… J’étais au niveau des prérequis.

* Avoir bien-sûr installé Android Studio
* Disposer d’un smartphone Android muni d’un câble pour les tests

**Créer un projet Android Studio (**link**)**

**Créer une activité Navigation Drawer Activity**

**Créer une activité Tabbed Activity**

**Concevoir la DrawerAndTabbed Activity**

Bien, cela n’a rien de sorcier, nous allons juste mélanger d’une certaine manière les codes générés par Android Studio lors de la création des deux activités.

La Tabbed Activity va se positionner à l’intérieur du contenu de la Navigation Drawer Activity.

Pour ce, nous allons modifier le **app/res/layout/app\_bar\_drawer.xml** de la sorte.

A la racine nous avons le **CoordinatorLayout**, qui a pour premier nœud enfant le **AppBarLayout**, ensuite un **include** et le **FloatingActionButton**.

1. Nous allons modifier la hauteur du **AppBarLayout** comme ceci :

android:layout\_height="100dp"

… et y ajouter le **TabLayout** qui se trouve dans **app/res/layout/activity\_tabbed.xml**

<android.support.design.widget.TabLayout  
 android:id="@+id/tabs"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />

… pour avoir finalement dans le **app\_bar\_drawer.xml**

<android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"  
 android:theme="@style/AppTheme.AppBarOverlay">  
  
 <android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
 <android.support.design.widget.TabLayout  
 android:id="@+id/tabs"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
</android.support.design.widget.AppBarLayout>

1. **content\_drawer.xml**

Après le **AppBarLayout** se trouvant dans le **app\_bar\_drawer.xml**, nous avons un **include** avec comme attribut **layout=’@layout/content\_drawer’.**

Le contenu principal du **Navigation Drawer** est dans ce fichier layout. Nous allons donc y mettre le **ViewPager** qui se trouve dans le **activity\_tabbed.xml**

Voici l’aspect final du **content\_drawer.xml :**

<android.support.v4.view.ViewPager android:id="@+id/container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto" />

1. **DrawerActivity.java** et **TabbedActivity.java**

Dans le **TabbedActivity.java**, Android Studio a généré deux classes **PlaceholderFragment** et **SectionsPagerAdapter**.

Nous allons copier ces deux classes et les coller dans le **DrawerActivity.java**.

Ensuite déclarer deux attributs privés de la classe **DrawerActivity** de type **SectionsPagerAdapter** et **ViewPager.**

private SectionsPagerAdapter mSectionsPagerAdapter;  
private ViewPager mViewPager;

Nous allons ensuite ajouter à la méthode **onCreate** du **DrawerActivity.java** ce bout de code :

mSectionsPagerAdapter = new SectionsPagerAdapter(getSupportFragmentManager());  
mViewPager = (ViewPager) findViewById(R.id.*container*);  
mViewPager.setAdapter(mSectionsPagerAdapter);  
TabLayout tabLayout = (TabLayout) findViewById(R.id.*tabs*);  
tabLayout.setupWithViewPager(mViewPager);

1. **fragment\_tabbed.xml**

Modifiez le fichier de la sorte :

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 tools:context="com.example.phareal.drawerandtabs.DrawerActivity$PlaceholderFragment">  
  
 <TextView  
 android:id="@+id/section\_label"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
  
</RelativeLayout>

Remarquez le changement de la valeur de l’attribut **tools** du **RelativeLayout**.

A partir d’ici nous n’avons plus rien à voir avec le **TabbedActivity.java**, nous allons donc le supprimer et lancer l’application pour un premier test.

Voilà ce que ça donne.

Le contenu textuel des sections se trouve dans le fichier **fragment\_tabbed.xml** et est modifié dynamiquement en fonction de la section sélectionnée.

*“Hello World from section: 2”* quand on sélectionne la section 2, “*Hello World from section: 3 “* quand on sélectionne la section 3, ainsi de suite.

Ce que nous allons faire c’est créer différents fichiers fragment et afficher le contenu en fonction de la section sélectionnée au lieu de modifier le contenu d’un seul fichier.

Créez trois fichiers layout **fragment\_1.xml, fragment\_2.xml** et **fragment\_3.xml**.

Vous pouvez y mettre n’importe quelle view mais dans ce cas nous allons juste afficher du texte.

**fragment\_1.xml**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 tools:context="com.example.phareal.drawerandtabs.DrawerActivity$PlaceholderFragment">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Texte statique de la Section 1"  
 android:textSize="22sp"/>  
  
</RelativeLayout>

**fragment\_2.xml**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 tools:context="com.example.phareal.drawerandtabs.DrawerActivity$PlaceholderFragment">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Texte statique de la Section 2"  
 android:textSize="22sp"/>  
  
</RelativeLayout>

**fragment\_3.xml**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 tools:context="com.example.phareal.drawerandtabs.DrawerActivity$PlaceholderFragment">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Texte statique de la Section 3"  
 android:textSize="22sp"/>  
  
</RelativeLayout>

Maintenant que nous avons nos trois (3) fichiers layout à afficher, rendons-nous dans le **onCreateView** du **PlaceholderFragment (DrawerActivity.java).**

Voici le code actuel :

@Override  
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 View rootView = inflater.inflate(R.layout.*fragment\_1*, container, false);  
 TextView textView = (TextView) rootView.findViewById(R.id.*section\_label*);  
 textView.setText(getString(R.string.*section\_format*, getArguments().getInt(*ARG\_SECTION\_NUMBER*)));  
 return rootView;  
}

Nous allons le modifier pour correspondre à :

@Override  
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 View rootView = inflater.inflate(R.layout.*fragment\_1*, container, false);  
  
 switch (getArguments().getInt(*ARG\_SECTION\_NUMBER*)) {  
 case 1:  
 rootView = inflater.inflate(R.layout.*fragment\_1*, container, false);  
 break;  
 case 2:  
 rootView = inflater.inflate(R.layout.*fragment\_2*, container, false);  
 break;  
 case 3:  
 rootView = inflater.inflate(R.layout.*fragment\_3*, container, false);  
 break;  
 }  
  
 return rootView;  
}

En fonction de la section sélectionnée (getArguments().getInt(*ARG\_SECTION\_NUMBER*)), on inflate le layout correspondant (inflater.inflate(R.layout.*fragment\_\**, container, false)).

Lancez l’application pour constater le résultat et laissez un feedback.

Vous pouvez télécharger le code du projet à partir de github : **link.**

En ajout ! Pour modifier le titre des tabs, rendez-vous dans **getPageTitle** de **SectionsPagerAdapter.**

Merci et rdv pour un autre article, vous pouvez également lire mes articles à propos du Web design, Android Design, Javascript, PHP et NodeJS.

Suivez mon fil sur les différents réseaux sociaux.