

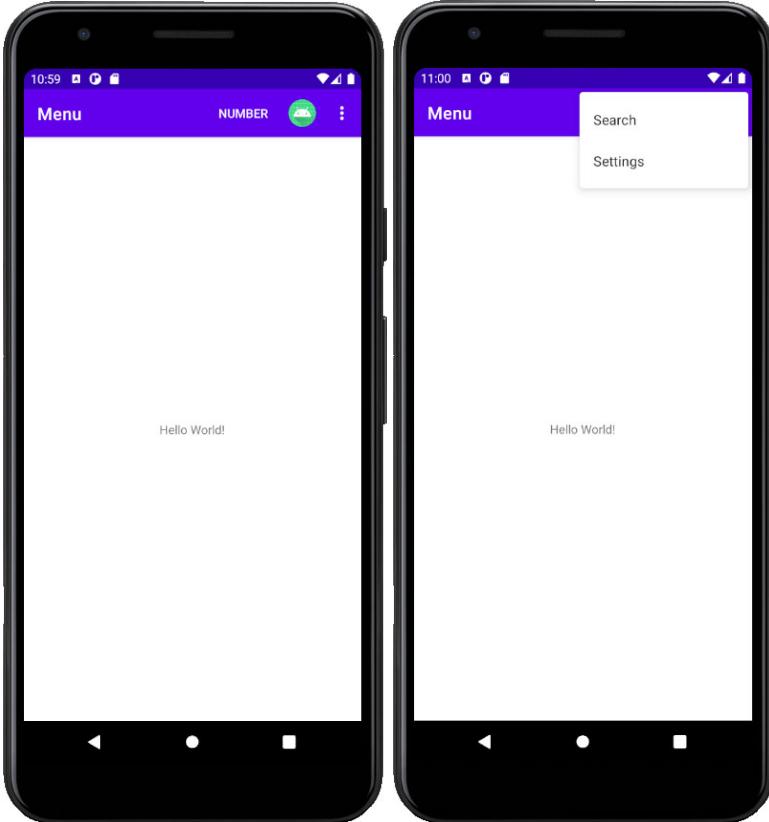
Creating an Action Bar (Menu)

Reference: <https://developer.android.com/training/appbar/setting-up>

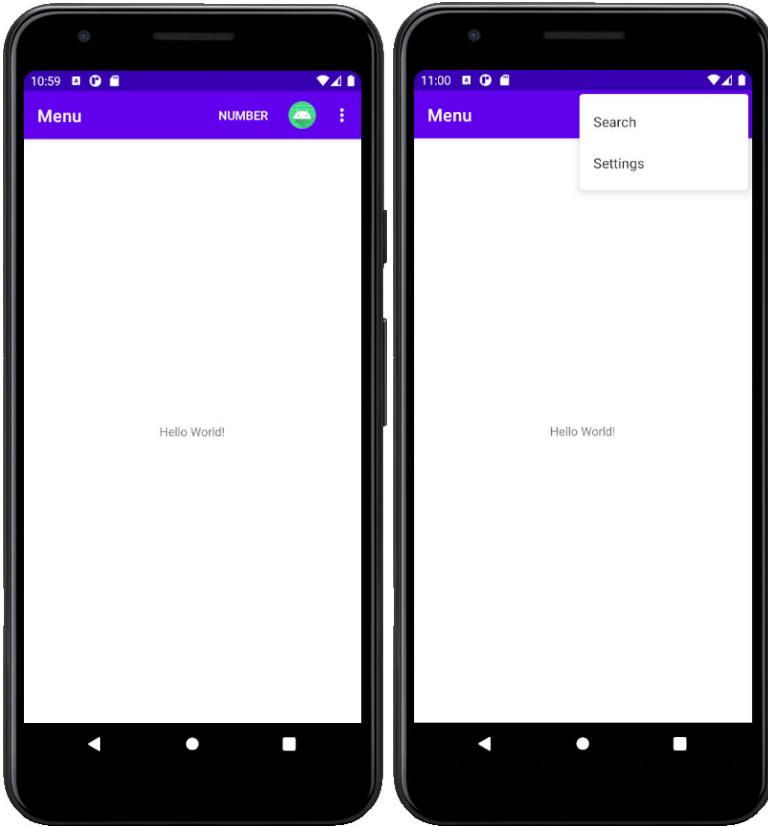
<https://developer.android.com/reference/android/app/ActionBar>

<https://www.geeksforgeeks.org/toolbar-in-android-with-example/>

This document describes how to do the following:



- [Basic action bar menu](#)
- [Customize the action bar](#)
- [Remove the ActionBar \(Menu bar\) Completely](#)
- [Add your own custom ActionBar \(Menu bar\)](#)



Basic action bar menu

1. Create a new **Empty Activity** project and name it **Menu**

Edit the *main_menu.xml* file

2. Create a new directory under the **res** folder and name it **menu**.
 - Right-click on the **res** folder
 - Select **New | Directory**
 - Type in **menu** for the name
3. Create a new menu xml file in the newly created **menu** folder and name it **main_menu**.
 - Right-click on the **menu** folder
 - Select **New | Menu Resource File**
 - Type in **main_menu** for the name
4. Type the following in the newly created **main_menu.xml** file.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/search"
        android:title="Search"
```

```

        app:actionViewClass="android.widget.SearchView"
        app:showAsAction="ifRoom"/>
    <item
        android:id="@+id/number"
        android:title="Number"
        app:showAsAction="always"/>
    <item
        android:id="@+id/alpha"
        android:title="Alpha"
        android:icon="@mipmap/ic_launcher"
        app:showAsAction="always"/>
    <item
        android:id="@+id/settings"
        android:title="Settings"
        app:showAsAction="never"/>
</menu>

```

Edit the MainActivity.java file

5. Override the **onCreateOptionsMenu** method to inflate the menu that is defined in `main_menu.xml`.
6. Override the **onOptionsItemSelected** method to respond to clicks on the items in the menu.
7. Here's the complete `ActivityMain.java` file

```

package com.example.menu;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.Menu;
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);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main_menu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        switch (id) {
            case android.R.id.home:// for the HomeAsUpIndicator icon
                Toast.makeText(this, "click app icon",

```

```

        Toast.LENGTH_SHORT).show();
            break;
        case R.id.search:
            Toast.makeText(this,"click search",
        Toast.LENGTH_SHORT).show();
            break;
        case R.id.number:
            Toast.makeText(this,"click number",
        Toast.LENGTH_SHORT).show();
            break;
        case R.id.alpha:
            Toast.makeText(this,"click alpha", Toast.LENGTH_SHORT).show();
            break;
        case R.id.settings:
            Toast.makeText(this,"click settings",
        Toast.LENGTH_SHORT).show();
            break;
        default:
            return super.onOptionsItemSelected(item);
    }
    return true;
}
}

```

Customize the action bar

Edit the themes.xml file

- In the manifests |AndroidManifest.xml file the following line defines the name of the theme to use

```
        android:theme="@style/Theme.Menu">
```

This name, **Theme.Menu** (where Menu is the name of your app), along with all of the other theme attributes are defined in the file **res | values | themes | themes.xml** as shown here

```

<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.Menu"
parent="Theme.MaterialComponents.DayNight.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/purple_500</item>
        <item name="colorPrimaryVariant">@color/purple_700</item>
        <item name="colorOnPrimary">@color/white</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/teal_200</item>
        <item name="colorSecondaryVariant">@color/teal_700</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item name="android:statusBarColor">
            tools:targetApi="l">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->

```

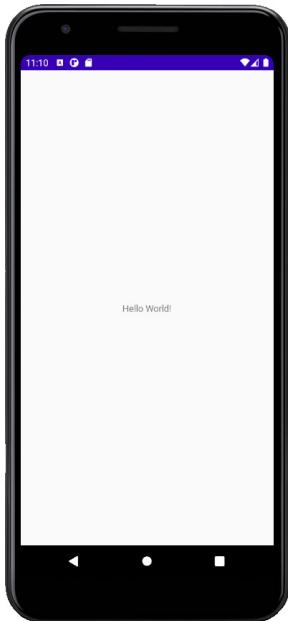
```
</style>
</resources>
```

- Notice the style `name="Theme.Menu"`
- You can change the color to whatever you want
- The `colorPrimary` is the color for the ActionBar that is below the status bar
- The color for the status bar at the very top is set to use the `colorPrimaryVariant` color

To remove the default action bar

9. To remove the action bar change the parent theme to `NoActionBar`

```
<style name="Theme.Menu" parent="Theme.AppCompat.Light.NoActionBar">
```



10. The `colorPrimary` is the color for the ActionBar that is below the status bar
11. The color for the status bar at the very top is set to use the `colorPrimaryVariant` color

To add your own custom action bar

Edit the activity_main.xml file

12. Add the `androidx.appcompat.widget.Toolbar`
13. Here's the complete `activity_main.xml` file

```
<?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">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        android:elevation="4dp"
        android:theme="@style/ThemeOverlay.AppCompat.ActionBar"
        android:popupTheme="@style/ThemeOverlay.AppCompat.Light"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

- Another version

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    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:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:theme="@style/Theme.MyApplication.AppBarOverlay">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/Theme.MyApplication.PopupOverlay" />

    </com.google.android.material.appbar.AppBarLayout>

</androidx.coordinatorlayout.widget.CoordinatorLayout>

```

Edit the MainActivity.java file

14. In the **onCreate** method add the following lines
 - Type **Toolbar** and select the **androidx.appcompat.widget**
 - Make sure that the **import androidx.appcompat.widget.Toolbar** is added and not **android.widget.Toolbar** or another one
 - Finish typing the line by declaring the **toolbar** variable and initializing it to the view
 - Call **setSupportActionBar** to use the toolbar as the ActionBar
 - Select an icon from the Vector Asset library and put it in the **drawable** folder. Name it **ic_icon**. See instruction in step 16 on how to select an icon from the library

- The rest of the lines are optional customizations.
- By default the app title is shown. Call **setTitle** to change it.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    Toolbar toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar); // use toolbar as ActionBar

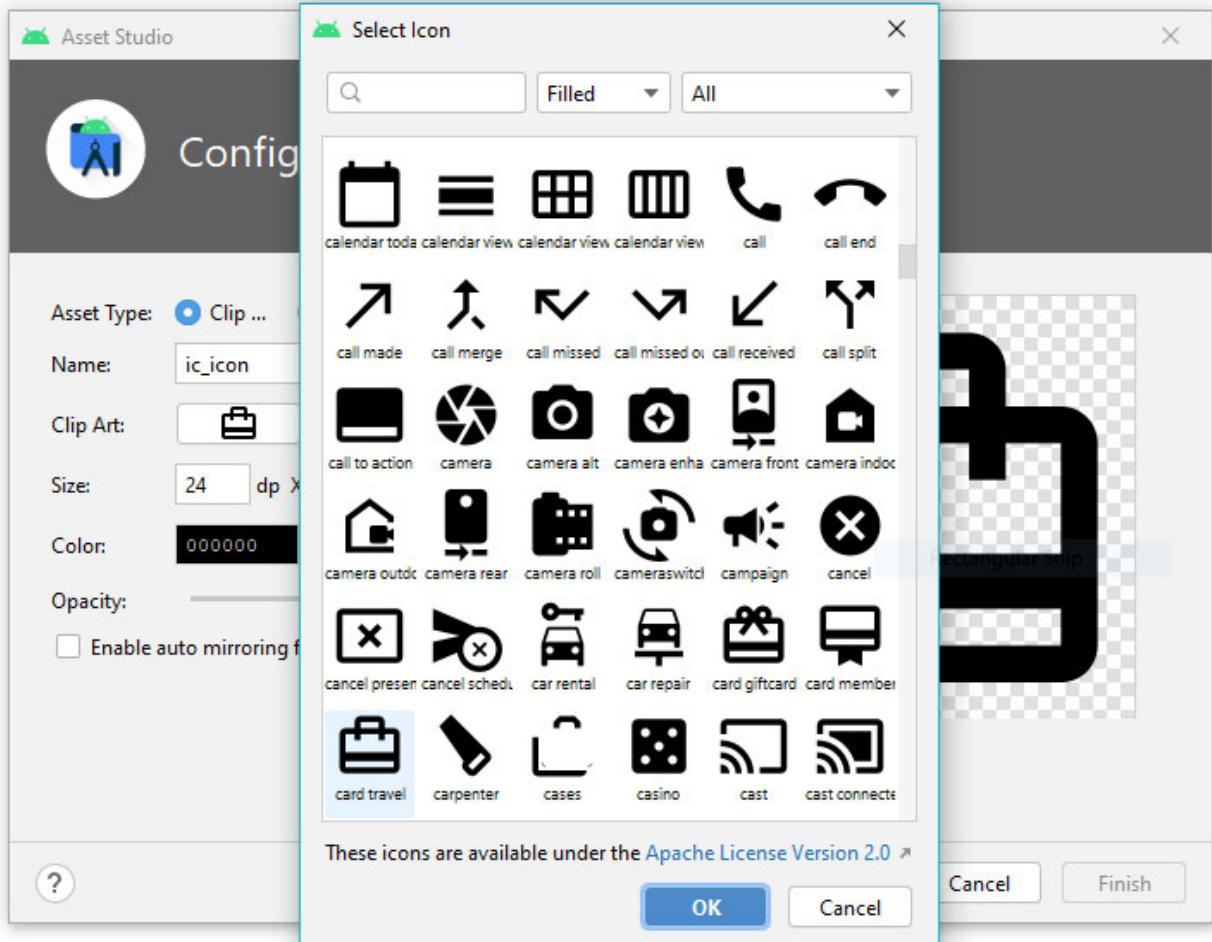
    // optional customization
    ActionBar actionBar = getSupportActionBar();

    actionBar.setLogo(R.drawable.ic_icon); // show app Logo
    actionBar.setDisplayUseLogoEnabled(true);

    actionBar.setHomeAsUpIndicator(R.drawable.ic_launcher_foreground);
    actionBar.setDisplayHomeAsUpEnabled(true); // show up icon

    actionBar.setTitle(""); // remove the app title
}
```

15. Everything in the **onCreateOptionsMenu** method and the **onOptionsItemSelected** method are the same as before
16. To use an icon from the library
 - Right-click on **res | drawable**
 - Select **New | Vector Asset**
 - Click on the icon next to **Clip Art**
 - Select the icon you want
 - Rename to **ic_icon**



Remove the ActionBar (Menu bar) Completely

Reference: <https://developer.android.com/training/appbar/setting-up>

Edit the AndroidManifest.xml file

17. Change the app theme to use custom theme defined in `style.xml`

```
    android:theme="@style/AppTheme"
```

18. Here's the complete `AndroidManifest.xml` file

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Edit the styles.xml file

19. Create `styles.xml` file under the `res | values` folder

- Right-click on the `res | values` folder
- Select **New | Value Resource File**
- Type in `styles` for the filename

20. Copy and paste the following code into `styles.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/teal_200</item>
        <item name="colorPrimaryDark">@color/teal_700</item>
        <item name="colorAccent">@color/purple_200</item>
    </style>
</resources>
```

21. To remove the action bar the parent theme must be

- ```
parent="Theme.AppCompat.Light.NoActionBar"
```
22. The colorPrimary is the color for the ActionBar that is below the status bar  
23. The colorPrimaryDark is the color for the status bar at the very top

## Add your own custom ActionBar (Menu bar)

Reference: <https://developer.android.com/training/appbar/setting-up>

<https://developer.android.com/reference/android/app/ActionBar>

<https://www.geeksforgeeks.org/toolbar-in-android-with-example/>

### Edit the activity\_main.xml file

24. Add a Toolbar object to the activity's layout

```
<?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/appBarLayout"
 android:layout_width="0dp"
 android:layout_height="wrap_content"
 app:layout_constraintEnd_toEndOf="parent"
 app:layout_constraintStart_toStartOf="parent"
 app:layout_constraintTop_toTopOf="parent">

 <androidx.constraintlayout.widget.ConstraintLayout
 android:layout_width="match_parent"
 android:layout_height="match_parent">

 <androidx.appcompat.widget.Toolbar
 android:id="@+id/toolbar"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:background="?attr/colorPrimary"
 android:minHeight="?attr/actionBarSize"
 android:theme="?attr/actionBarTheme"
 app:layout_constraintEnd_toEndOf="parent"
 app:layout_constraintStart_toStartOf="parent"
 app:layout_constraintTop_toTopOf="parent"
 app:title="Custom Menu"/>

 <!-- Right most TextView -->
 <TextView
 android:id="@+id/textView2"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_marginRight="10dp"
```

```

 android:text="0"
 android:textColor="#fffff"
 android:textSize="26sp"
 android:textStyle="bold"
 app:layout_constraintBottom_toBottomOf="parent"
 app:layout_constraintEnd_toEndOf="@+id/toolbar"
 app:layout_constraintTop_toTopOf="parent" />

 <!-- Right most ImageView -->
 <ImageView
 android:id="@+id/imageView2"
 android:layout_width="wrap_content"
 android:layout_height="24dp"
 android:layout_marginRight="9dp"
 android:src="@drawable/ic_one"
 app:layout_constraintBottom_toBottomOf="parent"
 app:layout_constraintEnd_toStartOf="@+id/textView2"
 app:layout_constraintTop_toTopOf="parent"
 />

 <!-- ImageView beside title of ToolBar -->
 <ImageView
 android:id="@+id/imageView"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_marginRight="150dp"
 app:layout_constraintBottom_toBottomOf="parent"
 app:layout_constraintEnd_toStartOf="@+id/imageView2"
 app:layout_constraintTop_toTopOf="parent"
 android:src="@drawable/ic_two"
 app:tint="#fffff" />

</androidx.constraintlayout.widget.ConstraintLayout>

</com.google.android.material.appbar.AppBarLayout>

<!-- TextView of the Activity -->
<TextView
 android:id="@+id/textView"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="Hello World!"
 app:layout_constraintBottom_toBottomOf="parent"
 app:layout_constraintLeft_toLeftOf="parent"
 app:layout_constraintRight_toRightOf="parent"
 app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

## Edit the MainActivity.java file

25. Make sure that it is the androidx toolbar that is imported and not another version of the toolbar

```
package com.example.menu;

import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
 TextView textView;
 int count = 0;

 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);

 // assigning ID of the toolbar to a variable
 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
 // using toolbar as ActionBar
 setSupportActionBar(toolbar);

 // optional customization
 ActionBar actionBar = getSupportActionBar();
 // show app logo
 actionBar.setDisplayShowHomeEnabled(true);
 actionBar.setDisplayUseLogoEnabled(true);
 actionBar.setLogo(R.drawable.ic_app_icon);
 // show title
 actionBar.setTitle("Hello");
 // show homeasup icon
 actionBar.setHomeAsUpIndicator(R.drawable.ic_launcher_foreground);
 actionBar.setDisplayHomeAsUpEnabled(true);

 // assigning ID of textView2 to a variable
 textView = (TextView) findViewById(R.id.textView2);

 // "on click" operations to be performed
 textView.setOnClickListener(new View.OnClickListener() {
 @Override

 // incrementing the value of textView
 public void onClick(View view) {
 count++;
 textView.setText("" + count);
 }
 });
 }
}
```

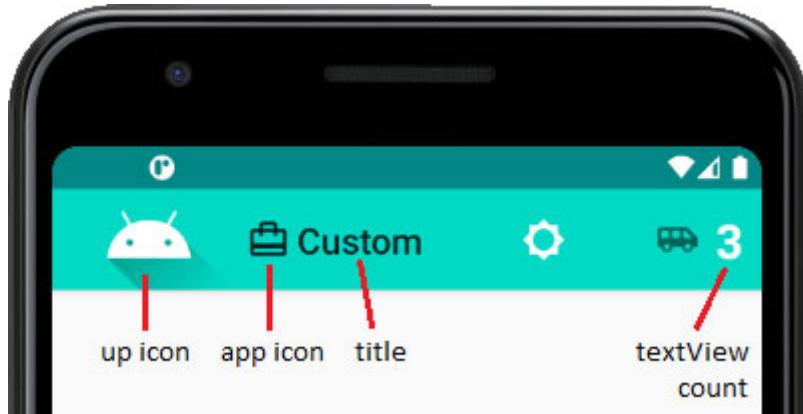
## **Icons**

26. Need to have three icons named **ic\_one**, **ic\_two** and **ic\_app\_icon** in the **drawable** folder.

## **Run it**

27. That's it. Run the app on an actual device.

Press the right-most textView. Each press will increment the count.



```
app:navigationIcon="@drawable/ic_launcher_foreground"
app:logo="@drawable/ic_launcher_foreground"
app:contentInsetLeft="0dp"
app:contentInsetStart="0dp"
app:contentInsetStartWithNavigation="0dp"
```