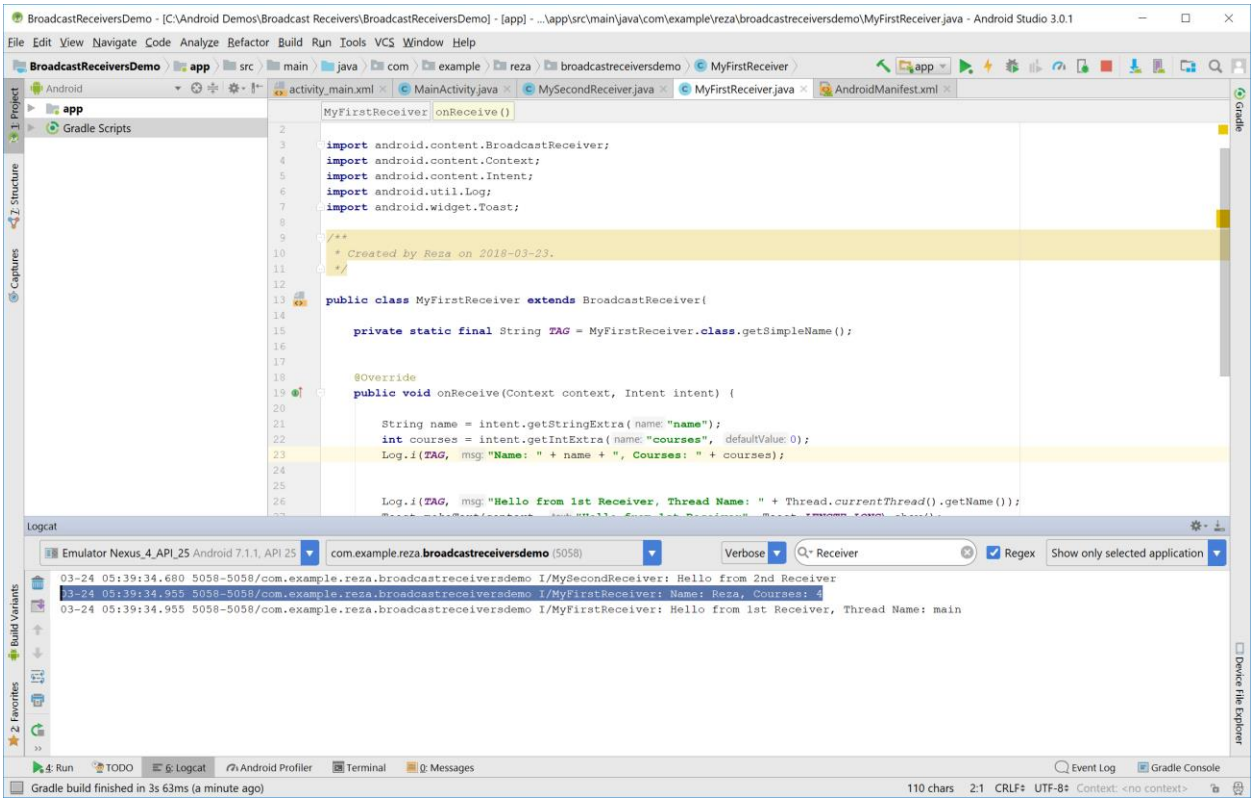
Note that lines 27 and 28 were added. Now let's go to our first receiver (go to MyFirstReceiver.java file) and retrieve this data as shown (lines 21, 22, and 23 were added):



Now let’s run our app and see if we can pass data from our main activity to our first receiver so run the app:

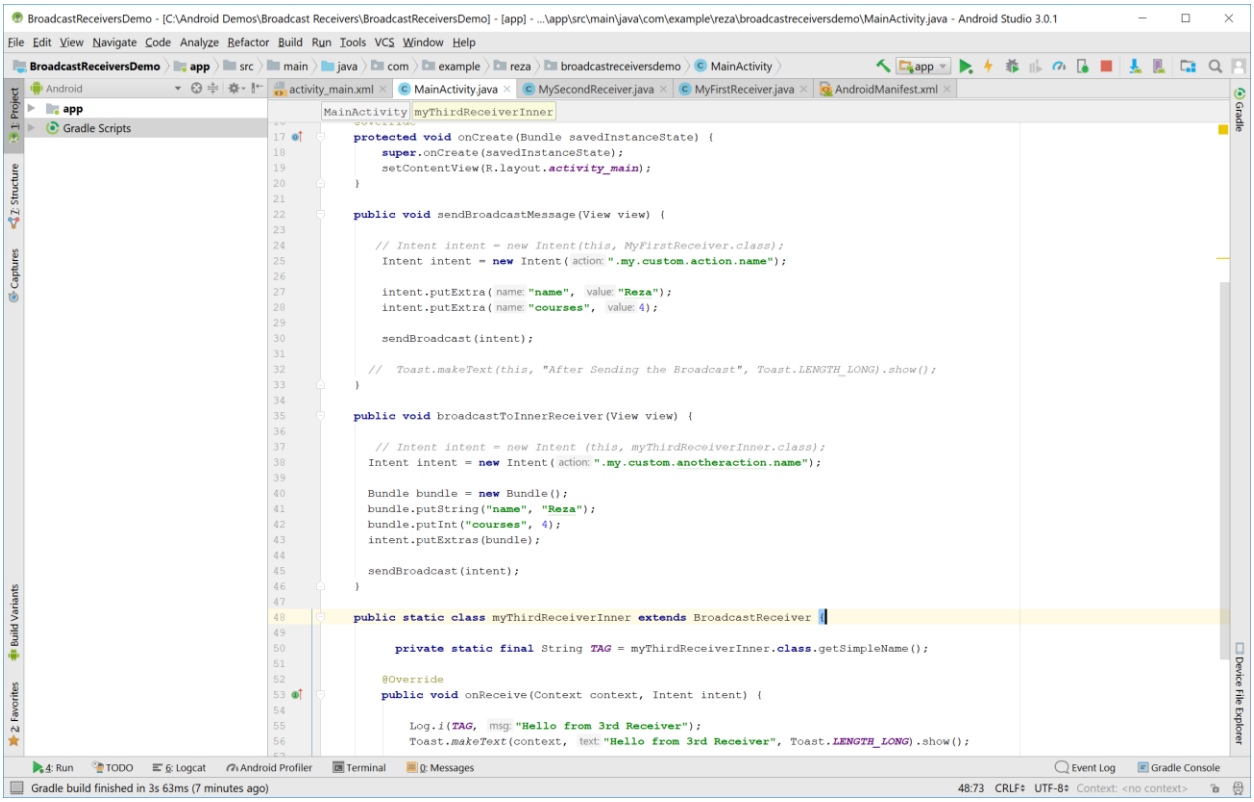


Click “Broadcast Now” button and here is the Logcat window once you do the broadcast:



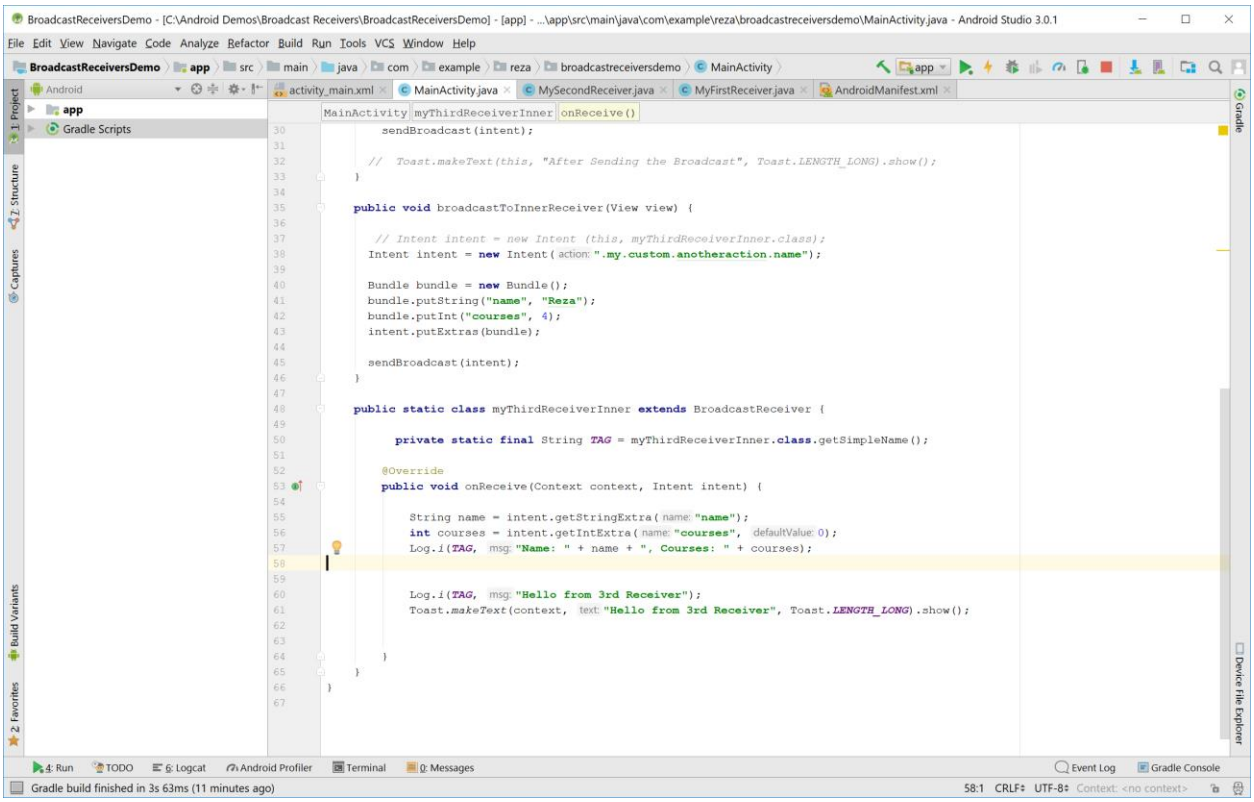
So we were able to send data from our main activity to our first receiver. Now here we used putExtra() method which behind the scene used the Bundle object.

You can use the Bundle object directly and use it to send data. Let's try that to send data to our third receiver so in MainActivity.java, modify the method broadcastToInnerReceiver() (which sends data to the third receiver) as shown:



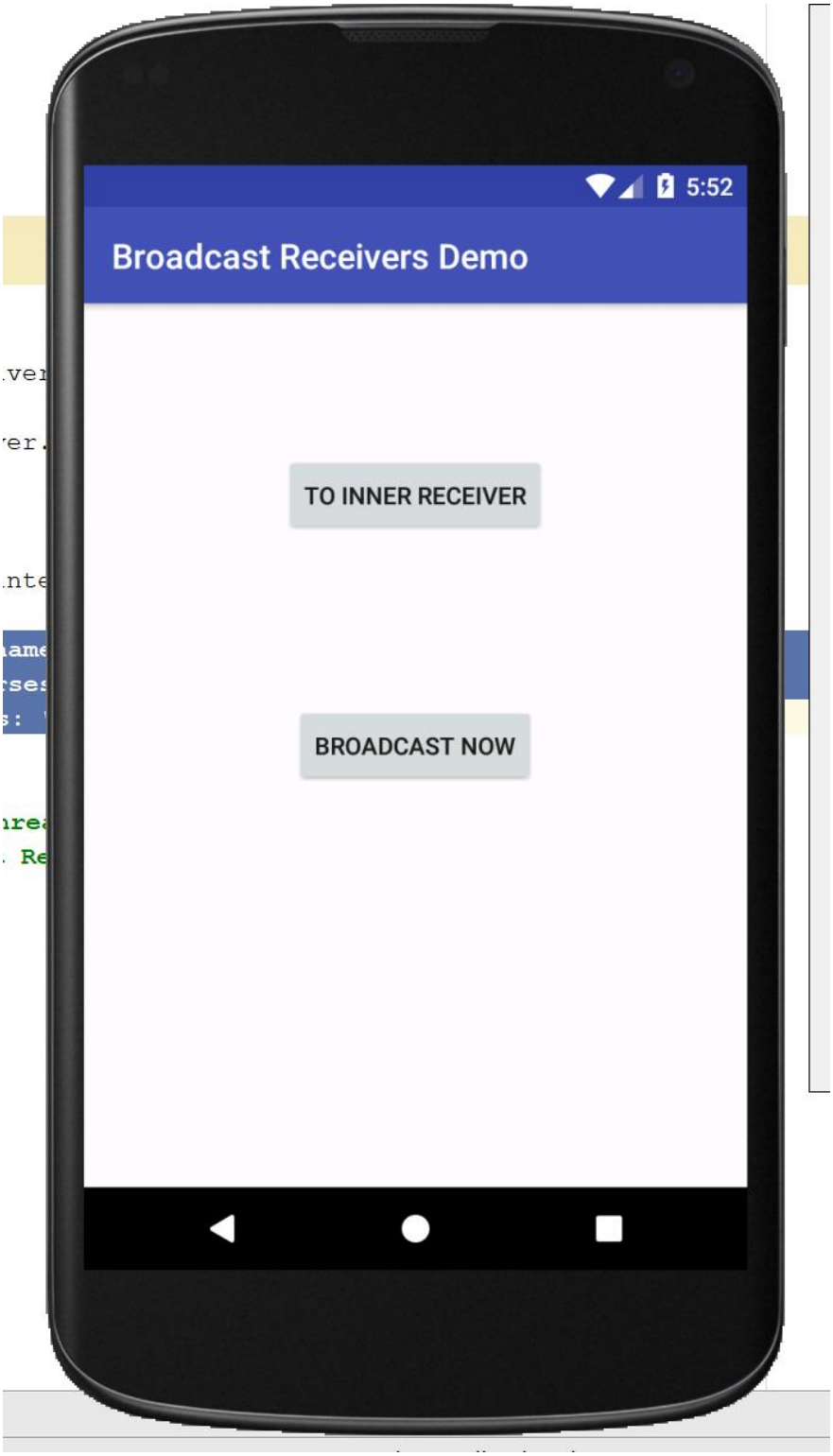
Lines 40, 41, 42, and 43 were added.

Since we are sending data to our third receiver which is actually an inner class inside the main activity then we go to that class and modify the code for its onReceive() method as shown so we can retrieve the data we just sent to it as shown:

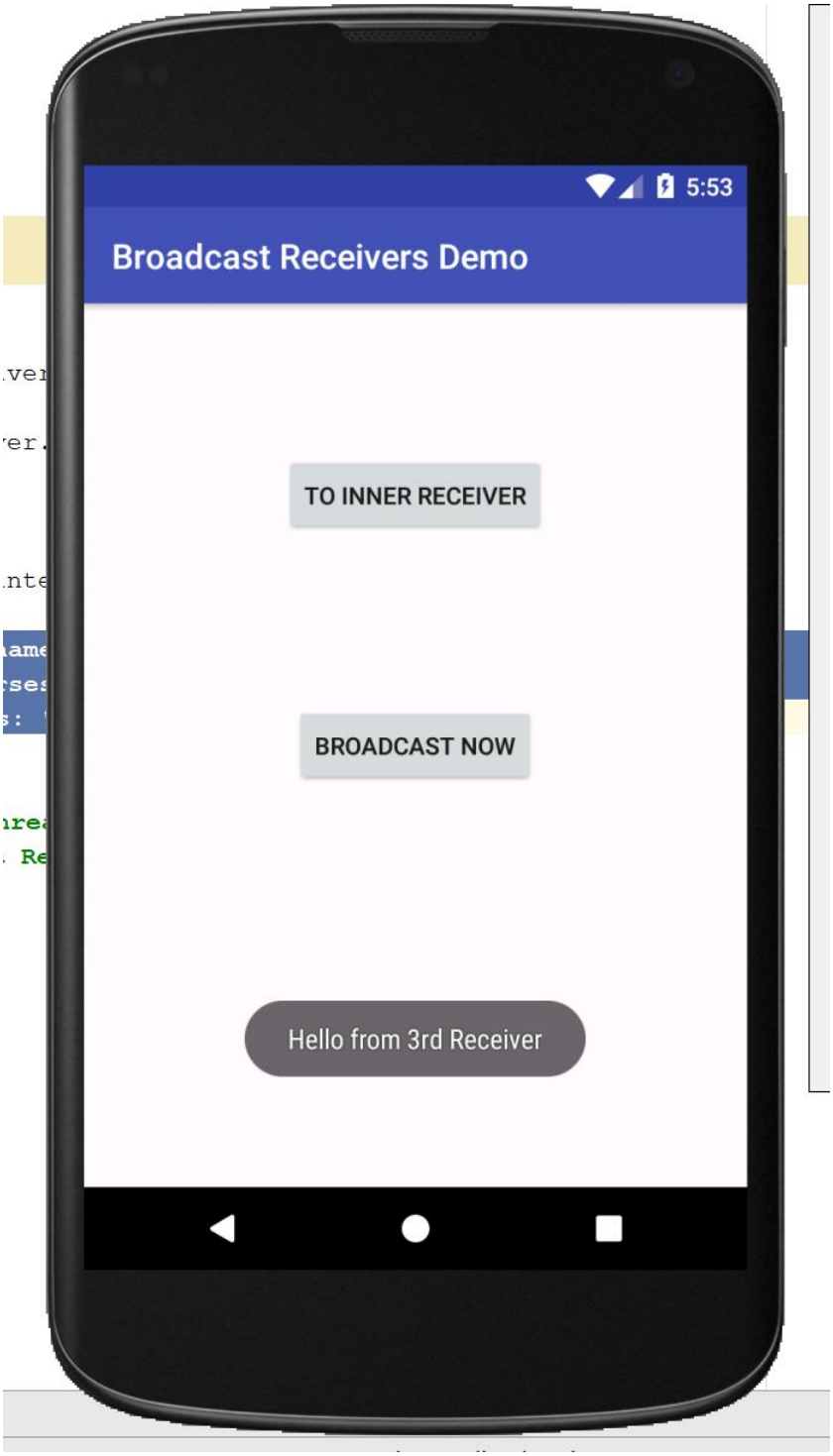


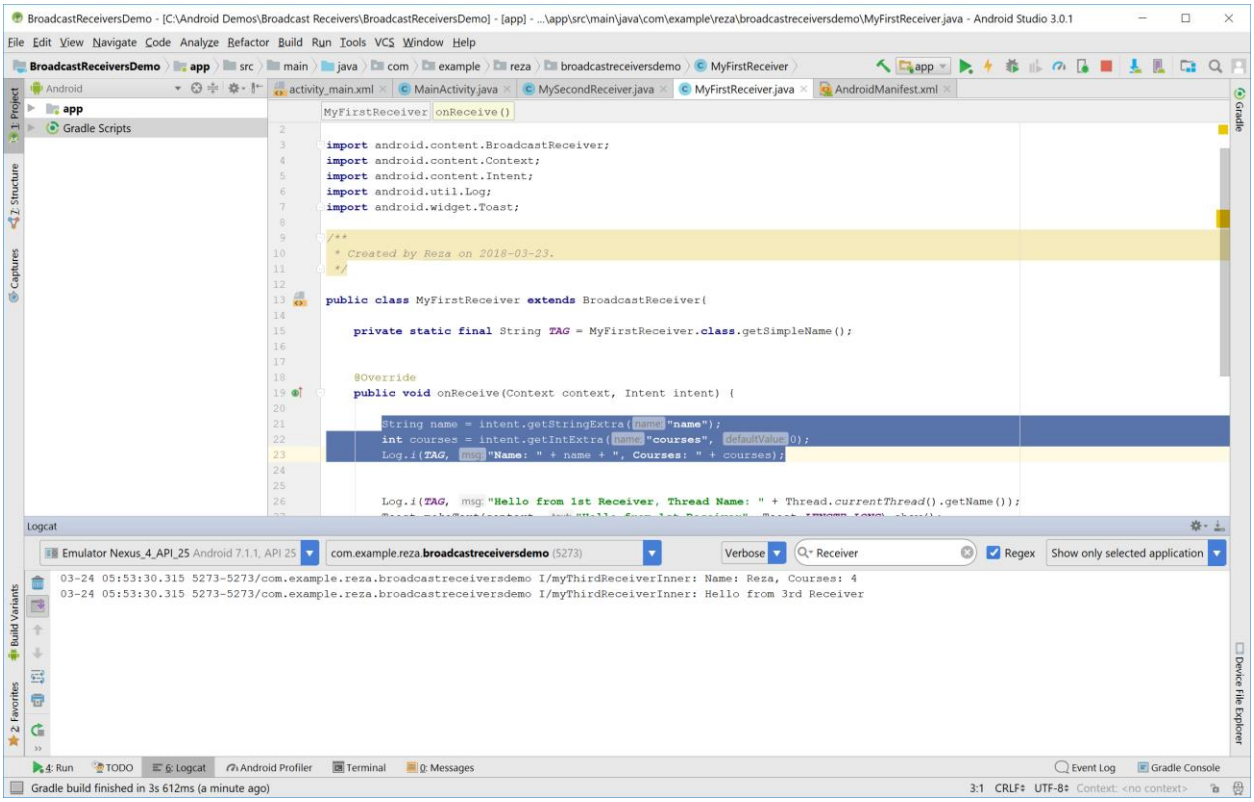
Lines 55, 56, and 57 were added. Note that they are identical to lines 21, 22, and 23 which we added to the onReceive() method of our first receiver.

Now let’s run the app to test our newly added code.



Click “To Inner Receiver” button to send broadcast to the third receiver:

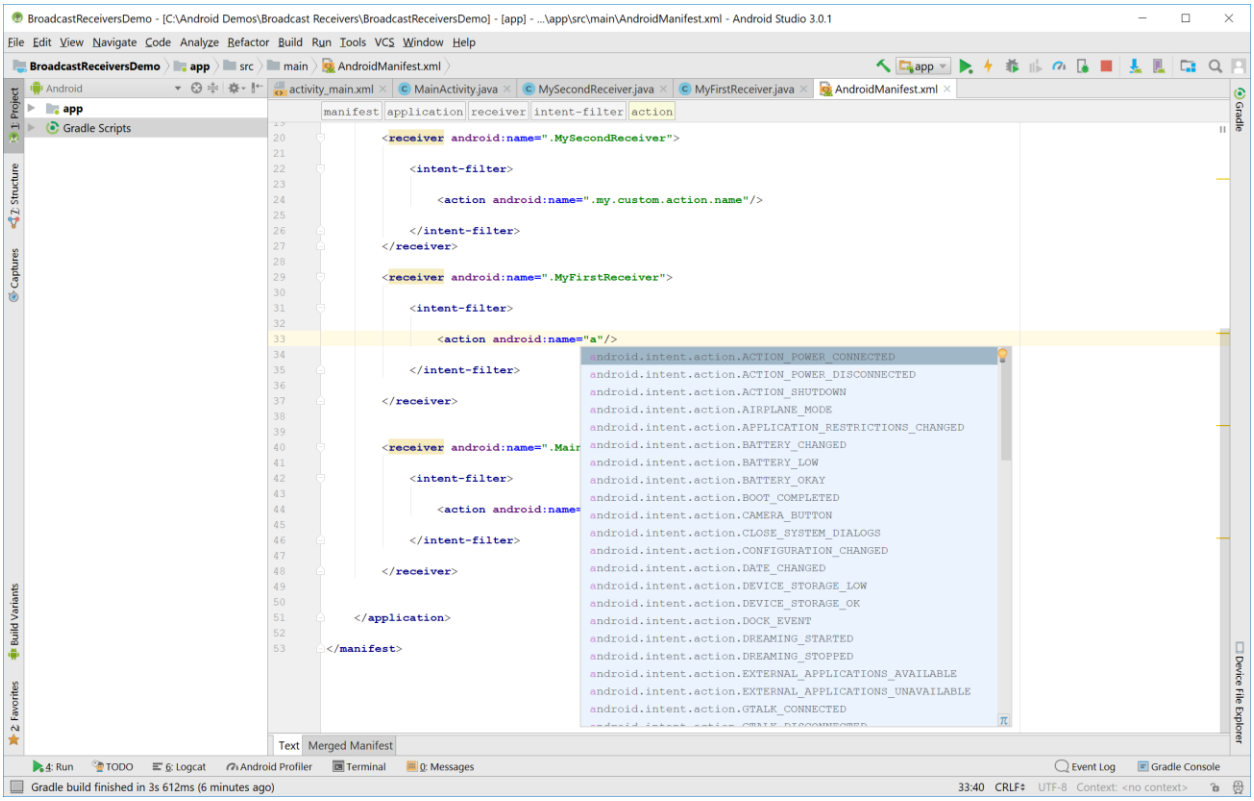




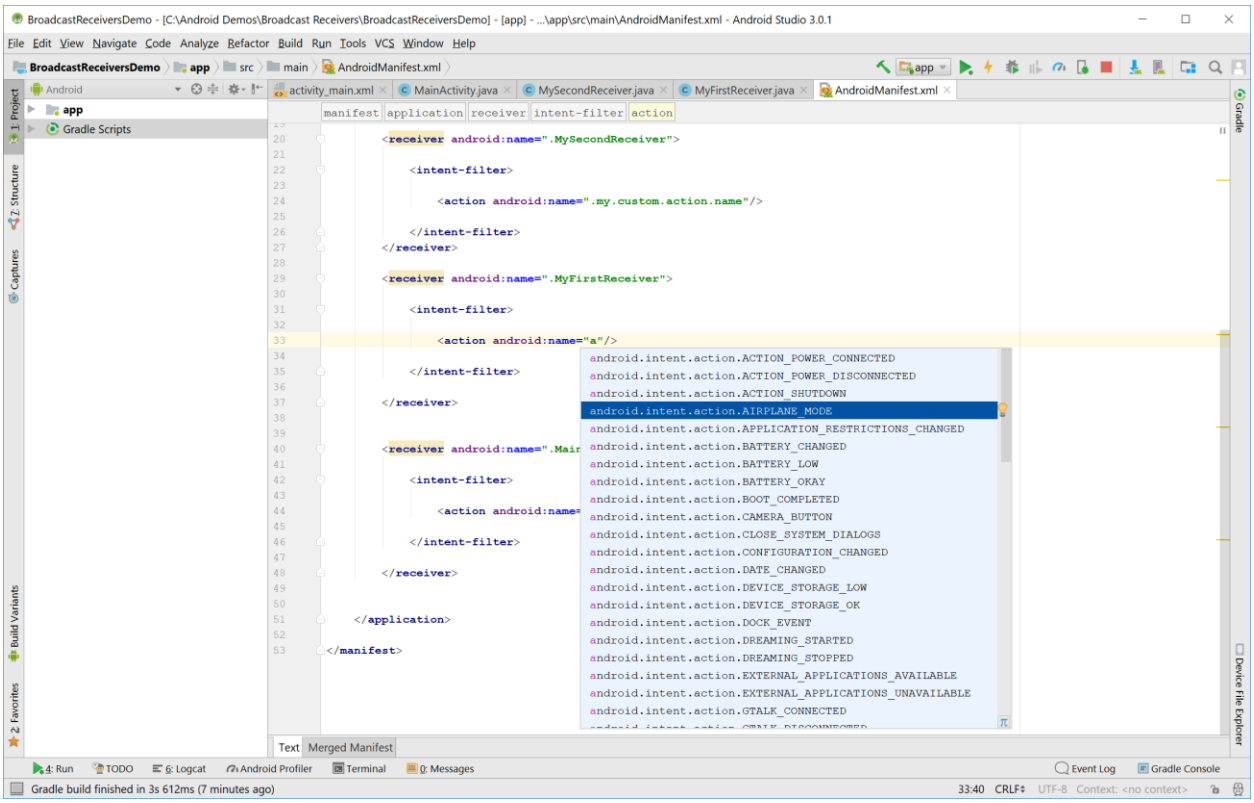
So once again we were able to send data from our activity to our receiver (third receiver here).

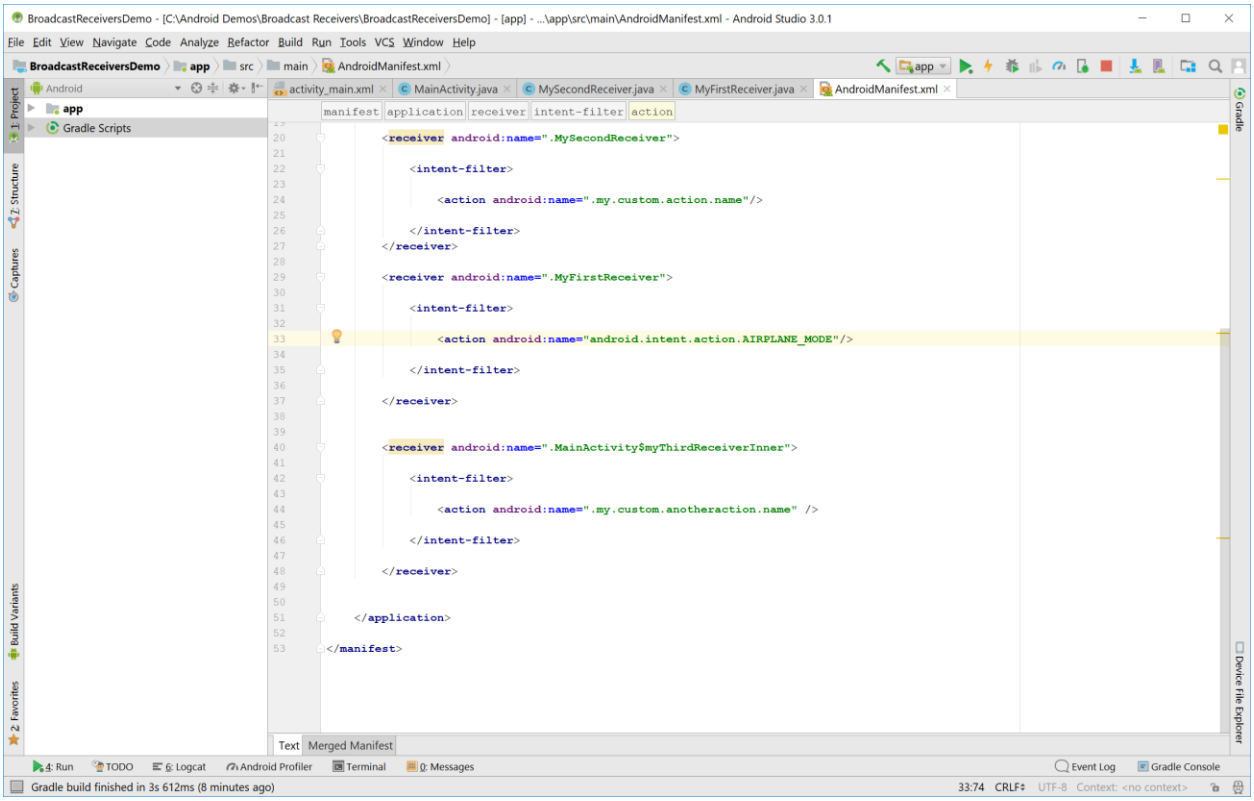
A few more practical examples:

In your manifest file remove the current action from the action tag and type “a” and you will see a list of many predefined actions in Android as shown:



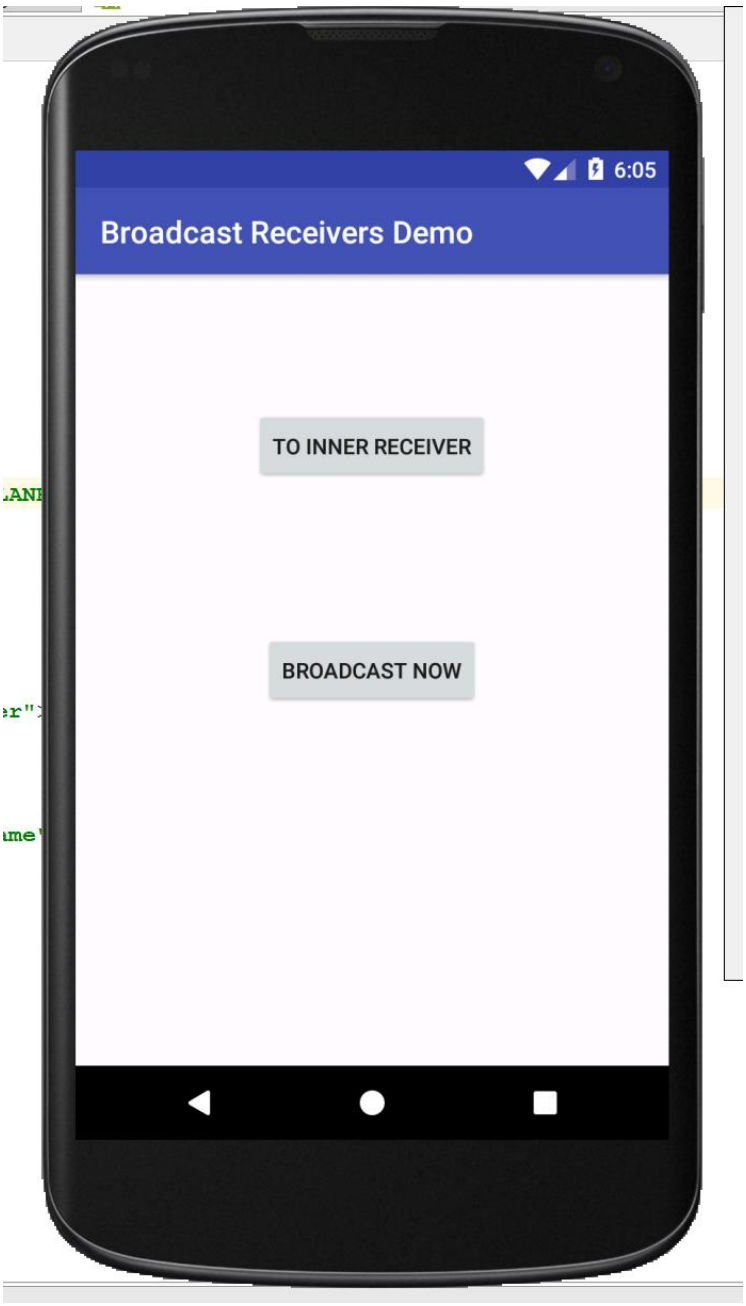
Pick the action airplane mode as shown:



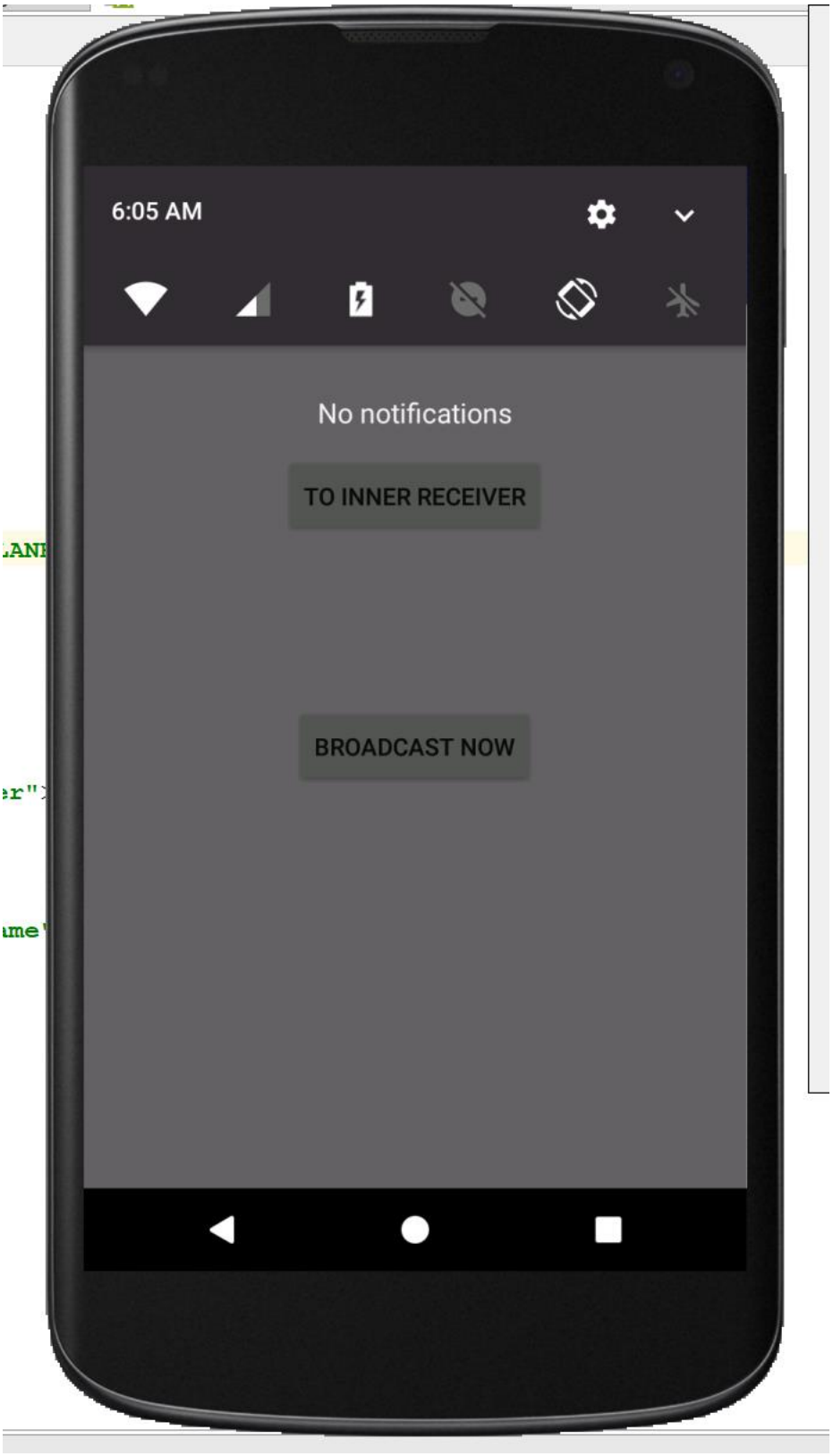


So now our first receiver is registered with the air plane mode action. So anytime you turn on the flight mode of your device, then the Android operating system will broadcast a message to all the applications that the flight mode has been turned on or it has been turned off if you turn it off and any broadcast receiver which is registered for this action will act accordingly.

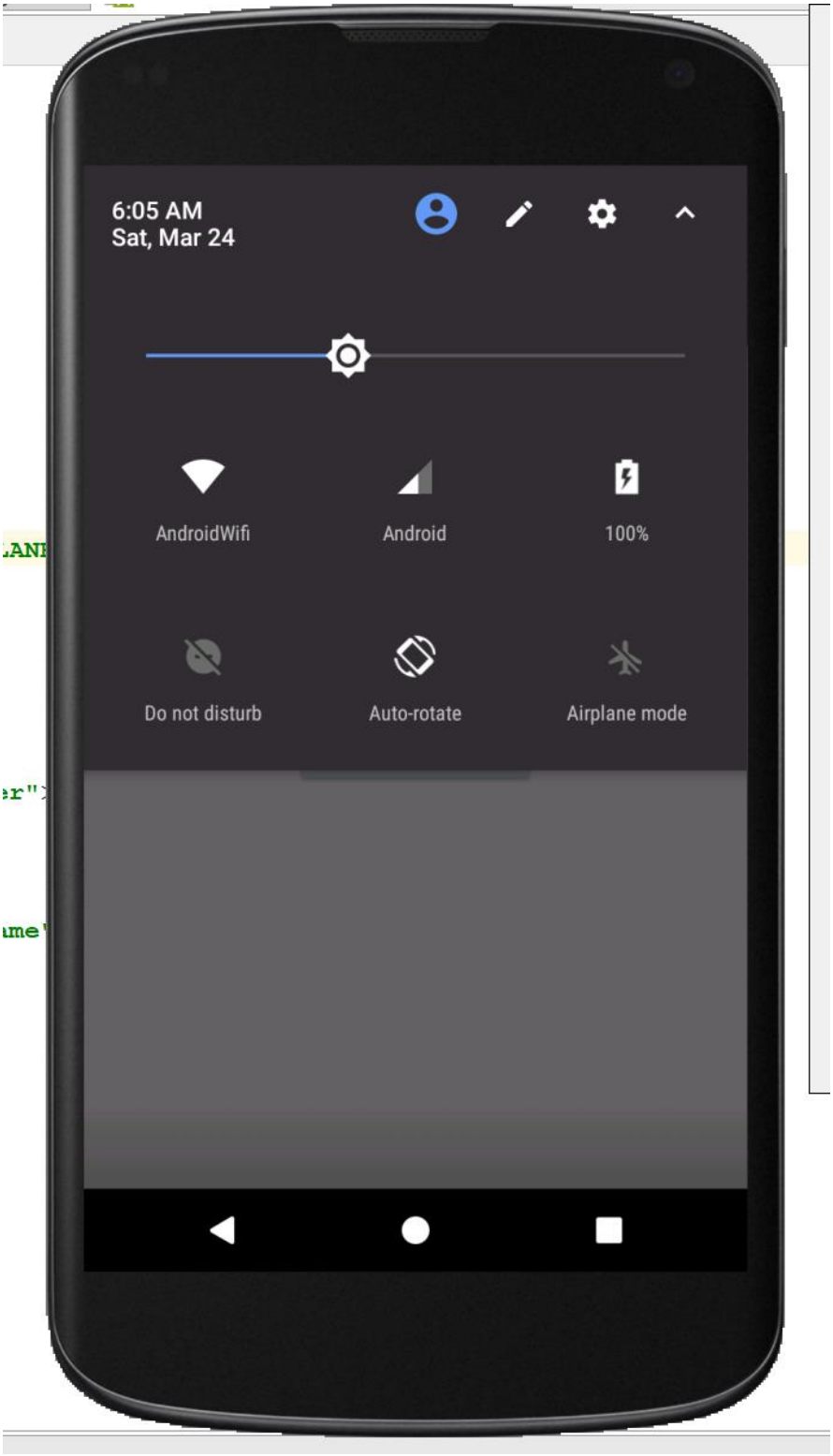
Now let's run our app to test this.



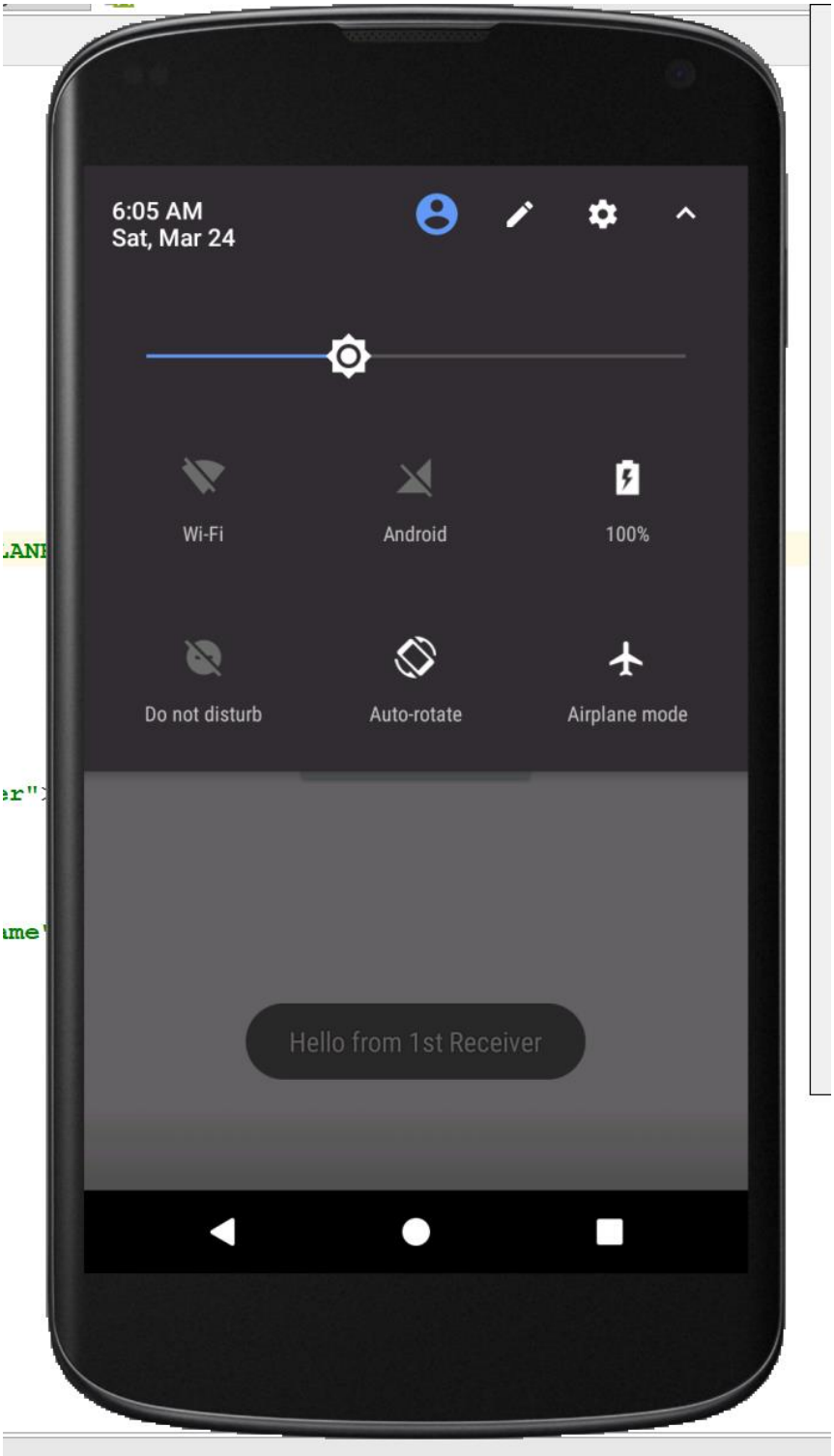
Pull down the notification panel:



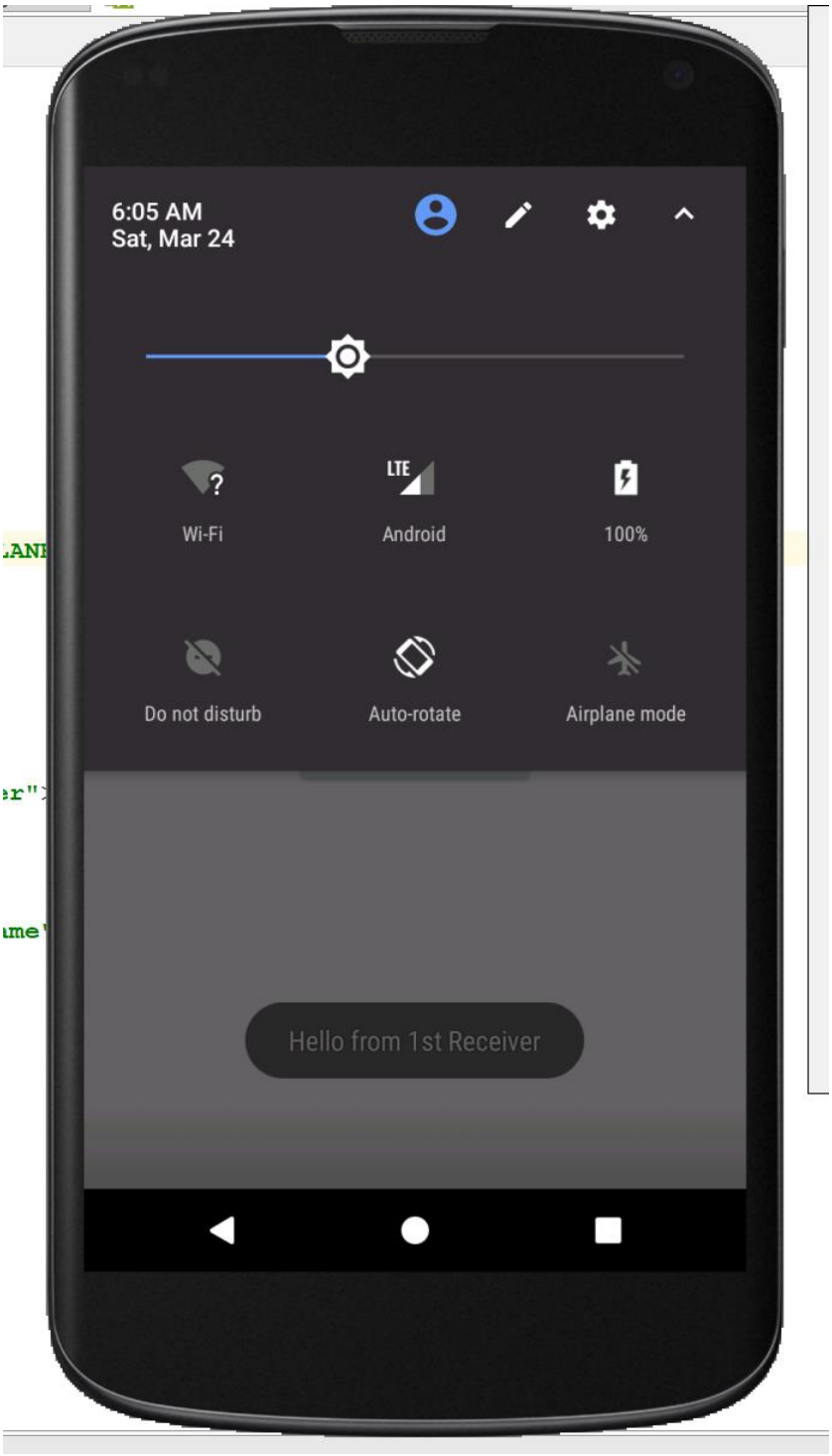
Click on the down arrow:



Click on the “Airplane mode” to turn it on and you see your first receiver responds as shown:



Now click on the “Airplane mode” again to turn the flight mode off and again you have a response from your first receiver as shown:



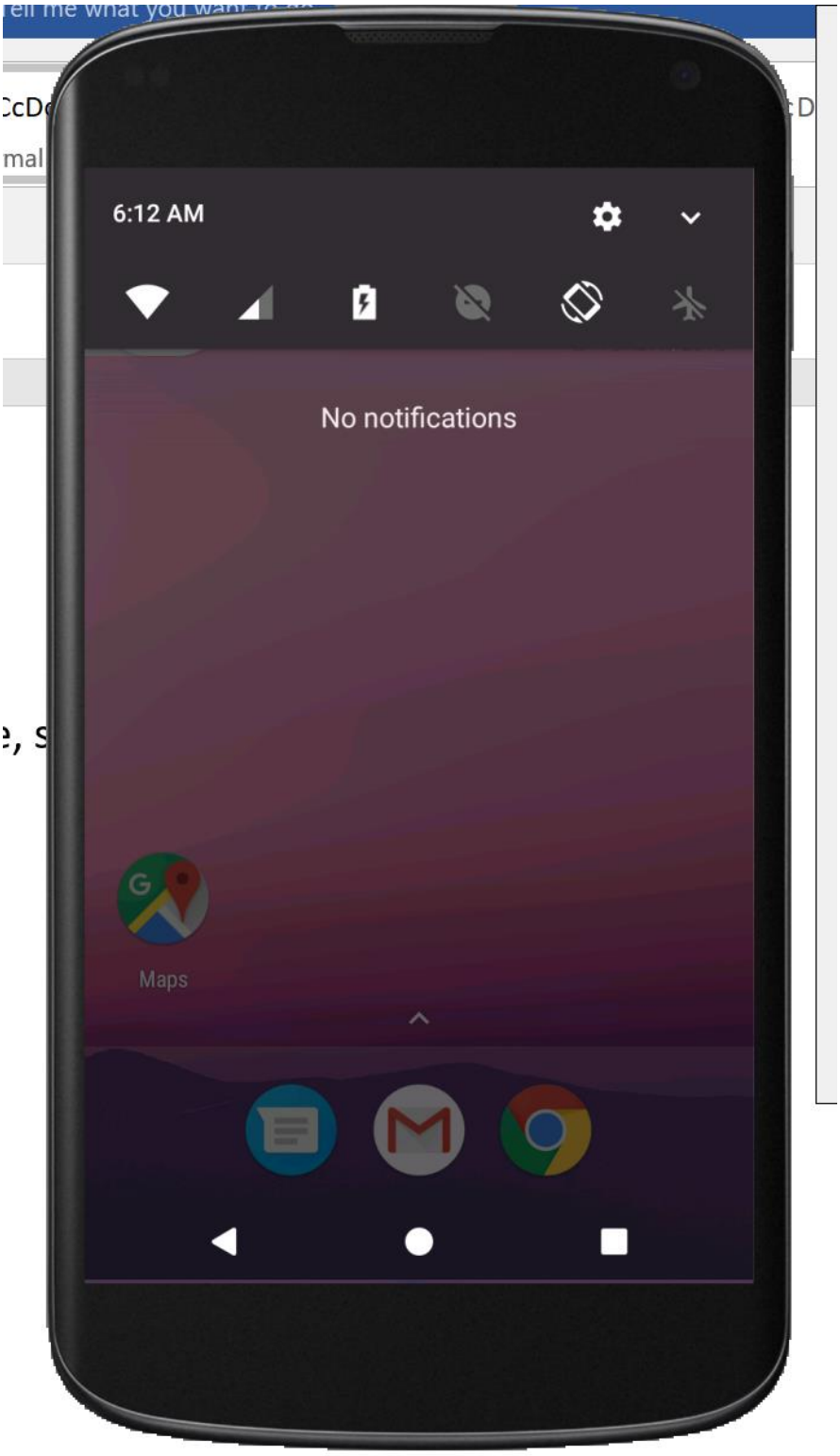
Now note that even if you shut down your application and then turn the flight mode on or off, your receiver still receives the broadcast as shown.

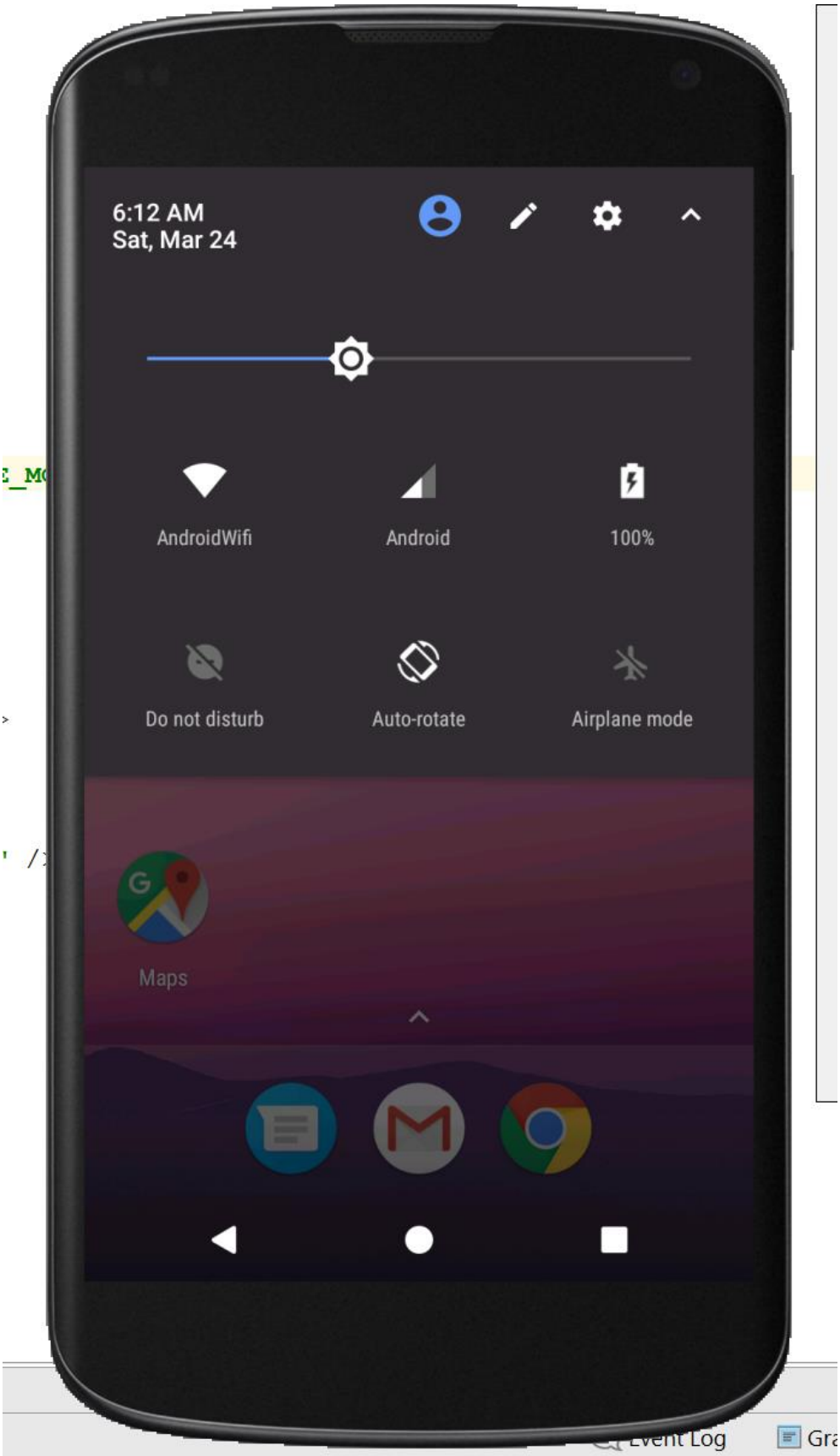
Run the app:

lig

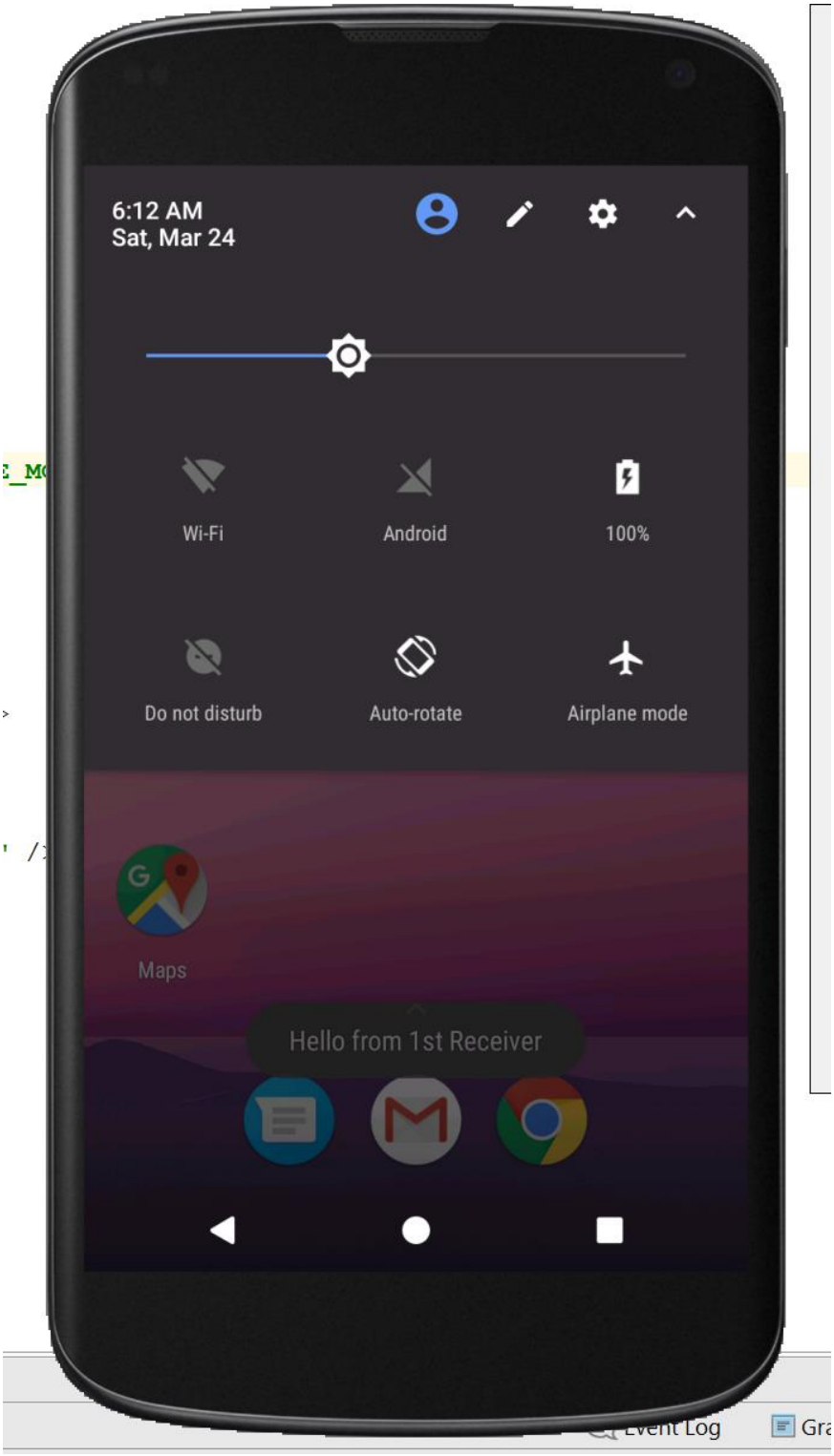
Click on the left arrow at the bottom to shut down the application:

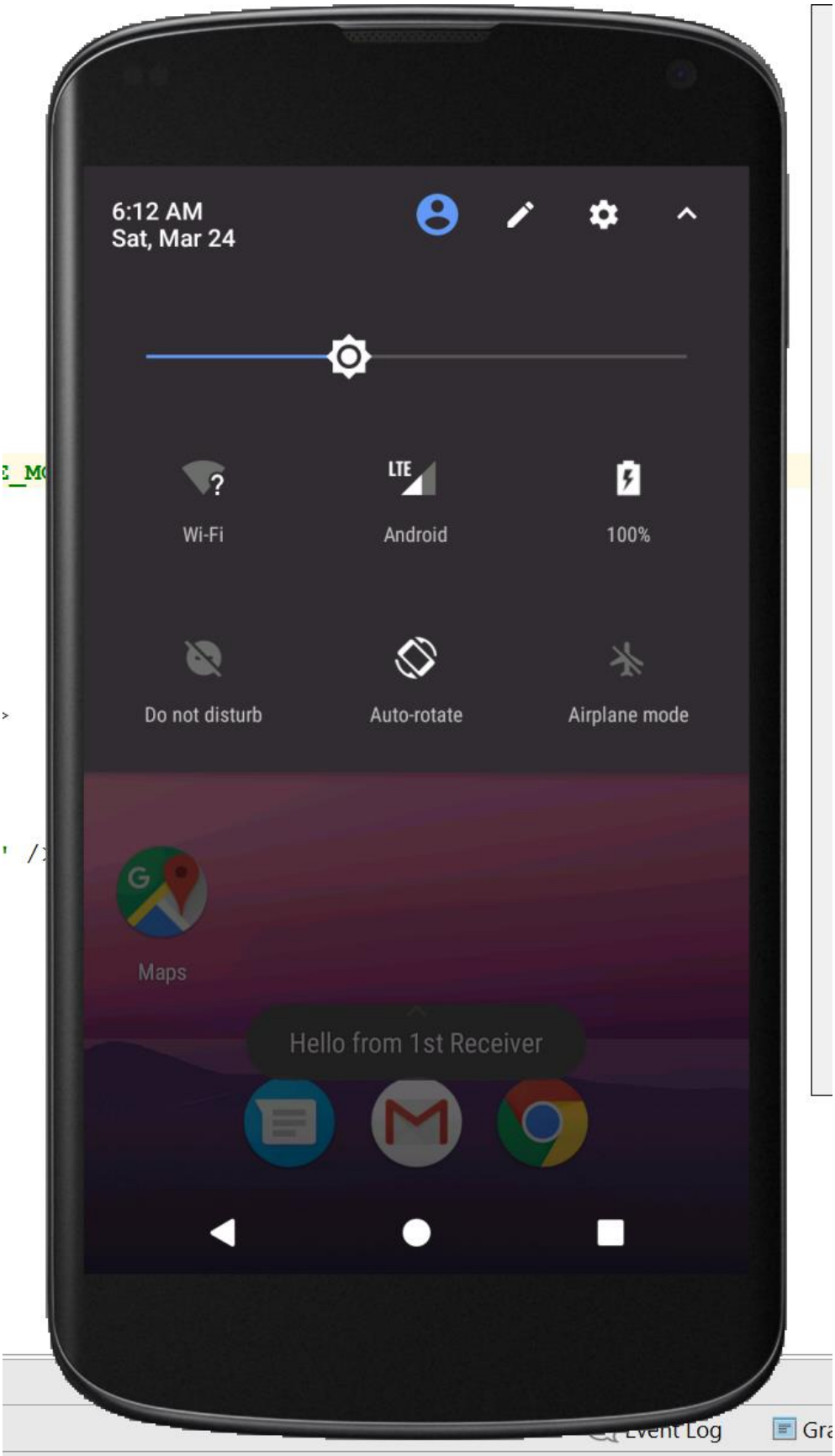
Now pull down your notification panel:




6:12 AM
Sat, Mar 24
AndroidWifi
Android
100%
Do not disturb
Auto-rotate
Airplane mode
Maps

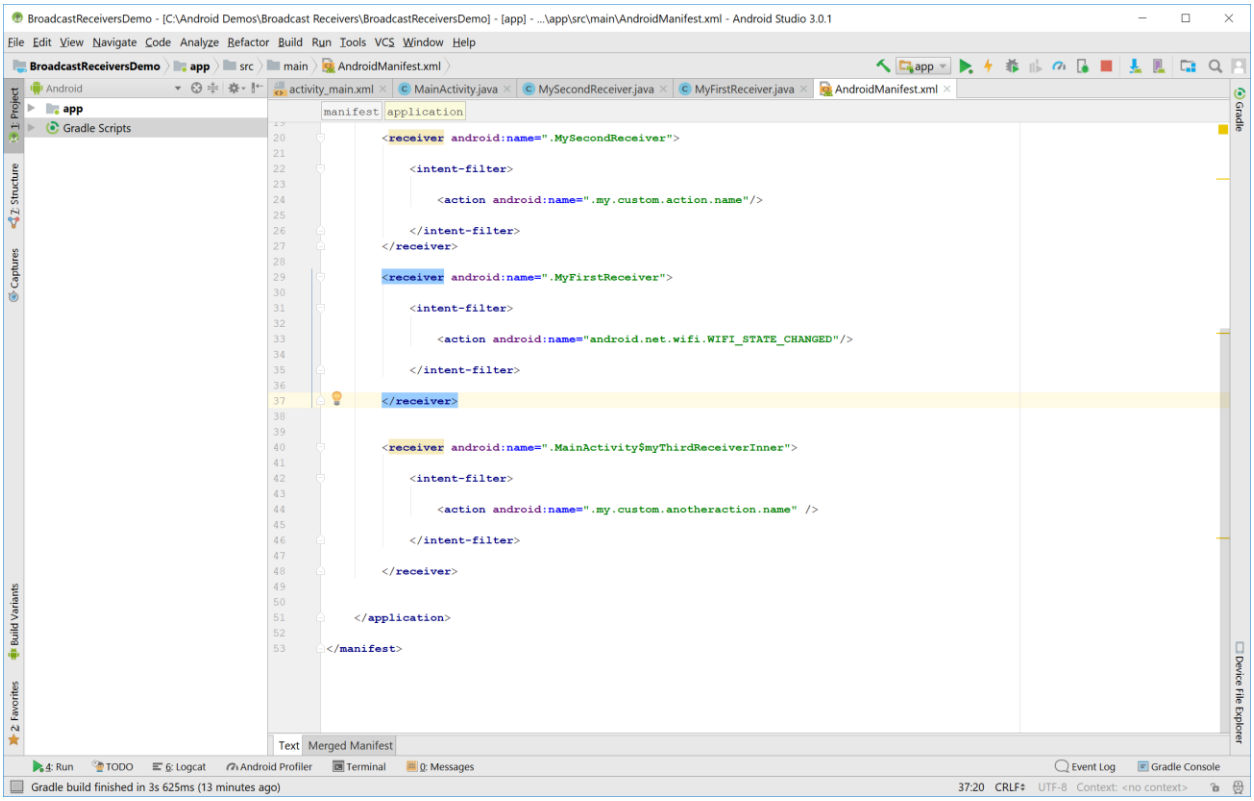
and turn on the flight mode, you will still get the response from your first receiver even if the app is not running as shown:



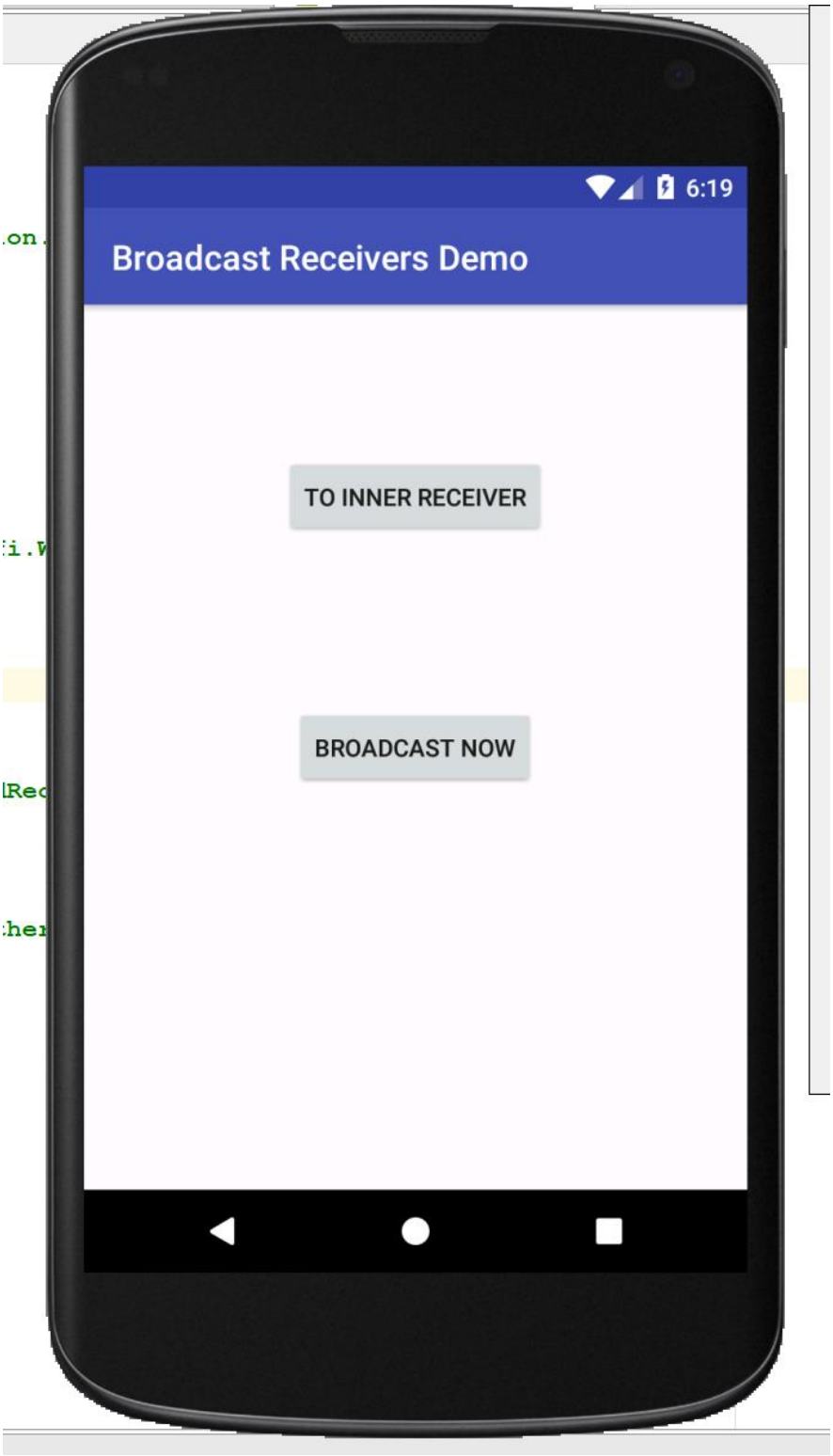

6:12 AM
Sat, Mar 24
Wi-Fi
Android
100%
Do not disturb
Auto-rotate
Airplane mode
Maps
Hello from 1st Receiver

So even if your application is not open and running, your application is still going to respond to any event to which your broadcast receiver has been registered.

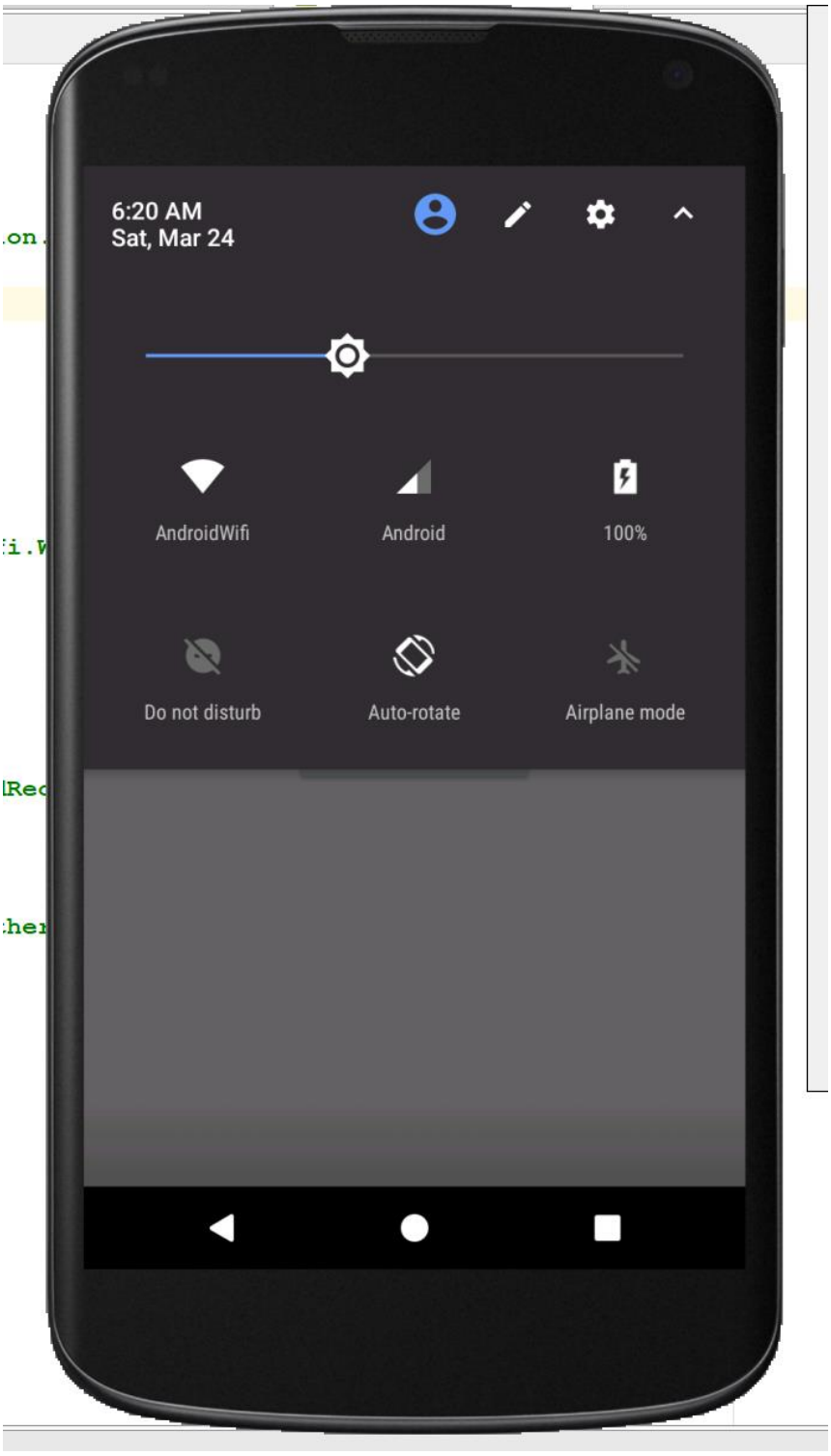
Now let's try a different event. Change the manifest file as shown (line 33):

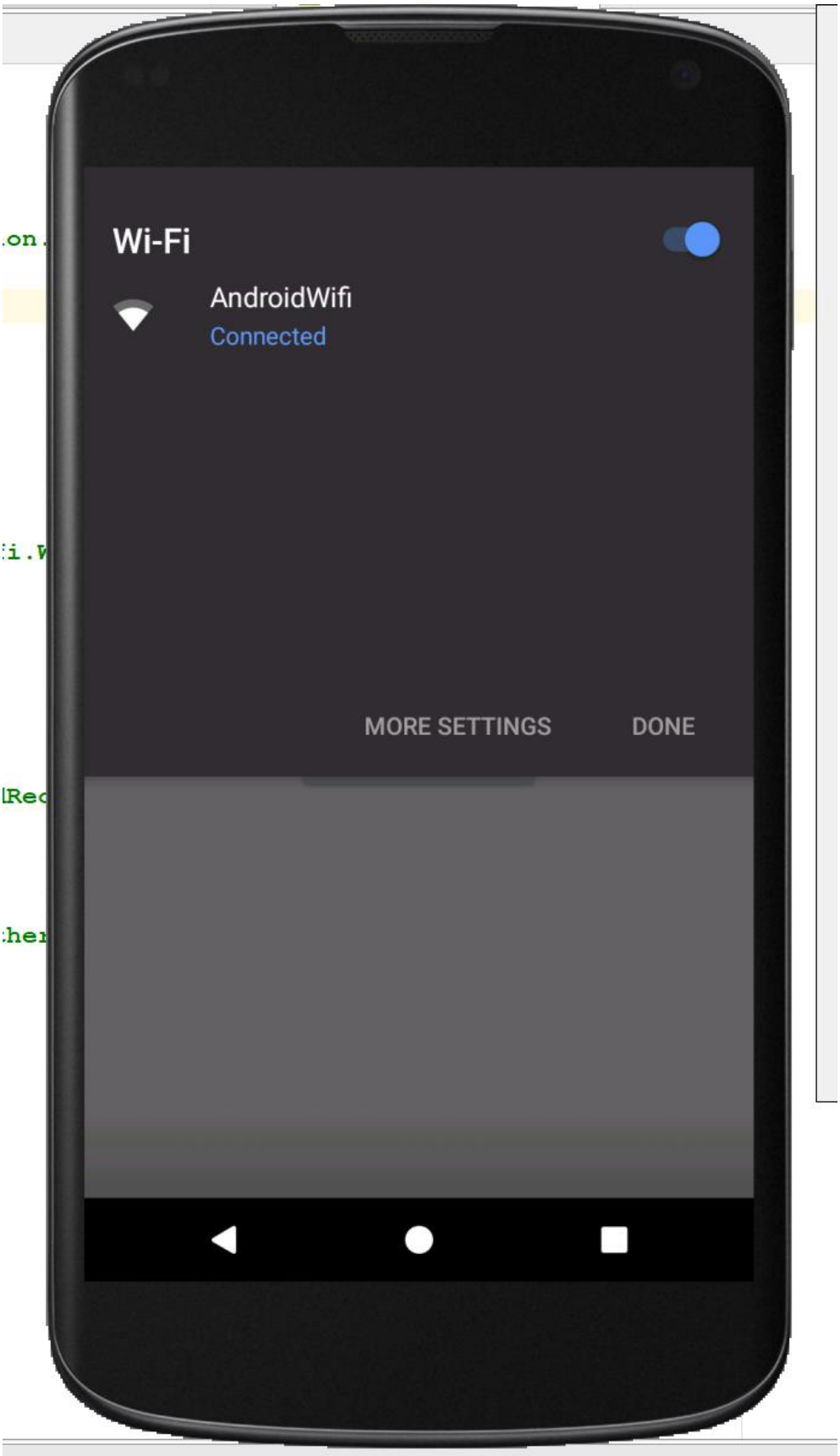


Now run the app to check if our receiver responds to the change in the state of Wi-Fi as shown:

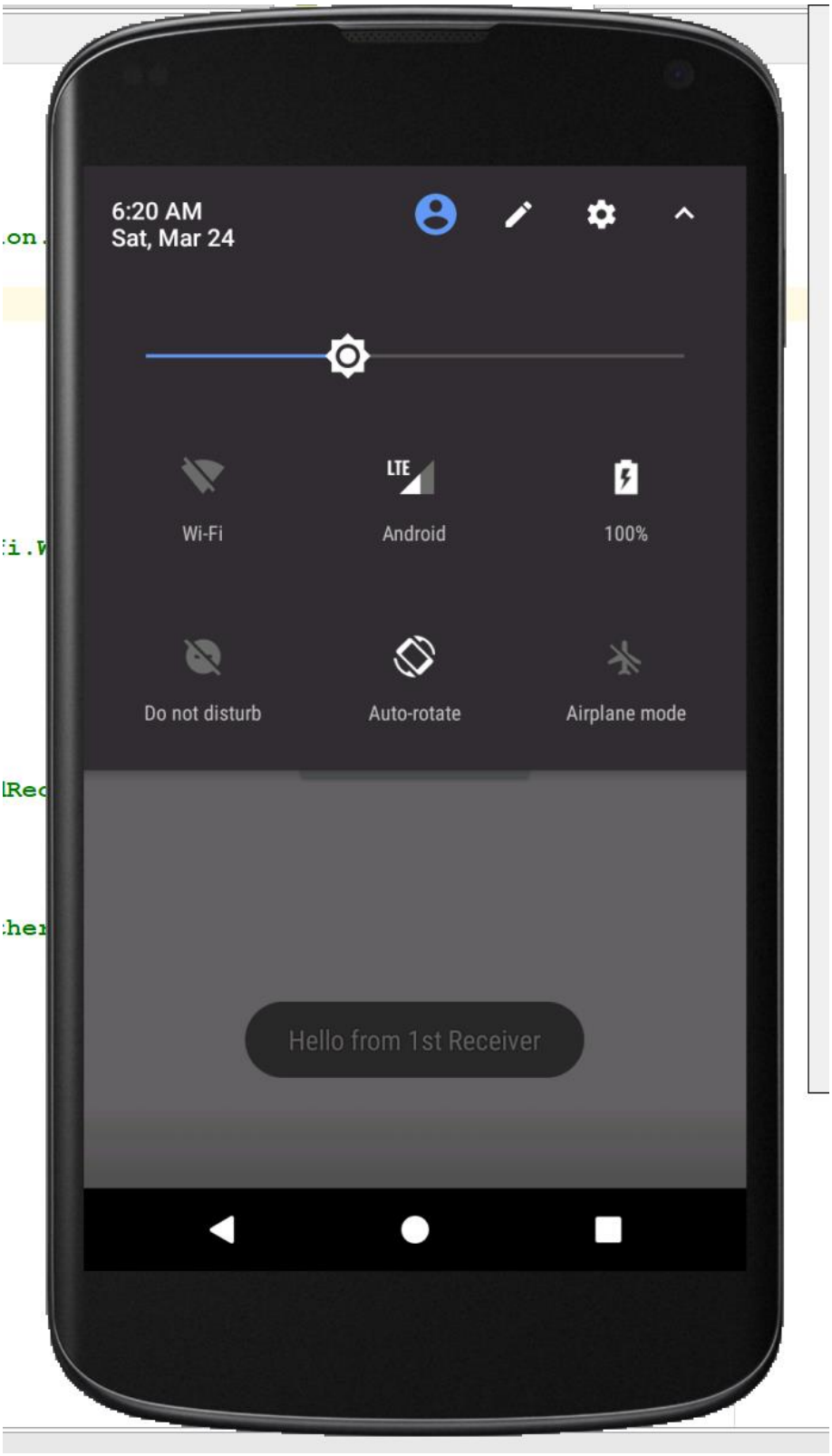


Pull down the menu:

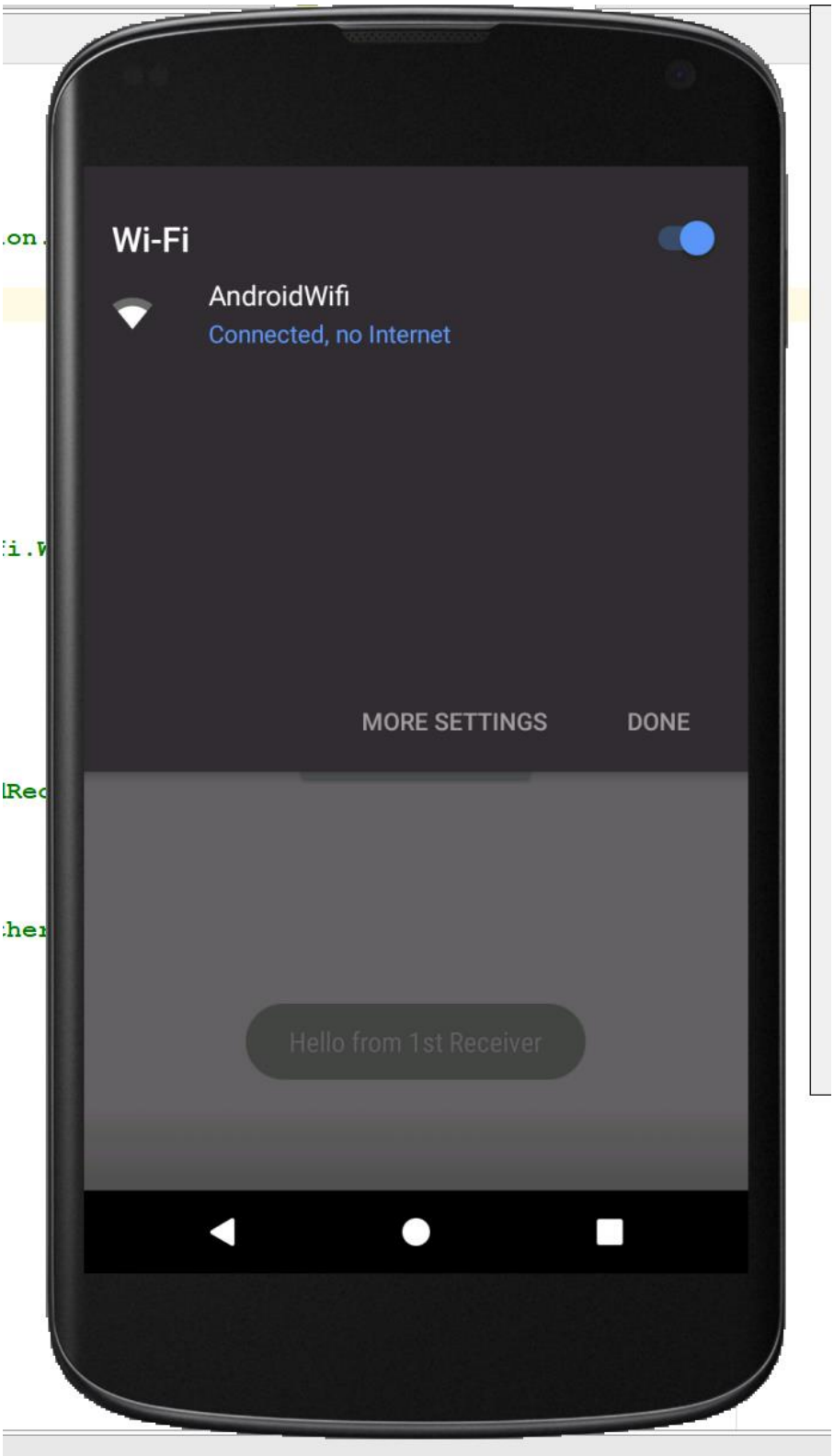


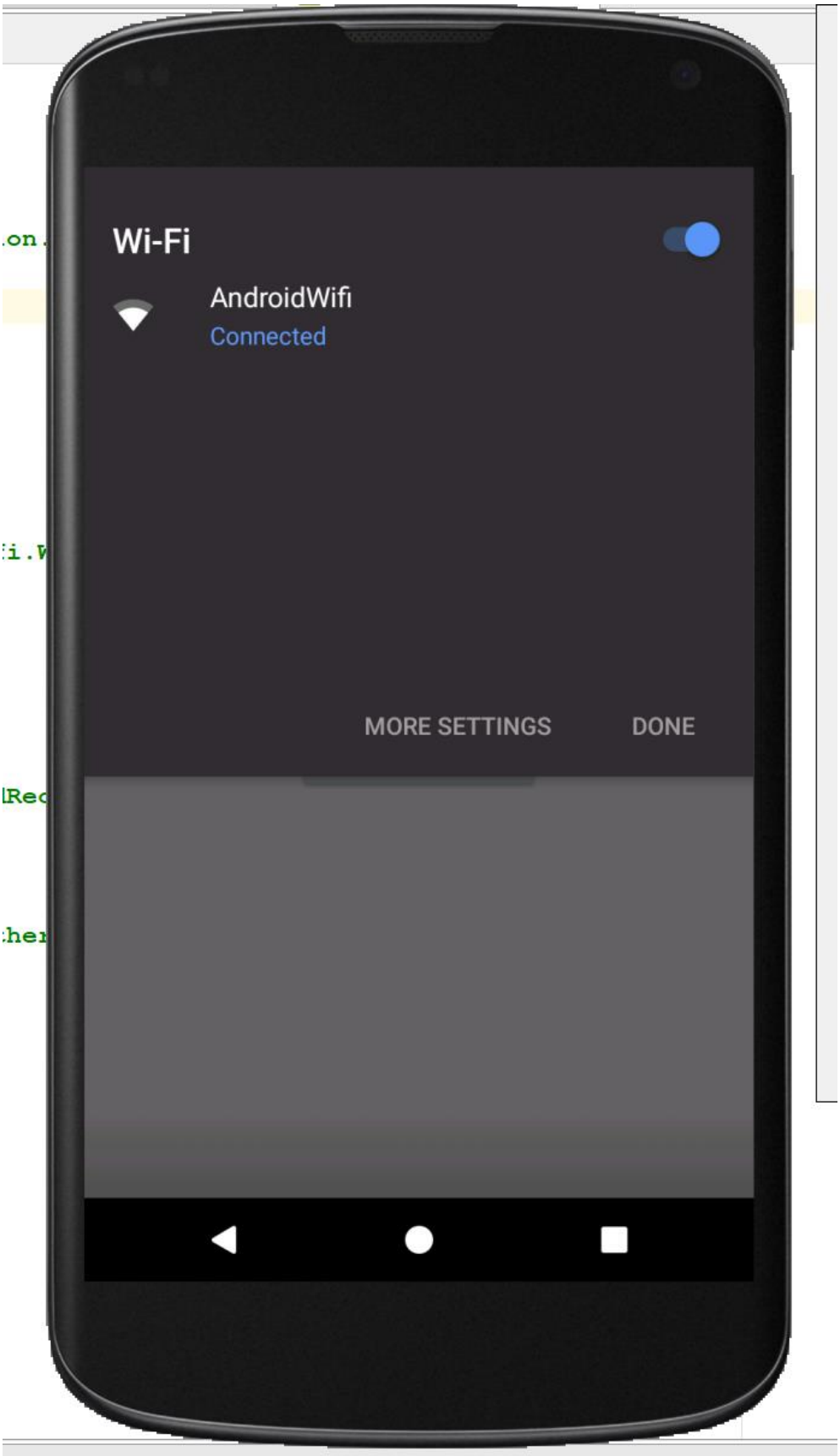

Wi-Fi
AndroidWifi
Connected
MORE SETTINGS
DONE

Turn off the Wi-Fi and your receiver responds to the event as shown:

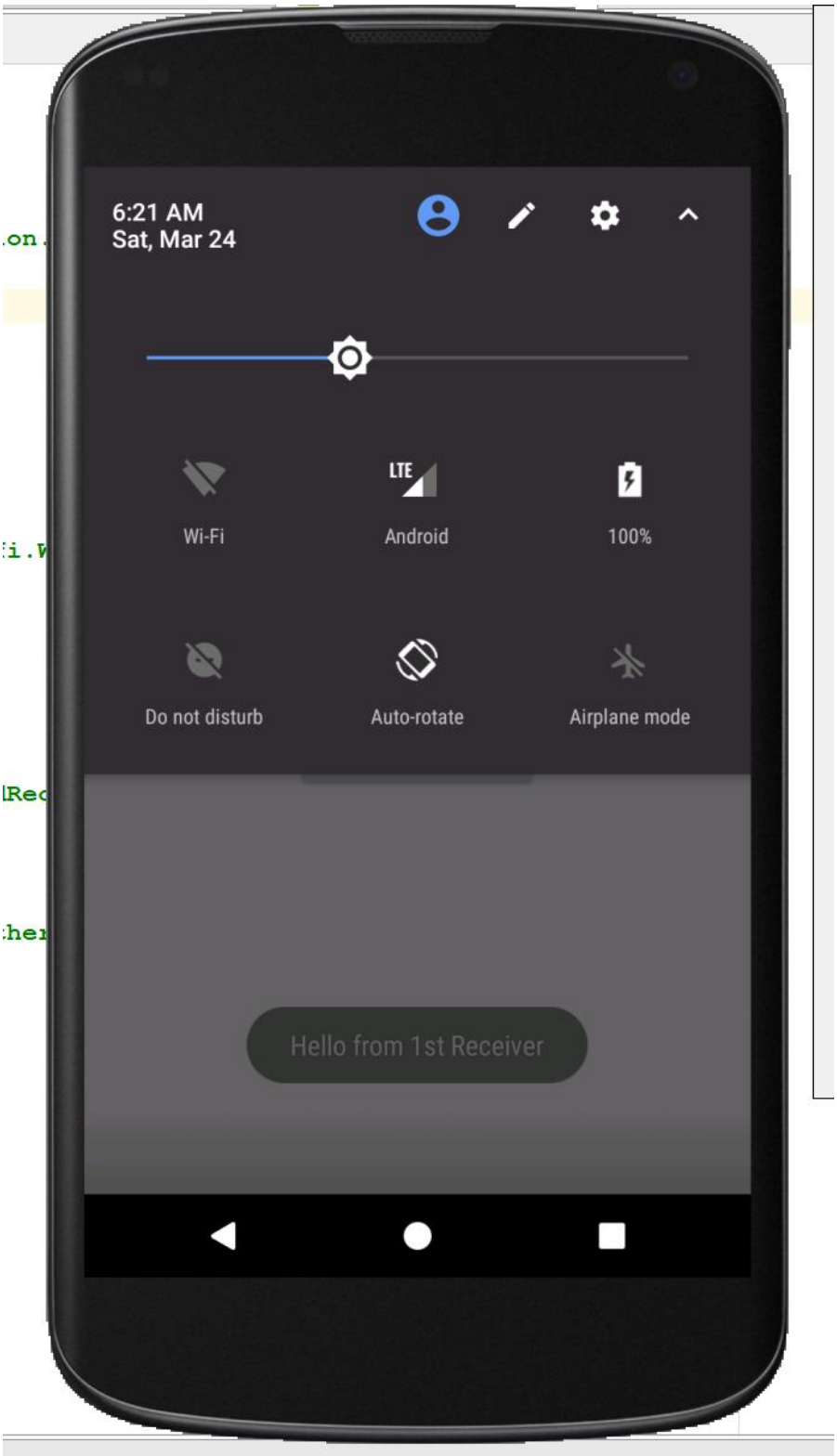


Turn it on again and again your receiver responds to the change in the state of Wi-Fi as shown:

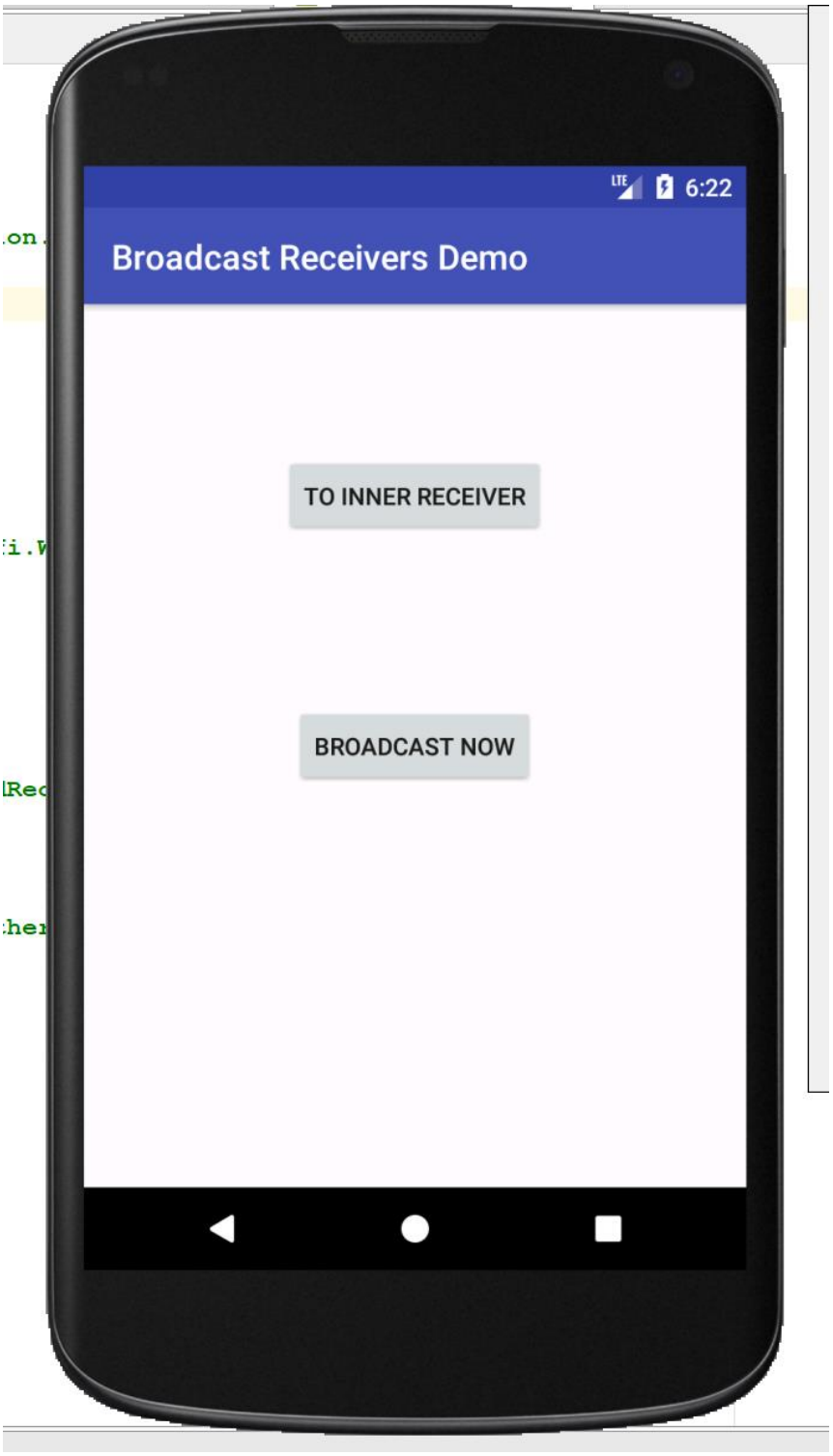



Wi-Fi
AndroidWifi
Connected
MORE SETTINGS
DONE

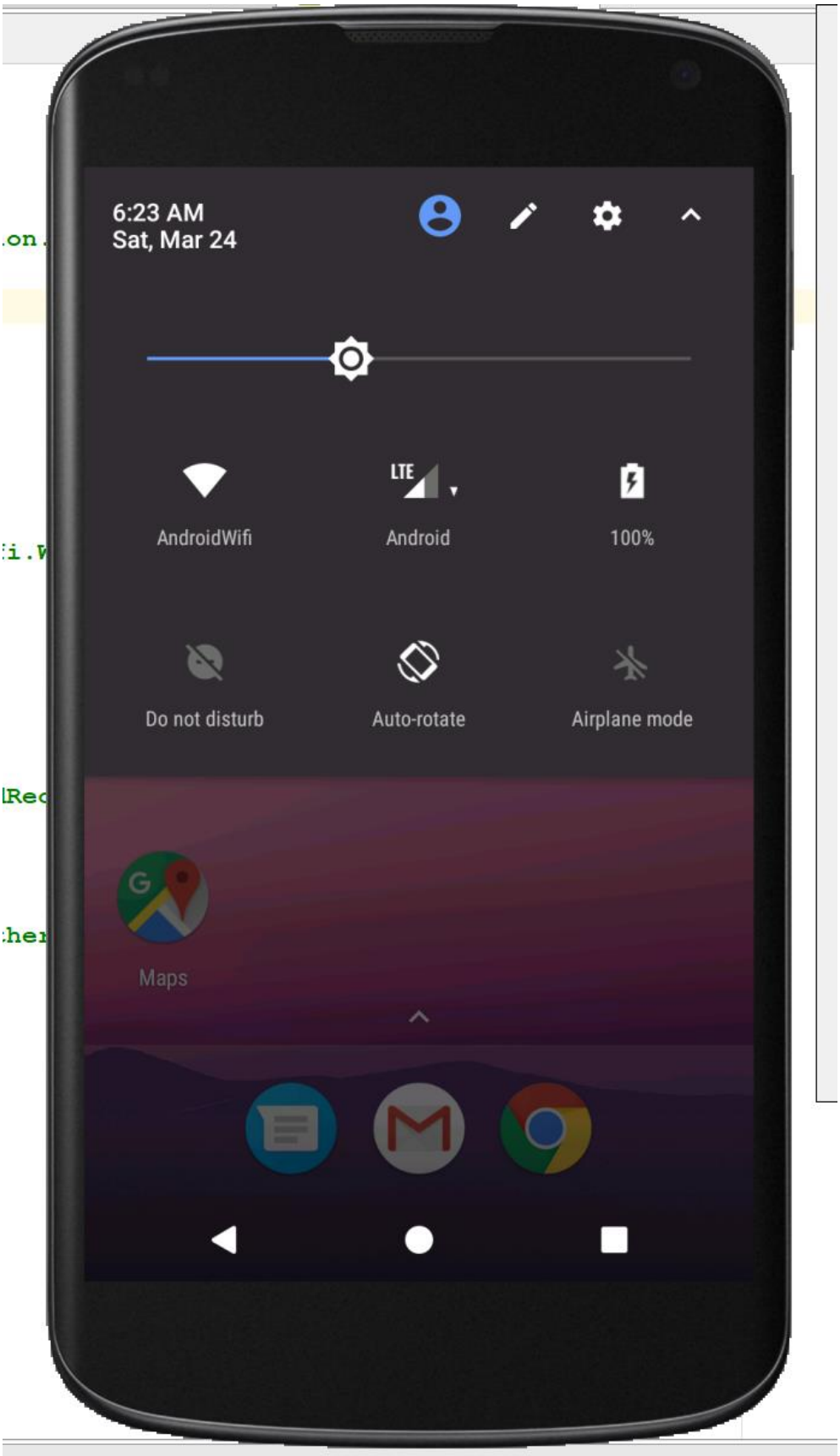
Turn it off again and another response from the receiver:

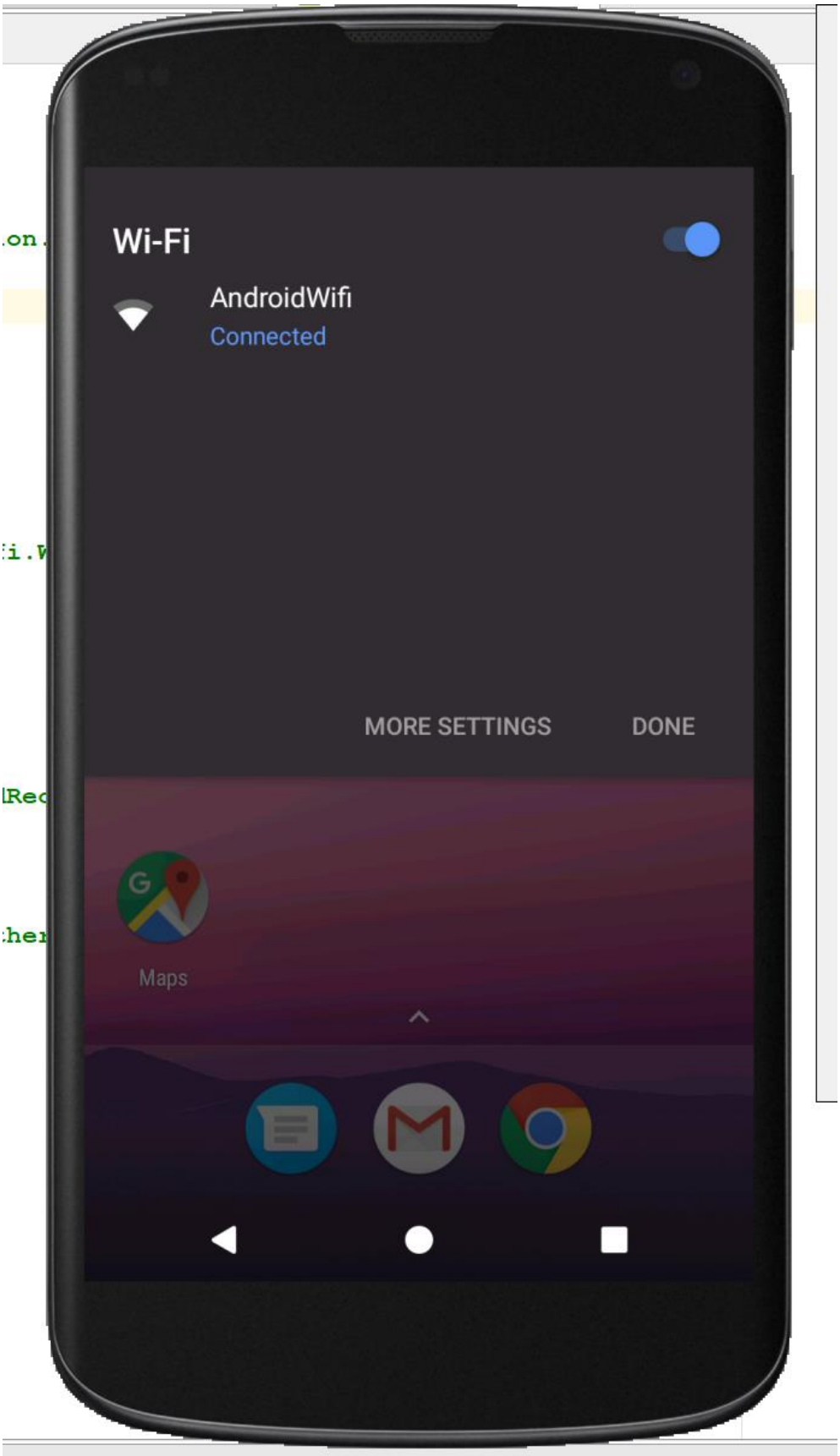


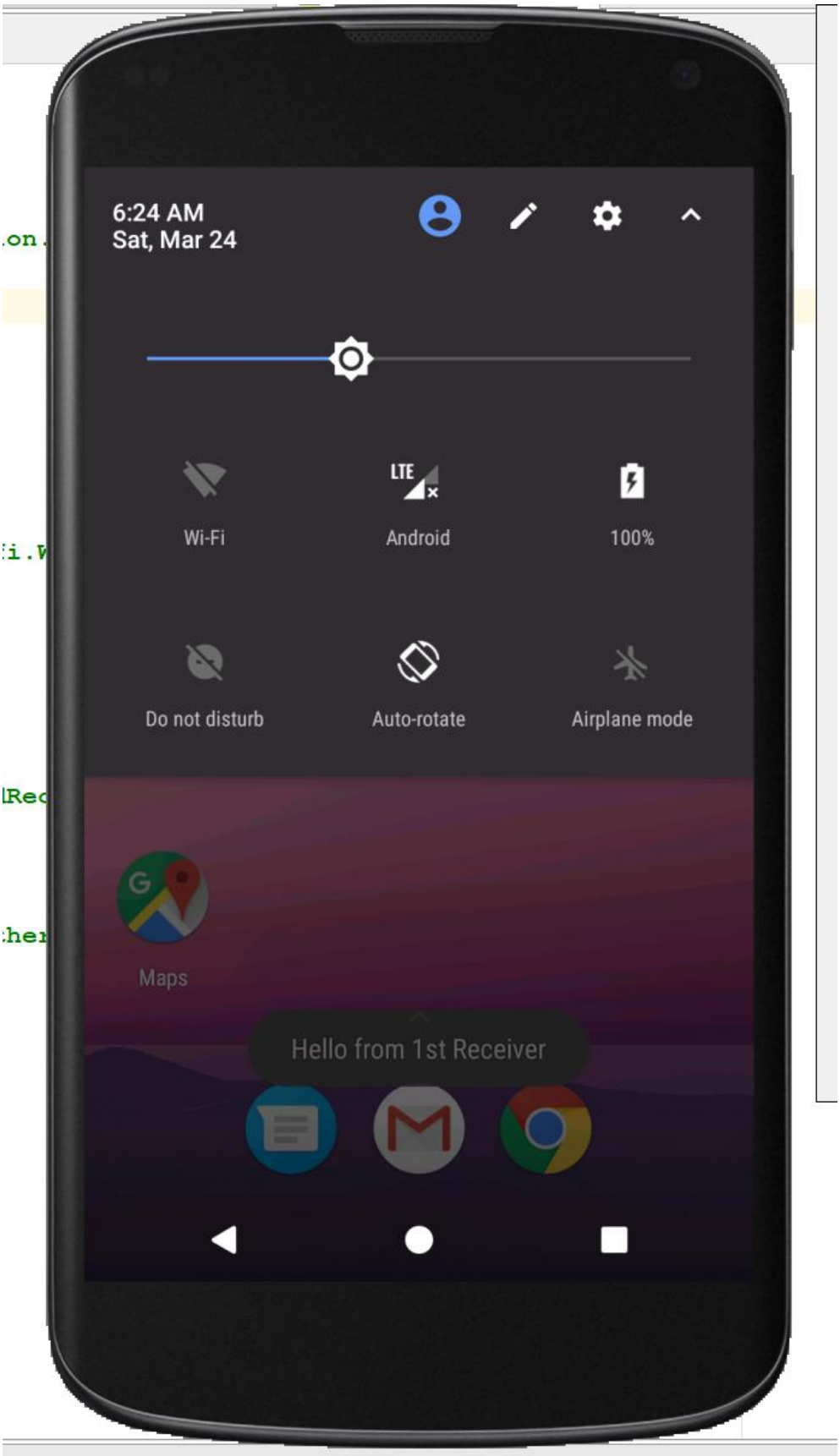
Now in your app:



Close the app by clicking on the left arrow and try to turn on and off the Wi-Fi and you'll see our first receiver still responds to the change in the state of Wi-Fi, another word to the action it is registered for even if the app is not running.


6:23 AM
Sat, Mar 24
AndroidWifi
Android
100%
Do not disturb
Auto-rotate
Airplane mode
Maps


Wi-Fi
AndroidWifi
Connected
MORE SETTINGS
DONE
Maps


6:24 AM
Sat, Mar 24
Wi-Fi
Android
100%
Do not disturb
Auto-rotate
Airplane mode
Maps
Hello from 1st Receiver