

22F-3853

Codes:

Boot complete file

```
// for static registration

package com.example.april12;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;

public class BootCompleteReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {

if(Intent.ACTION_BOOT_COMPLETED.equals(intent.getAction()))
    {
        Intent serviceIntent=new Intent(context,
BackgroundService.class);
        context.startService(serviceIntent);
    }
    }
}
```

Background service .java

```
// for static registration
package com.example.april12;

import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.widget.Toast;

import androidx.annotation.Nullable;

public class BackgroundService extends Service {
    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int
```

```

startId) {
    Toast.makeText(this, "Service started on
boot!", Toast.LENGTH_SHORT).show();
    return START_STICKY;
}
}

```

Main activity .java

```

package com.example.april12;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.provider.Settings;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    private BroadcastReceiver airplaneModeReceiver;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);

        airplaneModeReceiver=new BroadcastReceiver() {

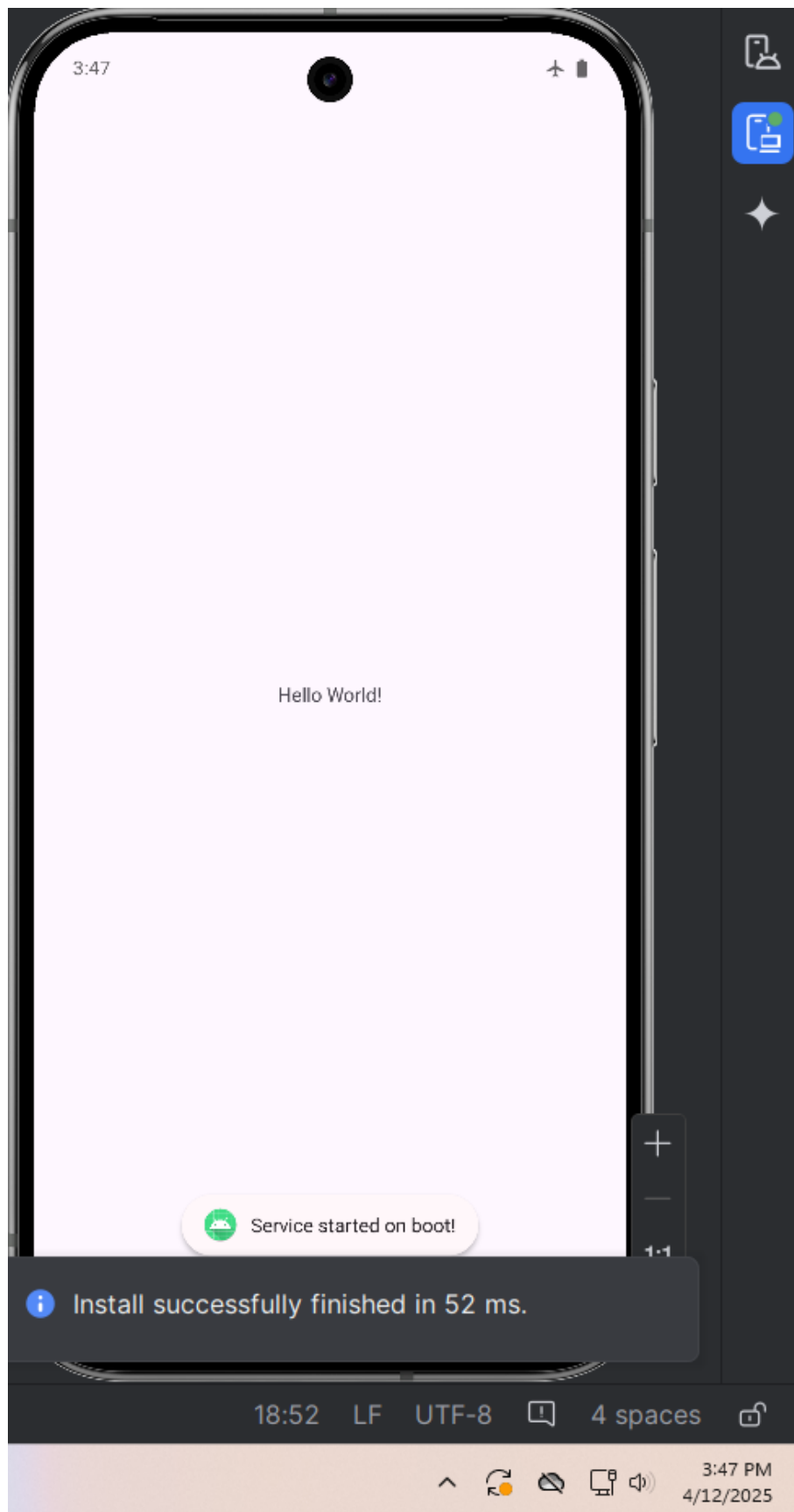
            @Override
            public void onReceive(Context context, Intent
intent) {

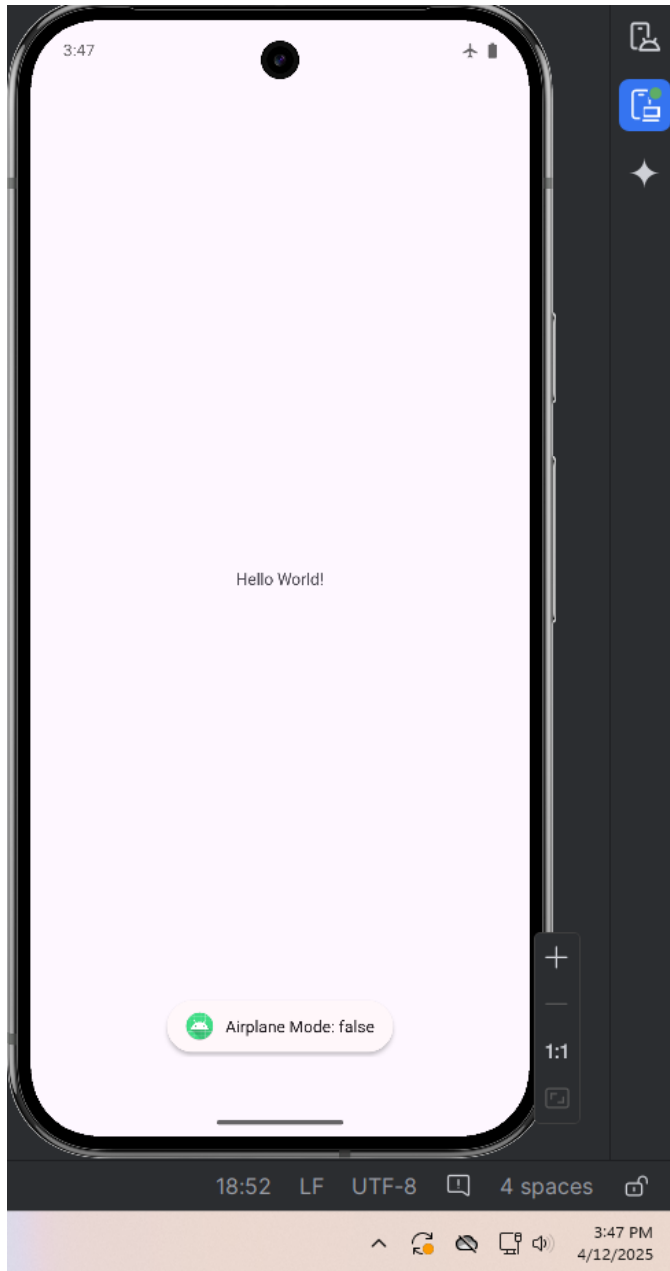
                boolean isAirplaneMode= Settings.System.getInt(
context.getContentResolver(), Settings.Global.AIRPLANE_MODE_ON, 0
) !=0;
                Toast.makeText(context, "Airplane Mode:
"+isAirplaneMode, Toast.LENGTH_SHORT).show();
            }
        };
    }
}

```

```
        IntentFilter filter =new  
IntentFilter(Intent.ACTION_AIRPLANE_MODE_CHANGED);  
        registerReceiver(airplaneModeReceiver,filter);  
    }  
}
```

22F-3853





```

package com.example.april12;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.telephony.PhoneStateListener;
import android.telephony.TelephonyManager;
import android.widget.Toast;

public class RecieverCall extends BroadcastReceiver {

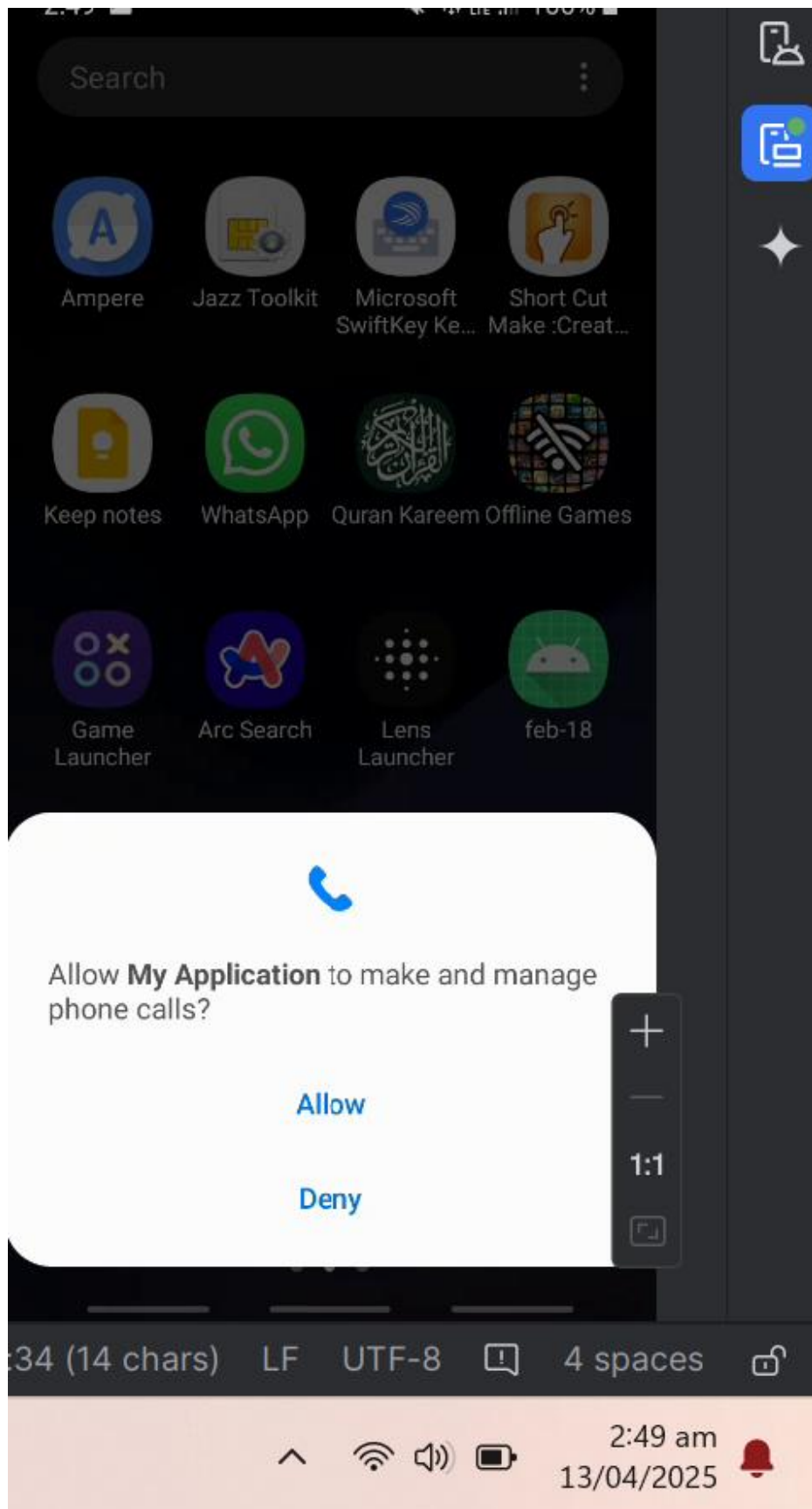
    Context context;
    @Override
    public void onReceive(Context context, Intent intent) {
        this.context=context;
        Toast.makeText(context,"call
recieved",Toast.LENGTH_SHORT).show();

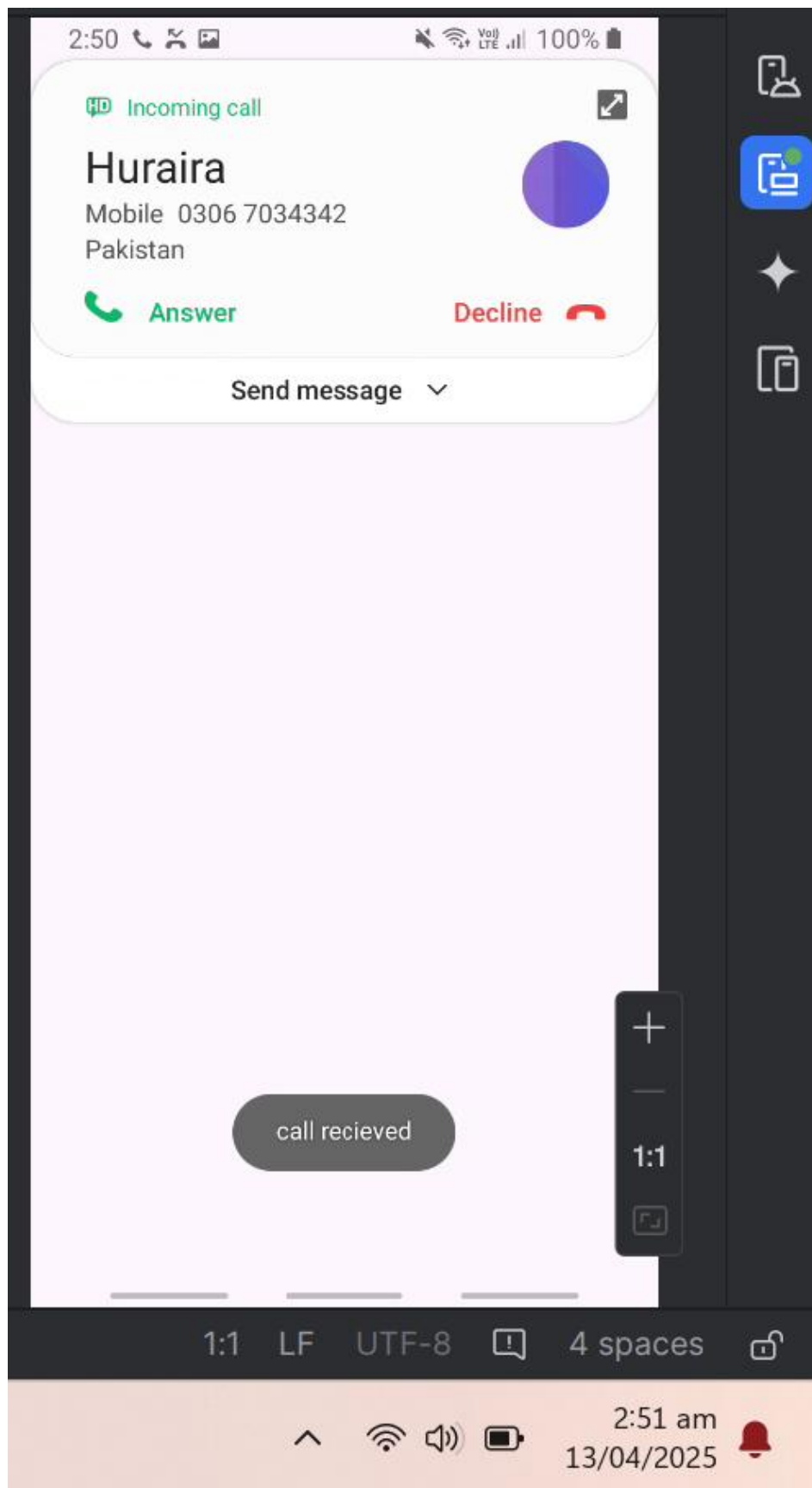
        TelephonyManager telephonyManager=(TelephonyManager)
context.getSystemService(Context.TELEPHONY_SERVICE);

    }

    private class MyPhoneStateListener extends
PhoneStateListener{
        @Override
        public void onCallStateChanged(int state, String
phoneNumber) {
            if(state==1)
            {
                String msg="New Phone call Event.Incoming
Number: "+phoneNumber;
                int duration=Toast.LENGTH_LONG;
                Toast
toast=Toast.makeText(context,msg,duration);
                toast.show();
            }
        }
    }
}

```





Home task

```
package com.example.myapplication;
```

```
import android.Manifest;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.pm.PackageManager;
import android.os.BatteryManager;
import android.os.Bundle;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
```

```
public class MainActivity2 extends AppCompatActivity {
```

```
    private ReceiverCall receiverCall;
    private BatteryStateReceiver batteryStateReceiver;
    private static final int REQUEST_PHONE_STATE_PERMISSION = 1;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main2);
```

```
        receiverCall = new ReceiverCall();
```

```
        // Initialize the BatteryStateReceiver instance
        batteryStateReceiver = new BatteryStateReceiver();
```

```
        checkAndRequestPermission();
```

```
    }
```

```
    //Method to check and request READ_PHONE_STATE permission
```

```

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.READ_PHONE_STATE)
        != PackageManager.PERMISSION_GRANTED) {
        // Permission is not granted, request it
        ActivityCompat.requestPermissions(
            this,
            new String[]{Manifest.permission.READ_PHONE_STATE},
            REQUEST_PHONE_STATE_PERMISSION
        );
    } else {

        registerPhoneStateReceiver();
    }
}

// Method to register the ReceiverCall for PHONE_STATE changes

private void registerPhoneStateReceiver() {
    IntentFilter filter = new IntentFilter("android.intent.action.PHONE_STATE");
    registerReceiver(receiverCall, filter);
}

@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);

    if (requestCode == REQUEST_PHONE_STATE_PERMISSION) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {

            registerPhoneStateReceiver();
        } else {

            System.out.println("READ_PHONE_STATE permission denied.");
        }
    }
}

@Override
protected void onStart() {
    super.onStart();

    if (ContextCompat.checkSelfPermission(this, Manifest.permission.READ_PHONE_STATE)

```

```

        == PackageManager.PERMISSION_GRANTED) {
            registerPhoneStateReceiver();
        }

        // Register the battery state receiver
        IntentFilter batteryFilter = new IntentFilter();
        batteryFilter.addAction(Intent.ACTION_BATTERY_LOW);
        batteryFilter.addAction(Intent.ACTION_BATTERY_OKAY);
        batteryFilter.addAction(Intent.ACTION_BATTERY_CHANGED);
        registerReceiver(batteryStateReceiver, batteryFilter);
    }

    @Override
    protected void onPause() {
        super.onPause();

        // Unregister the phone state receiver to avoid memory leaks
        unregisterReceiver(receiverCall);

        // Unregister the battery state receiver
        unregisterReceiver(batteryStateReceiver);
    }

    // Inner class to handle battery state changes

    private class BatteryStateReceiver extends BroadcastReceiver {

        @Override
        public void onReceive(Context context, Intent intent) {
            String action = intent.getAction();

            if (action != null) {
                switch (action) {
                    case Intent.ACTION_BATTERY_LOW:
                        // Show a Toast when the battery is critically low
                        Toast.makeText(context, "Battery is critically low!", Toast.LENGTH_LONG).show();
                        break;

                    case Intent.ACTION_BATTERY_OKAY:
                        // Show a Toast when the battery recovers from being low
                        Toast.makeText(context, "Battery is okay now.", Toast.LENGTH_LONG).show();
                        break;
                }
            }
        }
    }

```

```

case Intent.ACTION_BATTERY_CHANGED:
    // Extract battery level and charging status
    int level = intent.getIntExtra(BatteryManager.EXTRA_LEVEL, -1);
    int scale = intent.getIntExtra(BatteryManager.EXTRA_SCALE, -1);
    float batteryPct = level / (float) scale * 100;

    boolean isCharging = intent.getIntExtra(BatteryManager.EXTRA_STATUS, -1)
        == BatteryManager.BATTERY_STATUS_CHARGING;

    // Show a Toast with the current battery percentage and charging status
    String chargingStatus = isCharging ? " (Charging)" : " (Not Charging)";
    Toast.makeText(context, "Battery: " + (int) batteryPct + "%" + chargingStatus,
        Toast.LENGTH_SHORT).show();
    break;
}
}
}
}
}

```

