Design Support Library

基本介紹

程式碼：https://github.com/stone0705/DesignSupportTest

DrawerLayout：

抽屜式的layout 搭配NavigationView一起使用。

NavigationView：

抽屜中的內容，

app:headerLayout設定抽屜的標題

app:menu設定抽屜的選單

android:layout\_gravity=”start”設定抽屜拉出方向，start為左邊拉出，end為右邊拉出。

CoordinatorLayout：

主要負責協調底下view的互動效果。

為主要的配置Layout

AppBarLayout：

AppBar目前必須是CoordinatorLayout中的第一個子View

控制滾動事件(layout\_scrollFlag)

scrollFlag有三種：

1.enterAlways：只要往上滑就會顯示出來

2.enterAlwaysCollapsed：必須拉到最頂部往上滑才會顯示

3.exitUnitCollpsed：當到達最小高度就會消失

ToolBar ：

小標題列，取代actionBar

CollapsingToolbarLayout：

在可見時會是比較大的標題列，可透過註冊過的捲軸類型layout收縮

在它底下的view設定為parallax的在收起來之後會消失，設定為pin的會保留。

TabLayout：

有可以切換內頁的小標籤頁組成的layout

FloatingActionButton：

浮動式的小按鈕，預設的表現是會被snackbar往上推。

SnackBar：

出現在最底部 會推動浮動按鈕，類似於toast的效果，可以做出簡單的訊息通知與點選功能。

TextInputLayout：

裡面要包一個EditView，Hint會再點擊後跑到上面去。

常用的方法有如下：

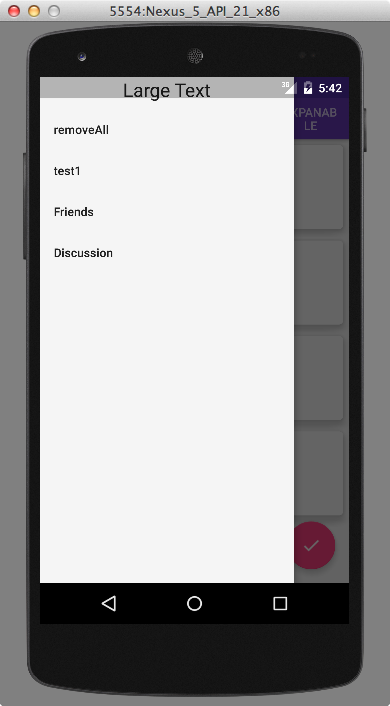
setHint()：設置提示。

getEditText()：得到TextInputLayout中的EditView。

setErrorEnabled():設置是否顯示錯誤訊息。

setError()：設置錯誤訊息內容。

圖文介紹

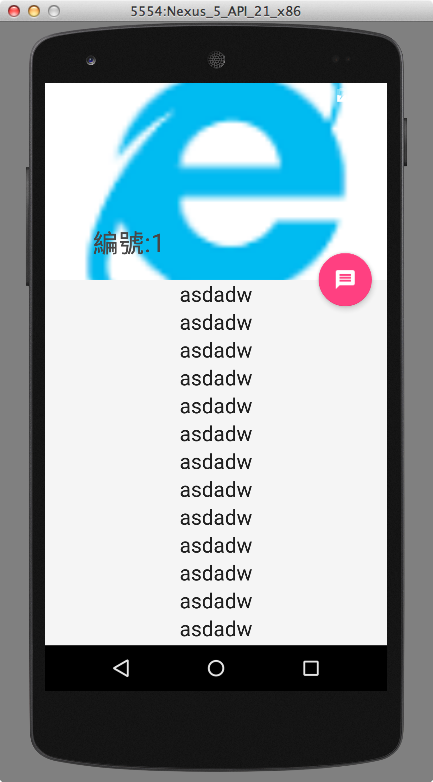


左邊拉出的就是NavigationView，LargeText的位置是header，底下為menu部分。

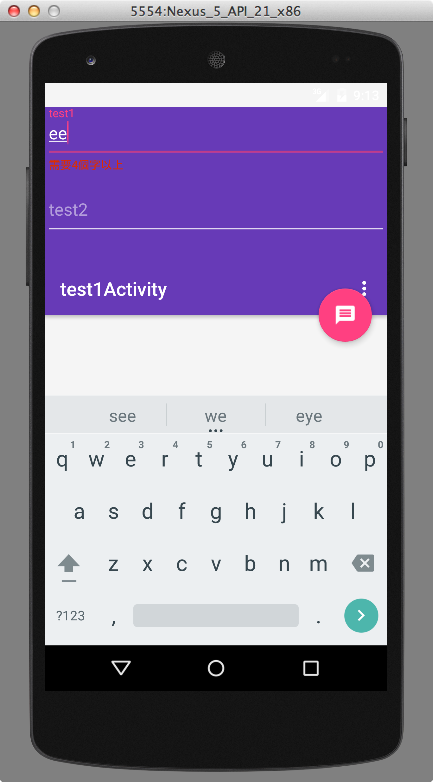
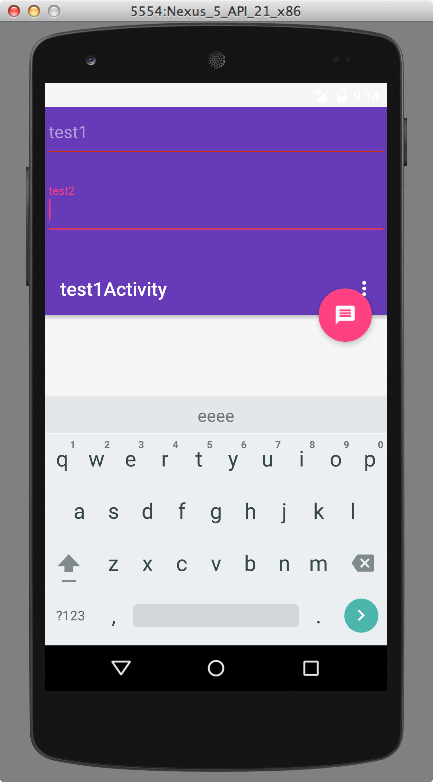


左圖上方部分為一個AppBarLayout，裡面包含一個ToolBar和一個TabLayout，TabLayout底下有一個ViewPager和一個FloatActionButton。

右圖為ToolBar收縮之後的圖片。



左圖上面為一個CollapsingToolbarLayout，包含一個ImageView跟一個ToolBar，往上推之後會收縮成一個ToolBar。

上方2個為TextInputLayout，各包含一個EditText，點選時Hint會漂浮到EditText上方，如果有設置錯誤訊息則會顯示在下方。



圖片下方的test橫條為SnackBar，橫拉之後可以消除。

Demo 程式

主要分為三個頁面

第一個頁面為MainActivity，按下浮動按鈕後會在當下的viewpager增加新的項目，點linear的tab下的項目可以跳到第二個頁面，在MainActivity畫面的左方橫拉可以拉出NavigationView點下test1可以到第三個頁面

讀取的xml檔為main\_layout與content\_layout。

main\_layout主要包含DrawerLayout與NavigationView，NavgationView讀取的header是drawer\_header，讀取的menu是nav\_menu。

content\_layout包含CoordinatorLayoutlinear﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽，AppBarLayout，ToolBar，ToolBar，TabLayout，FloatingActionButton，ViewPager。

讀取的Java檔是MainActivity

在main\_layout主要設置的是關於抽屜的部分，把抽屜的header跟menu還有抽屜從左邊拉或從右邊拉設定好。

content\_layout主要設定layout之間的動作關係，把ToolBar的Flag設定為enterAlwaysCollapsed，讓ToolBar可以收縮，然後ViewPager設定表現為app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"讓他跟AppBarLayout連接起來。

Java檔的部分，首先要先設定ToolBar顯示的內容setSupportActionBar(toolbar);

，然後是把ViewPager底下放進一些Fragment然後把ViewPager放到TabLayout裡面，FloatingActionButton的listener底下放Snackbar.make(v,"test",Snackbar.LENGTH\_LONG).show()，v表示View，它會自動往上搜尋到最高的parentView，中間為顯示的文字，後面為顯示的時間，可以設定為短，長，不會消失，也可以設定Action來處理按下後的反應。

第二個頁面為 DetailActivity目的為展現CollapsingToolbarLayout效果。

讀取的xml檔是acitvity\_Detail主要包含CoordinatorLayoutlinear﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽，AppBarLayout，CollapsingToolbarLayout，ImageView，ToolBar，FloatingActionButton，RecyclerView。

讀取的Java檔為DetailActivity。

acitvity\_Detail主要設定CollapsingToolbarLayout收縮關係，CollapsingToolbarLayout設定app:layout\_scrollFlags="scroll|exitUntilCollapsed"

，在ImageView設定app:layout\_collapseMode="parallax"，表示在收縮的時候ImageView是會被收縮的，ToolBar則設定為pin表示為不會收縮，在FloatingActionButton設定app:layout\_anchor="@id/appbar" 讓他跟AppBarLayout產生連結，在收縮時一起消失，RecyclerView也跟第一個頁面的ViewPager一樣設定關係，讓他在捲軸的時候可以收縮CollapsingToolbarLayout。

Java的部分，也是先設定ToolBar的內容，然後要設定CollapsingToolbarLayout的文字標題setTitle，也可以設定文字顏色等等。

第三個頁面是test1Activity，目的為表現TextInputLayout效果。

讀取的xml檔為acitvity\_test1，包含2個TextInputLayout等等

讀取的Java為test1Activity。

在java檔可以在TextInputLayout上面setHint跟setError，一個是設定hint訊息，一個是設定錯誤訊息，在EditText上面addTextChangedListener來改變錯誤訊息是否顯示。

已知問題

在api23的時候會有一些BUG，拉出抽屜時會崩潰，snackBar橫拉消除時會崩潰，CollapsingToolbarLayout在收縮時點擊會崩潰。

暫時解決方法：

抽屜崩潰問題可以把CoordinatorLayout添加setOnTouchListener裡面直接return true

CollapsingToolbarLayout 可以把CollapsingToolbarLayout添加setOnTouchListener也是return true

snackBar問題無解

 但是在api22的時候不會有上述問題