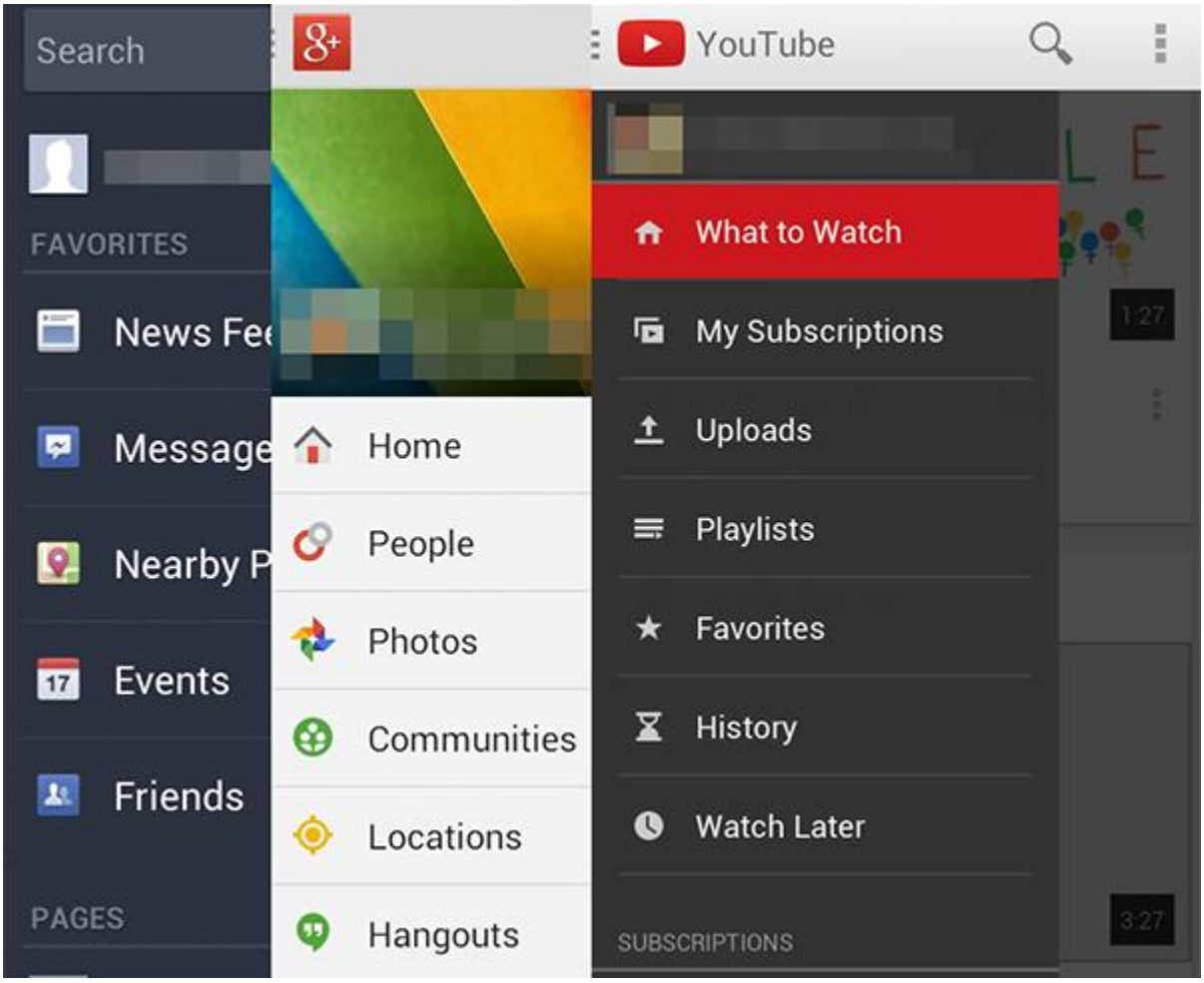
SLIDING MENU

In most of the android applications like Facebook , Google plus ,you tube, amazon kindle , true caller .. etc you have seen a side menu which appears on click of an icon on top left corner or by dragging on to the screen from left to right.

Some examples of sliding menu are as follows:



Following are the series of steps to be followed for coding a sliding menu in your android application.

Step 1 : Create all variables and arrays containing the menu options names & icons in the **strings.xml** file.

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

    <string name="app_name">Sliding Menu</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>
    <string-array name="rivers">
        <item >Brahmaputra</item>
        <item >Ganges</item>
        <item >Kaveri</item>
        <item >Godavari</item>
        <item >Ganga</item>
        <item >Nile </item>
        <item >Yamuna</item>
    </string-array>
    <string name="drawer_open">Open navigation drawer</string>
    <string name="drawer_close">Close navigation drawer</string>


</resources>
```

The navigation drawer is a panel that displays the app's main navigation options on the left edge of the screen. It is hidden most of the time, but is revealed when the user swipes a finger from the left edge of the screen or, while at the top level of the app, the user touches the app icon in the action bar

Step 2 : Create a drawerLayout.
To add a navigation drawer, declare your user interface with a DrawerLayout object as the root view of your layout. Inside the DrawerLayout, add one view that contains the main content for the screen (your primary layout when the drawer is hidden) and another view that contains the contents of the navigation drawer.

For example, the following layout uses a DrawerLayout with two child views: a FrameLayout to contain the main content (populated by a Fragment at runtime), and a ListView for the navigation drawer.

```
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <!-- The main content view -->
    <FrameLayout
        android:id="@+id/content_frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
    <!-- The navigation drawer -->
    <ListView android:id="@+id/drawer_list"
        android:layout_width="240dp"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:choiceMode="singleChoice"
        android:divider="@android:color/transparent"
        android:dividerHeight="0dp"
        android:background="#111"/>
</android.support.v4.widget.DrawerLayout>
```

Step 3: Initialize the Drawer List.
In your activity, one of the first things to do is initialize the navigation drawer's list of items. How you do so depends on the content of your app, but a navigation drawer often consists of a ListView, so the list should be populated by an Adapter (such as ArrayAdapter)

Step 4 : Handle Navigation Click Events.
When the user selects an item in the drawer's list, the system calls onItemClick() on the OnItemClickListener given to setOnItemClickListener().

```
// Setting item click listener for the listview mDrawerList
mDrawerList.setOnItemClickListener(new OnItemClickListener() {

    @Override
    public void onItemClick(AdapterView<?> parent,
                View view,
                int position,
                long id) {

        // Getting an array of rivers
        String[] rivers = getResources().getStringArray(R.array.rivers);

        //Currently selected river
        mTitle = rivers[position];


        // Creating a fragment object
        RiverFragment rFragment = new RiverFragment();

        // Creating a Bundle object
        Bundle data = new Bundle();

        // Setting the index of the currently selected item of mDrawerList
        data.putInt("position", position);

        // Setting the position to the fragment
        rFragment.setArguments(data);

        // Getting reference to the FragmentManager
        FragmentManager fragmentManager  = getFragmentManager();

        // Creating a fragment transaction
        FragmentTransaction ft = fragmentManager.beginTransaction();

        // Adding a fragment to the fragment transaction
        ft.replace(R.id.content_frame, rFragment);

        // Committing the transaction
        ft.commit();

        // Closing the drawer
        mDrawerLayout.closeDrawer(mDrawerList);

    }
});
```

Step 5 : Set a Listener for Open and Close Events of the menu.
To listen for drawer open and close events, call setDrawerListener() on your DrawerLayout and pass it an implementation of DrawerLayout.DrawerListener. This interface provides callbacks for drawer events such as onDrawerOpened() and onDrawerClosed().

However, rather than implementing the DrawerLayout.DrawerListener, if your activity includes the action bar, you can instead extend the ActionBarDrawerToggle class. The ActionBarDrawerToggle implements

DrawerLayout.DrawerListener so you can still override those callbacks, but it also facilitates the proper interaction behavior between the action bar icon and the navigation drawer

Step 6 : Create new xml files each for the sliding menu and another for the fragment that will be shown after any particular item from the menu is clicked.

To understand in a more lucid way please go through the code & follow the comments.

The complete project & .apk is available along with this documentation.

https://drive.google.com/open?id=0B4rCFkKCsCeKQS0tNlpaWE1ZbGc