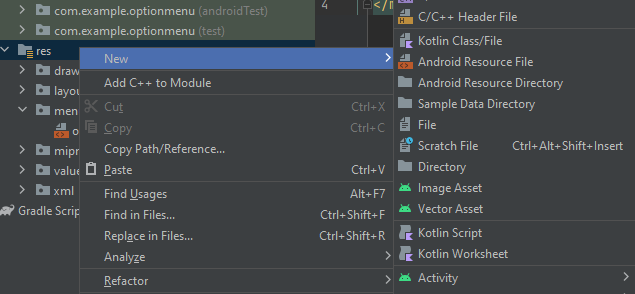
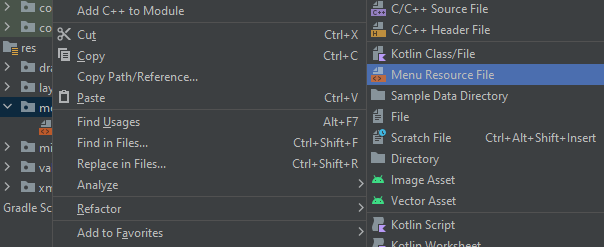
1.click res and create a new android resource directory

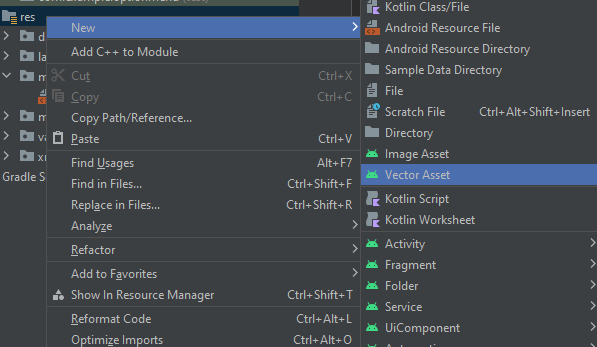
2.directory name as our choice and choose type as menu

3.now click menu ->new menu resource file name it as optionmenu.xml





To add icon using vector asset



options menu.xml

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android">

<item android:id="@+id/search\_id"

android:title="search"

android:icon="@drawable/search\_icon"/>

<item android:id="@+id/share\_id"

android:title="share"

android:icon="@drawable/share\_icon"/>

<item android:id="@+id/download\_id"

android:title="download"

android:icon="@drawable/download\_icon"

/>

</menu>

Main.java

package com.example.optionmenu;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.view.menu.MenuBuilder;

import android.annotation.SuppressLint;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuInflater;

import android.view.MenuItem;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

@SuppressLint("RestrictedApi")

@Override

public boolean onCreateOptionsMenu(Menu menu) {

MenuInflater inflater=getMenuInflater();

inflater.inflate(R.menu.optionmenu,menu);

if(menu instanceof MenuBuilder) {

MenuBuilder m = (MenuBuilder) menu; //to view icon with menu

m.setOptionalIconsVisible(true);

}

return super.onCreateOptionsMenu(menu);

}

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

Toast.makeText(this, "selected item is "+item.getTitle(), Toast.LENGTH\_SHORT).show();

switch(item.getItemId())

{

case R.id.search\_id:

return true;

case R.id.share\_id:

return true;

case R.id.download\_id:

return true;

default:

return super.onOptionsItemSelected(item); //must

}

}

}

Output

