

[2CEIT5PE5: MOBILE APPLICATION DEVELOPMENT]

Practical: 9

AIM: What is Difference between ListView and RecyclerView? Create SMS Android Application that can receive SMS by BroadcastReceiver and Interface, Send SMS and read & display all SMS of Inbox by using RecyclerView.

Submitted By: Jesika Pargi
Enrollment number: 21012022030



**Ganpat
University**

॥ विद्यया समाजोत्कर्षः ॥

**U.V. Patel
College of
Engineering**

**Department of Computer
Engineering/Information Technolog**

RecyclerView	ListView
<ul style="list-style-type: none"> The RecyclerView's adaptor forces us to use the ViewHolder pattern. 	<ul style="list-style-type: none"> The ListView doesn't give that kind of protection by default.
<ul style="list-style-type: none"> Supported Vertical Scrolling and Horizontal Scrolling 	<ul style="list-style-type: none"> Supported Only Vertical Scrolling
<ul style="list-style-type: none"> Use of less memory. 	<ul style="list-style-type: none"> More memory is used for a long list. Sometimes devices get hanged.
<ul style="list-style-type: none"> Animations using ItemAnimator are easy and smooth. Animations like list appearance and disappearance, adding or removing particular views, and so on. 	<ul style="list-style-type: none"> It's complex to use Animation and hard to handle it.
<ul style="list-style-type: none"> Dividers between items are not shown by default. 	<ul style="list-style-type: none"> Dividers between items are shown by default.
<ul style="list-style-type: none"> Use ItemDecorations to add margins and draw on or under an item View. 	<ul style="list-style-type: none"> ItemDecorations require customization
<ul style="list-style-type: none"> RecyclerView is a more flexible control for handling a "list of data" 	<ul style="list-style-type: none"> ListView is a less flexible control for handling a "list of data"

sms_item_view.xml:

```

<?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="horizontal"
    android:layout_marginHorizontal="10dp"
    android:padding="20dp">

    <ImageView
        android:id="@+id/icon"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:padding="5dp"
        android:background="@drawable/shape"
        android:src="@drawable/ic_baseline_person_24"
        />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textViewPhoneNo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Medium Text"
            android:textSize="14dp"
            android:textStyle="bold"

            android:textAppearance="?android:attr/textAppearanceMedium"
            android:layout_marginLeft="10dp"
            android:layout_marginTop="5dp"
            android:padding="2dp"
            android:textColor="#4d4d4d" />

        <TextView
            android:id="@+id/textViewMessage"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="TextView"
            android:layout_marginLeft="10dp"
            android:textSize="12dp"/>
    </LinearLayout>
</LinearLayout>

```

activity_main.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/app_bar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent">
        <androidx.appcompat.widget.Toolbar
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            app:title="Message Application"
            app:titleTextColor="@color/white"/>
        </com.google.android.material.appbar.AppBarLayout>
        <com.google.android.material.card.MaterialCardView
            android:id="@+id/card"
            android:layout_width="0dp"
            android:layout_height="0dp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintBottom_toTopOf="@id/l1"
            app:layout_constraintTop_toBottomOf="@id/app_bar"
            android:layout_margin="10dp"
        >
            <ListView
                android:id="@+id/listView"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"

                android:backgroundTint="@color/cardview_shadow_end_color"
                android:layout_margin="10dp"
            />
        </com.google.android.material.card.MaterialCardView>
        <LinearLayout
            android:id="@+id/l1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:weightSum="100"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintBottom_toBottomOf="parent"
            android:orientation="horizontal"
            android:layout_margin="20dp">

            <LinearLayout
                android:layout_weight="90"

```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/phoneNoInput"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Mobile No"
    app:endIconMode="custom"

app:endIconDrawable="@drawable/ic_baseline_phone_android_24"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox">

<com.google.android.material.textfield.TextInputEditText
    android:id="@+id/phoneNo"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:inputType="phone"
    />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/msgInput"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Message"
    app:endIconMode="custom"

app:endIconDrawable="@drawable/ic_baseline_message_24"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox">

<com.google.android.material.textfield.TextInputEditText
    android:id="@+id/msg"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:inputType="textLongMessage"
    />

</com.google.android.material.textfield.TextInputLayout>
</LinearLayout>
```

```
<com.google.android.material.button.MaterialButton
    android:id="@+id/sendButton"
    app:fabSize="normal"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:layout_marginHorizontal="10dp"
    android:layout_marginTop="5dp"
    app:icon="@drawable/ic_baseline_send_24"
    app:iconSize="20dp"
    android:elevation="5dp"
    app:cornerRadius="15dp"
/>

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.kt:

```
package com.example.madpractical9_21012022030

import android.Manifest
import android.content.IntentFilter
import android.content.pm.PackageManager
import android.net.Uri
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.provider.Telephony
import android.telephony.SmsManager
import android.widget.ListView
import androidx.appcompat.app.AlertDialog
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import
com.example.madpractical9_21012022030.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    private lateinit var binding : ActivityMainBinding
    private val SMS_PERMISSION_CODE = 100
    private lateinit var lv: ListView
    private lateinit var al: ArrayList<SMSView>
    private lateinit var smsReceiver: SMSBroadcastReceiver
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
    }
}
```

```

        lv = binding.listView
        al = ArrayList()

        if (checkRequestPermission()) {
            loadSMSInbox()
        }

        smsReceiver = SMSBroadcastReceiver()
        registerReceiver(smsReceiver, IntentFilter(
            Telephony.Sms.Intents.SMS_RECEIVED_ACTION))
        smsReceiver.setListener(ListenerImplement())
        binding.sendButton.setOnClickListener{
            val phone = binding.phoneNo.text.toString()
            val msg = binding.msg.text.toString()
            sendSms(phone, msg)

            val builder : AlertDialog.Builder =
AlertDialog.Builder(this@MainActivity)
            builder.setTitle("Sent SMS")

            builder.setMessage("SMS is sent.\nPhone No :
$phone \n\n Message : $msg")
            builder.setCancelable(true)
            builder.setPositiveButton("OK", null);
            builder.show()
        }
    }

    private fun sendSms(sPhoneNo: String?, sMsg: String?) {
        if (!checkRequestPermission()) {
            return
        }
        val smsmanager = SmsManager.getDefault()
        if (smsmanager != null) {
            smsmanager.sendTextMessage(sPhoneNo, null, sMsg, null, null)
        }
    }

    inner class
ListenerImplement: SMSBroadcastReceiver.Listener {
        override fun onTextReceived(sPhoneNo: String?, sMsg:
String?) {
            val builder : AlertDialog.Builder =
AlertDialog.Builder(this@MainActivity)
            builder.setTitle("New SMS Received")
            builder.setMessage("From : $sPhoneNo\n\n Message
: $sMsg")
            builder.setCancelable(true)
            builder.setPositiveButton("OK", null);

```

```

        builder.show()
        loadSMSInbox()
    }
}

private val isSMSReadPermission: Boolean
    get() = ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_SMS) ==
        PackageManager.PERMISSION_GRANTED
private val isSMSWritePermission: Boolean
    get() = ContextCompat.checkSelfPermission(this,
Manifest.permission.SEND_SMS) ==
        PackageManager.PERMISSION_GRANTED

private fun requestSMSPermission() {
    if
(ActivityCompat.shouldShowRequestPermissionRationale
        (this, Manifest.permission.READ_SMS)) {
        // You may display a non-blocking explanation
here,
        read more in the documentation:
        //
https://developer.android.com/training/permissions/requesting.html
    }

    ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.READ_SMS,
        Manifest.permission.SEND_SMS,
        Manifest.permission.RECEIVE_SMS),
        SMS_PERMISSION_CODE)
}
private fun checkRequestPermission(): Boolean {
    return if (!isSMSReadPermission ||
!isSMSWritePermission) {
        requestSMSPermission()
        false
    } else true
}

private fun loadSMSInbox() {
    if (!checkRequestPermission()) return
    val uriSMS = Uri.parse("content://sms/inbox")
    val c = contentResolver.query(uriSMS, null, null,
null, null)
    al.clear()
    while (c!!.moveToNext()) {
        al.add(SMSView(c.getString(2), c.getString(12)))
    }
    lv.adapter = SMSViewAdapter(this, al)
}

```



```

    }

    override fun onDestroy() {
        unregisterReceiver(smsReceiver)
        super.onDestroy()
    }
}

```

SMSBroadcastReceiver.kt:

```

package com.example.madpractical9_21012022030

import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.os.Build
import android.provider.Telephony

class SMSBroadcastReceiver : BroadcastReceiver() {

    interface Listener {
        fun onTextReceived(sPhoneNo: String?, sMsg: String?)
    }

    private var listener: Listener? = null
    fun setListener(lis: Listener) {
        listener = lis
    }

    override fun onReceive(context: Context, intent: Intent) {
        if (intent.action ==
Telephony.Sms.Intents.SMS_RECEIVED_ACTION) {
            var sPhoneNo = ""
            var sSMSBody = ""
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                for (smsMessage in
Telephony.Sms.Intents.getMessagesFromIntent(intent)) {
                    sPhoneNo =
smsMessage.displayOriginatingAddress
                    sSMSBody += smsMessage.messageBody
                }
                if (listener != null) {
                    listener?.onTextReceived(sPhoneNo, sSMSBody)
                }
            }
        }
    }
}

```

SMSView.kt:

```

package com.example.madpractical9_21012022030

```

```
class SMSView(val phoneNo:String, val msg:String) {
}
```

SMSViewAdapter.kt:

```
package com.example.madpractical9_21012022030

import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.View.inflate
import android.view.ViewGroup
import android.widget.AdapterView
import com.example.madpractical9_21012022030.databinding.SmsItemViewBinding

class SMSViewAdapter(context: Context, private val
array:ArrayList<SMSView>):
    ArrayAdapter<SMSView>(context,array.size,array) {
    override fun getView(position: Int, convertView: View?,
parent: ViewGroup): View {
        //val currentItemView:View =
        LayoutInflater.from(context)
            .inflate(R.layout.sms_item_view, parent, false)
        // get the position of the view from the ArrayAdapter
        val currentSms: SMSView? = getItem(position)
        val binding =
        SmsItemViewBinding.inflate(LayoutInflater.from(context))
        binding.textViewPhoneNo.text = currentSms!!.phoneNo
        binding.textViewMessage.text = currentSms.msg
        return binding.root
        //return super.getView(position, convertView, parent)
    }
}
```

Manifest:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
package="com.example.madpractical9_20012021068" >

    <uses-permission
android:name="android.permission.RECEIVE_SMS" />
    <uses-permission
```

```

android:name="android.permission.READ_SMS" />
    <uses-permission
android:name="android.permission.SEND_SMS" />

    <application
        android:allowBackup="true"

android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Practical 9"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"

android:theme="@style/Theme.MADPractical9_20012021068"
        tools:targetApi="31" >
        <receiver
            android:name=".SMSBroadcastReceiver"
            android:enabled="true"
            android:exported="true" >
        </receiver>

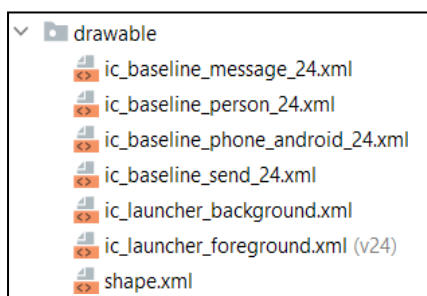
        <activity
            android:name=".MainActivity"
            android:exported="true" >
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>

```

Resources:



Output:

