Lab 26: Android Notification using Kotlin

Introduction

Even when the programme is not active, Android Notifications deliver quick, pertinent information about the action that took place. The emblem, title, and a portion of the content text are displayed in the notice. Using the NotificationCompat.Builder object, Android notification properties are set.

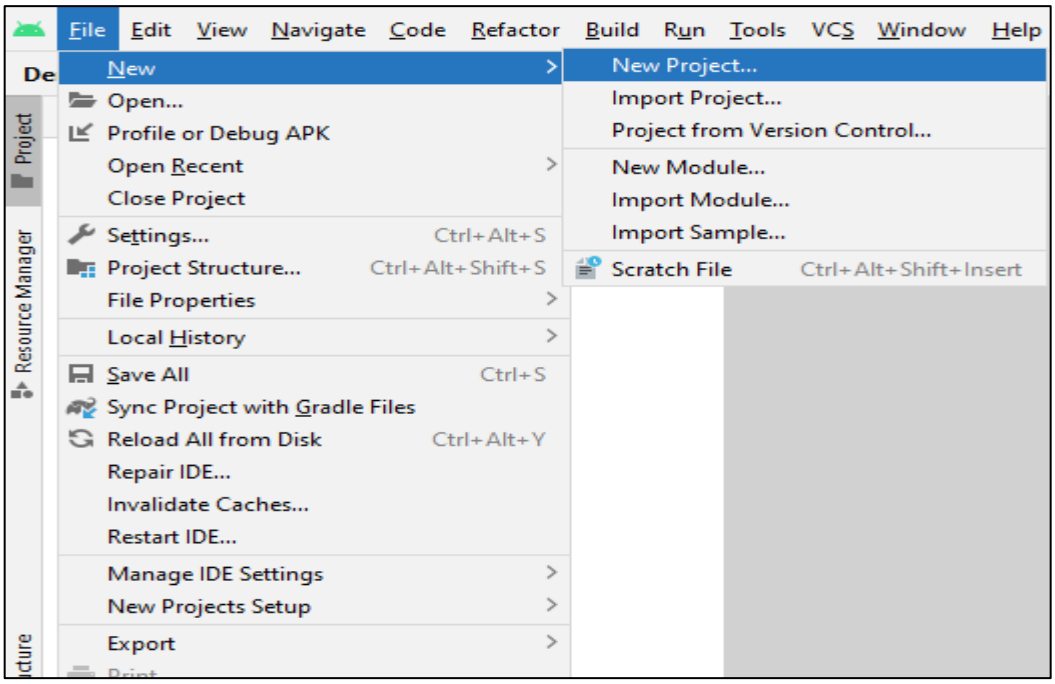
Some of the notification properties are mention below:

- **setSmallIcon()**
- **setContentTitle()**
- **setContentText()**
- **setAutoCancel()**
- **setPriority()**

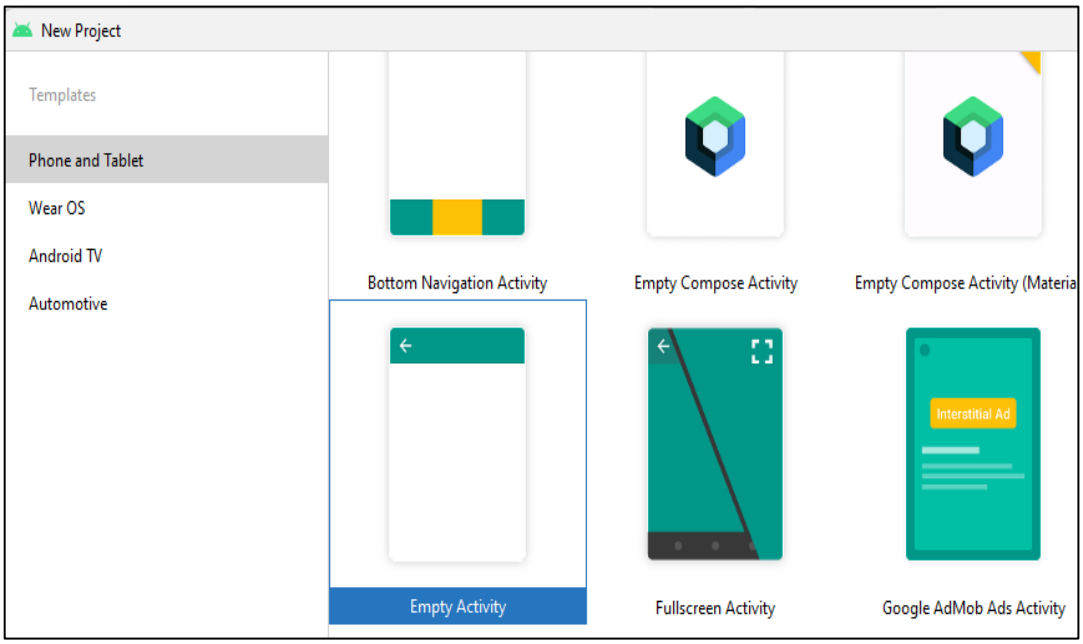
Let's get Started:

In this experiment we will develop an Android App to demonstrate the use of Android Notification Manager.

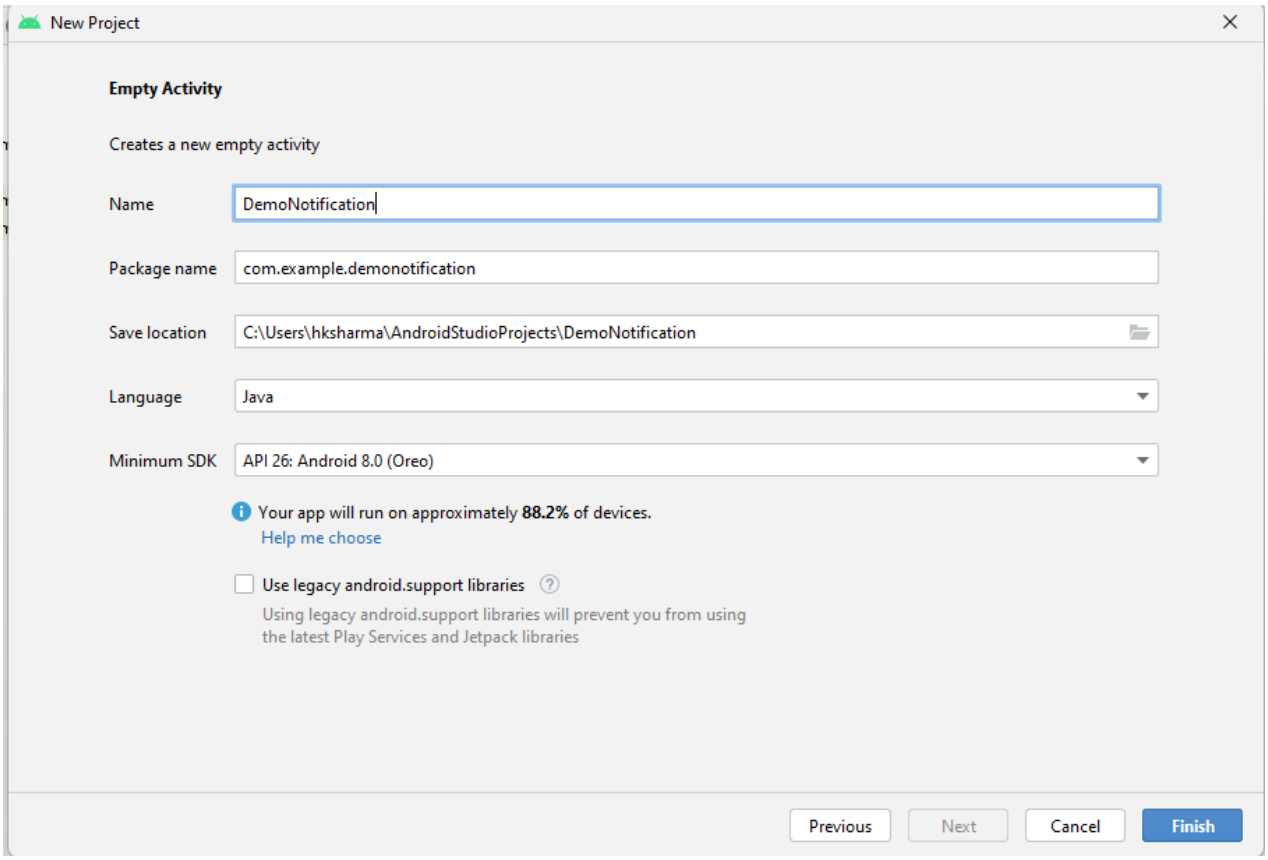
Step 1: Create a New Project in Android Studio as shown below



Step 2: Select Empty Activity as shown below



Step 3: Provide a Project Name as shown below



Step 4: Update MainActivity.kt as per the code given below

```
package com.example.demonotificationkotlin
import android.app.Notification
import android.app.NotificationChannel
import android.app.NotificationManager
import android.app.PendingIntent
import android.content.Context
import android.content.Intent
import android.graphics.BitmapFactory
import android.graphics.Color
import android.os.Build
import android.os.Bundle
import android.widget.Button
import android.widget.RemoteViews
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    // declaring variables
    lateinit var notificationManager: NotificationManager
    lateinit var notificationChannel: NotificationChannel
    lateinit var builder: Notification.Builder
    private val channelId = "i.apps.notifications"
    private val description = "Test notification"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val btn = findViewById<Button>(R.id.btn)
        notificationManager = getSystemService(Context.NOTIFICATION_SERVICE)
as NotificationManager
        btn.setOnClickListener {
            val intent = Intent(this, afterNotification::class.java)
            val pendingIntent = PendingIntent.getActivity(this, 0, intent,
PendingIntent.FLAG_UPDATE_CURRENT)
            val contentView = RemoteViews(packageName,
R.layout.activity_after_notification)
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                notificationChannel = NotificationChannel(channelId,
description, NotificationManager.IMPORTANCE_HIGH)
                notificationChannel.enableLights(true)
                notificationChannel.lightColor = Color.GREEN
                notificationChannel.enableVibration(false)

notificationManager.createNotificationChannel(notificationChannel)

                builder = Notification.Builder(this, channelId)
                    .setContent(contentView)
                    .setSmallIcon(R.drawable.ic_launcher_background)
                    .setLargeIcon(BitmapFactory.decodeResource(this.resources,
R.drawable.ic_launcher_background))
                    .setContentIntent(pendingIntent)
            } else {
```

```
                builder = Notification.Builder(this)
                    .setContent(contentView)
                    .setSmallIcon(R.drawable.ic_launcher_background)
                    .setLargeIcon(BitmapFactory.decodeResource(this.resources,
R.drawable.ic_launcher_background))
                    .setContentIntent(pendingIntent)
            }
            notificationManager.notify(1234, builder.build())
        }
    }
}
```

Step 5: activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="Send Notification" />

</RelativeLayout>
```

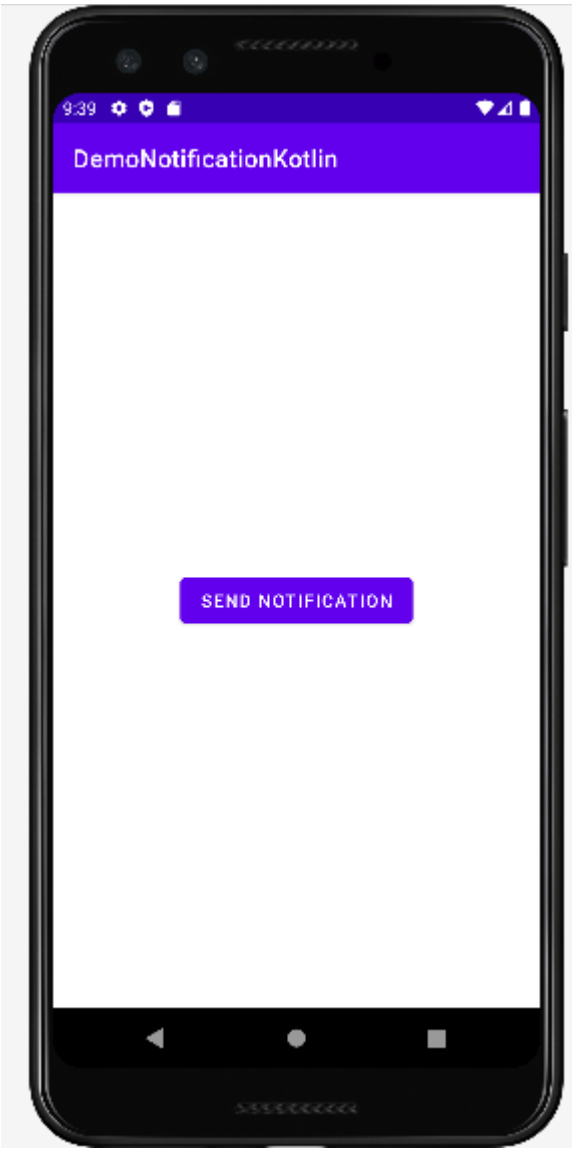
Step 6: create afterNotification.kt and add code in activity_after_notification.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".afterNotification">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="Welcome To Snap Notification"
        android:textSize="15sp"
        android:textStyle="bold" />

</RelativeLayout>
```

Step 7: Output



Voila!! We have successfully completed this lab.