zaini1983@gmail.com

ANDROID BASIC PART (RESOURCE-ACTIVITY-BROADCASTRECEIVER)

Resource

- Non java file
- It Consist of various thing that our code use.
- Located in res directory

Resource

- drawable/
  - Image files like .png, .jpg, .gif or XML files that are compiled into bitmaps, state lists, shapes, animation drawable. They are saved in res/drawable/ and accessed from the **R.drawable** class.
- layout/
  - XML files that define a user interface layout. They are saved in res/layout/ and accessed from the **R.layout** class.
- menu/
  - XML files that define application menus, such as an Options Menu, Context Menu, or Sub Menu. They are saved in res/menu/ and accessed from the **R.menu** class.
- values/
  - XML files that contain simple values, such as strings, integers, and colors.

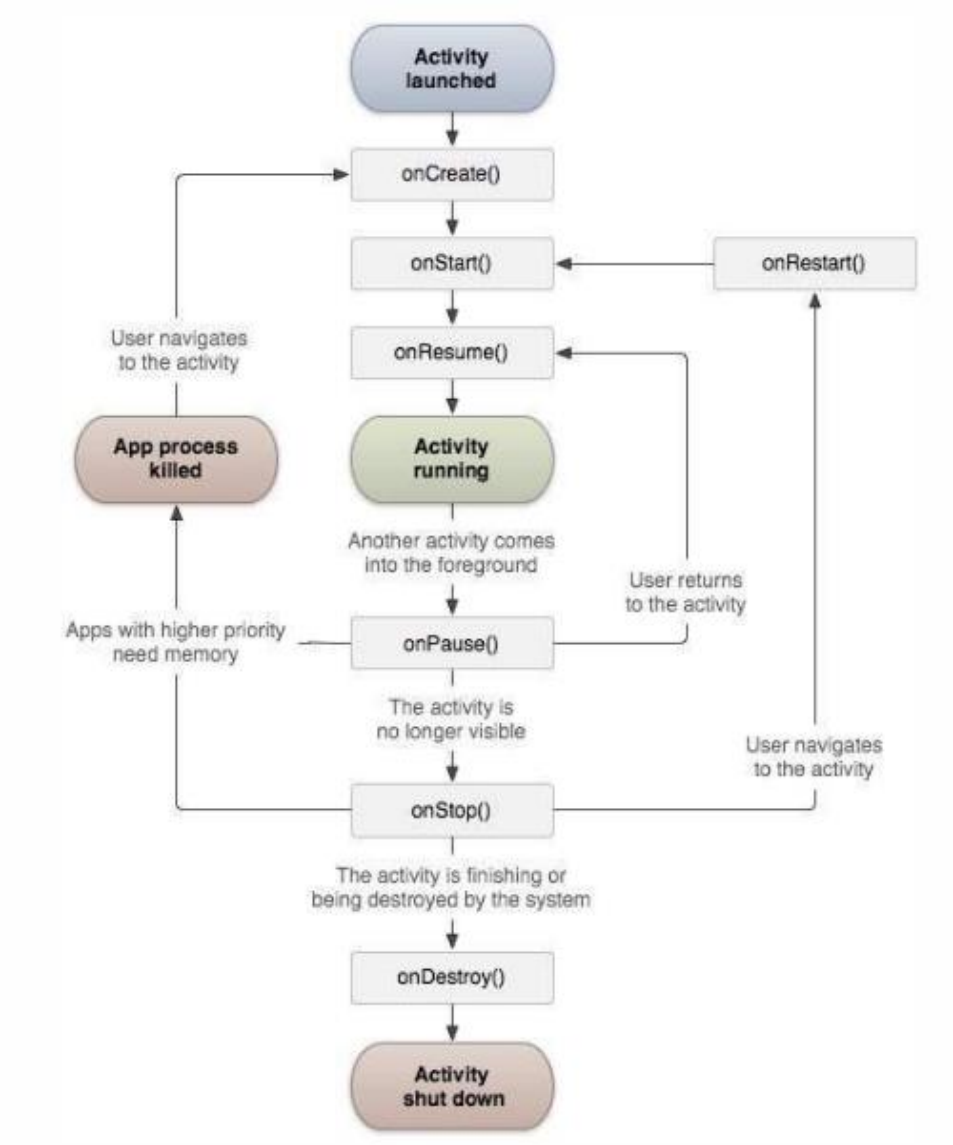
Accessing Resource

- From Code : It can be accessed from R class
- From XML : @[folder]/[resource_id]

Activity

- When we use C, C++ or java ,we start our program from main function.
- Activity is similar to main function.
- An activity class loads all the UI component using the XML file available in *res/layout* folder of the project.
  - setContentView(R.layout.activity_main);

Activity Live Cycle



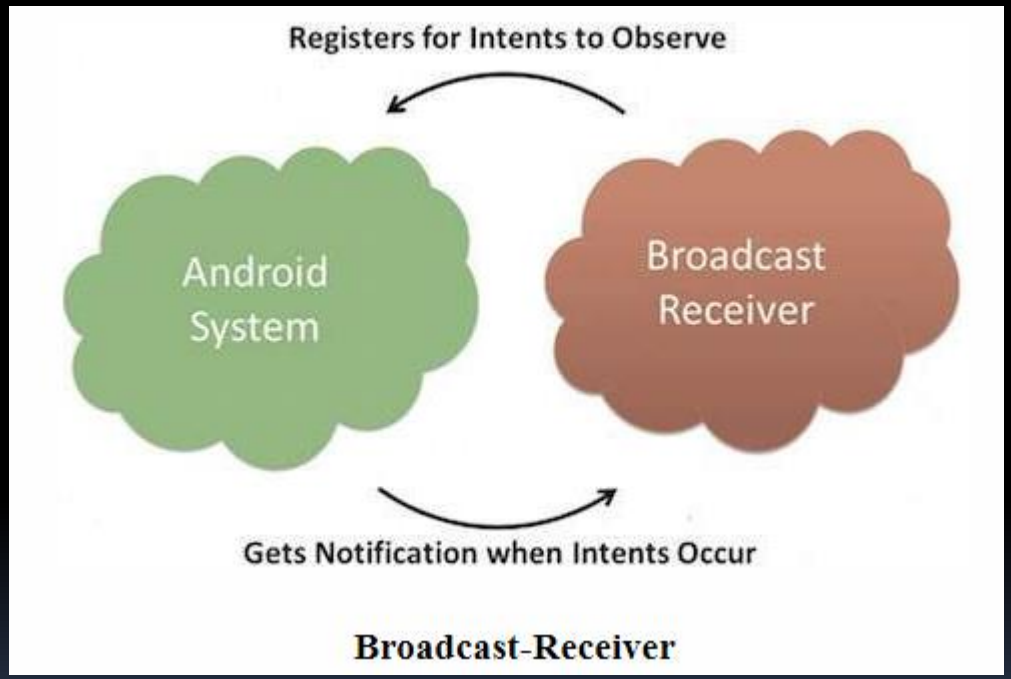
Broadcast Receiver

- Sometimes we need our application to intercept external/internal message when particular events happened. For ex: boot, battery low, or may be custom events

BR’s Important Step

- Creating BR
- Registering BR
- Broadcast (optional, in case when we are going to implement custom BR)

BR’s Illustration



Creating BR

- By implementing BroadcastReceiver Class and overriding onReceive method

```
public class MyReceiver extends BroadcastReceiver {
   @Override
   public void onReceive(Context context, Intent intent) {
      Toast.makeText(context, "Intent Detected.", Toast.LENGTH_LONG).show();
   }
}
```

Registering BR

- An application listens for specific broadcast intents by registering a broadcast receiver inside receiver tag on *AndroidManifest.xml* file.

Ex. Registering BR to Intercept BOOT_COMPLETED intent

- It can be done through intent-filter tag inside *AndroidManifest.xml* file

```
<application
   android:icon="@drawable/ic_launcher"
   android:label="@string/app_name"
   android:theme="@style/AppTheme" >
   <receiver android:name="MyReceiver">

      <intent-filter>
         <action android:name="android.intent.action.BOOT_COMPLETED">
         </action>
      </intent-filter>

   </receiver>
</application>
```

Broadcasting Custom intent

- It can be done by using sendBroadcast method inside activity

```
public void broadcastIntent(View view) {
    Intent intent = new Intent();
    intent.setAction("com.tutorialspoint.CUSTOM_INTENT");
    sendBroadcast(intent);
}
```

Assignment

- Create an apllication that can show notification when device rebooted
- Create an application that can detect another apps events