

Step-by-Step Guide for Lifecycle Logging in Android

Step 1: Create a New Android Studio Project

1. Open Android Studio.
2. Click on "Start a new Android Studio project".
3. Choose "Empty Activity" and click "Next".
4. Name your application (e.g., LifecycleLoggerApp), select the save location, and choose the language as Java.
5. Click "Finish" to create the project.

Step 2: Create a New Activity

1. Right-click on `app/src/main/java/your_package_name` and select `New > Activity > Empty Activity`.
2. Name your new activity (e.g., `SecondActivity`) and click "Finish".

Step 3: Override Lifecycle Methods

1. Open `MainActivity.java` and override the lifecycle methods to log transitions.
2. Repeat the same for `SecondActivity.java`.

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

import android.os.Bundle;
import android.util.Log;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private static final String TAG = "MainActivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(TAG, "onCreate called");
    }

    @Override
    protected void onStart() {
        super.onStart();
        Log.d(TAG, "onStart called");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.d(TAG, "onResume called");
    }
}
```

```
    }

    @Override
    protected void onPause() {
        super.onPause();
        Log.d(TAG, "onPause called");
    }

    @Override
    protected void onStop() {
        super.onStop();
        Log.d(TAG, "onStop called");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d(TAG, "onDestroy called");
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d(TAG, "onRestart called");
    }
}
```

Similarly, for SecondActivity.java:

```
package com.example.myapplication;
import android.os.Bundle;
import android.util.Log;
import androidx.appcompat.app.AppCompatActivity;

public class SecondActivity extends AppCompatActivity {

    private static final String TAG = "SecondActivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        Log.d(TAG, "onCreate called");
    }

    @Override
    protected void onStart() {
        super.onStart();
        Log.d(TAG, "onStart called");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.d(TAG, "onResume called");
    }

    @Override
    protected void onPause() {
        super.onPause();
        Log.d(TAG, "onPause called");
    }

    @Override
    protected void onStop() {
        super.onStop();
        Log.d(TAG, "onStop called");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d(TAG, "onDestroy called");
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d(TAG, "onRestart called");
    }
}
```

Step 4: Create Layouts for the Activities

1. Open `res/layout/activity_main.xml` and add a Button to navigate to `SecondActivity`.

```
2. <?xml version="1.0" encoding="utf-8"?>
  <LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp">

    <Button
      android:id="@+id/buttonNavigate"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Go to Second Activity"/>

  </LinearLayout>
```

Open `res/layout/activity_second.xml` and add a `TextView`.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:gravity="center"
  android:padding="16dp">

  <TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="This is the Second Activity"
    android:textSize="18sp"/>

</LinearLayout>
```

Step 5: Add Navigation Logic

1. Open `MainActivity.java` and add the logic to navigate to `SecondActivity` when the Button is clicked.

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

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
```

```

private static final String TAG = "MainActivity";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Log.d(TAG, "onCreate called");

    Button buttonNavigate = findViewById(R.id.buttonNavigate);
    buttonNavigate.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this,
SecondActivity.class);
            startActivity(intent);
        }
    });
}

@Override
protected void onStart() {
    super.onStart();
    Log.d(TAG, "onStart called");
}

@Override
protected void onResume() {
    super.onResume();
    Log.d(TAG, "onResume called");
}

@Override
protected void onPause() {
    super.onPause();
    Log.d(TAG, "onPause called");
}

@Override
protected void onStop() {
    super.onStop();
    Log.d(TAG, "onStop called");
}

@Override
protected void onDestroy() {
    super.onDestroy();
    Log.d(TAG, "onDestroy called");
}

@Override
protected void onRestart() {
    super.onRestart();
    Log.d(TAG, "onRestart called");
}
}

```

Step 6: Run the Application and Observe Logs

1. Connect your Android device or start an emulator.
2. Click on the "Run" button in Android Studio or select "Run" > "Run 'app'" from the menu.
3. Open the Logcat window in Android Studio (View > Tool Windows > Logcat).
4. Observe the log messages for lifecycle method calls.
5. Interact with the application (e.g., rotate the device, press the home button, navigate to `SecondActivity`, and back) and observe how the lifecycle methods are logged.