Lab 3 – BroadcastReceiver

KUAN-TING LAI

2022/09/26

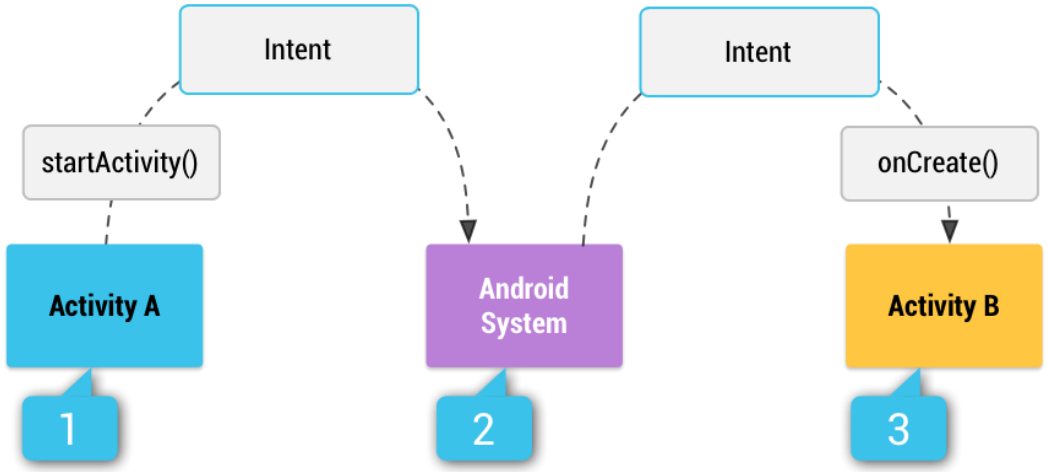
Broadcast Overview

- Publish-subscribe pattern
- Use Intents to communicate
- System broadcasts
- Local broadcasts

Ref: https://developer.android.com/guide/components/activities/intro-activities#java

Intent and Intent Filters

- A message object used to invoke other components
- Starting an activity
- Starting a service
- Delivering a broadcast



Ref: https://developer.android.com/guide/components/intents-filters

Intent Filters

- <intent-filter>
  - action
  - category
  - data

```
<action android:name="string" />
<category android:name="string" />
<data android:mimeType="string" android:scheme="string" android:host="string" android:port="string"
android:path="string" android:pathPrefix="string" android:pathPattern="string" />
```

```
URI= <scheme>://<host>:<port>/[<path>|<pathPrefix>|<pathPattern>]
```

Note:

When an intent is sent, the system checks each component in the Manifest for Intent filters and launches the appropriate component. If the component does not have a filter, it can only receive an intent with a clearly specified component.

System Broadcasts

Event Constant	Description
android.intent.action.BATTERY_CHANGED	Sticky broadcast containing the charging state, level, and other information about the battery.
android.intent.action.BATTERY_LOW	Indicates low battery condition on the device.
android.intent.action.BATTERY_OKAY	Indicates the battery is now okay after being low.
android.intent.action.BOOT_COMPLETED	This is broadcast once, after the system has finished booting.
android.intent.action.BUG_REPORT	Show activity for reporting a bug.
android.intent.action.CALL	Perform a call to someone specified by the data.
android.intent.action.CALL_BUTTON	The user pressed the "call" button to go to the dialer or other appropriate UI for placing a call.
android.intent.action.DATE_CHANGED	The date has changed.
android.intent.action.REBOOT	Have the device reboot.

Ref: https://stackoverflow.com/questions/4031740/where-is-the-all-android-broadcast-intent-list

System Broadcast Changes

- Android 9 (API level 28)
  - NETWORK_STATE_CHANGED_ACTION broadcast doesn't receive information about the user's location or personally identifiable data.
  - Wi-Fi don't contain SSIDs, BSSIDs, connection information, or scan results. To get this information, call getConnectionInfo() instead.
- Android 8.0 (API level 26)
  - Cannot use the manifest to declare a receiver for most implicit broadcasts. You can still use a context-registered receiver when the user is actively using your app.
- Android 7.0
  - Android 7.0 (API level 24) and higher don't send the following system broadcasts:
    - ACTION_NEW_PICTURE
    - ACTION_NEW_VIDEO
  - You must register the CONNECTIVITY_ACTION broadcast using registerReceiver(BroadcastReceiver, IntentFilter). Declaring a receiver in the manifest doesn't work.

Ref: https://developer.android.com/guide/components/broadcasts

Manifest-declared receivers

- Specify receiver and intent-filter

```
<receiver android:name=".MyBroadcastReceiver" android:exported="true">
   <intent-filter>
      <action android:name="android.intent.action.BOOT_COMPLETED"/>
      <action android:name="android.intent.action.INPUT_METHOD_CHANGED" />
   </intent-filter>
</receiver>
```

Note:

If your app targets API level 26 or higher, you cannot use the manifest to declare a receiver for implicit broadcasts (broadcasts that do not target your app specifically), except for a few implicit broadcasts that are exempted from that restriction. In most cases, you can use scheduled jobs instead.

BroadcastReceiver

```
public class MyBroadcastReceiver extends BroadcastReceiver {
    private static final String TAG = "MyBroadcastReceiver";
    @Override
    public void onReceive(Context context, Intent intent) {
        StringBuilder sb = new StringBuilder();
        sb.append("Action: " + intent.getAction() + "\n");
        sb.append("URI: " + intent.toUri(Intent.URI_INTENT_SCHEME).toString() + "\n");
        String log = sb.toString();
        Log.d(TAG, log);
        Toast.makeText(context, log, Toast.LENGTH_LONG).show();
    }
}
```

Context Registered Receivers

- Create an instance of MyBroadcastReceiver

```
BroadcastReceiver br = new MyBroadcastReceiver();
```

- Create an IntentFilter

```
IntentFilter filter = new IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION);
filter.addAction(Intent.ACTION_AIRPLANE_MODE_CHANGED);
this.registerReceiver(br, filter);
```

Note: To register for local broadcasts, call **~~LocalBroadcastManager.registerReceiver(BroadcastReceiver, IntentFilter)~~** instead.

Register Receiver in onCreate()

AndroidManifest.xml | MainActivity.java | MyBroadcastReceiver.java | SelfReceiver.java | activity_main.xml

```java
package com.lab3.broadcast;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        BroadcastReceiver br = new MyBroadcastReceiver();
        IntentFilter filter = new IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION);
        filter.addAction(Intent.ACTION_AIRPLANE_MODE_CHANGED);
        this.registerReceiver(br, filter);
    }
```

MyBroadcastReceiver.java

```java
public class MyBroadcastReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        StringBuilder sb = new StringBuilder();
        sb.append("Action: " + intent.getAction() + "\n");
        sb.append("URI: " + intent.toUri(Intent.URI_INTENT_SCHEME).toString() + "\n");
        String log = sb.toString();
        Toast.makeText(context, log, Toast.LENGTH_LONG).show();
    }
}
```

Effects on Process State

- If we declare receivers in Manifest.xml, after onReceive(), the system can kill the process to reclaim memory anytime
- Use goAsync() to flag we need more time to finish task

```
public class MyBroadcastReceiver extends BroadcastReceiver {
    private static final String TAG = "MyBroadcastReceiver";
    @Override
    public void onReceive(Context context, Intent intent) {
        final PendingResult pendingResult = goAsync();
        Task asyncTask = new Task(pendingResult, intent);
        asyncTask.execute();
    }
    private static class Task extends AsyncTask {
        private final PendingResult pendingResult;
        private final Intent intent;
        private Task(PendingResult pendingResult, Intent intent) {
            this.pendingResult = pendingResult;
            this.intent = intent;
        }
        @Override
        protected String doInBackground(String... strings) {
            StringBuilder sb = new StringBuilder();
            sb.append("Action: " + intent.getAction() + "\n");
            sb.append("URI: " + intent.toUri(Intent.URI_INTENT_SCHEME).toString() + "\n");
            String log = sb.toString();
            Log.d(TAG, log);
            return log;
        }
        @Override
        protected void onPostExecute(String s) {
            super.onPostExecute(s);
            // Must call finish() so the BroadcastReceiver can be recycled.
            pendingResult.finish();
        }
    }
}
```

An example of goAsync()

Sending Broadcasts

- sendOrderedBroadcast(Intent, String)
  - Send to one receiver at a time
- sendBroadcast(Intent)
- LocalBroadcastManager.sendBroadcast

Sending with permissions

- Restrict receiver app to have permission

```
sendBroadcast(new Intent("com.example.NOTIFY"), Manifest.Manifest.permission.SEND_SMS);
```

- Receiving APP must have the permission

```
<receiver android:name=".MyBroadcastReceiver"
    android:permission="android.permission.SEND_SMS">
    <intent-filter>
        <action android:name="android.intent.action.AIRPLANE_MODE"/>
    </intent-filter>
</receiver>
```

Send Custom Event

- Add intent filter in AndroidManifest.xml (Another app)

```
<action android:name="com.lab3.broadcast.CUSTOM_INTENT"/>
```

- Add sendCustomInent in MainActivity.java

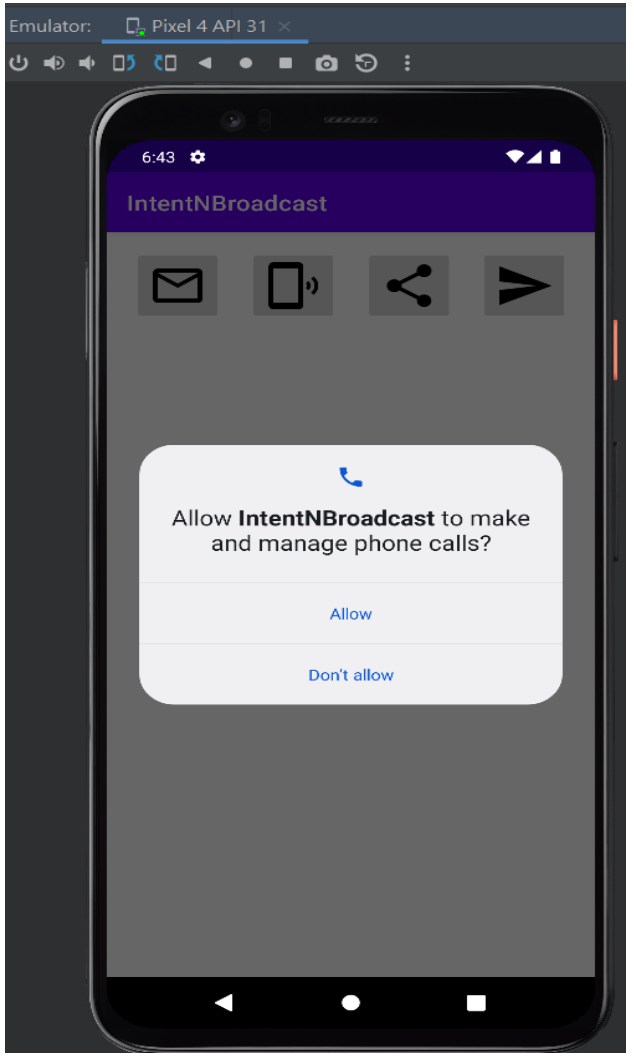
```
/*
 When click this button, it won't happen anything unless you register a BroadcastReciver() to listen for the
"com.lab3.broadcast.CUSTOM_INTENT"
 */
public void sendCustomIntent(View view)
{
    Intent intent = new Intent();
    intent.setAction("com.lab3.broadcast.CUSTOM_INTENT");
    intent.putExtra("message", "Custom Intent Test");
    intent.addFlags(Intent.FLAG_INCLUDE_STOPPED_PACKAGES);
    sendBroadcast(intent);
}
```

Note:

FLAG_INCLUDE_STOPPED_PACKAGES : can match apps that are currently stopped.
FLAG_EXCLUDE_STOPPED_PACKAGES : can’t match apps that are currently stopped.

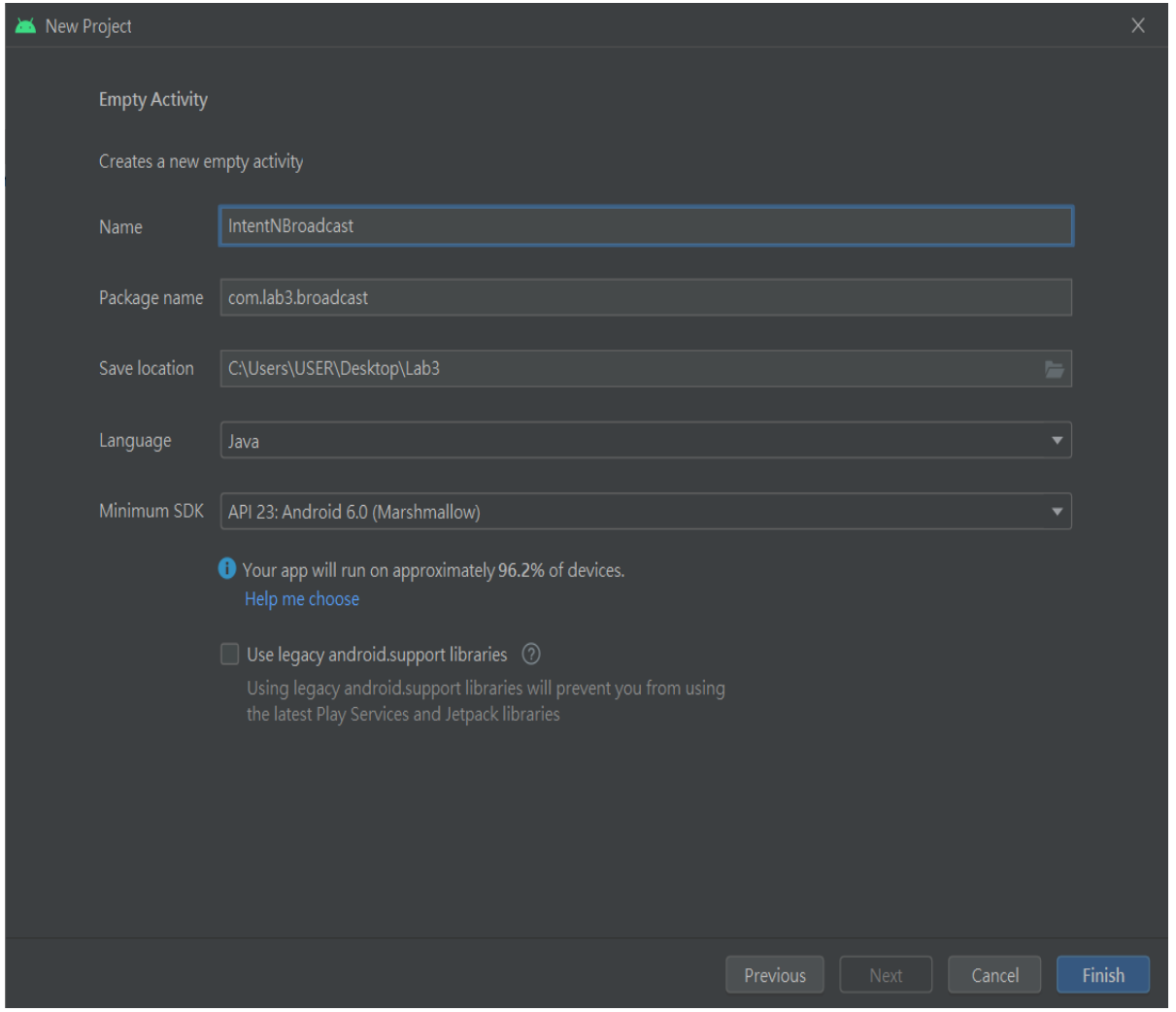
Today's Lab

- Receive system broadcast (Network Connectivity)
  - Show a toast when network connectivity changes, e.g. airplane mode
- Use Intent to …
  - Send an email
  - Make a call
  - Share an image
  - Send & receive your custom Intent



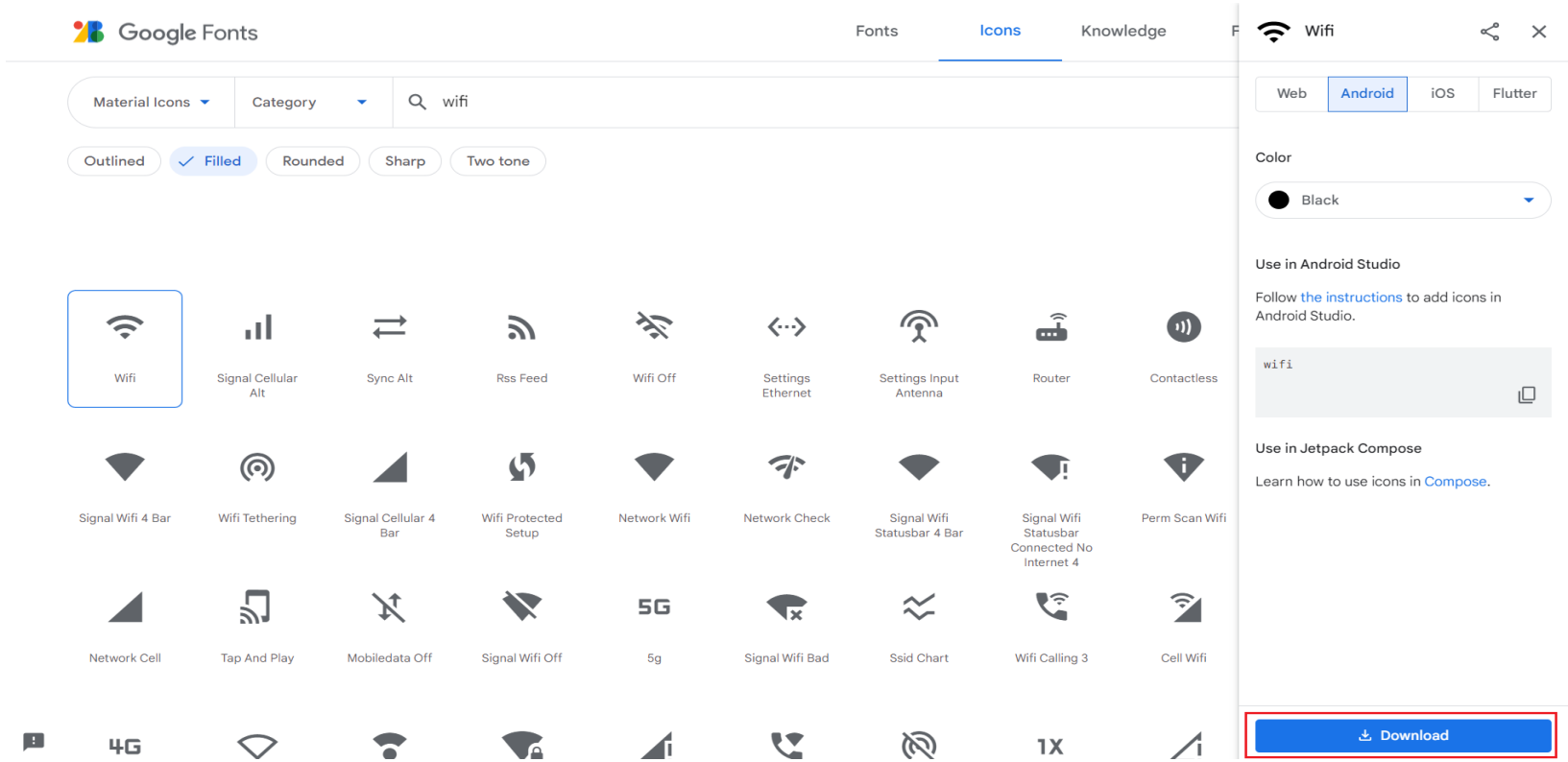
Create a New Project

- Create a new project names IntentNBroadcast
- Select **Empty Activity**
- Use default class name “MainActivity”
- Finish



Download icons

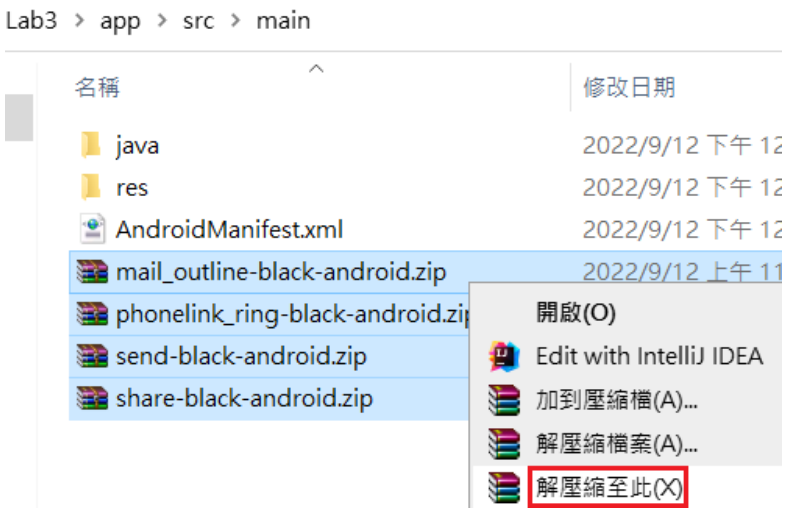
- https://fonts.google.com/icons?selected=Material+Icons
- Select your icon
- Select Android & Black at right side



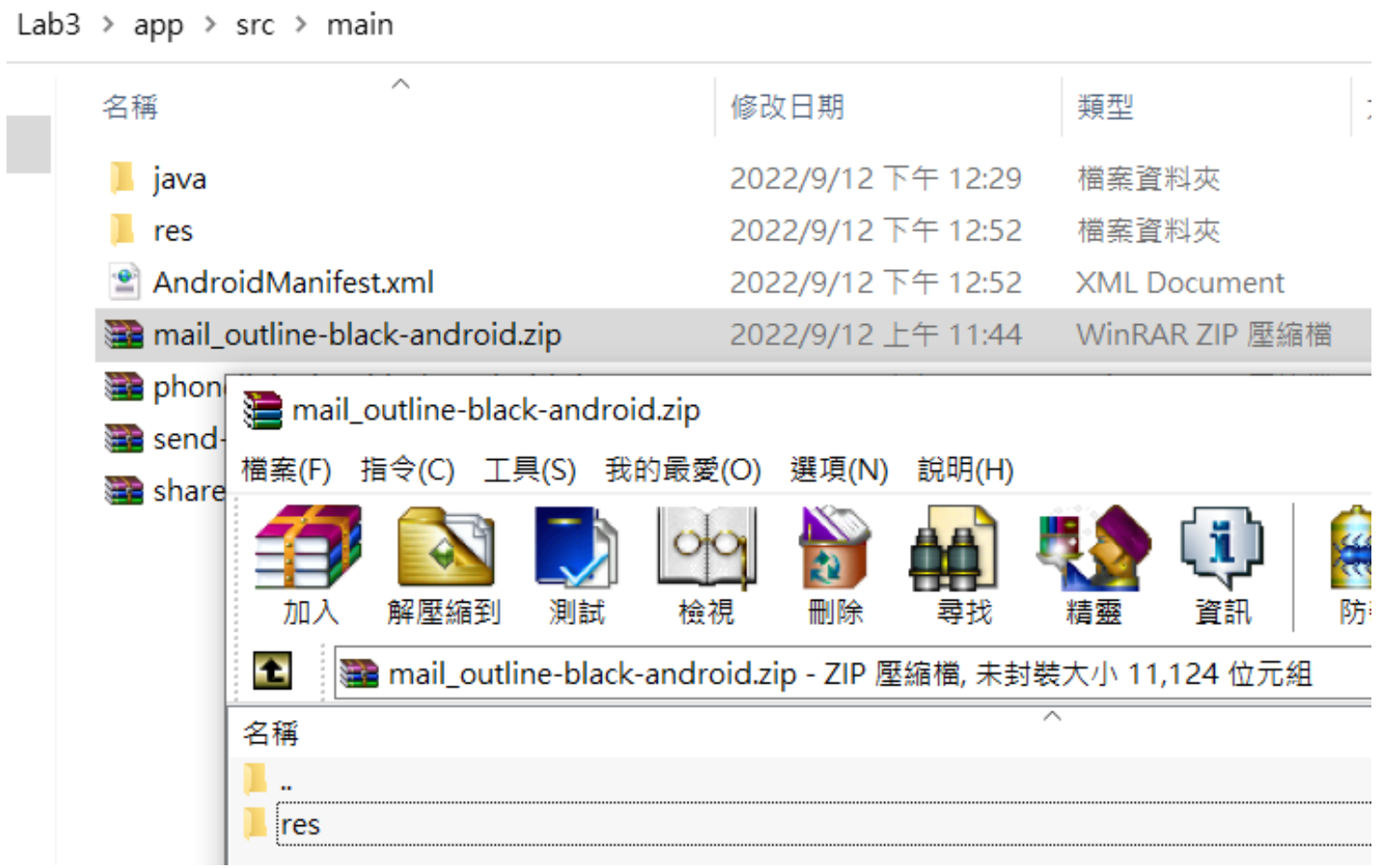
- mail_outline
- phonelink_ring
- send
- share

Add icons to projects

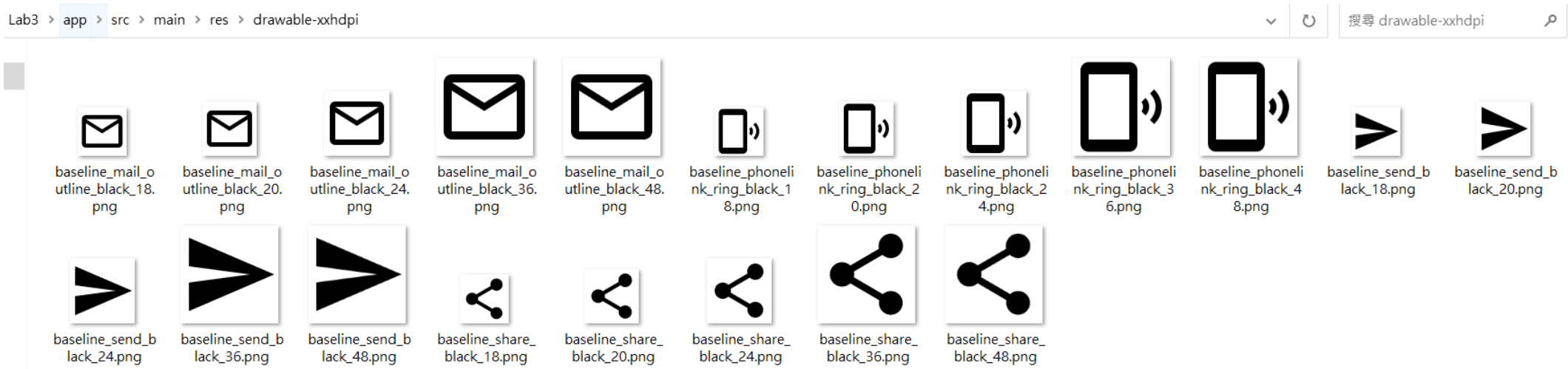
- Copy all icon zip files to “./app/src/main”
- Decompress all zip files
- Images with different sizes and resolutions will be put in corresponding folders
- File > Reload All from Disk



Data structure in image zip file

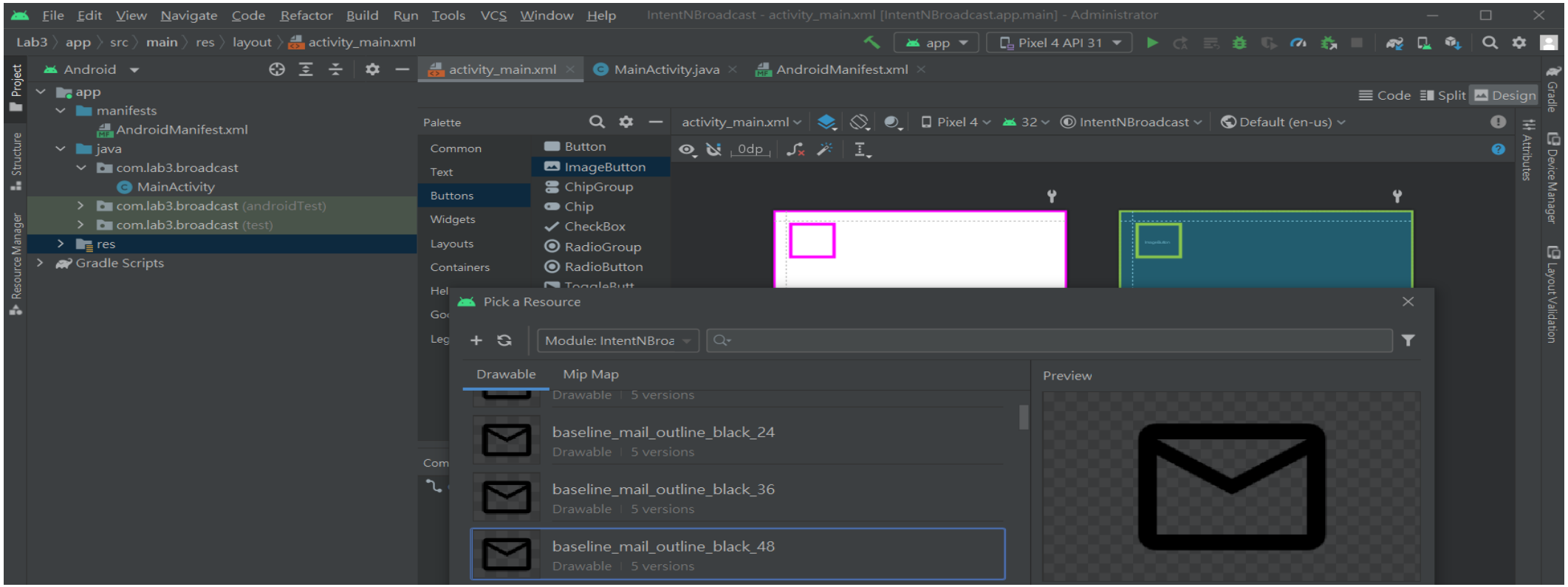


Images will be extracted to Folders Automatically!



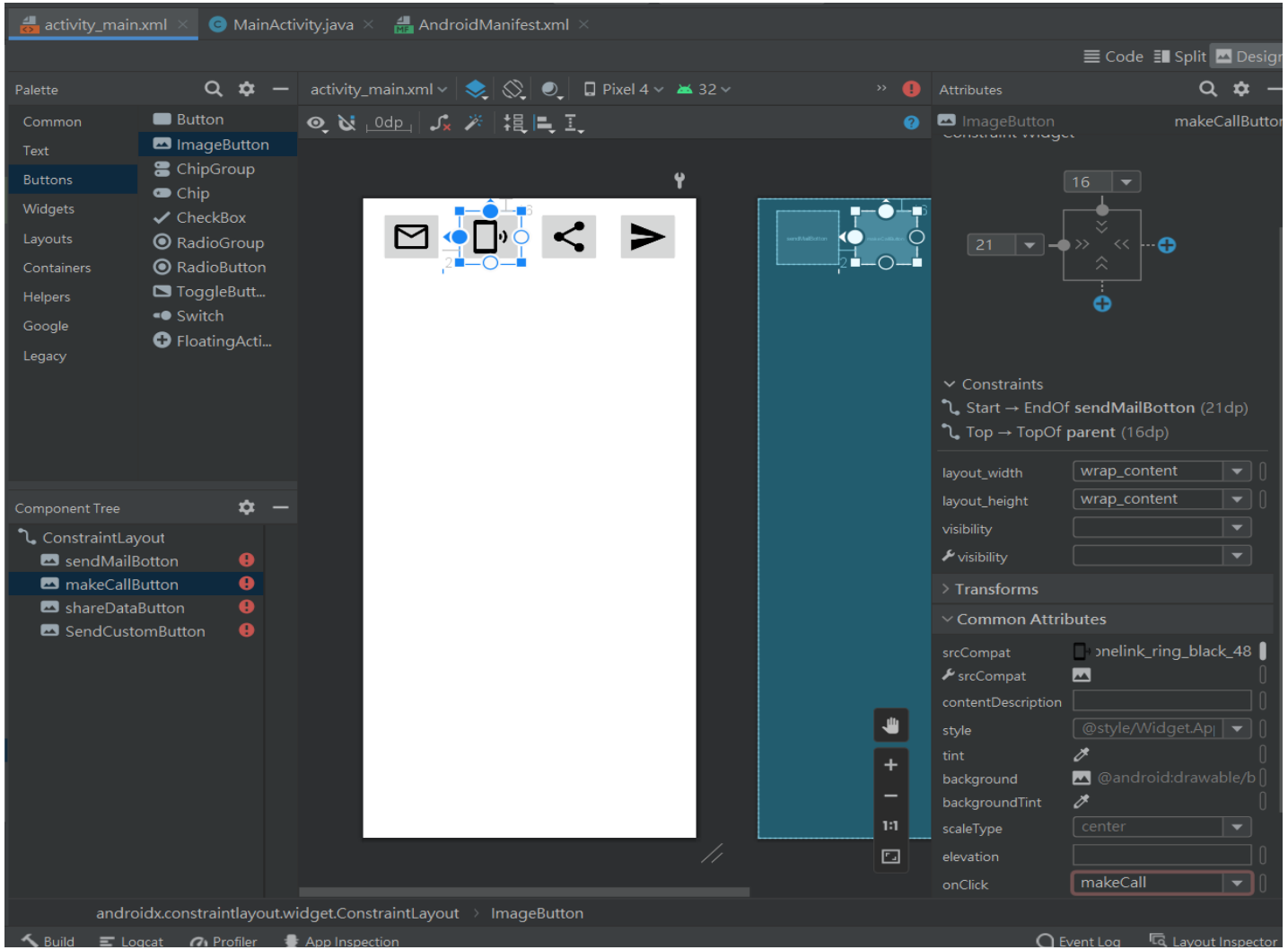
Create ImageButtons

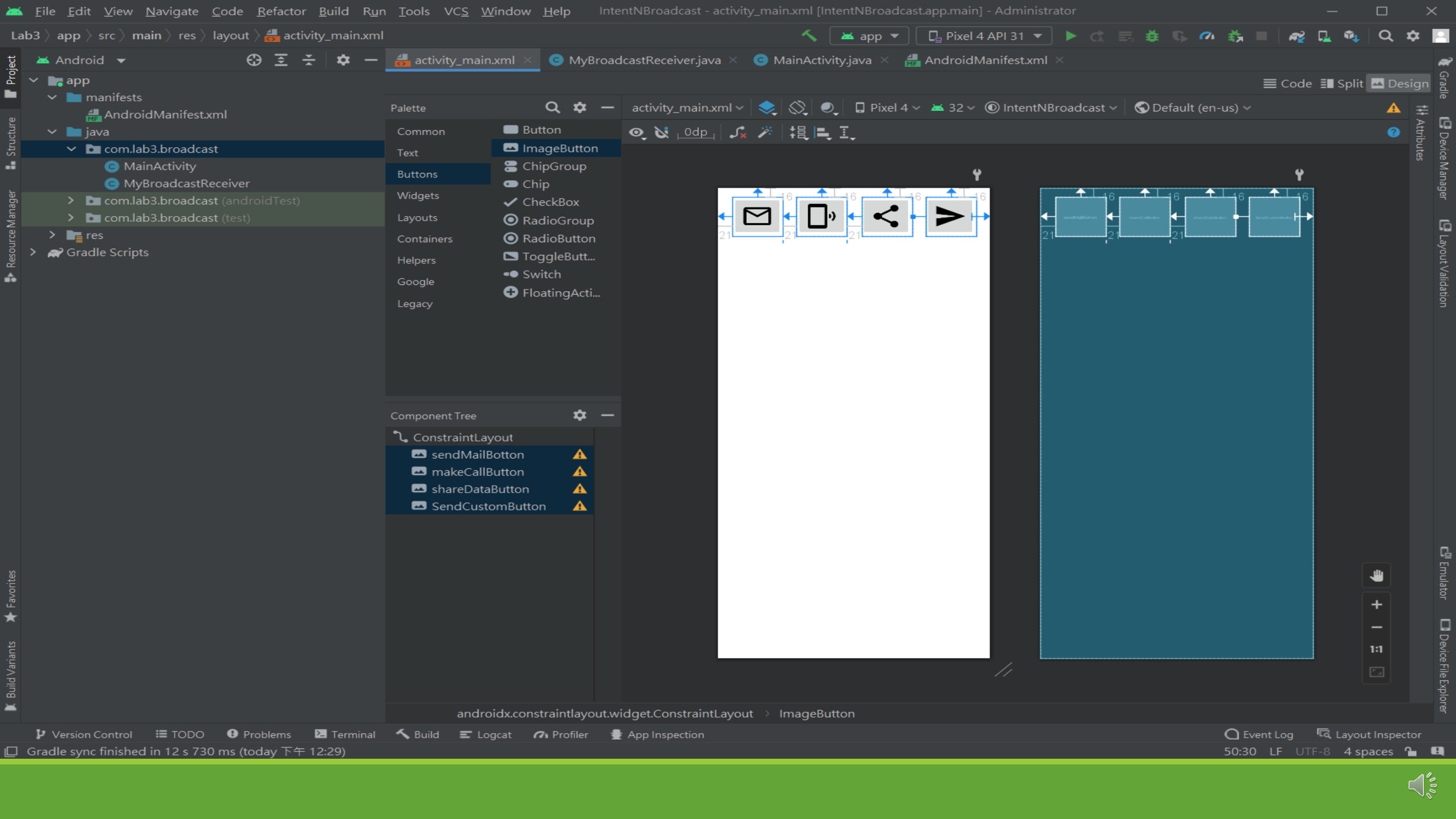
- Open “activity_main.xml”
- Drag ImageButton to the Blueprint
- Select baseline_mail_outline_black_48



Create ImageButtons

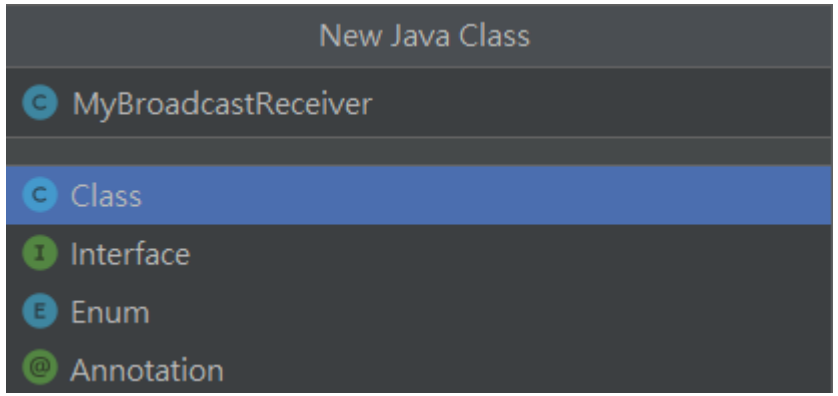
- Create 4 ImageButtons
  - sendMailButton
  - makeCallButton
  - shareImageButton
  - customBroadcastButton

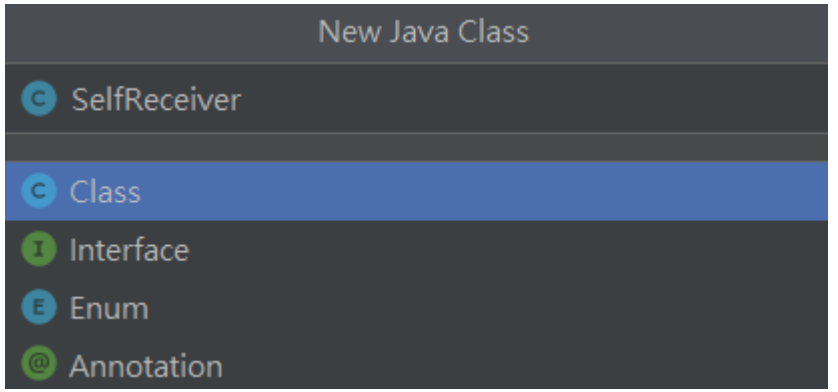



File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
IntentNBroadcast - activity_main.xml [IntentNBroadcast.app.main] - Administrator
Lab3 app src main res layout activity_main.xml
app
Pixel 4 API 31
Android
activity_main.xml
MyBroadcastReceiver.java
MainActivity.java
AndroidManifest.xml
Code Split Design
app
manifests
AndroidManifest.xml
java
com.lab3.broadcast
MainActivity
MyBroadcastReceiver
com.lab3.broadcast (androidTest)
com.lab3.broadcast (test)
res
Gradle Scripts
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
Button
ImageButton
ChipGroup
Chip
CheckBox
RadioGroup
RadioButton
ToggleButt...
Switch
FloatingActi...
activity_main.xml
Pixel 4
32
IntentNBroadcast
Default (en-us)
0dp
Component Tree
ConstraintLayout
sendMailBotton
makeCallButton
shareDataButton
SendCustomButton
androidx.constraintlayout.widget.ConstraintLayout
ImageButton
Version Control
TODO
Problems
Terminal
Build
Logcat
Profiler
App Inspection
Event Log
Layout Inspector
Gradle sync finished in 12 s 730 ms (today 下午 12:29)
50:30 LF UTF-8 4 spaces

Create BroadcastReceiver

- Create MyBroadcastReceiver.java
- Create SelfReceiver.java





Add the Code in Receiver

- MyBroadcastReceiver.java

```
public class MyBroadcastReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        StringBuilder sb = new StringBuilder();
        sb.append("Action: " + intent.getAction() + "\n");
        sb.append("URI: " + intent.toUri(Intent.URI_INTENT_SCHEME).toString() + "\n");
        String log = sb.toString();
        Toast.makeText(context, log, Toast.LENGTH_LONG).show();
    }
}
```

- SelfReceiver.java

```
public class SelfReceiver extends BroadcastReceiver {
    public void onReceive(Context context, Intent intent) {
        String msg = intent.getStringExtra("message");
        Toast.makeText(context, msg, Toast.LENGTH_SHORT).show();
    }
}
```

Add Receiver in AndroidManifest.xml

```
<application …>
    <activity
        android:name=".MainActivity"
        android:exported="true">
        …
    </activity>
    <receiver android:name=".MyBroadcastReceiver" android:enabled="true" android:exported="true">
        <intent-filter>
            <action android:name="android.net.wifi.WIFI_STATE_CHANGED"/>
            <action android:name="android.net.wifi.STATE_CHANGE"/>
        </intent-filter>
    </receiver>
    <receiver android:name=".SelfReceiver" android:enabled="true" android:exported="true" />
</application>
```

Register Receiver in MainActivity.java onCreate()

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    BroadcastReceiver br = new MyBroadcastReceiver();
    IntentFilter filter = new IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION);
    filter.addAction(Intent.ACTION_AIRPLANE_MODE_CHANGED);
    this.registerReceiver(br, filter);
}
```

Add function *SendMailMessage()* in MainActivity.java

```
public void sendMailMessage(View view)
{
    String mailto = "mailto:bob@example.org" +
            "?cc=" + "alice@example.com" +
            "&subject=" + Uri.encode("No Subject") +
            "&body=" + Uri.encode("");
    Intent emailIntent = new Intent(Intent.ACTION_SENDTO);
    emailIntent.setData(Uri.parse(mailto));
    try {
        startActivity(emailIntent);
    } catch (ActivityNotFoundException e) {
        //TODO: Handle case where no email app is available
    }
}
```

Add makeCall(View view)

```
public void makeCall(View view) {
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED)
    {
        ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.CALL_PHONE}, 1);
    }
    else
    {
        String number = "23454568678";
        Intent intent = new Intent(Intent.ACTION_CALL);
       // setData傳入的是Uri，用於數據的過濾，setData可以被用來尋找目標組件
        intent.setData(Uri.parse("tel:" + number));
        startActivity(intent);
    }
}
```

Add Permission

- Add permission in AndroidManifest.xml

```
<uses-permission android:name="android.permission.CALL_PHONE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<application …>
...
</application>
```

Add ShareData()

- Add shareData(View view) in MainActivity.java

```
public void shareData(View view)
{
    // https://stackoverflow.com/questions/20333186/how-to-share-image-texttogether-using-action-send-in-android
    Uri imageUri = Uri.parse("android.resource://" + getPackageName() + "/drawable/" + "ic_launcher");
    Intent shareIntent = new Intent();
    shareIntent.setAction(Intent.ACTION_SEND);
    shareIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    shareIntent.putExtra(Intent.EXTRA_TEXT, "Hello World");
    shareIntent.putExtra(Intent.EXTRA_STREAM, imageUri);
    shareIntent.setType("image/png");
    startActivity(Intent.createChooser(shareIntent, "Share with"));
}
```

Add sendSelfBroadcast()

- Add sendSelfBroadcast(View view) in MainActivity.java

```
public void sendSelfBroadcast(View view)
{
    Intent intent = new Intent(MainActivity.this, SelfReceiver.class);
    intent.putExtra("message", "Send Self Intent Test");
    sendBroadcast(intent);
}
```

Set *onClick()*

- Select sendMailBotton, set *onClick* as “sendMailMessage”
- Select makeCallButton, set *onClick* as “makeCall”
- Select shareDataButton, set *onClick* as “shareData”
- Select SendCustomButton, set *onClick* as “sendSelfBroadcast”

Final Result

- Click any ImageButton
- The App match the intent will pop up