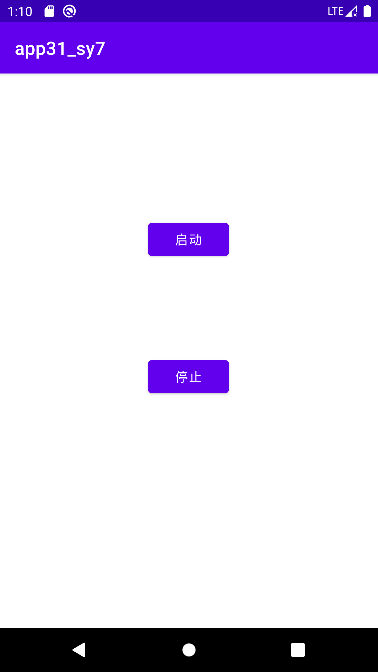
实验八：服务与广播

设计项目A和项目B，在A中设计“启动”和“停止”两个按钮，点击“启动”时启动B项目中的计数服务(非绑定，在日志中输出计数过程)，点击“停止”时停止计数过程并结束B项目中的服务。

public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 }  
  
 public void start(View view) {  
 Intent intent = new Intent();  
 intent.setAction("hello");  
 sendBroadcast(intent);  
  
 }  
  
 public void stop(View view) {  
 Intent intent2 = new Intent();  
 intent2.setAction("hello2");  
 sendBroadcast(intent2);  
 }  
}



*<?*xml version="1.0" encoding="utf-8"*?>*<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.shf.app32\_sy7">  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.App32\_sy7">  
 <receiver  
 android:name=".MyStopReceiver"  
 android:enabled="true"  
 android:exported="true">  
 </receiver>  
  
 <service  
 android:name=".MyService"  
 android:enabled="true"  
 android:exported="true">  
 </service>  
  
 <receiver  
 android:name=".MyReceiver"  
 android:enabled="true"  
 android:exported="true">  
 </receiver>  
  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN"/>  
  
 <category android:name="android.intent.category.LAUNCHER"/>  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

package com.shf.app32\_sy7;  
  
import android.content.Intent;  
import android.content.IntentFilter;  
import androidx.appcompat.app.AppCompatActivity;  
import android.os.Bundle;  
  
public class MainActivity extends AppCompatActivity {  
 private MyReceiver receiver;  
 private MyStopReceiver stopReceiver;  
  
 public static Intent *intent1*;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 *intent1* = new Intent(this, MyService.class);  
  
*// 实例化广播接受者* receiver = new MyReceiver();  
*// 实例化意图过滤器* String action = "hello";  
 IntentFilter filter = new IntentFilter(action);  
*// 动态注册广播接受者* registerReceiver(receiver,filter);  
  
*// 实例化广播接受者* stopReceiver = new MyStopReceiver();  
*// 实例化意图过滤器* String action2 = "hello2";  
 IntentFilter filter2 = new IntentFilter(action2);  
*// 动态注册广播接受者* registerReceiver(stopReceiver,filter2);  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
*// 取消注册广播接受者* unregisterReceiver(receiver);  
 unregisterReceiver(stopReceiver);  
 }  
}

public class MyReceiver extends BroadcastReceiver {  
  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 *// TODO: This method is called when the BroadcastReceiver is receiving  
 // an Intent broadcast.  
// throw new UnsupportedOperationException("Not yet implemented");* context.startService(MainActivity.*intent1*);  
 }  
}

public class MyStopReceiver extends BroadcastReceiver {  
  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 *// TODO: This method is called when the BroadcastReceiver is receiving  
 // an Intent broadcast.  
// throw new UnsupportedOperationException("Not yet implemented");* context.stopService(MainActivity.*intent1*);  
 }  
}

public class MyService extends Service {  
  
 boolean running = true;  
 int i = 0;  
  
 public MyService() {  
 }  
  
 @Override  
 public IBinder onBind(Intent intent) {  
 *// TODO: Return the communication channel to the service.* throw new UnsupportedOperationException("Not yet implemented");  
 }  
  
 @Override  
 public void onCreate() {  
 super.onCreate();  
  
 new Thread(new Runnable() {  
  
 @Override  
 public void run() {  
 while (i < 200 && running) {  
 i++;  
 Log.i("i的值为：", String.valueOf(i));  
 try {  
 Thread.sleep(1000);  
 } catch (InterruptedException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 }  
 }).start();  
 }  
  
 @Override  
 public void onDestroy() {  
 running = false;  
  
 Log.*i*("程序停止", "程序停止onDestroy");  
  
 super.onDestroy();  
 }  
}

