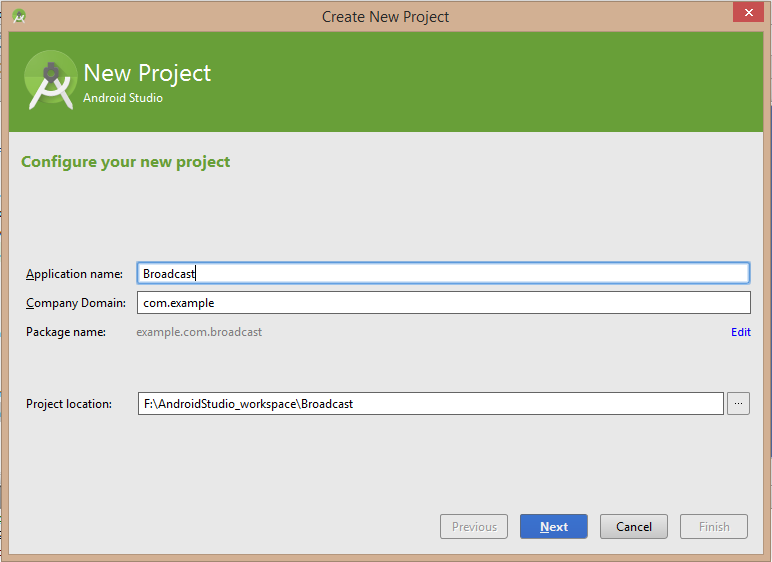
Lab: Android Broadcast Service (Android Studio)

**Step 1:** Create a simple Android Application using Android Studio. Follow the option **File -> NewProject**. Now name your application as **Broadcast** using the wizard window as follows:



**Step 2:**

Following is the content of the modified main activity file *java/com.example.broadcast/MainActivity.java*. This file can include each of the fundamental lifecycle methods. We have added *broadcastIntent()* method to broadcast a custom intent.

package example.com.broadcast;

import android.os.Bundle;

import android.app.Activity;

import android.view.Menu;

import android.content.Intent;

import android.view.View;

public class MainActivity extends Activity {

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

//getMenuInflater().inflate(R.menu.activity\_main, menu);

return true;

}

// broadcast a custom intent.

public void broadcastIntent(View view)

{

Intent intent = new Intent();

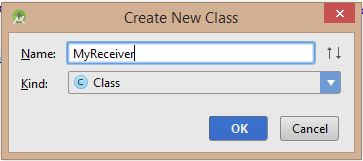
intent.setAction("com.tutorialspoint.CUSTOM\_INTENT");

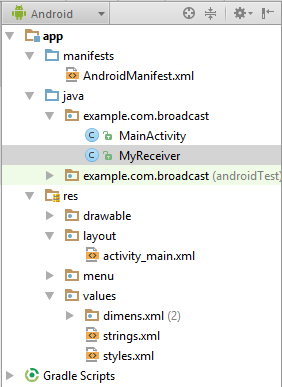
sendBroadcast(intent);

}

}

Create a new class in the *com.example.broadcast* package. Following is the content of *java/com.example.broadcast/MyReceiver.java*.





package example.com.broadcast;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

import android.widget.Toast;

public class MyReceiver extends BroadcastReceiver {

@Override

public void onReceive(Context context, Intent intent) {

Toast.makeText(context, "Intent Detected.", Toast.LENGTH\_LONG).show();

}

}

Following will the modified content of *AndroidManifest.xml* file. Here we have added <service.../> tag to include our service:

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="example.com.broadcast" >

<application

android:allowBackup="true"

android:icon="@drawable/ic\_launcher"

android:label="@string/app\_name"

android:theme="@style/AppTheme" >

<activity

android:name=".MainActivity"

android:label="@string/title\_activity\_main" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER"/>

</intent-filter>

</activity>

<receiver android:name="MyReceiver">

<intent-filter>

<action android:name="com.tutorialspoint.CUSTOM\_INTENT">

</action>

</intent-filter>

</receiver>

</application>

</manifest>

Following will be the content of *res/layout/activity\_main.xml* file to include a button to broadcast our custom intent:

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools" android:layout\_width="match\_parent"

android:layout\_height="match\_parent" android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin" tools:context=".MainActivity">

<Button android:id="@+id/btnStartService"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="@string/broadcast\_intent"

android:onClick="broadcastIntent"/>

</RelativeLayout>

Following will be the content of *res/values/strings.xml* to define two new constants:

<?xml version="1.0" encoding="utf-8"?>

<resources>

<string name="app\_name">Broadcast</string>

<string name="hello\_world">Hello world!</string>

<string name="action\_settings">Settings</string>

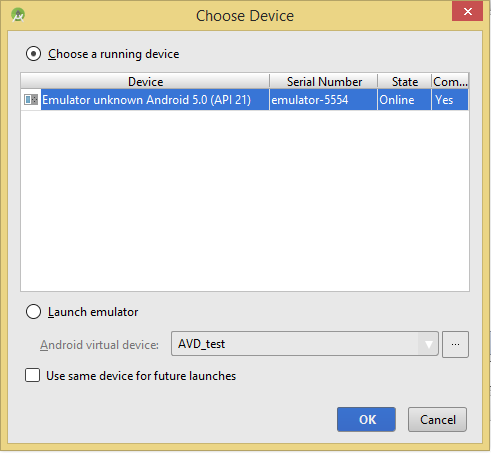
<string name="title\_activity\_main">MainActivity</string>

<string name="broadcast\_intent">Broadcast Intent</string>

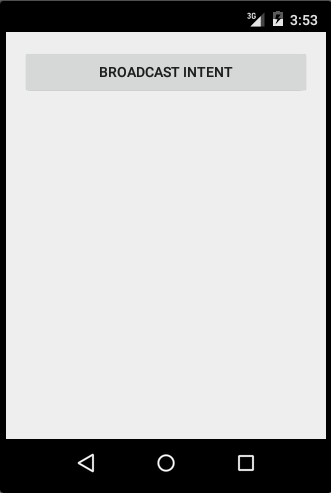
</resources>

**Step 3:**

Let's try to run our modified **Hello World application** we just modified. I assume you had created your AVD while doing environment setup. At first, start your existed **AVD**, and then click **Run** Eclipse Run Icon icon from the toolbar. Choose this AVD, and click “OK”:



Then the AVD will display like as following:



Now to broadcast our custom intent, let's click on Broadcast Intent button, this will broadcast our custom intent "com.tutorialspoint.CUSTOM\_INTENT" which will be intercepted by our registered BroadcastReceiver ie. MyReceiver and as per our implemented logic a toast will appear on the bottom of the simulator as follows:

