Andhra Pradesh State Skill Development Corporation

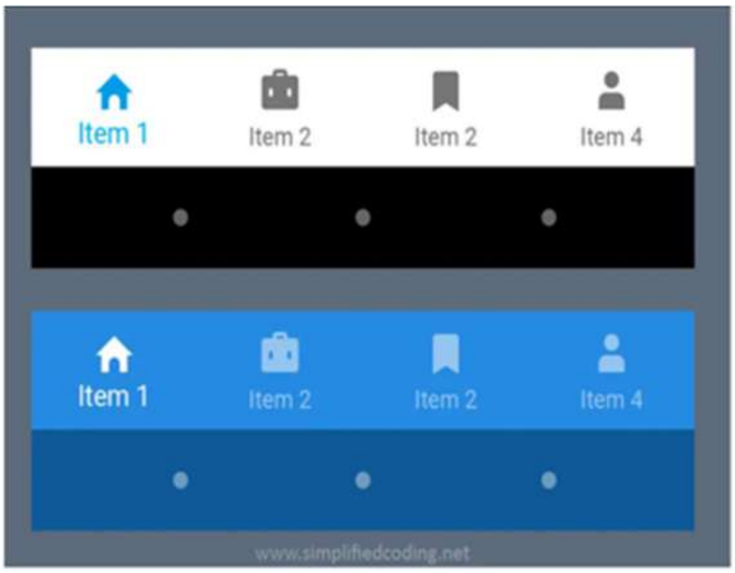
e Skill AP
Learn Anytime Anywhere
Andhra Pradesh State Skill Development Corporation

ANDROID APPLICATION DEVELOPMENT

BOTTOM NAVIGATION

BOTTOM NAVIGATION ANDROID

In Android, there are many options for making navigation easy in your application for the user. Bottom Navigation Bar is one of them. Bottom Navigation Bar always stays at the bottom of your application and provides navigation between the views of your application.

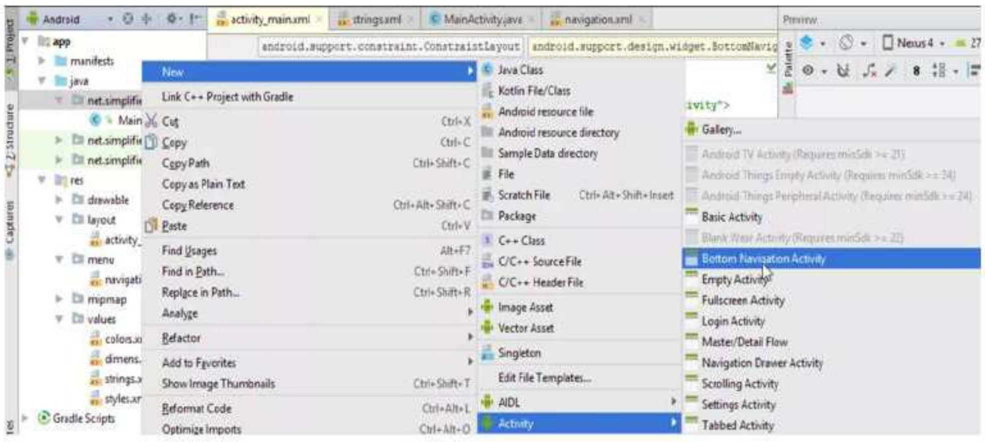


Bottom Navigation Android

Using Bottom Navigation in Android

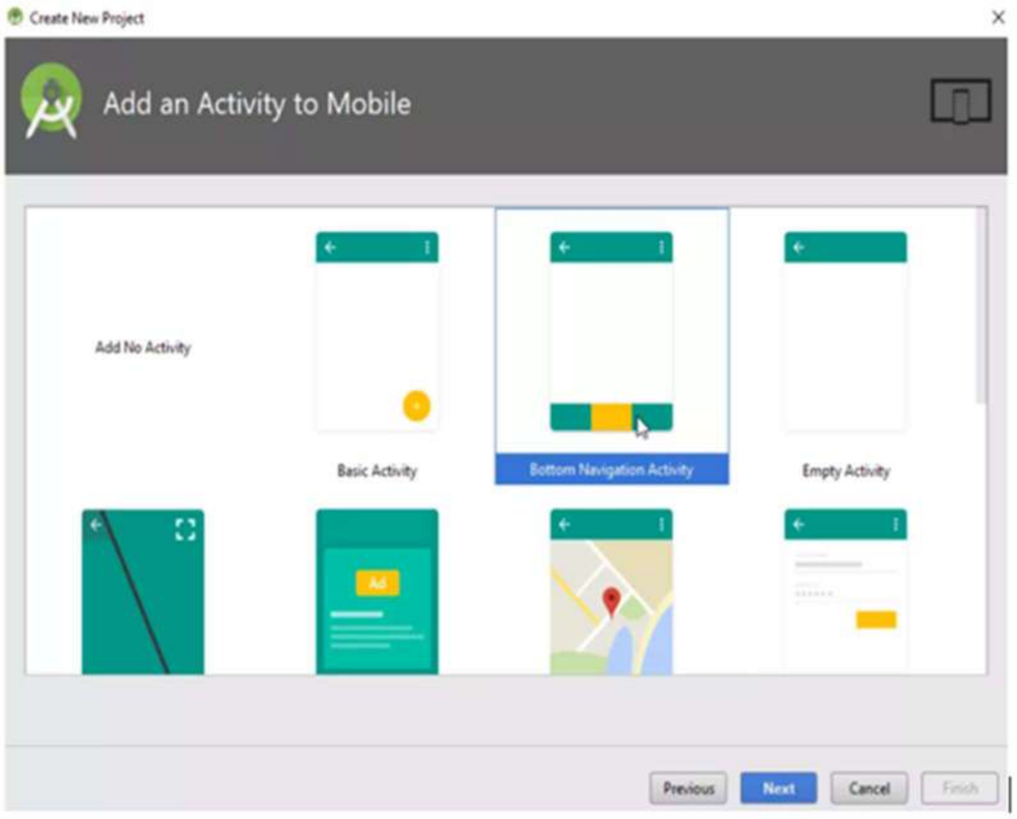
It is very easy, you can create a Bottom Navigation Activity in your project by following ways.

1. you just need to use , if you want to do it in XML.
2. Android Studio also have a predefined template for creating BottomNavigationView, when you create a new Activity you can select Bottom Navigation Activity, as shown in the image.



Bottom Navigation Android Activity

3. While creating an Android Project you can select Bottom Navigation Activity from the template.



Bottom Navigation Android Activity from Templates

Bottom Navigation Android Example

Now lets see everything in an Android Project.

Defining Colors and Strings

• I have created a project named BottomNavigationExample using a Bottom Navigation Activity. You can choose Empty Activity, it will not make any difference.
• Now first, we will define the colors. You can change the colors to anything you want. So define the following in your colors.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#278be3</color>
    <color name="colorPrimaryDark">#0f5998</color>
    <color name="colorAccent">#4075a2</color>
    <color name="colorNavIcon">#dae9f6</color>
    <color name="colorNavText">#01294b</color>
</resources>
```

• Now come to strings.xml and define the following strings.

```
<resources>

    <string name="app_name">Bottom Navigation Android Example</string>

    <string name="title_home">Home</string>
    <string name="title_dashboard">Dashboard</string>
    <string name="title_notifications">Notifications</string>
    <string name="title_profile">Profile</string>

</resources>
```

Customizing Bottom Navigation View

• Now come to activity_main.xml and modify it as shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/a
pk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="net.simplifiedcoding.bottomnavigationexample.MainActivity">

    <FrameLayout
        android:id="@+id/fragment_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="56dp"
        android:text="@string/title_home"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <android.support.design.widget.BottomNavigationView
        android:id="@+id/navigation"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        app:itemIconTint="@color/colorNavIcon"
        app:itemTextColor="@color/colorNavText"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:menu="@menu/navigation" />

</android.support.constraint.ConstraintLayout>
```

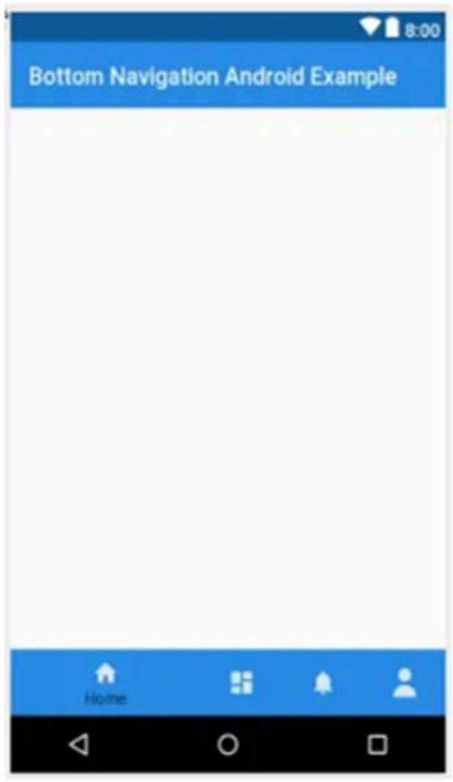
Remember for we can use the following properties.
• app:itemIconTint : for defining the Item Icon color.
• app:itemTextColor : for defining the Item Text Color.

• android:background : for defining the Bottom Navigation View background.
• app:menu : for defining the menu that we need to display in the Bottom Navigation View.

Now, if you created the project using Bottom Navigation Activity template, a menu file named navigation.xml is created by default inside the menu folder. If it is not created you can create it manually as well. Here we define all the menu items that we need to display in the Bottom Navigation Bar.

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

   <item
      android:id=”@+id/navigation_home”
      android:icon=”@drawable/ic_home_black_24dp”
      android:title=”@string/title_home” />

   <item
      android:id=”@+id/navigation_dashboard”
      android:icon=”@drawable/ic_dashboard_black_24dp”
      android:title=”@string/title_dashboard” />

   <item
      android:id=”@+id/navigation_notifications”
      android:icon=”@drawable/ic_notifications_black_24dp”
      android:title=”@string/title_notifications” />

   <item
      android:id=”@+id/navigation_profile”
      android:icon=”@drawable/ic_profile_black_24dp”
      android:title=”@string/title_profile” />

</menu>
```

• So, the above codes will produce the following Layout for our Main Activity.



• If you have used the predefined template for creating Bottom Navigation View, then you will have some codes in the MainActivity.java by default. We will remove those codes and change the MainActivity as shown below, so that we can learn everything.

```
Package net.simplifiedcoding.bottomnavigationexample;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

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

}
```

Creating Fragments

For each of the view we need to create a layout resource file and a fragment class to inflate the view. As we have 4 different views to be switched we need to create 4 layout resource files and 4 java classes. Every class and resource file is almost the same as we don't have anything other than a simple TextView in the screens.

Creating Layout Resource Files

• Create fragment_home.xml,fragment_dashboard.xml,fragment_notifications.xml and fragment_profile.xml.

• All the files will have the following code in it.

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="Home"
        android:textAppearance="@style/Base.TextAppearance.AppCompat.Large" />

</RelativeLayout>
```

• Don’t forget to change the text value of the TextView in each file.

Creating Fragments

• Now create java classes named, HomeFragment.java, DashboardFragment.java, NotificationsFragment.java and ProfileFragment.java.
• Each file will contain the following code in it.

```
package net.simplifiedcoding.bottomnavigationexample;

import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

/**
 * Created by Belal on 1/23/2018.
 */

public class HomeFragment extends Fragment {
    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullab
le Bundle savedInstanceState) {
        //just change the fragment_dashboard
        //with the fragment you want to inflate
        //like if the class is HomeFragment it should have R.layout.home_fragment
        //if it is DashboardFragment it should have R.layout.fragment_dashboard
        return inflater.inflate(R.layout.fragment_home, null);
    }
}
```

• Don’t forget to change the layout resource id (R.layout.file_name) with the layout that you want to display for the fragment.

Switching Fragments

• Now we will switch the screens or fragments when the bottom navigation menu is clicked. We also need to load some fragment initially which is HomeFragment in this case.
• First we will create a method to switch the fragment. I have created the following method named loadFragment() which is taking Fragment is an object.

```
  private boolean loadFragment(Fragment fragment) {
      //switching fragment
      if (fragment != null) {
          getSupportFragmentManager()
                  .beginTransaction()
                  .replace(R.id.fragment_container, fragment)
                  .commit();
          return true;
      }
     return false;
  }
```
```

• We will call the above method inside onCreate() to load the default fragment on starting.

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

    //loading the default fragment
    loadFragment(new HomeFragment());

    //getting bottom navigation view and attaching the listener
    BottomNavigationView navigation = findViewById(R.id.navigation);
    navigation.setOnNavigationItemSelectedListener(this);
}
```
```

• In onCreate() we also defined the BottomNavigationView object. We initialized it using findViewById() method and we attached the listener to detect the Navigation Item Selection.
• Now, we need to implement OnNavigationItemSelectedListener interface in the activity class.

```
//implement the interface OnNavigationItemSelectedListener in your activity class
public class MainActivity extends AppCompatActivity implements BottomNavigationView.OnNavigationItemSelectedListener {
```

• With the above interface we will get method, and it will be called whenever we will tap on an option from the Bottom Navigation View.

```
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        return true;
    }
```

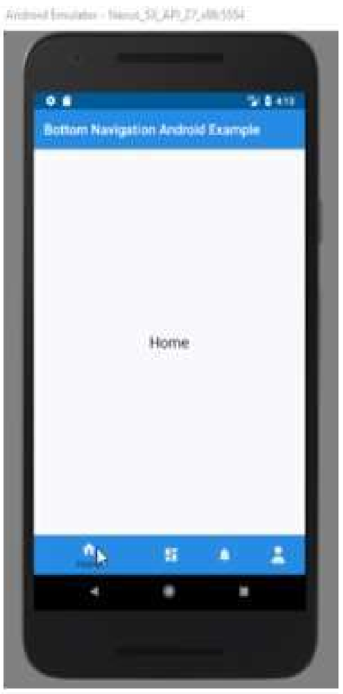
• In the above method we will switch the fragments.

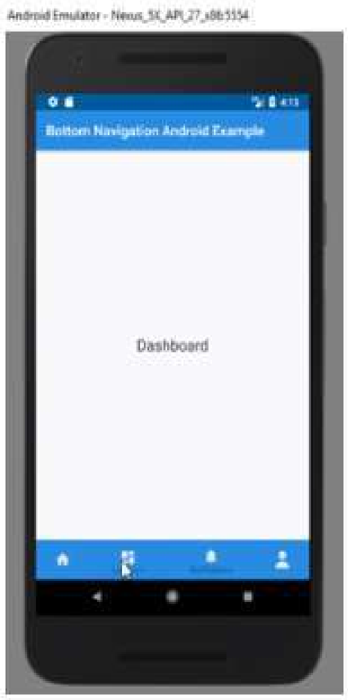
```
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        Fragment fragment = null;

        switch (item.getItemId()) {
            case R.id.navigation_home:
                fragment = new HomeFragment();
                break;

            case R.id.navigation_dashboard:
                fragment = new DashboardFragment();
                break;

            case R.id.navigation_notifications:
                fragment = new NotificationsFragment();
                break;

            case R.id.navigation_profile:
                fragment = new ProfileFragment();
                break;
        }

        return loadFragment(fragment);
    }
```

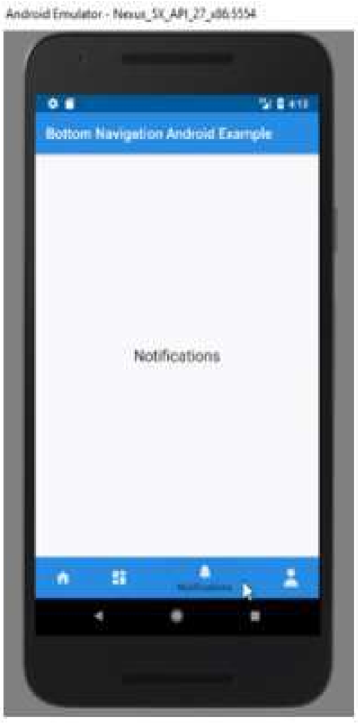
• So, the final code that we have for the MainActivity.java is.

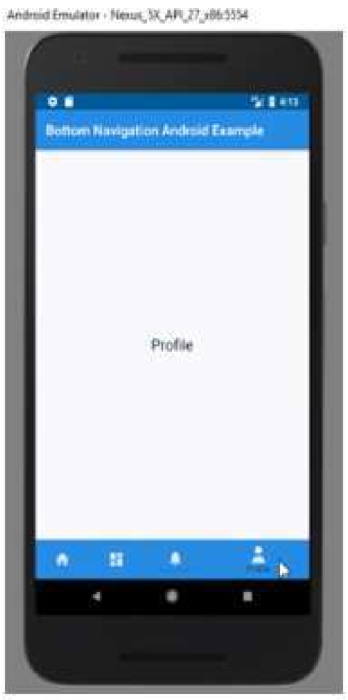
```
package net.simplifiedcoding.bottomnavigationexample;

import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.design.widget.BottomNavigationView;
import android.support.v4.app.Fragment;
import android.support.v7.app.AppCompatActivity;
import android.view.MenuItem;

//implement the interface OnNavigationItemSelectedListener in your activity class
public class MainActivity extends AppCompatActivity implements BottomNavigationView.OnN
avigationItemSelectedListener {


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //loading the default fragment
        loadFragment(new HomeFragment());

        //getting bottom navigation view and attaching the listener
        BottomNavigationView navigation = findViewById(R.id.navigation);
        navigation.setOnNavigationItemSelectedListener(this);
    }


    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        Fragment fragment = null;

        switch (item.getItemId()) {
            case R.id.navigation_home:
                fragment = new HomeFragment();
                break;

            case R.id.navigation_dashboard:
                fragment = new DashboardFragment();
                break;

            case R.id.navigation_notifications:
                fragment = new NotificationsFragment();
                break;

            case R.id.navigation_profile:
                fragment = new ProfileFragment();
                break;
        }

        return loadFragment(fragment);
    }

    private boolean loadFragment(Fragment fragment) {
        //switching fragment
        if (fragment != null) {
            getSupportFragmentManager()
                    .beginTransaction()
                    .replace(R.id.fragment_container, fragment)
                    .commit();
            return true;
        }
        return false;
    }
}
```

• Now, you can try running the application.


Android Emulator - Nexus_5X_API_27_x86:5554
Bottom Navigation Android Example
Home


Android Emulator - Nexus_5X_API_27_x86:5554
Bottom Navigation Android Example
Dashboard


Android Emulator - Nexus_5X_API_27_x86:5554
Bottom Navigation Android Example
Notifications


Android Emulator - Nexus_5X_API_27_x86:5554
Bottom Navigation Android Example
Profile

• As you can see it is working absolutely fine.
• Now you can design your fragments as you want, you can create some complex User Interface there, you can fetch some data from network to display it using a RecyclerView.