

# TabLayout dans Android Studio avec Kotlin

## Introduction

L'interface utilisateur des applications mobiles nécessite souvent une navigation fluide entre plusieurs vues. **TabLayout** et **ViewPager2** permettent de créer une interface avec des onglets (tabs) pour naviguer entre plusieurs fragments.

## 1. Architecture globale

L'architecture de l'application repose sur une structure modulaire comprenant :

1. Activity principale : Elle contient le TabLayout et le ViewPager2.
2. Fragments : Chaque onglet est représenté par un Fragment distinct (ex. : Login et Signup).
3. Adaptateur (ViewPagerAdapter) : Il lie les fragments au ViewPager2 pour afficher les vues correspondantes.

## 2. Structure du projet

```
/res
├── /drawable      # Images et formes graphiques
├── /layout        # Fichiers XML pour Les écrans
│   ├── activity_main.xml # Interface principale avec TabLayout et ViewPager2
│   ├── fragment_login_tab.xml # UI pour l'onglet "Login"
│   └── fragment_signup_tab.xml # UI pour l'onglet "Signup"
├── /values
│   ├── colors.xml # Définition des couleurs
│   └── themes.xml # Thèmes et styles
/src
├── /com/example/tablayment_kotlin
│   ├── MainActivity.kt # Activité principale
│   ├── ViewPagerAdapter.kt # Gestionnaire de fragments pour ViewPager2
│   ├── LoginTabFragment.kt # Fragment pour l'onglet "Login"
│   └── SignupTabFragment.kt # Fragment pour l'onglet "Signup"
```

## 3. Composants graphiques

Voici les principaux éléments graphiques de notre interface :

### 3.1. Colors.xml

On définit les couleurs utilisées dans l'application :

```
<resources>
    <color name="lavender">#8692f7</color>
    <color name="white">#FFFFFF</color>
</resources>
```

### 3.2. Themes.xml

On applique les couleurs aux thèmes de l'application :

```
<resources xmlns:tools="http://schemas.android.com/tools">
  <style name="Theme.MyApp"
    parent="Theme.MaterialComponents.DayNight.NoActionBar">
    <item name="colorPrimary">@color/lavender</item>
    <item name="colorOnPrimary">@color/white</item>
  </style>
</resources>
```

### 3.3. Activity\_main.xml

Ce fichier définit l'interface principale avec TabLayout et ViewPager2.

```
<!-- TabLayout -->
<com.google.android.material.tabs.TabLayout
  android:id="@+id/tab_layout"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"/>

<!-- ViewPager2 -->
<androidx.viewpager2.widget.ViewPager2
  android:id="@+id/view_pager"
  android:layout_width="match_parent"
  android:layout_height="match_parent"/>
```

### 3.4. Fragments (Exemple : SignupTabFragment)

Chaque fragment correspond à un onglet :

```
class SignupTabFragment : Fragment() {

  override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
  ): View? {
    return inflater.inflate(R.layout.fragment_signup_tab, container, false)
  }
}
```

## 4. Fonctionnement global

Voici les différentes étapes du fonctionnement de TabLayout avec ViewPager2 :

### 1. Initialisation de l'interface principale

- Création d'un TabLayout pour afficher les onglets (ex. : "Login" et "Signup").
- Ajout d'un ViewPager2 pour permettre le défilement entre les fragments.

### 2. Création et gestion des fragments

- Définition des fragments LoginTabFragment et SignupTabFragment.
- Chaque fragment contient son propre contenu et interface utilisateur.

### 3. Création d'un adaptateur (ViewPagerAdapter)

- Développement d'un adaptateur qui associe les fragments à ViewPager2.
- Cet adaptateur gère le nombre d'onglets et leur affichage dynamique.

### 4. Association du ViewPager2 avec le TabLayout

- Attribution de l'adaptateur au ViewPager2.
- Ajout des onglets dans le TabLayout avec les titres correspondants.

### 5. Synchronisation de la navigation

- Lorsqu'un onglet est sélectionné dans le TabLayout, le ViewPager2 affiche le fragment correspondant.
- Inversement, lors du balayage horizontal dans ViewPager2, l'onglet actif change automatiquement.

### 6. Améliorations possibles

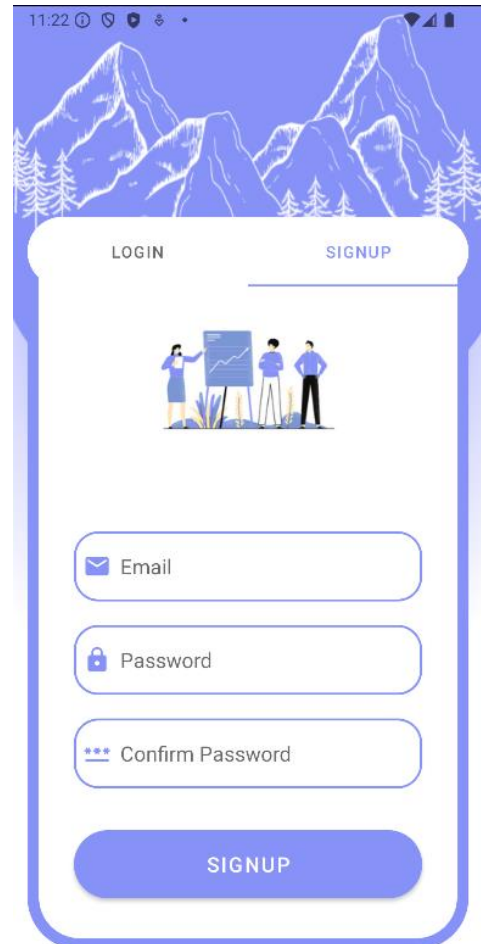
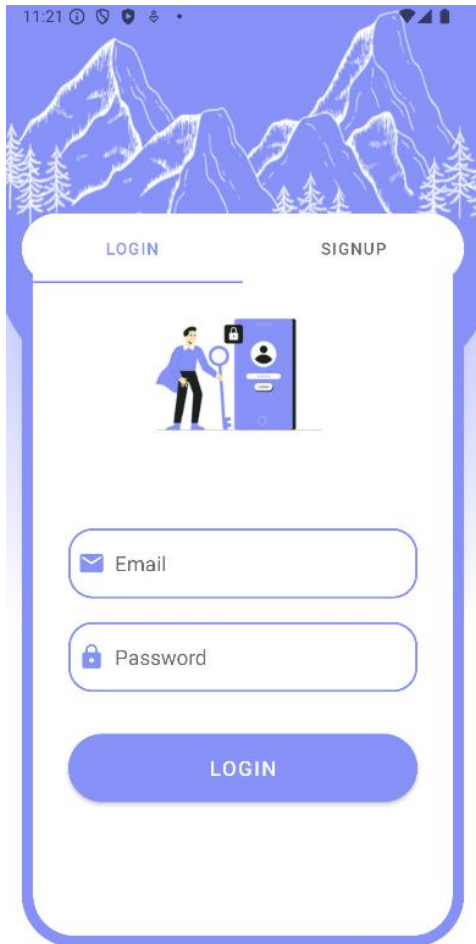
- Personnalisation des icônes et couleurs des onglets.
- Ajout de nouvelles pages ou fonctionnalités (ex. : connexion avec Firebase/SQLite).
- Gestion des transitions et animations pour une meilleure expérience utilisateur.

Avec ce fonctionnement, l'utilisateur peut naviguer facilement entre les différentes sections de l'application.

# TP TabLayout sous Android :

## Guide Complet

**Objectif** : Créer une interface utilisateur avec deux onglets (**Login et Signup**), en utilisant **TabLayout** et **ViewPager2** sous Android en Kotlin.



### 1. Configuration des couleurs et du thème

Modifier **colors.xml** (**res/values/colors.xml**)

Ajoute les couleurs personnalisées qui seront utilisées dans l'application :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="lavender">#8692f7</color>
</resources>
```

### Modifier themes.xml (res/values/themes.xml)

Configure le thème de l'application en utilisant lavender comme couleur principale :




```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->

    <style name="Theme.TabLayout_Kotlin" parent="Theme.MaterialComponents.DayNight.NoActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/lavender</item>
        <item name="colorPrimaryVariant">@color/lavender</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">?attr/colorPrimaryVariant</item>

    </style>
</resources>
```

## 2. Ajout des icônes Vector Asset

Dossier : res/drawable

- Étape : Clic droit → New → Vector Asset
- Icônes à ajouter :
  - mail 
  - lock 
  - password 
- Code couleur : #8692F7

## 3. Ajout des fichiers XML pour le style des composants

### Fichier tab\_bkg.xml (res/drawable/tab\_bkg.xml)

Personnalise l'arrière-plan des onglets :

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid
        android:color="@color/white"/>
    <corners
        android:radius="40dp"/>
</shape>
```

### Fichier viewPager\_bkg.xml (res/drawable/viewpager\_bkg.xml)

Personnalise l'arrière-plan du ViewPager2 :

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid
        android:color="@color/white"/>
    <corners
        android:radius="40dp"/>
    <stroke
        android:color="@color/lavender"
        android:width="10dp"/>
</shape>
```

## 4. Interface utilisateur dans activity\_main.xml

### Modifier activity\_main.xml (res/layout/activity\_main.xml)

```
<?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"
    android:background="@drawable/mainbkg"
    tools:context=".MainActivity">
    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="380dp"
        android:layout_height="0dp"
```

```

        android:layout_marginBottom="20dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHeight_percent=".78"
        app:layout_constraintVertical_bias="1">
<androidx.viewpager2.widget.ViewPager2
    android:layout_width="match_parent"
    android:layout_height="630dp"
    android:id="@+id/view_pager"
    app:layout_constraintVertical_bias="0"
    android:background="@drawable/viewpager_bkg"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"/>
<com.google.android.material.tabs.TabLayout
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:id="@+id/tab_layout"
    app:layout_constraintVertical_bias="0"
    android:background="@drawable/tab_bkg"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

## 5. Création des Fragments

### Créer SignupTabFragment.kt

```

class SignupTabFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_signup_tab, container, attachToRoot: false)
    }
}

```

## Créer fragment\_signup\_tab.xml Ajoute un formulaire d'inscription :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".SignupTabFragment">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_marginTop="20dp"
            android:padding="20dp">
            <ImageView
                android:layout_width="wrap_content"
                android:layout_height="200dp"
                android:src="@drawable/signimg"/>
            <EditText
                android:layout_width="300dp"
                android:layout_height="60dp"
                android:layout_gravity="center"
                android:id="@+id/signup_email"
                android:layout_marginTop="30dp"
                android:padding="8dp"
                android:hint="Email"
                android:drawableLeft="@drawable/baseline_email_24"
                android:drawablePadding="8dp"
                android:background="@drawable/edittext_bkg"/>
            <EditText
                android:layout_width="300dp"
                android:layout_height="60dp"
                android:layout_gravity="center"
                android:id="@+id/signup_password"
                android:layout_marginTop="20dp"
                android:padding="8dp"
                android:hint="Password"
                android:inputType="textPassword"
                android:drawableLeft="@drawable/baseline_lock_24"
                android:drawablePadding="8dp"
                android:background="@drawable/edittext_bkg"/>
        </LinearLayout>
    </LinearLayout>
</LinearLayout>
```



```

<EditText
    android:layout_width="300dp"
    android:layout_height="60dp"
    android:layout_gravity="center"
    android:id="@+id/signup_confirm"
    android:layout_marginTop="20dp"
    android:padding="8dp"
    android:hint="Confirm Password"
    android:inputType="textPassword"
    android:drawableLeft="@drawable/baseline_password_24"
    android:drawablePadding="8dp"
    android:background="@drawable/edittext_bkg"/>
<Button
    android:layout_width="300dp"
    android:layout_height="70dp"
    android:layout_gravity="center"
    android:layout_marginTop="30dp"
    android:id="@+id/signup_button"
    android:textSize="18sp"
    android:text="Signup"
    app:cornerRadius="30dp"
/>
</LinearLayout>
</LinearLayout>
</LinearLayout>

```

### Créer LoginTabFragment.kt

```

class LoginTabFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_login_tab, container, attachToRoot: false)
    }
}

```

### Créer fragment\_login\_tab.xml Ajoute un formulaire de connexion :

```

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

```

```
android:orientation="vertical"
tools:context=".LoginTabFragment">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_marginTop="20dp"
        android:padding="20dp">
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="200dp"
            android:src="@drawable/loginimg"/>
        <EditText
            android:layout_width="300dp"
            android:layout_height="60dp"
            android:layout_gravity="center"
            android:id="@+id/login_email"
            android:layout_marginTop="30dp"
            android:padding="8dp"
            android:hint="Email"
            android:drawableLeft="@drawable/baseline_email_24"
            android:drawablePadding="8dp"
            android:background="@drawable/edittext_bkg"/>
        <EditText
            android:layout_width="300dp"
            android:layout_height="60dp"
            android:layout_gravity="center"
            android:id="@+id/login_password"
            android:layout_marginTop="20dp"
            android:padding="8dp"
            android:hint="Password"
            android:inputType="textPassword"
            android:drawableLeft="@drawable/baseline_lock_24"
            android:drawablePadding="8dp"
            android:background="@drawable/edittext_bkg"/>

    <Button
        android:layout_width="300dp"
        android:layout_height="70dp"
        android:layout_gravity="center"
        android:layout_marginTop="30dp"
        android:id="@+id/signup_button"
        android:textSize="18sp"
        android:text="Login"
```

```
        app:cornerRadius = 30dp" />
    </LinearLayout>
</LinearLayout>
```

## 6. Création de l'Adapter pour ViewPager2

Créer ViewPagerAdapter.kt

```
package com.example.signlogintab

import androidx.fragment.app.Fragment
import androidx.fragment.app.FragmentManager
import androidx.lifecycle.Lifecycle
import androidx.viewpager2.adapter.FragmentStateAdapter
import com.example.tablayout_kotlin.SignupTabFragment
import com.example.tablayout_kotlin.LoginTabFragment
class ViewPagerAdapter(
    fragmentManager: FragmentManager,
    lifecycle: Lifecycle
) : FragmentStateAdapter(fragmentManager, lifecycle) {

    override fun getItemCount(): Int = 2

    override fun createFragment(position: Int): Fragment {
        return if (position == 1) {
            SignupTabFragment()
        } else {
            LoginTabFragment()
        }
    }
}
```

## 7.Configuration de MainActivity.kt

```
import com.example.signlogintab.ViewPagerAdapter

class MainActivity : AppCompatActivity() {

    private lateinit var tabLayout: TabLayout
    private lateinit var viewPager2: ViewPager2
    private lateinit var adapter: FragmentStateAdapter

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        tabLayout = findViewById(R.id.tab_layout)
        viewPager2 = findViewById(R.id.view_pager)
        tabLayout.addTab(tabLayout.newTab().setText("Login"))
        tabLayout.addTab(tabLayout.newTab().setText("Signup"))
        val fragmentManager: FragmentManager = supportFragmentManager
        adapter = ViewPagerAdapter(fragmentManager, lifecycle)
        viewPager2.adapter = adapter
        tabLayout.addOnTabSelectedListener(object : TabLayout.OnTabSelectedListener {
            override fun onTabSelected(tab: TabLayout.Tab?) {
                tab?.let { viewPager2.currentItem = it.position }
            }
            override fun onTabUnselected(tab: TabLayout.Tab?) {}
            override fun onTabReselected(tab: TabLayout.Tab?) {}
        })
        viewPager2.registerOnPageChangeCallback(object : ViewPager2.OnPageChangeCallback() {
            override fun onPageSelected(position: Int) {
                tabLayout.selectTab(tabLayout.getTabAt(position))
            }
        })
    }
}
```