

Design et ergonomie

Adrian Bracigliano

Objectifs

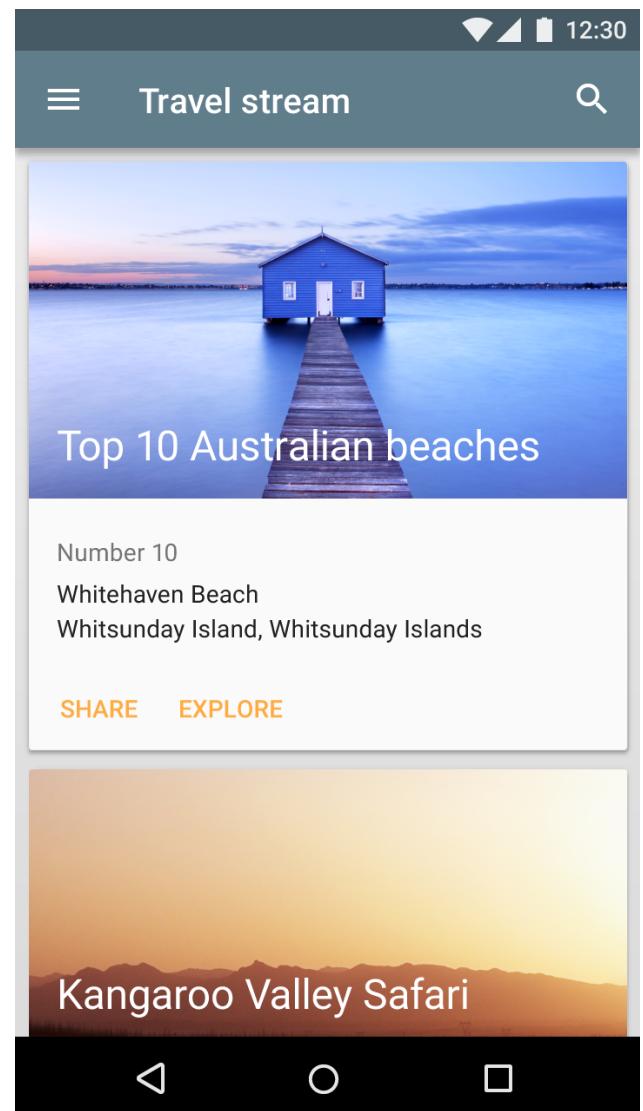
- Se baser sur le Material design pour concevoir des écrans

Material Design

- Introduit par Google en 2014
- Ensemble de règles et de bonnes pratiques pour concevoir des écrans (Application Mobile, site Web, etc...)
- Utilisé sur Android à partir de la version 5.0

Material Design : Quelques principes

- Les différents blocs constituant l'interface sont comme des feuilles de papier superposées les unes au dessus des autres



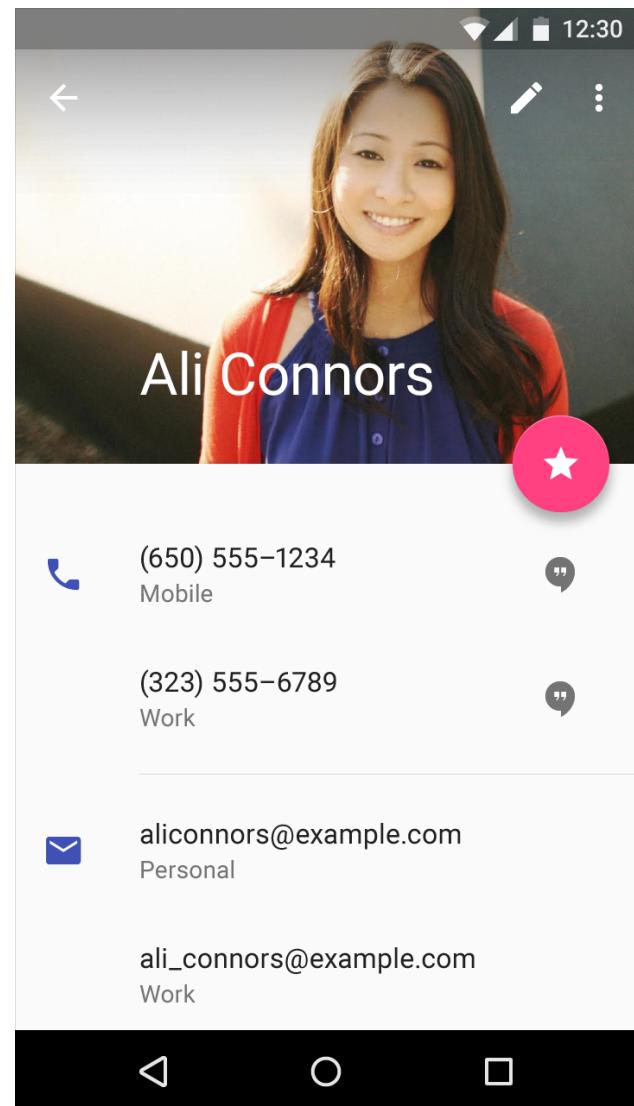
Material Design : Quelques principes

- Les différents blocs constituant l'interface sont comme des feuilles de papier superposées les unes au dessus des autres
- Les animations ont une place importante dans l'interface et ont un véritable intérêt



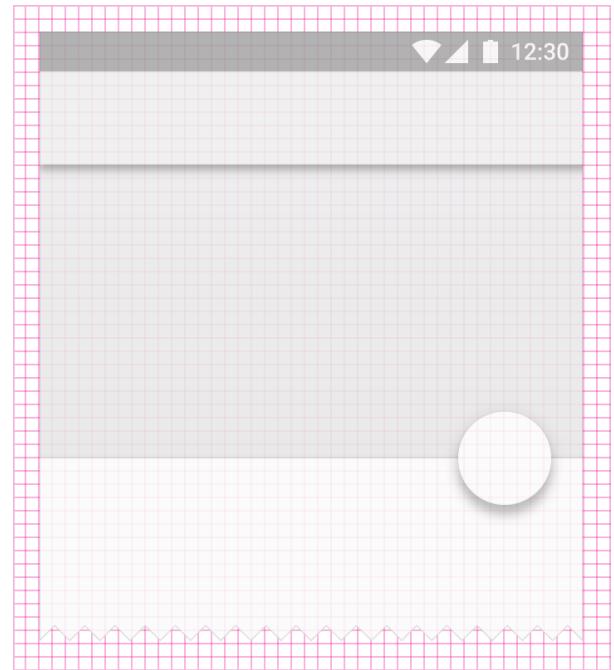
Material Design : Quelques principes

- Les différents blocs constituant l'interface sont comme des feuilles de papier superposées les unes au dessus des autres
- Les animations ont une place importante dans l'interface et ont un véritable intérêt
- Les images sont au centre de l'interface



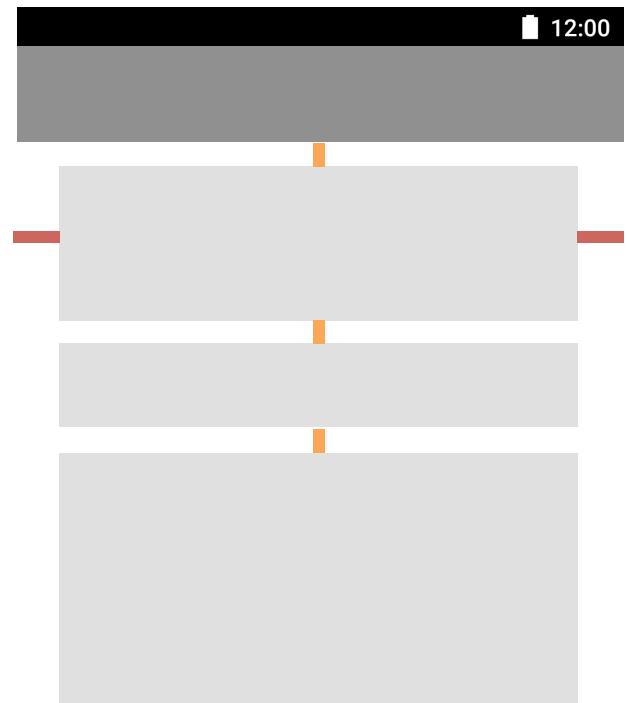
Material Design : Espacement

- L'espacement de chaque élément d'un écran est basé sur une grille de 4dp



Material Design : Espacement

- L'espacement de chaque élément d'un écran est basé sur une grille de 4dp
- L'espace horizontal de chaque côté d'un écran est de 16dp
- L'espace vertical entre chaque bloc d'éléments est de 8dp



Ressources Material Design : Icônes

<https://material.io/icons/>

Search

Action



3d rotation



accessibility



accessible



account balance



account balance ...



account box



account circle



add shopping cart



alarm



alarm add



alarm off



alarm on



all out



android



announcement



aspect ratio



assessment



assignment



assignment ind



assignment late



assignment return



assignment return...



assignment turne...



autorenew



backup



book



bookmark



bookmark border



bug report



build



cached



camera enhance



card giftcard



card membership



card travel



change history



check circle



chrome reader mo...



class



code



compare arrows



copyright



credit card



dashboard



date range

Ressources Material Design : Couleurs

<https://material.io/guidelines/style/color.html>

≡ Style – Color



Red	
500	#F44336
50	#FFEBEE
100	#FFCDD2
200	#EF9A9A
300	#E57373
400	#EF5350
500	#F44336
600	#E53935
700	#D32F2F
800	#C62828
900	#B71C1C
A100	#FF8A80
A200	#FF5252

Pink	
500	#E91E63
50	#FCE4EC
100	#F8BBDO
200	#F48FB1
300	#F06292
400	#EC407A
500	#E91E63
600	#D81B60
700	#C2185B
800	#AD1457
900	#880E4F
A100	#FF80AB
A200	#FF4081

Purple	
500	#9C27B0
50	#F3E5F5
100	#E1BEE7
200	#CE93D8
300	#BA68C8
400	#AB47BC
500	#9C27B0
600	#8E24AA
700	#7B1FA2
800	#6A1B9A
900	#4A148C
A100	#EA80FC
A200	#E040FB

Ressources Material Design : Typographie

<https://material.io/guidelines/style/typography.html>

English and English-like scripts

Latin, Greek, and Cyrillic.

The basic set of styles are based on a typographic scale of 12, 14, 16, 20, and 34.

Display 4

Display 3

Display 2

Display 1

Headline

Title

Subheading

Body 2

Body 1

Caption

Button

Light 112sp

Regular 56sp

Regular 45sp

Regular 34sp

Regular 24sp

Medium 20sp

Regular 16sp (Device), Regular 15sp (Desktop)

Medium 14sp (Device), Medium 13sp (Desktop)

Regular 14sp (Device), Regular 13sp (Desktop)

Regular 12sp

MEDIUM (ALL CAPS) 14sp

Material Design :

Zoom sur quelques composants

- Tabs
- Navigation Drawer
- Boutons
- App bar
- Thème

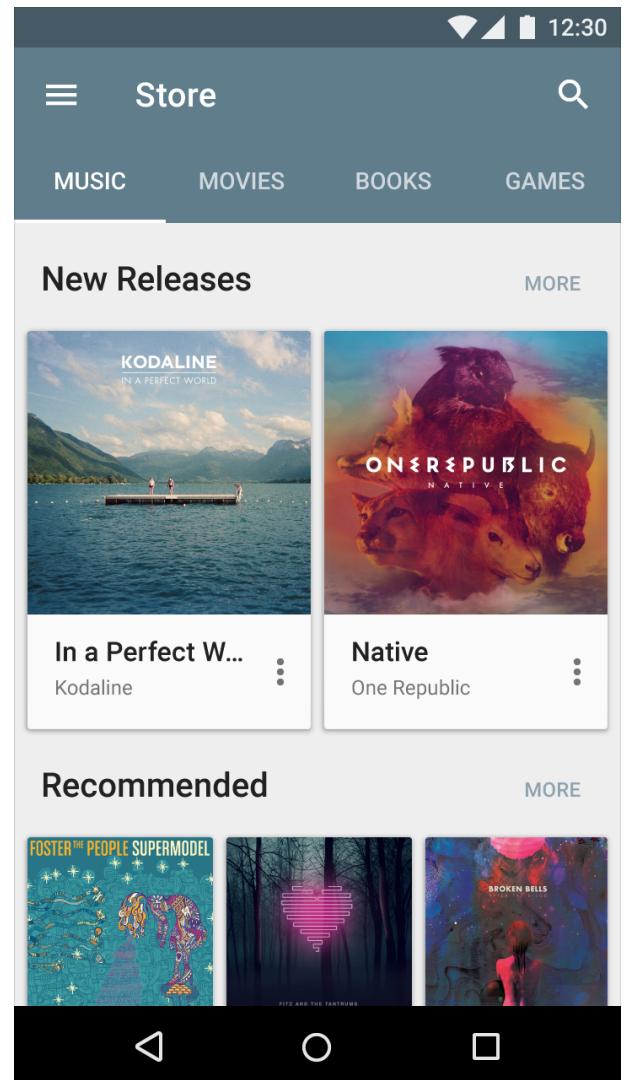
Tabs

- Permet de naviguer horizontalement entre plusieurs écrans d'importance égale



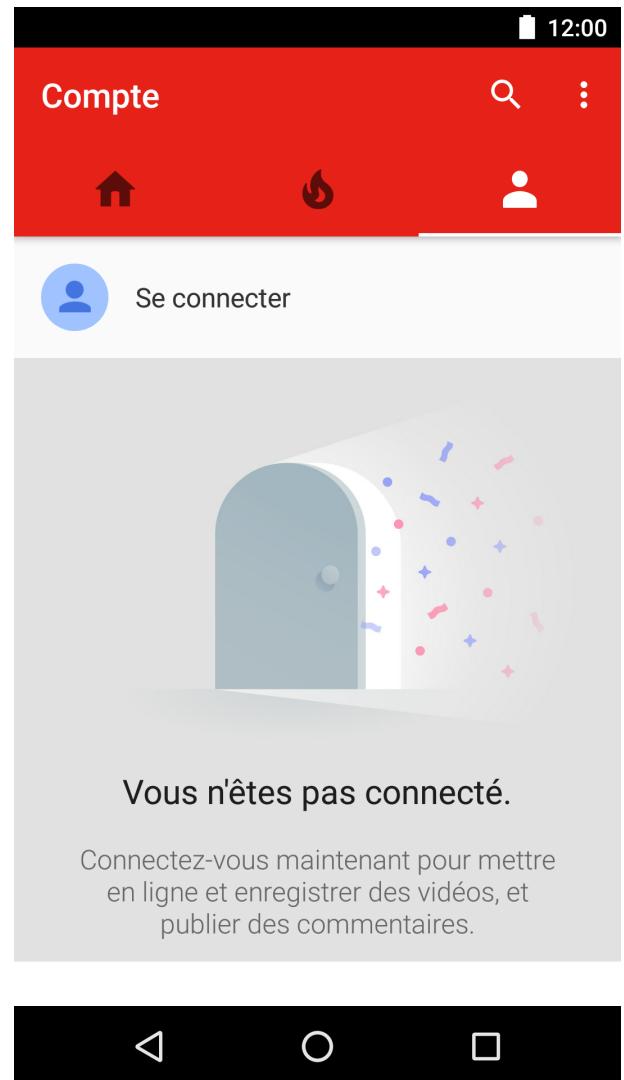
Tabs

- Permet de naviguer horizontalement entre plusieurs écrans d'importance égale
- Pour afficher du contenu pouvant être découpé en plusieurs sous-catégories :
 - Le titre de l'app bar représente la catégorie et chaque tab représente une sous-catégorie



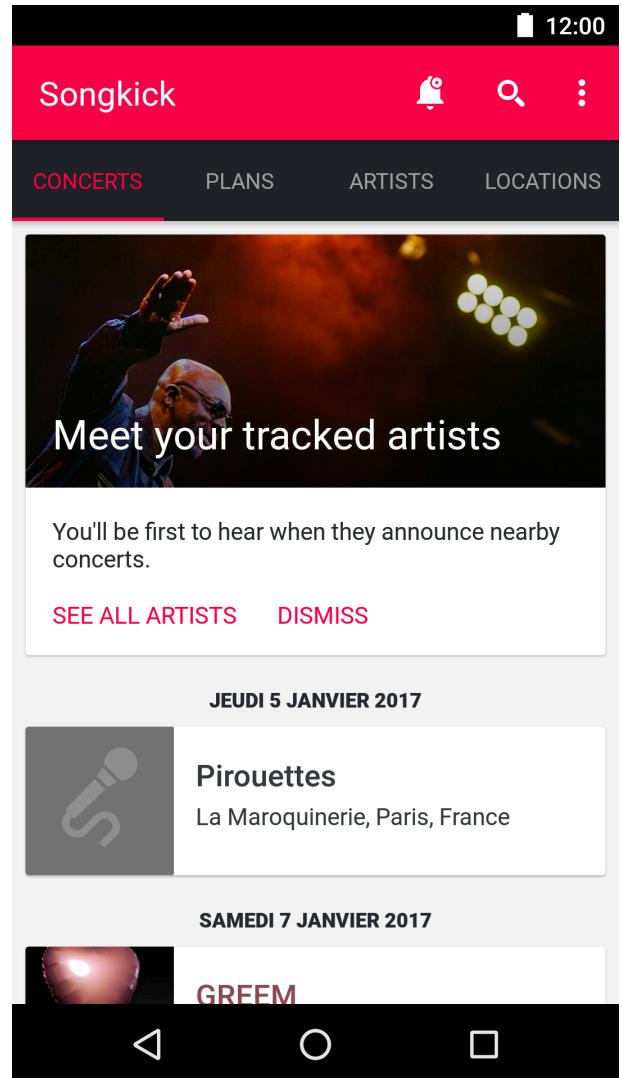
Tabs

- Permet de naviguer horizontalement entre plusieurs écrans d'importance égale
- Pour afficher du contenu pouvant être découpé en plusieurs sous-catégories :
 - Le titre de l'app bar représente la catégorie et chaque tab représente une sous-catégorie
- Pour une application avec quelques écrans en premier niveau de hiérarchie



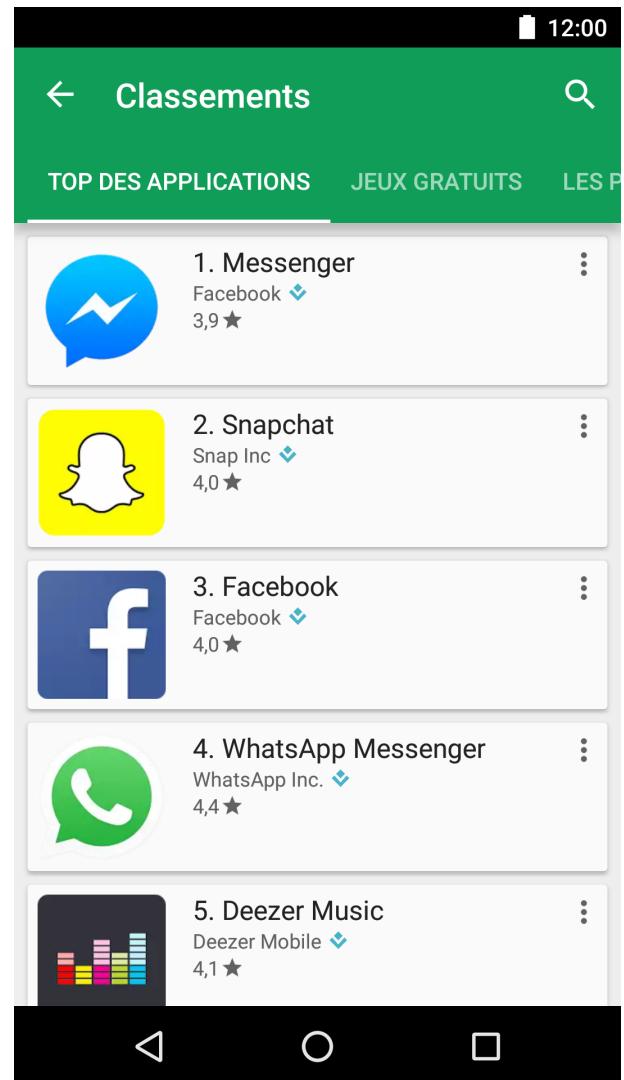
Tabs

- Style des tabs :
 - Taille fixe



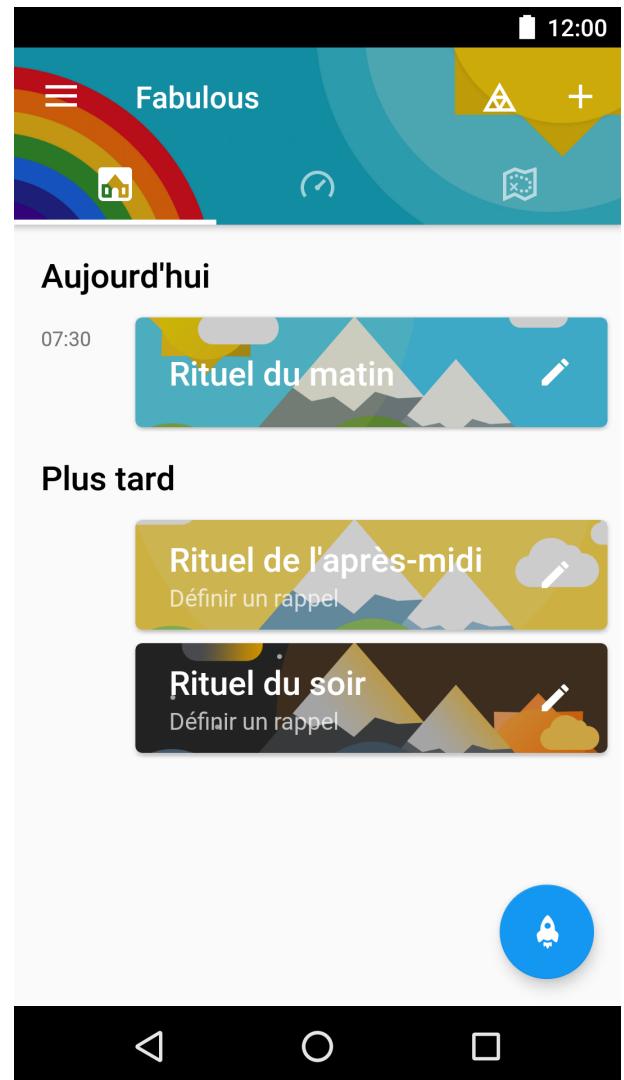
Tabs

- Style des tabs :
 - Taille fixe
 - Taille variable



Tabs

- Style des tabs :
 - Taille fixe
 - Taille variable
 - Icônes seulement



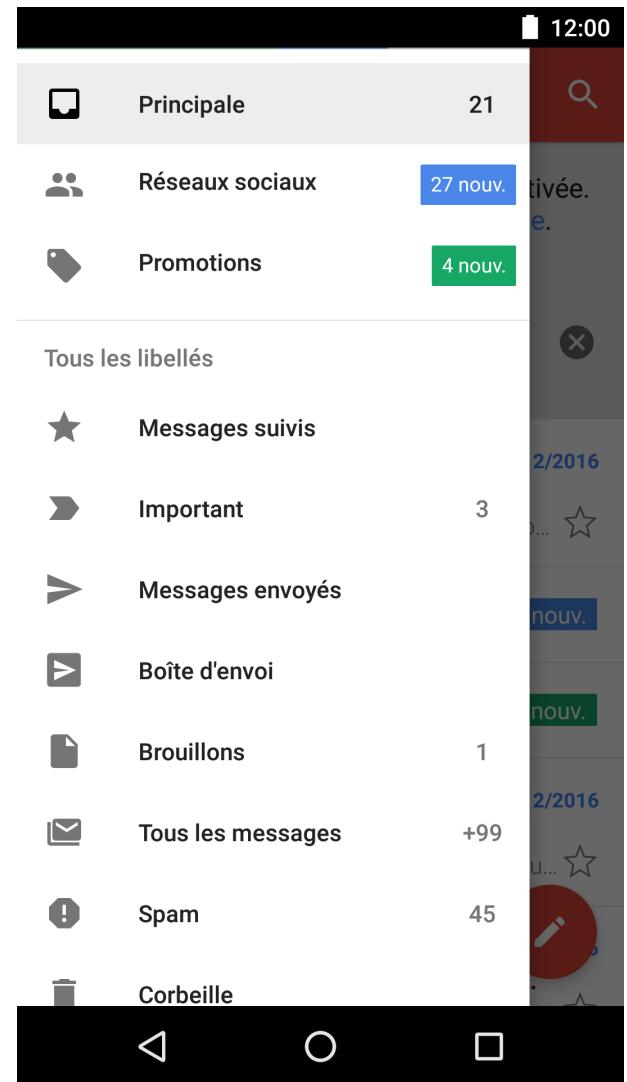
Navigation drawer

- Ecran de navigation constitué d'un header et d'une liste d'entrées s'affichant par la gauche au clic sur sur icône menu



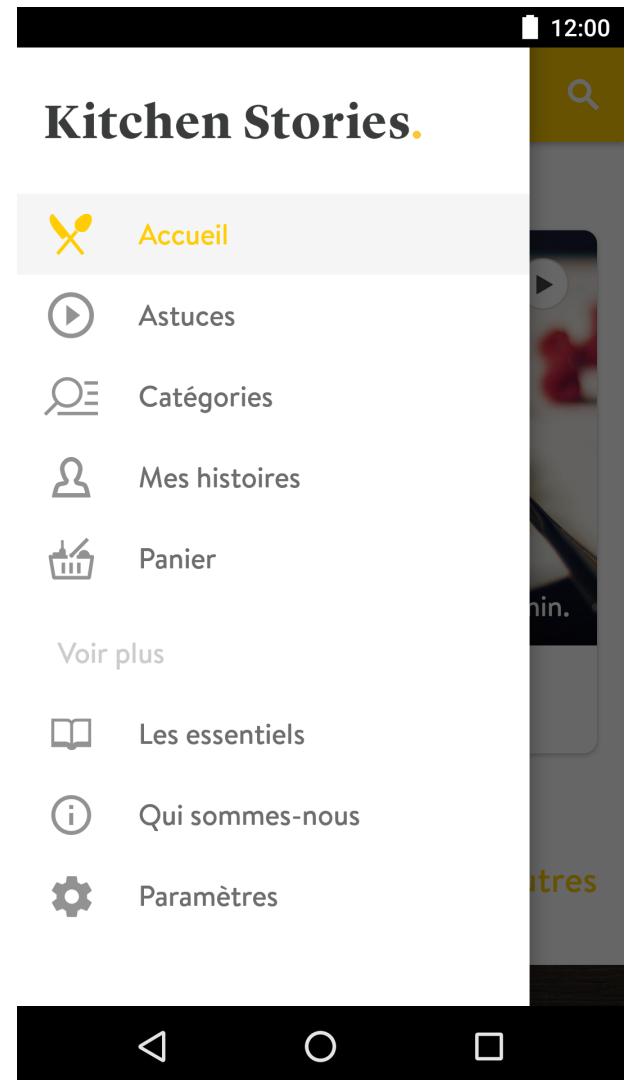
Navigation drawer

- Ecran de navigation constitué d'un header et d'une liste d'entrées s'affichant par la gauche au clic sur sur icône menu
- Pour pouvoir naviguer rapidement entre des écrans qui n'ont aucune relation les uns avec les autres



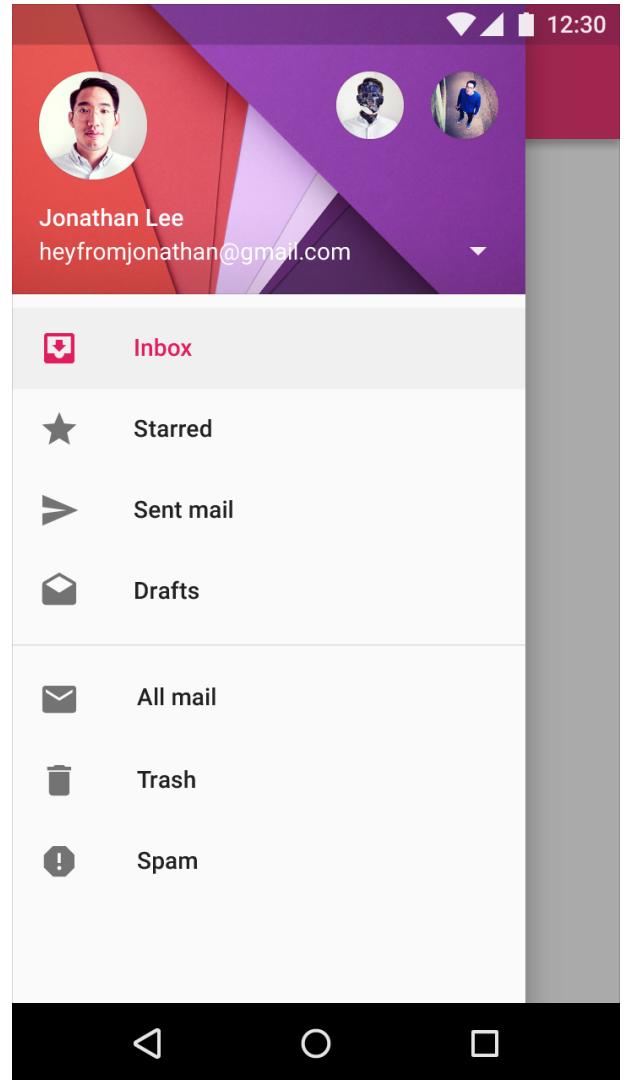
Navigation drawer

- Ecran de navigation constitué d'un header et d'une liste d'entrées s'affichant par la gauche au clic sur sur icône menu
- Pour pouvoir naviguer rapidement entre des écrans qui n'ont aucune relation les uns avec les autres
- Pour une application possédant beaucoup d'écrans au premier niveau de hierarchie



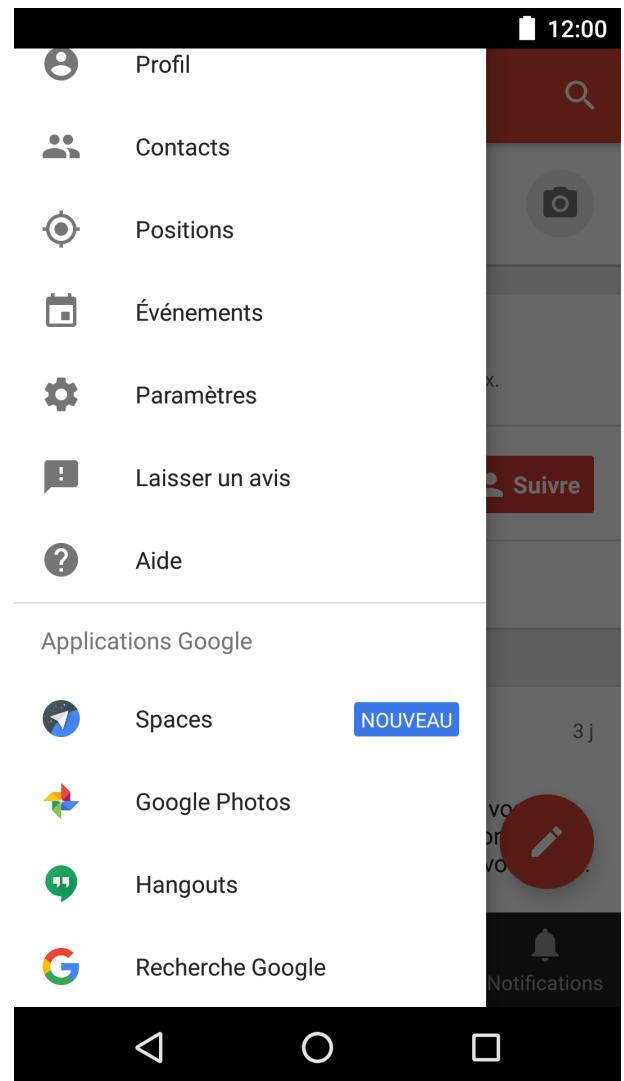
Navigation drawer

- Ecran de navigation constitué d'un header et d'une liste d'entrées s'affichant par la gauche au clic sur sur icône menu
- Pour pouvoir naviguer rapidement entre des écrans qui n'ont aucune relation les uns avec les autres
- Pour une application possédant beaucoup d'écrans au premier niveau de hiérarchie
- Autres intérêts :
 - Gestion d'un profil utilisateur



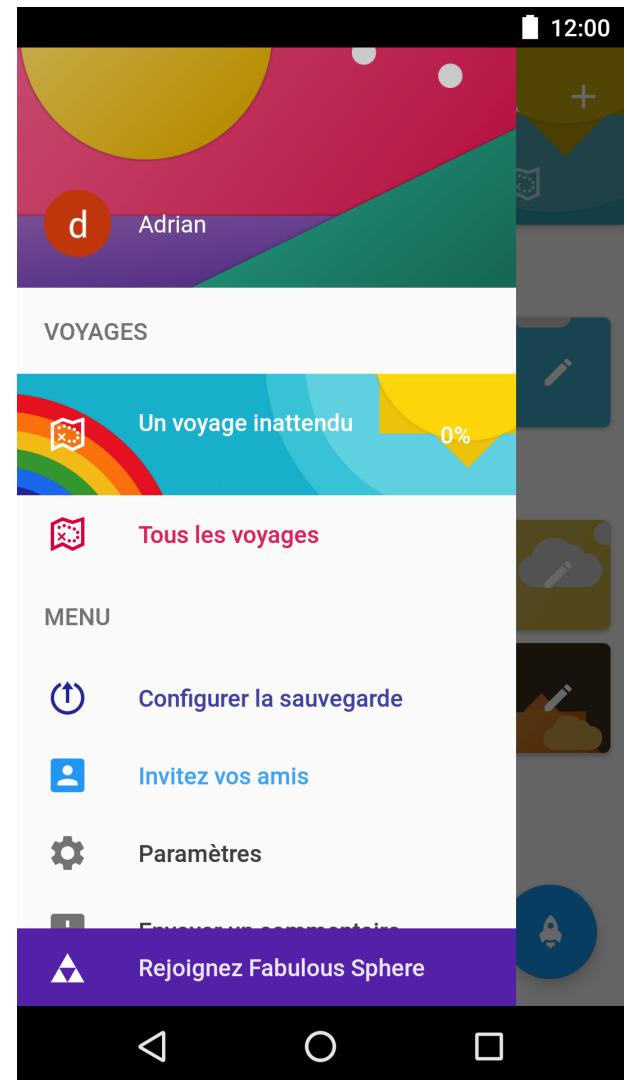
Navigation drawer

- Ecran de navigation constitué d'un header et d'une liste d'entrées s'affichant par la gauche au clic sur sur icône menu
- Pour pouvoir naviguer rapidement entre des écrans qui n'ont aucune relation les uns avec les autres
- Pour une application possédant beaucoup d'écrans au premier niveau de hiérarchie
- Autres intérêts :
 - Gestion d'un profil utilisateur
 - Liens vers d'autres applications



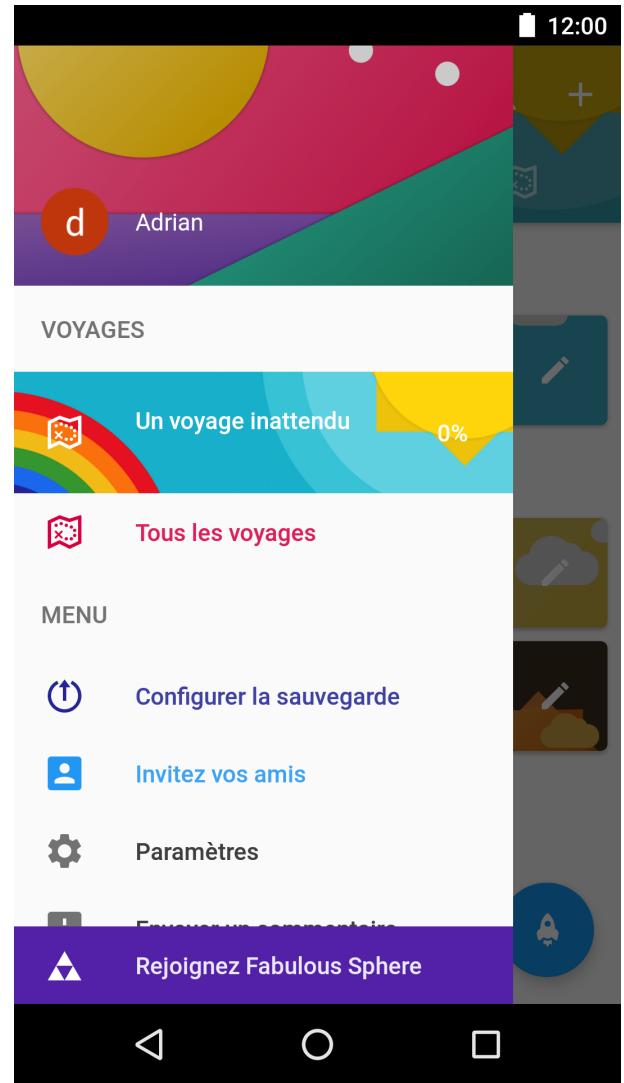
Navigation drawer

- Ecran de navigation constitué d'un header et d'une liste d'entrées s'affichant par la gauche au clic sur sur icône menu
- Pour pouvoir naviguer rapidement entre des écrans qui n'ont aucune relation les uns avec les autres
- Pour une application possédant beaucoup d'écrans au premier niveau de hiérarchie
- Autres intérêts :
 - Gestion d'un profil utilisateur
 - Liens vers d'autres applications
 - Promotion d'une offre premium



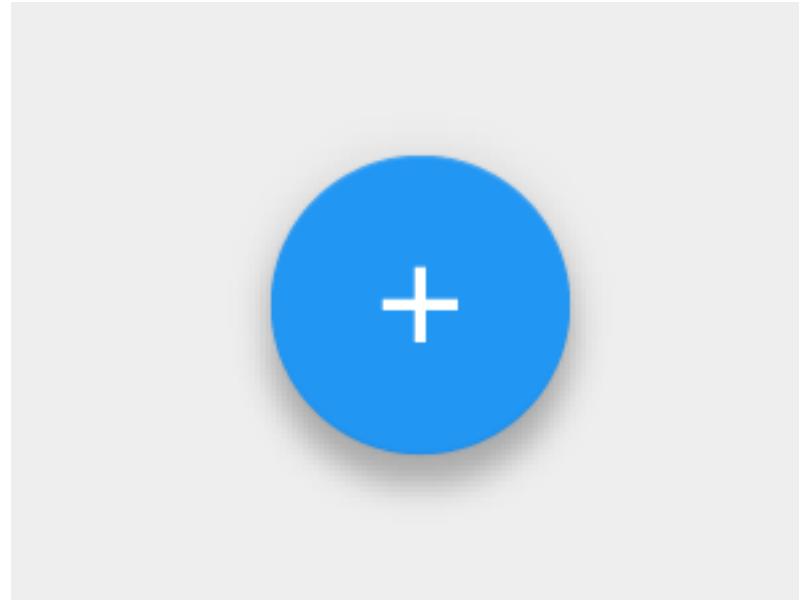
Navigation drawer

- Ecran de navigation constitué d'un header et d'une liste d'entrées s'affichant par la gauche au clic sur l'icône menu
- Pour pouvoir naviguer rapidement entre des écrans qui n'ont aucune relation les uns avec les autres
- Pour une application possédant beaucoup d'écrans au premier niveau de hiérarchie
- Autres intérêts :
 - Gestion d'un profil utilisateur
 - Liens vers d'autres applications
 - Promotion d'une offre premium
- Si possible, toujours préférer un système de navigation plus simple (Mauvaises accessibilités, l'information est cachée par défaut)



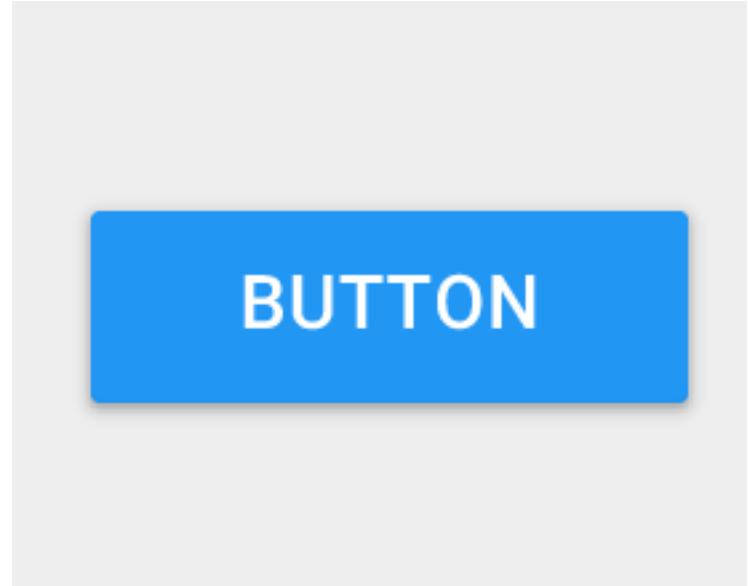
Les boutons

- Floating Action Button



Les actions

- Floating Action Button
- Raised Button



Les actions

- Floating Action Button
- Raised Button
- Flat Button



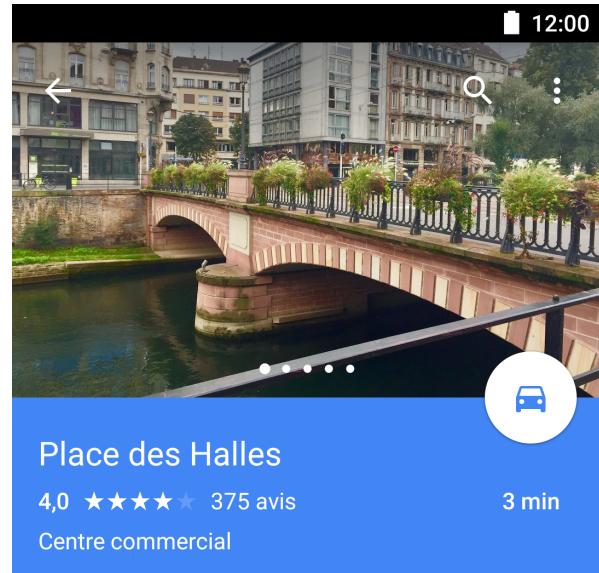
Floating Action Button (fab)

- Bouton d'action situé dans le coin bad droit d'un écran ou au niveau d'une app bar étendue



Floating Action Button (fab)

- Bouton d'action situé dans le coin bas droit d'un écran ou au niveau d'une app bar étendue



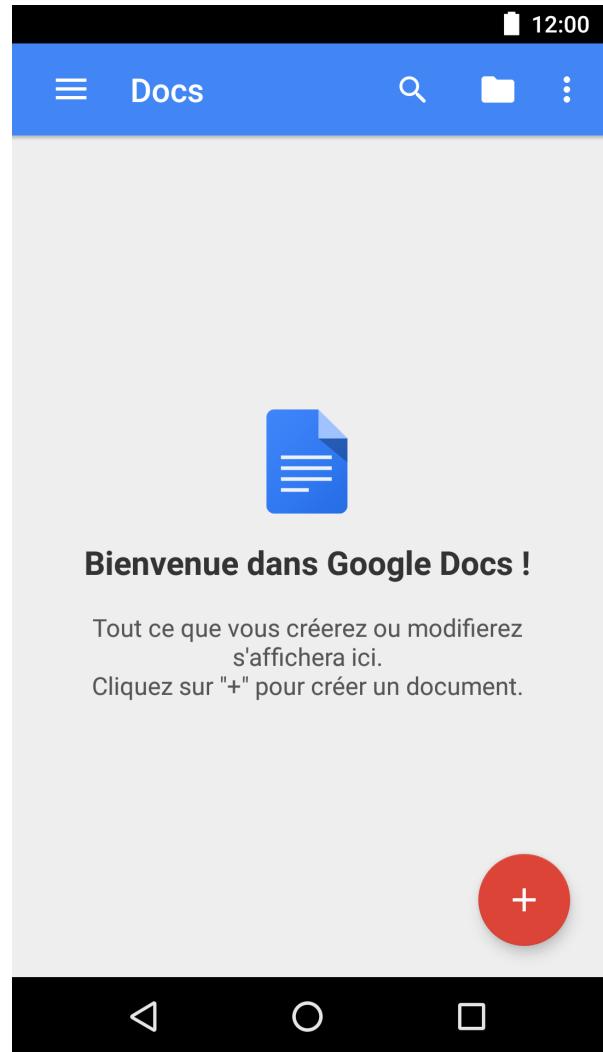
Grand centre commercial avec boutiques de mode, d'électroménager et de services, et lieux de restauration.

24 Place des Halles, 67000 Strasbourg



Floating Action Button (fab)

- Bouton d'action situé dans le coin bas droit d'un écran ou au niveau d'une app bar étendue
- A utiliser si une action doit être mise en avant sur un écran par rapport à toutes les autres



Floating Action Button (fab)

- Bouton d'action situé dans le coin bas droit d'un écran ou au niveau d'une app bar étendue
- A utiliser si une action doit être mise en avant sur un écran par rapport à toutes les autres
- A utiliser pour des actions positives (Créer, partager, éditer)

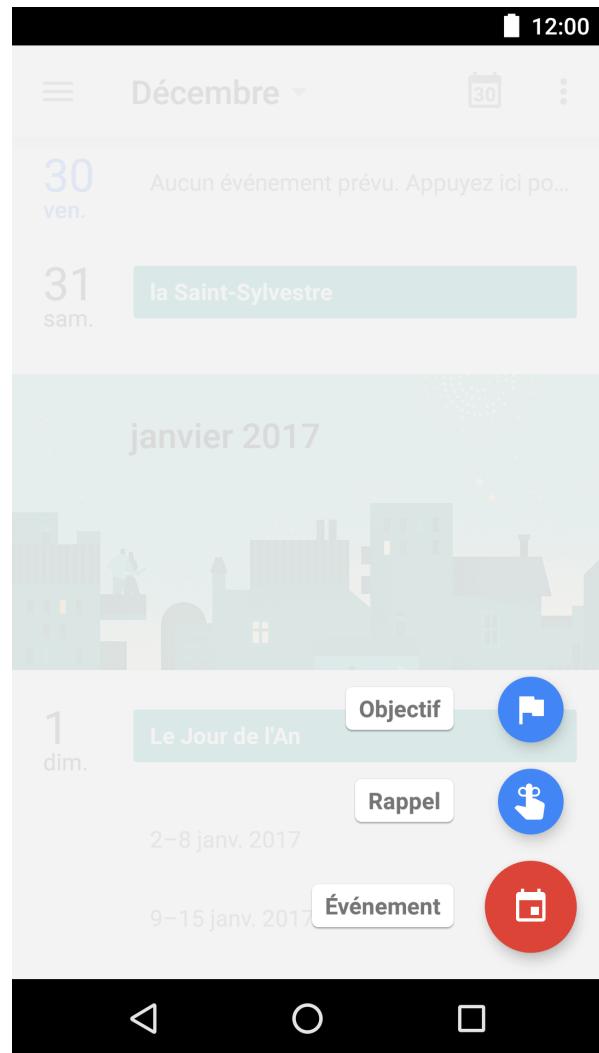


**Designing a New Look
for Instagram, Inspired
by the Community**

Today we announced a new look for Instagram, inside and outside the app. We created a new Instagram app.

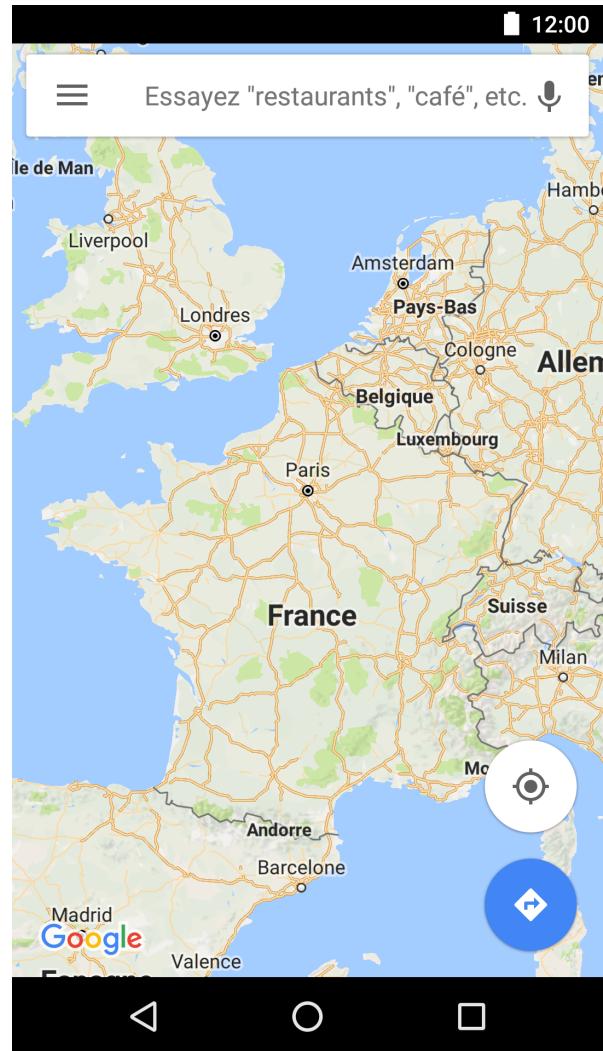
Floating Action Button (fab)

- Bouton d'action situé dans le coin bas droit d'un écran ou au niveau d'une app bar étendue
- A utiliser si une action doit être mise en avant sur un écran par rapport à toutes les autres
- A utiliser pour des actions positives (Créer, partager, éditer)
- Peut se décomposer en sous actions liés à l'action parent



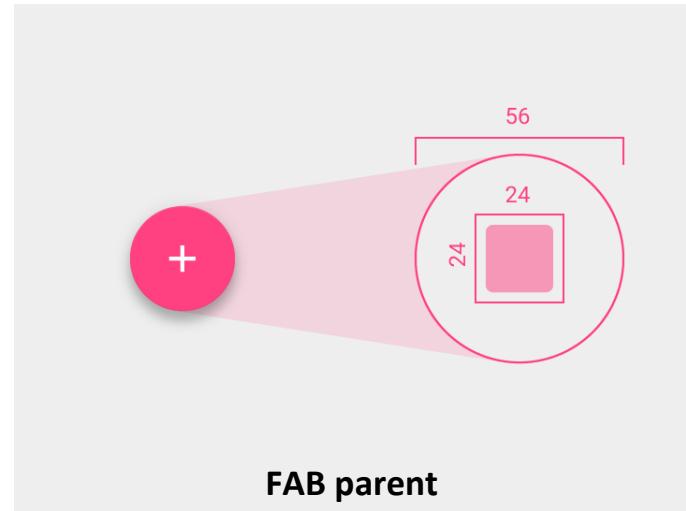
Floating Action Button (fab)

- Bouton d'action situé dans le coin bas droit d'un écran ou au niveau d'une app bar étendue
- A utiliser si une action doit être mise en avant sur un écran par rapport à toutes les autres
- A utiliser pour des actions positives (Créer, partager, éditer)
- Peut se décomposer en sous actions liés à l'action parent
- Un écran peut contenir maximum deux FAB

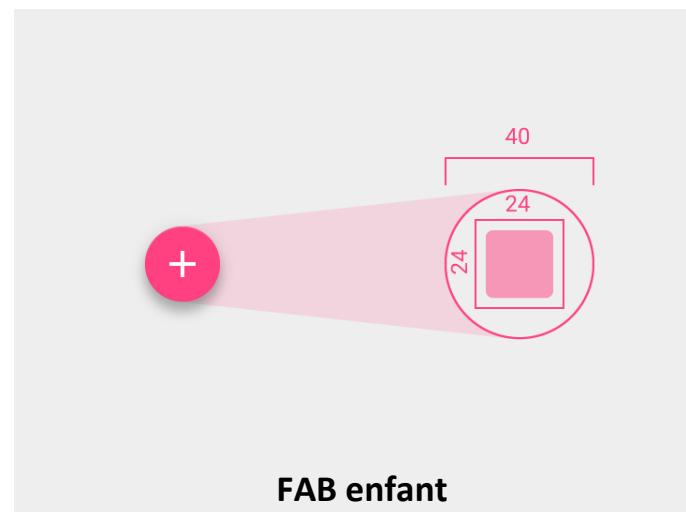


Floating Action Button (fab)

- Bouton d'action situé dans le coin bas droit d'un écran ou au niveau d'une app bar étendue
- A utiliser si une action doit être mise en avant sur un écran par rapport à toutes les autres
- A utiliser pour des actions positives (Créer, partager, éditer)
- Peut se décomposer en sous actions enfants liés à l'action parent
- Un écran peut contenir maximum deux FAB parents
- Un FAB peut avoir deux tailles différentes



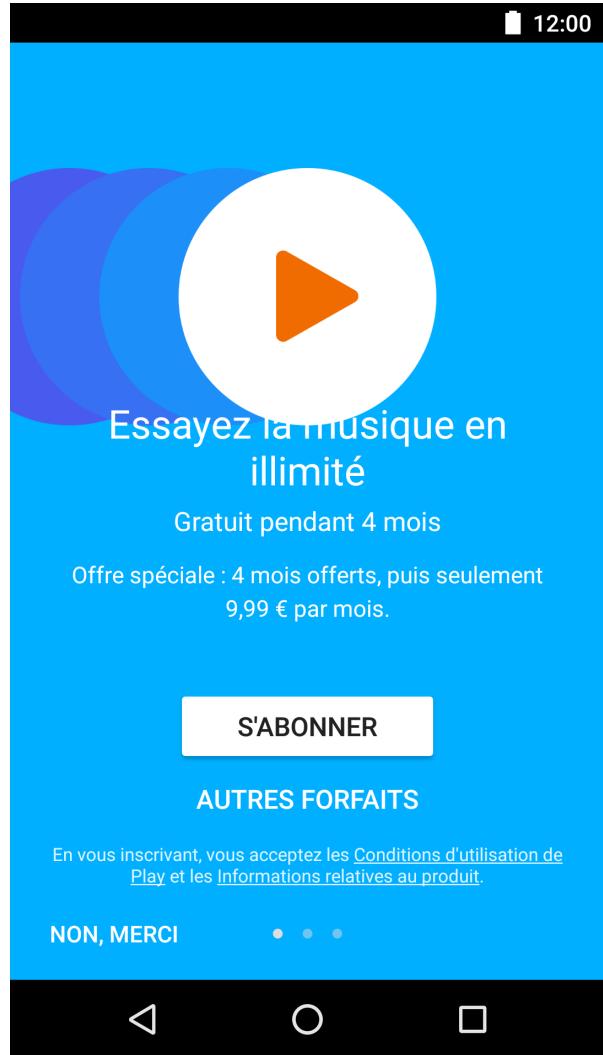
FAB parent



FAB enfant

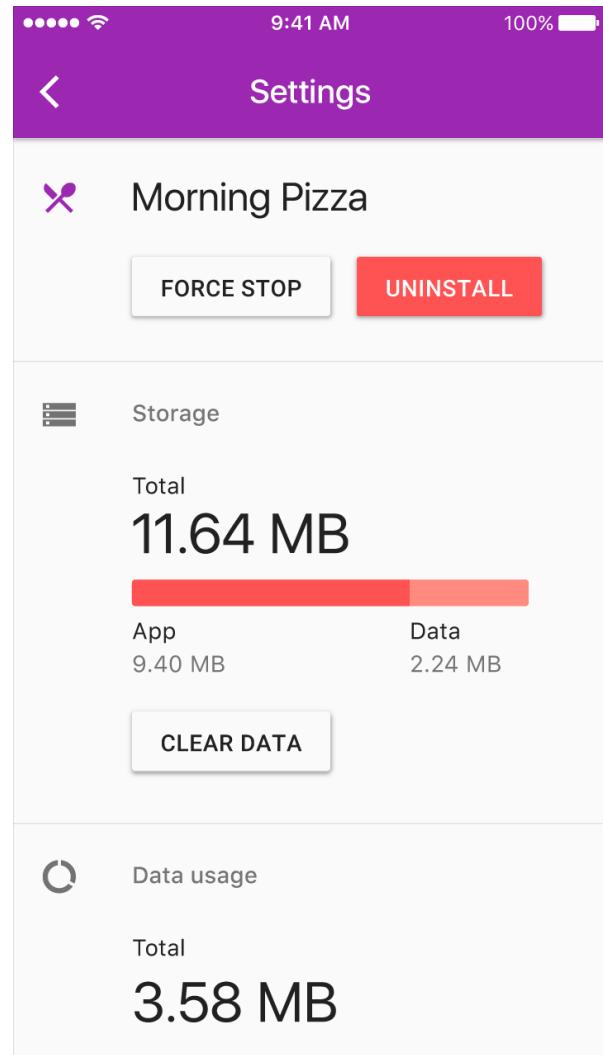
Raised Button

- Pour mettre en avant une action sur un écran chargé



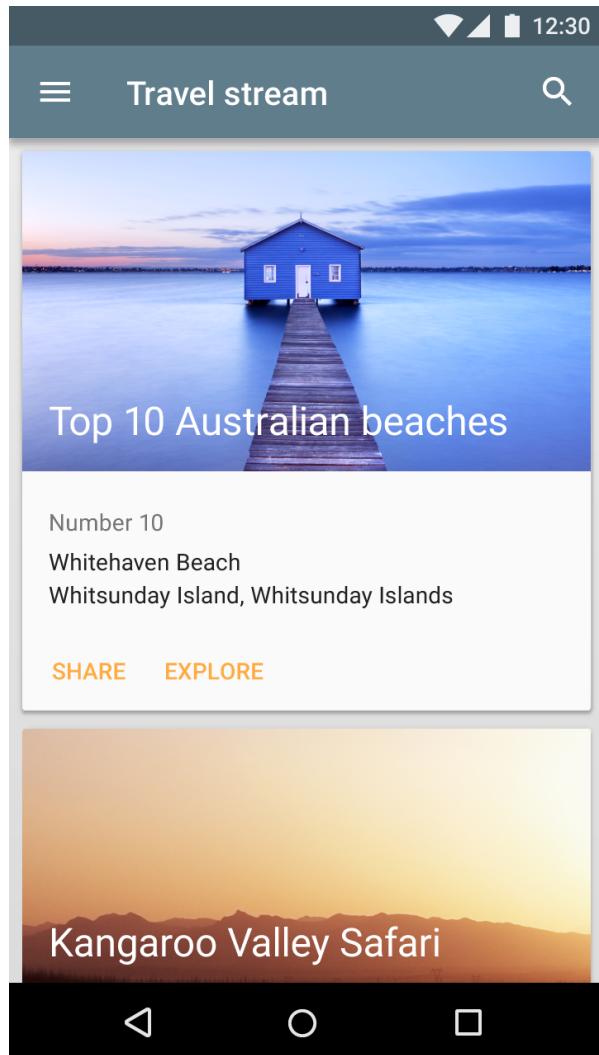
Raised Button

- Pour mettre en avant une action sur un écran chargé
- Pour réaliser une action par rapport à un contenu affiché sur une partie de l'écran



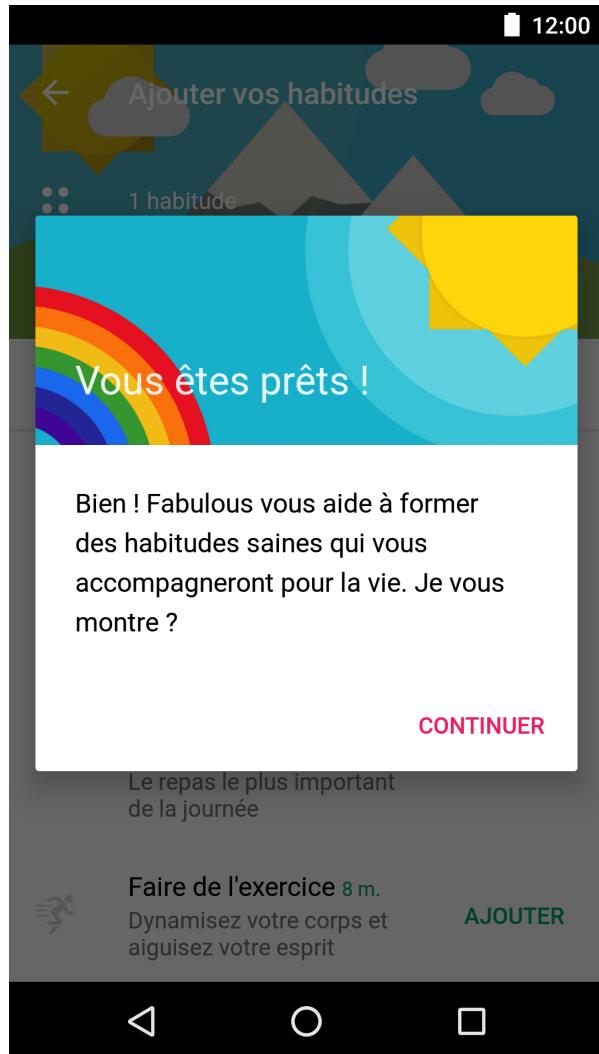
Flat Button

- Pour des actions qu'on veut afficher :
 - Sur des cartes



Flat Button

- Pour des actions qu'on veut afficher :
 - Sur des cartes
 - Sur des popups



Flat Button

- Pour des actions qu'on veut afficher :
 - Sur des cartes
 - Sur des popups
 - Dans des toolbars



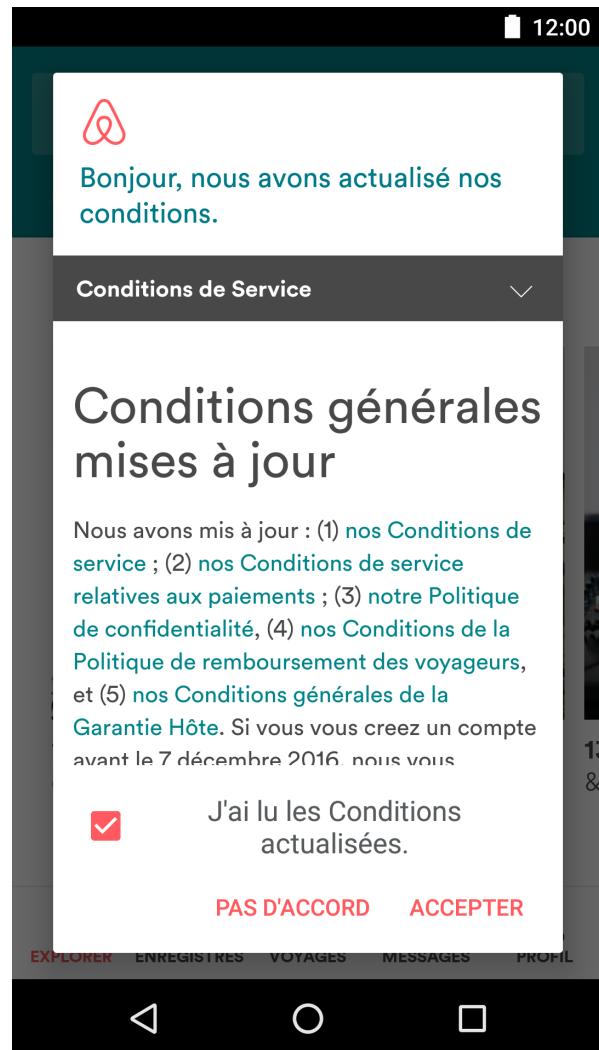
Flat Button

- Pour des actions qu'on veut afficher :
 - Sur des cartes
 - Sur des popups
 - Dans des toolbars
 - Dans le footer persistant d'un écran



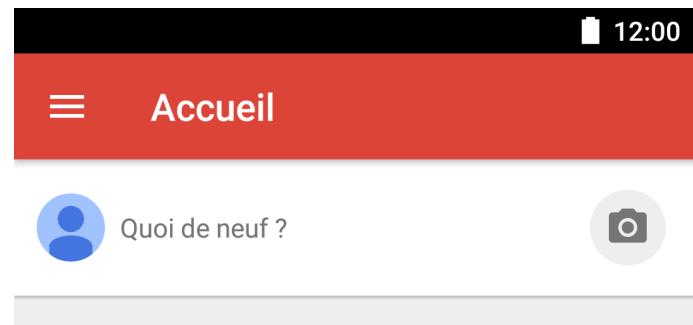
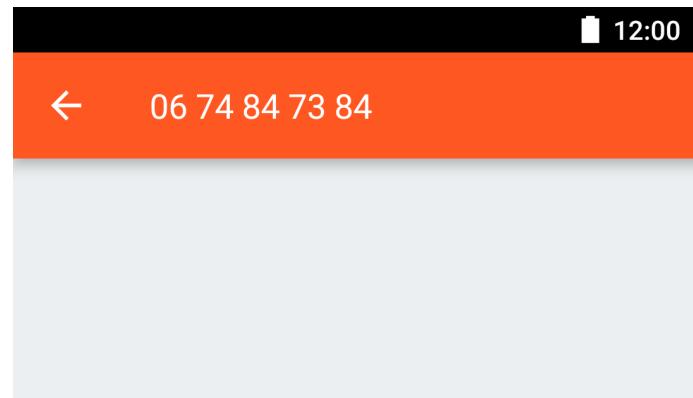
Flat Button

- Pour des actions qu'on veut afficher :
 - Sur des cartes
 - Sur des popups
 - Dans des toolbars
 - Dans le footer persistant d'un écran
- L'action porte sur le contenu dans lequel le bouton est ancré



App bar

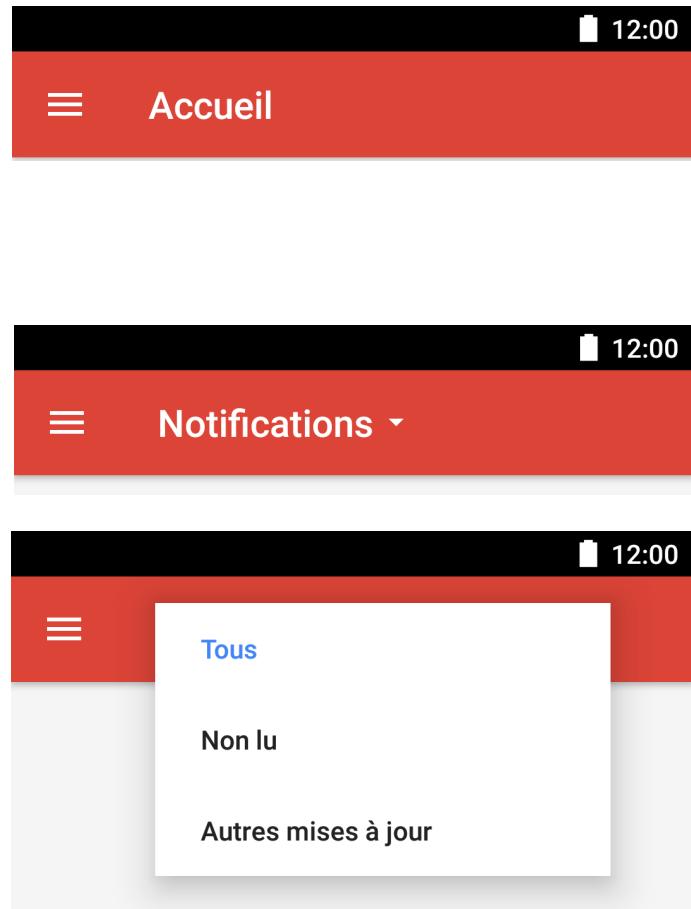
- Barre située en haut d'un écran
- Pour proposer des boutons de navigation pour retourner sur l'écran précédent ou pour afficher un menu



Vous connaissez peut-être...
Pour voir les posts de vos proches, suivez leur flux.

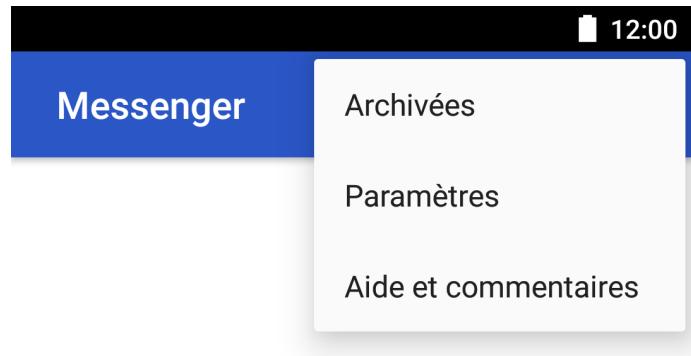
App bar

- Barre située en haut d'un écran
- Pour proposer des boutons de navigation pour retourner sur l'écran précédent ou pour afficher un menu
- Pour afficher un titre présentant le contenu de l'écran ou un spinner pour filtrer le contenu de la page



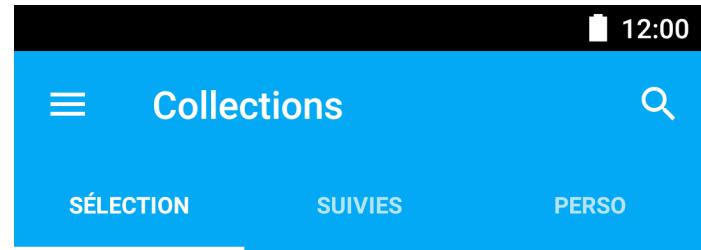
App bar

- Barre située en haut d'un écran
- Pour proposer des boutons de navigation pour retourner sur l'écran précédent ou pour afficher un menu
- Pour afficher un titre présentant le contenu de l'écran ou un spinner pour filtrer le contenu de la page
- Pour proposer des actions de recherche, de partage, de paramétrage etc... en relation avec le contenu de l'écran



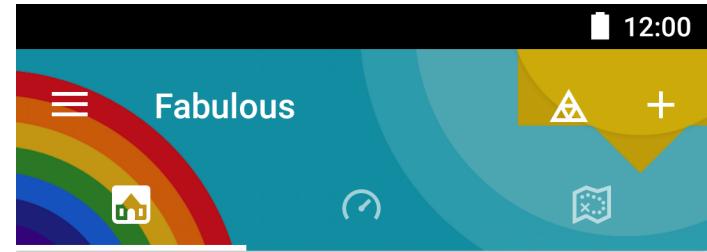
App bar

- Barre située en haut d'un écran
- Pour proposer des boutons de navigation pour retourner sur l'écran précédent ou pour afficher un menu
- Pour afficher un titre présentant le contenu de l'écran ou un spinner pour filtrer le contenu de la page
- Pour proposer des actions de recherche, de partage, de paramétrage etc... en relation avec le contenu de l'écran
- Utilisé pour l'affichage des tabs



App bar

- Entièrement personnalisable permettant de compléter le branding de l'application



App bar

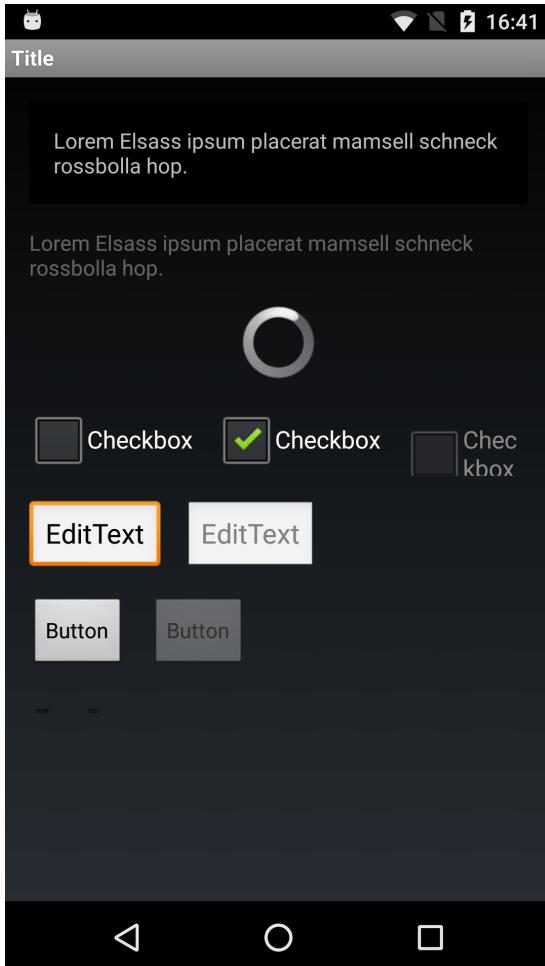
- Entièrement customisable permettant de compléter le branding de l'application
- Peut se collapse et s'expand pour mettre en avant du contenu



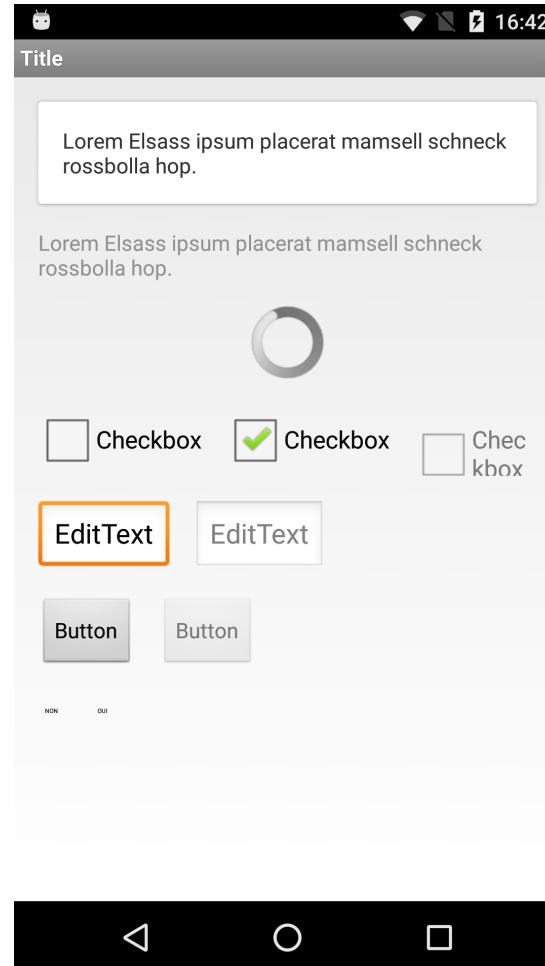
Thème et style

- Ensemble de propriétés et couleurs pour afficher tous les éléments d'une application
- Chaque widget (Button, EditText etc...) possède son propre style
- Chaque style possède une liste d'items semblables aux propriétés que peut avoir une view dans un layout
- Un thème/style peut hériter d'un autre thème/style pour modifier uniquement certaine propriété
- Il est possible définir son propre thème et ses propres styles dans un fichier styles.xml

Theme

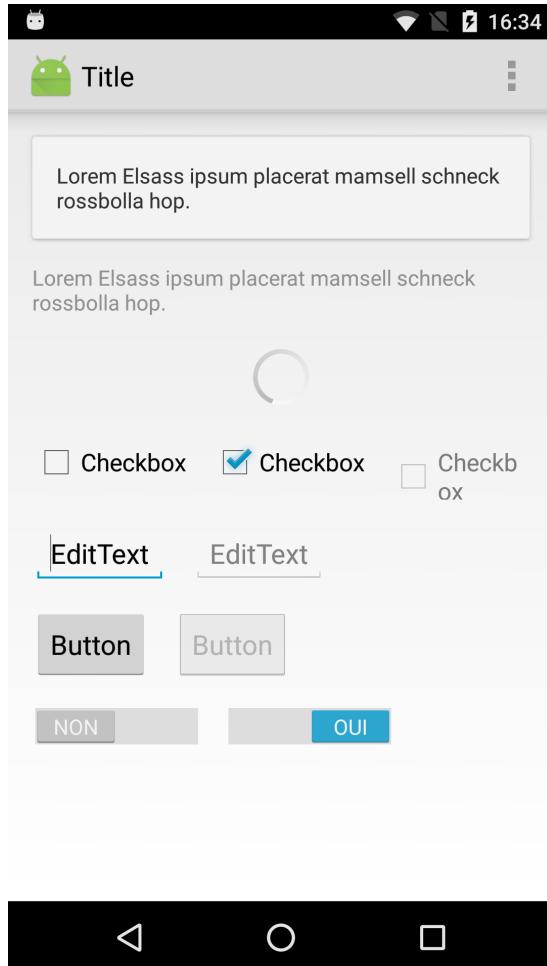


Theme

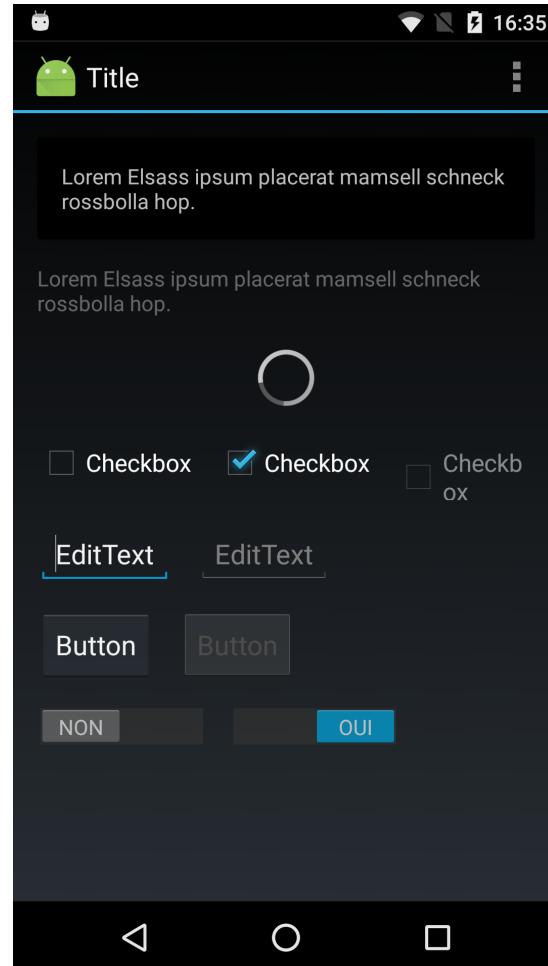


Theme.Light

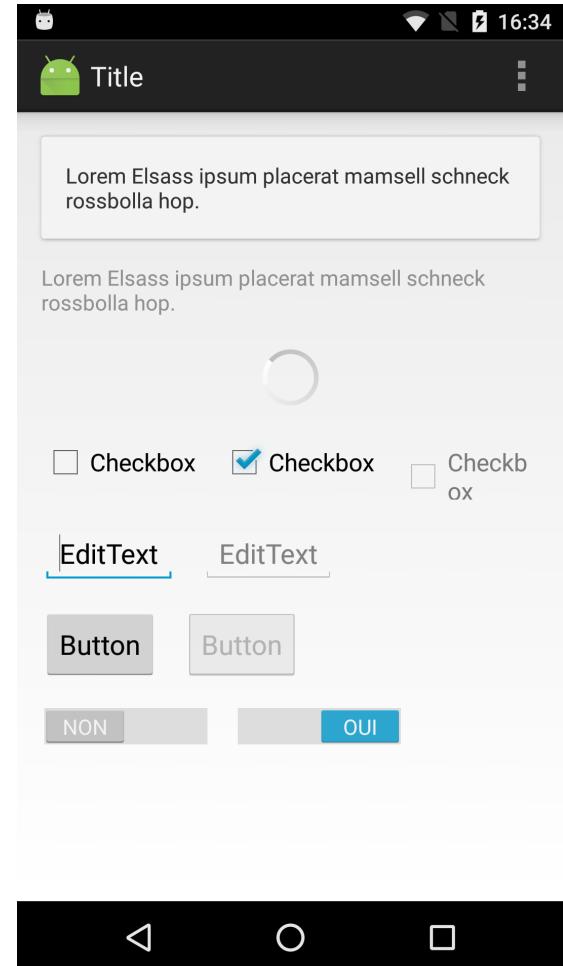
Theme Holo - Android 3.0 +



Theme.Holo.Light

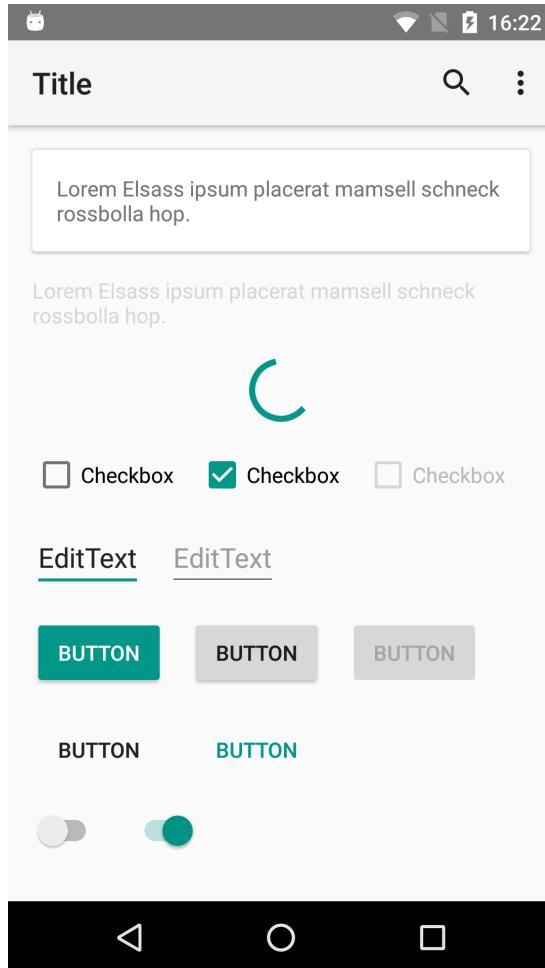


Theme.Holo

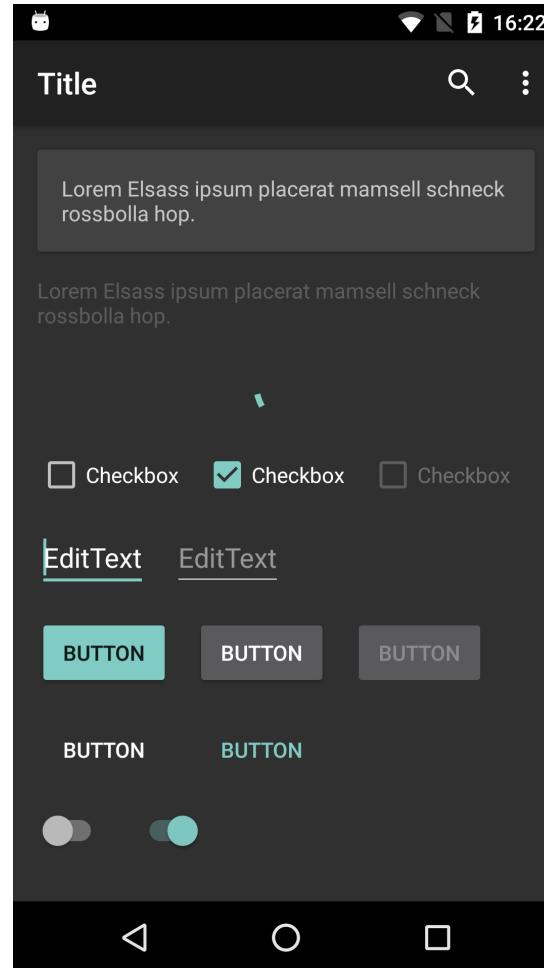


Theme.Holo.Light.DarkActionBar

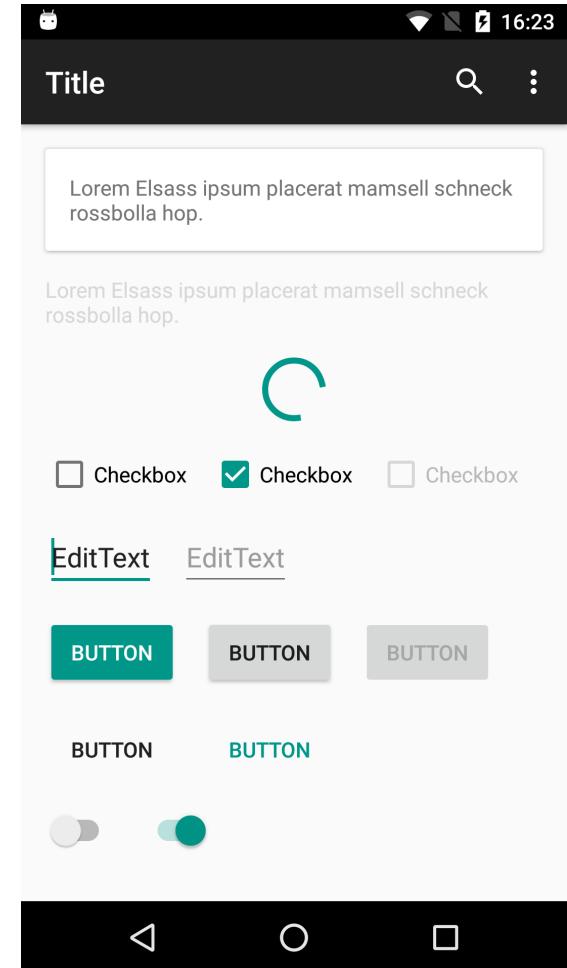
Theme Material - Android 5.0 +



Theme.Material.Light



Theme.Material



Theme.Material.Light.DarkActionBar

```
<style name="Theme">

    <item name="isLightTheme">false</item>
    <item name="colorForeground">@color/bright_foreground_dark</item>
    <item name="colorForegroundInverse">@color/bright_foreground_dark_inverse</item>
    <item name="colorBackground">@color/background_dark</item>
    <item name="colorBackgroundFloating">?attr/colorBackground</item>
    <item name="colorBackgroundCacheHint">?attr/colorBackground</item>

    <item name="buttonStyle">@style/Widget.Button</item>
    <item name="switchStyle">@style/Widget.CompoundButton.Switch</item>
    <item name="absListViewStyle">@style/Widget.AbsListView</item>
    <item name="autoCompleteTextViewStyle">@style/Widget.AutoCompleteTextView</item>
    <item name="checkboxStyle">@style/Widget.CompoundButton.CheckBox</item>
    <item name="checkedTextViewStyle">@style/Widget.CheckedTextView</item>
    <item name="dropDownListViewStyle">@style/Widget.ListView.DropDown</item>
    <item name="editTextStyle">@style/Widget.EditText</item>

    ...
</style>

<style name="Base.Widget.AppCompat.Button" parent="android:Widget">
    <item name="android:background">@drawable/abc_btn_default_mtrl_shape</item>
    <item name="android:textAppearance">?android:attr/textAppearanceButton</item>
    <item name="android:minHeight">48dip</item>
    <item name="android:minWidth">88dip</item>
    <item name="android:focusable">true</item>
    <item name="android:clickable">true</item>
    <item name="android:gravity">center_vertical|center_horizontal</item>
</style>
```

Exemple déclaration d'un thème et d'un style

```
<style name="Theme.Light">
    <item name="isLightTheme">true</item>
    <item name="windowBackground">@drawable/screen_background_selector_light</item>
    <item name="windowClipToOutline">false</item>

    <item name="colorBackground">@color/background_light</item>
    <item name="colorForeground">@color/bright_foreground_light</item>
    <item name="colorForegroundInverse">@color/bright_foreground_light_inverse</item>

    <item name="colorPrimaryDark">@color/legacy_light_primary_dark</item>
    <item name="colorPrimary">@color/legacy_light_primary</item>
    <item name="colorControlActivated">@color/legacy_light_control_activated

    ...
</style>

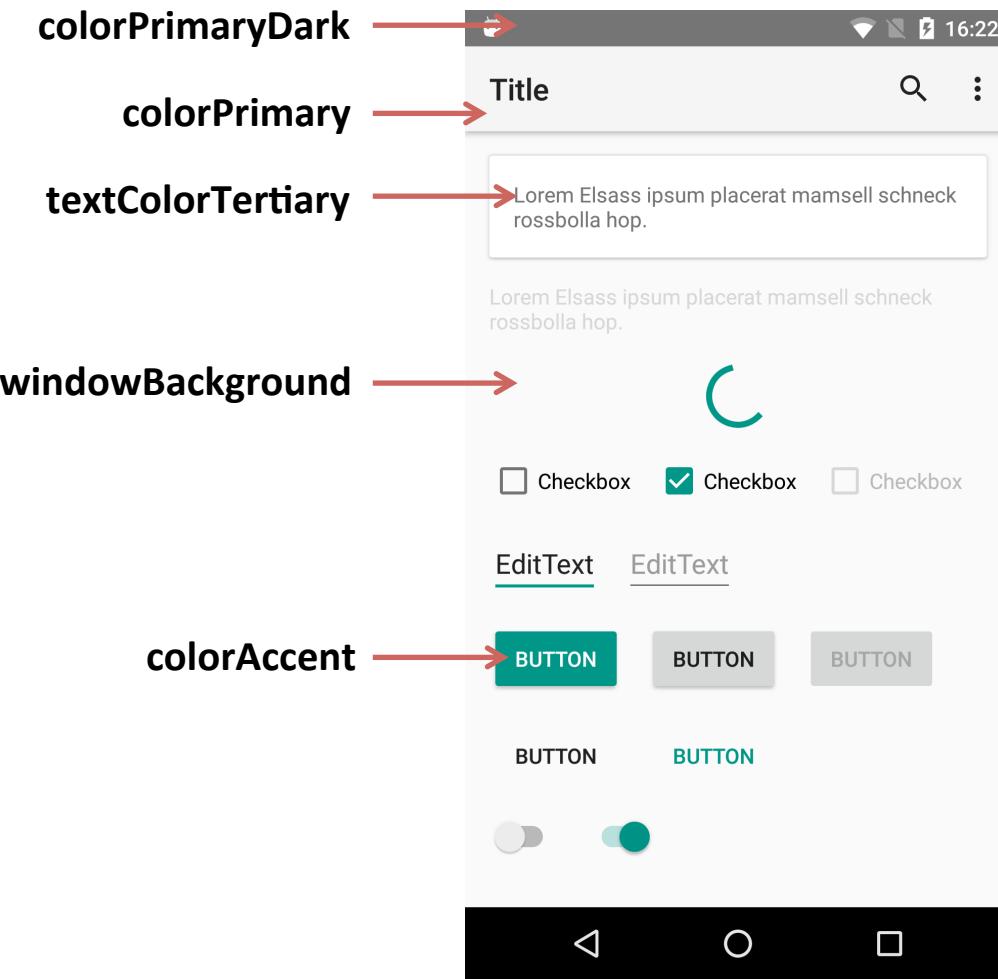
<style name="Platform.AppCompat.Light" parent="android:Theme.Light">
    <item name="android:windowNoTitle">true</item>

    <item name="android:colorForeground">@color/foreground_material_light</item>
    <item name="android:colorForegroundInverse">@color/foreground_material_dark</item>
    <item name="android:colorBackground">@color/background_material_light</item>
    <item name="android:disabledAlpha">@dimen/abc_disabled_alpha_material_light</item>
    <item name="android:backgroundDimAmount">0.6</item>
    <item name="android:windowBackground">@color/background_material_light</item>

    ...
</style>
```

Les deux façons pour hériter d'un thème

Création d'un thème custom

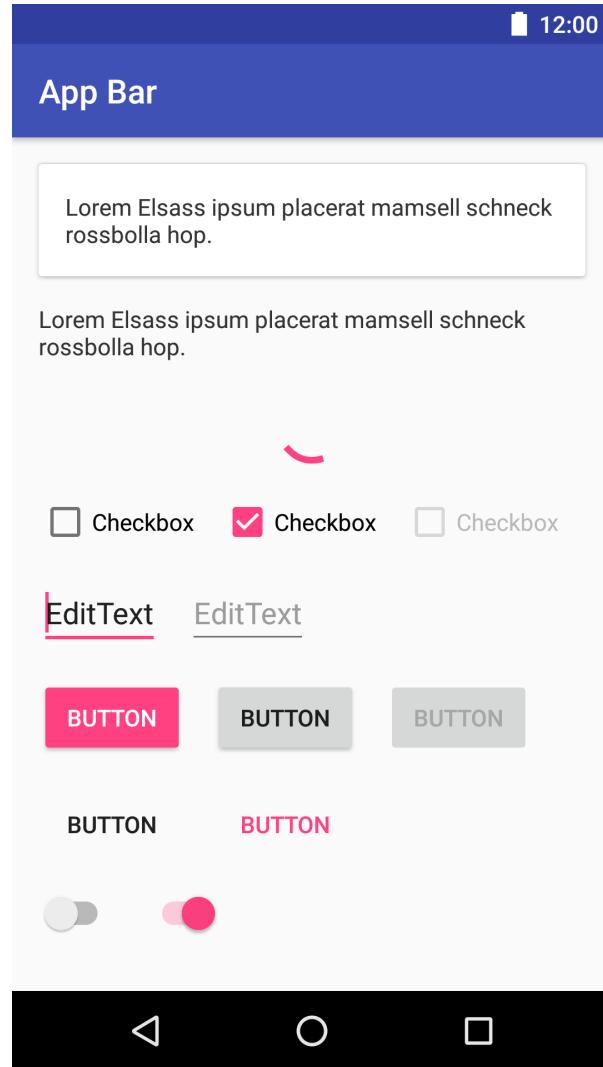


```
colors.xml
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
    <color name="textColorPrimary">#333333</color>
</resources>

styles.xml
<resources>
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
        <item name="android:textColorTertiary">@color/textColorPrimary</item>
    </style>
</resources>

AndroidManifest.xml
<activity android:name=".MyActivity" android:theme="@style/AppTheme" />
```

Création d'un thème custom



Librairies de support Android

Android Support v4

- Permet d'utiliser les fragments, le nouveau système de permission, les ViewPager (Pour l'utilisation des tabs), etc... sur les téléphones sous Android 2.3+

```
build.gradle
```

```
dependencies {  
    compile 'com.android.support:support-v4:25.1.0'  
}
```

Android Support CardView / RecyclerView

- Permet d'utiliser les CardViews et les Recyclers sur les téléphones sous Android 2.3+

build.gradle

```
dependencies {  
    compile 'com.android.support:cardview-v7:25.1.0'  
    compile 'com.android.support:recyclerview-v7:25.1.0'  
}
```

CardView

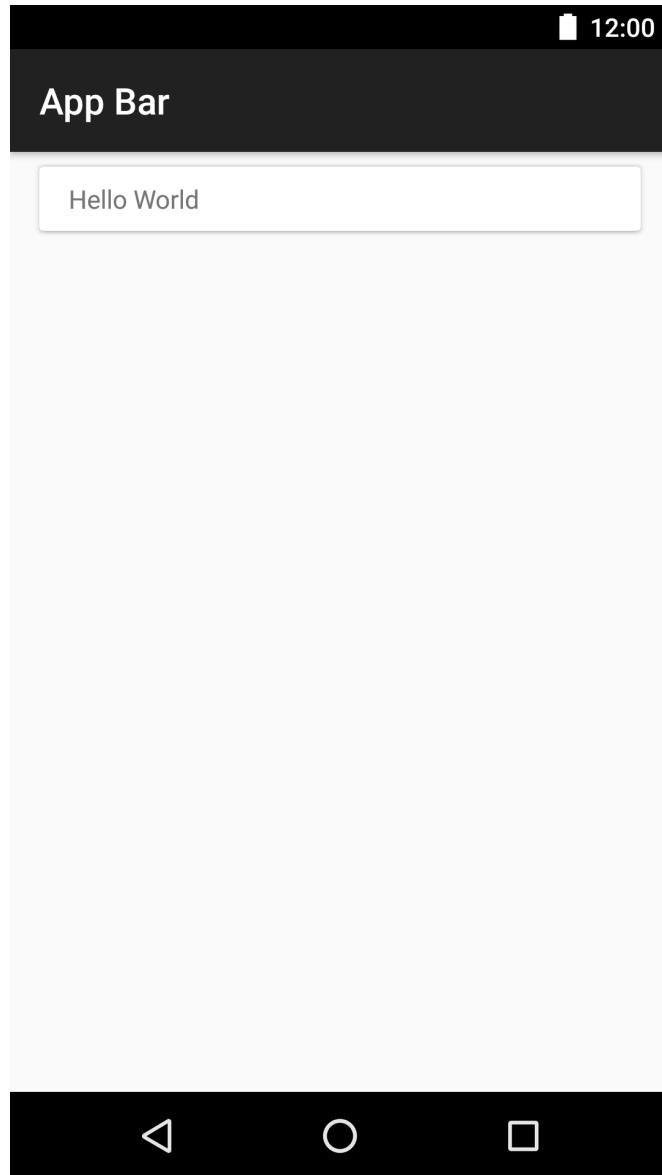
- Hérite du layout FrameLayout
- Propriétés particulières :
 - app:cardBackgroundColor : Couleur de fond de la CardView
 - app:contentPadding : Padding de la CardView
- Toujours affecter un minimum de margins à une CardView pour l'affichage de l'élévation

```
<android.support.v7.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="@dimen/spacing_medium"
    android:layout_marginLeft="@dimen/spacing_big"
    android:layout_marginRight="@dimen/spacing_big"
    android:layout_marginTop="@dimen/spacing_medium"
    app:cardBackgroundColor="@color/white"
    app:contentPaddingBottom="@dimen/spacing_medium"
    app:contentPaddingLeft="@dimen/spacing_big"
    app:contentPaddingRight="@dimen/spacing_big"
    app:contentPaddingTop="@dimen/spacing_medium">

    <!-- Contenu de la CardView -->
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/text" />
    </LinearLayout>
</android.support.v7.widget.CardView>
```

Ajout d'une CardView dans un layout



**Affichage d'une CardView sur
téléphones Android 2.3+**

Android AppCompat v7

- Permet d'utiliser les thèmes Material Design et l'App bar sur les téléphones Android 2.3+
- Dépend de la librairie Support v4

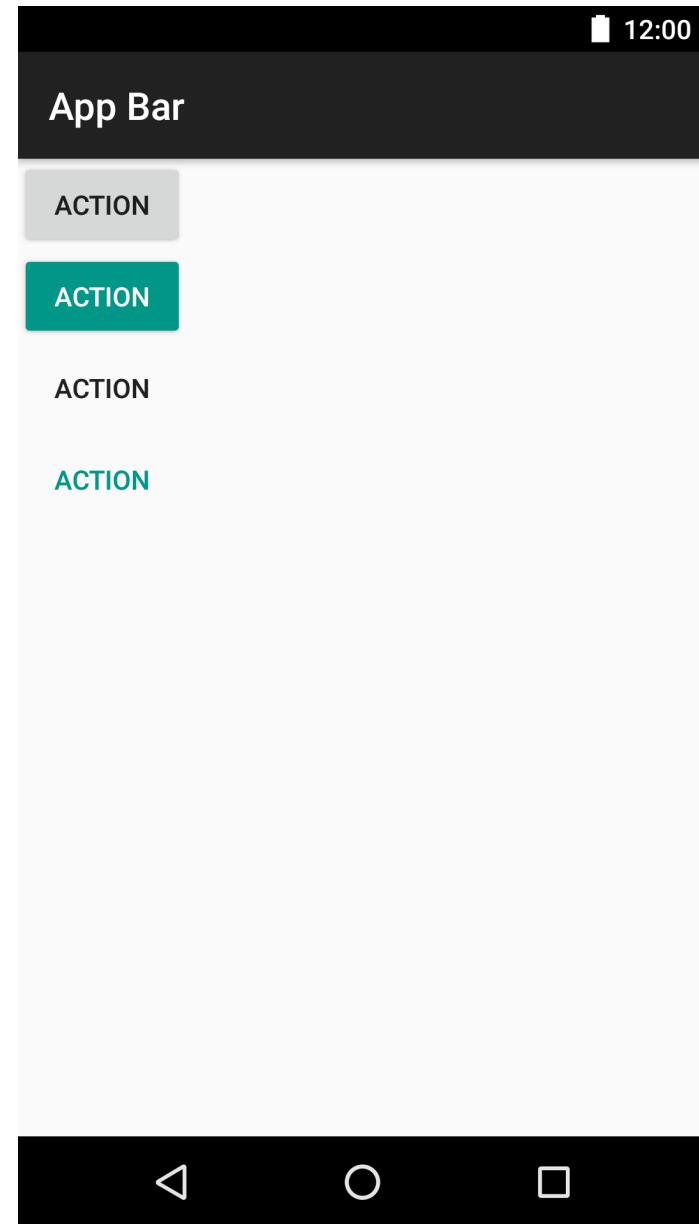
```
build.gradle
dependencies {
    compile 'com.android.support:appcompat-v7:25.1.0'
}
```

```
<Button
    style="@style/Widget.AppCompat.Button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/action" />

<Button
    style="@style/Widget.AppCompat.Button.Colored"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/action" />

<Button
    style="@style/Widget.AppCompat.Button.Borderless"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/action" />

<Button
    style="@style/Widget.AppCompat.Button.Borderless.Colored"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/action" />
```



**Utilisation des styles Material
Design**

```
styles.xml
<resources>
    <style name="AppTheme1" parent="Theme.AppCompat"/>
    <style name="AppTheme2" parent="Theme.AppCompat.Light"/>
    <style name="AppTheme3" parent="Theme.AppCompat.Light.DarkActionBar"/>
</resources>
```

```
AndroidManifest.xml
<activity android:name=".MyActivity" android:theme="@style/AppTheme3" />
```

```
MyActivity.java
public class MyActivity extends AppCompatActivity {
    ...
}
```

Mise en place d'un thème Material Design sur une Activity



Affichage de l'App bar sur
téléphones Android 2.3+

Ajout d'icônes d'action dans l'App bar

- Créer un fichier xml menu dans le dossier res/menu
- Chaque action est représentée par un noeud item
- Principales propriétés d'un item :
 - id : Permet d'identifier l'action
 - title : Texte de l'action
 - icon : Icône de l'action
 - showAsAction : Détermine comment afficher l'action
- Ajouter le menu dans l'Activity

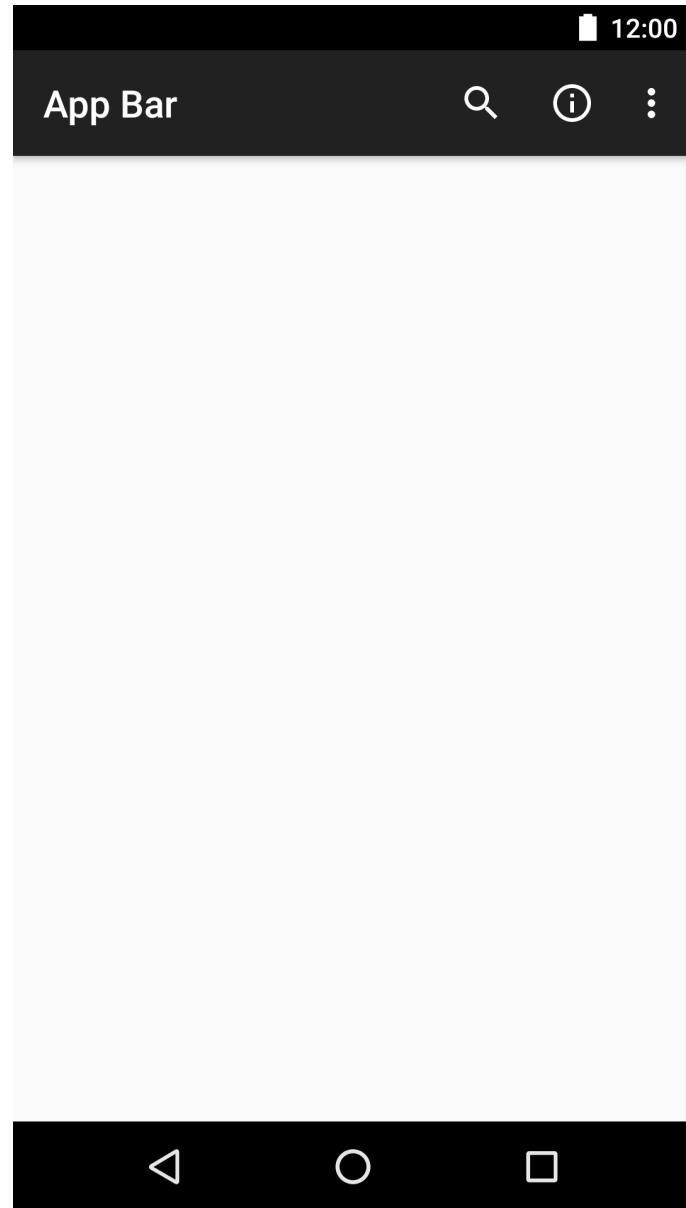
```
res/menu/my_menu.xml
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">

    <!-- Action toujours affichée -->
    <item
        android:id="@+id/ic_search"
        android:icon="@drawable/ic_search"
        android:title="@string/search"
        app:showAsAction="always" />

    <!-- Action affichée si place disponible dans l'App bar -->
    <item
        android:id="@+id/ic_about"
        android:icon="@drawable/about"
        android:title="@string/about"
        app:showAsAction="ifRoom" />

    <!-- Action toujours cachée sous "..." -->
    <item
        android:id="@+id/ic_logout"
        android:title="@string/logout"
        app:showAsAction="never" />
</menu>
```

```
public class MyActivity extends AppCompatActivity {  
  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        // Mise en place du menu  
        getMenuInflater().inflate(R.menu.my_menu, menu);  
        return true;  
    }  
  
    @Override  
    public boolean onOptionsItemSelected(MenuItem item) {  
        // Gestion des clics sur les icônes du menu  
        switch (item.getItemId()) {  
            case R.id.ic_search:  
                break;  
            case R.id.ic_about:  
                break;  
            ...  
        }  
        return true;  
    }  
}
```



Affichage de l'App bar avec
icônes d'action

Limite de l'App bar

- Par défaut, une App bar ne peut pas :
 - Intégrer des tabs
 - Afficher un spinner
 - Etre expandable / collapsable
 - Contenir un FAB

Introduction de la Toolbar

- Généralisation de l'App bar
- Permet de définir l'App bar à partir d'un layout et de la gérer à la main
- Permet de contourner les limites de l'App bar

Android Design

- Permet d'utiliser les comportements et les composants Material Design (FAB, navigation drawer, tabs etc...) sur les téléphones Android 2.3+
- Dépend de la librairie AppCompat

```
build.gradle
dependencies {
    compile 'com.android.support:design:25.1.0'
}
```

AndroidManifest.xml

```
<activity android:name=".MyActivity" android:theme="@style/AppTheme" />
```

styles.xml

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar"/>
```

Utilisation d'une ToolBar comme App bar

```
AndroidManifest.xml
<activity android:name=".MyActivity" android:theme="@style/AppTheme" />

styles.xml
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar"/>

my_layout.xml
<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"/>

</android.support.design.widget.AppBarLayout>
```

Utilisation d'une ToolBar comme App bar

```
AndroidManifest.xml
<activity android:name=".MyActivity" android:theme="@style/AppTheme" />
```

```
styles.xml
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar"/>
```

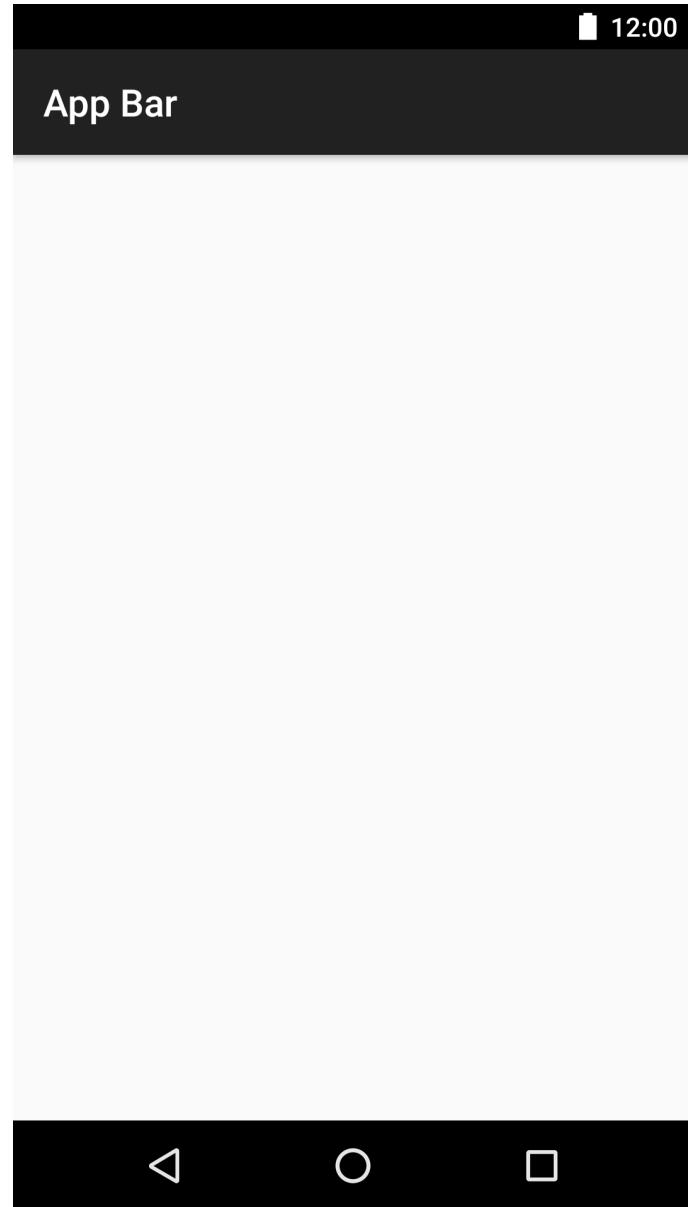
```
my_layout.xml
<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"/>

</android.support.design.widget.AppBarLayout>
```

```
MyActivity.java
@Override
protected void onCreate(@Nullable Bundle bundle) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.my_layout);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    setTitle("App Bar");
}
```

Utilisation d'une ToolBar comme App bar



Affichage d'une Toolbar comme
AppBar

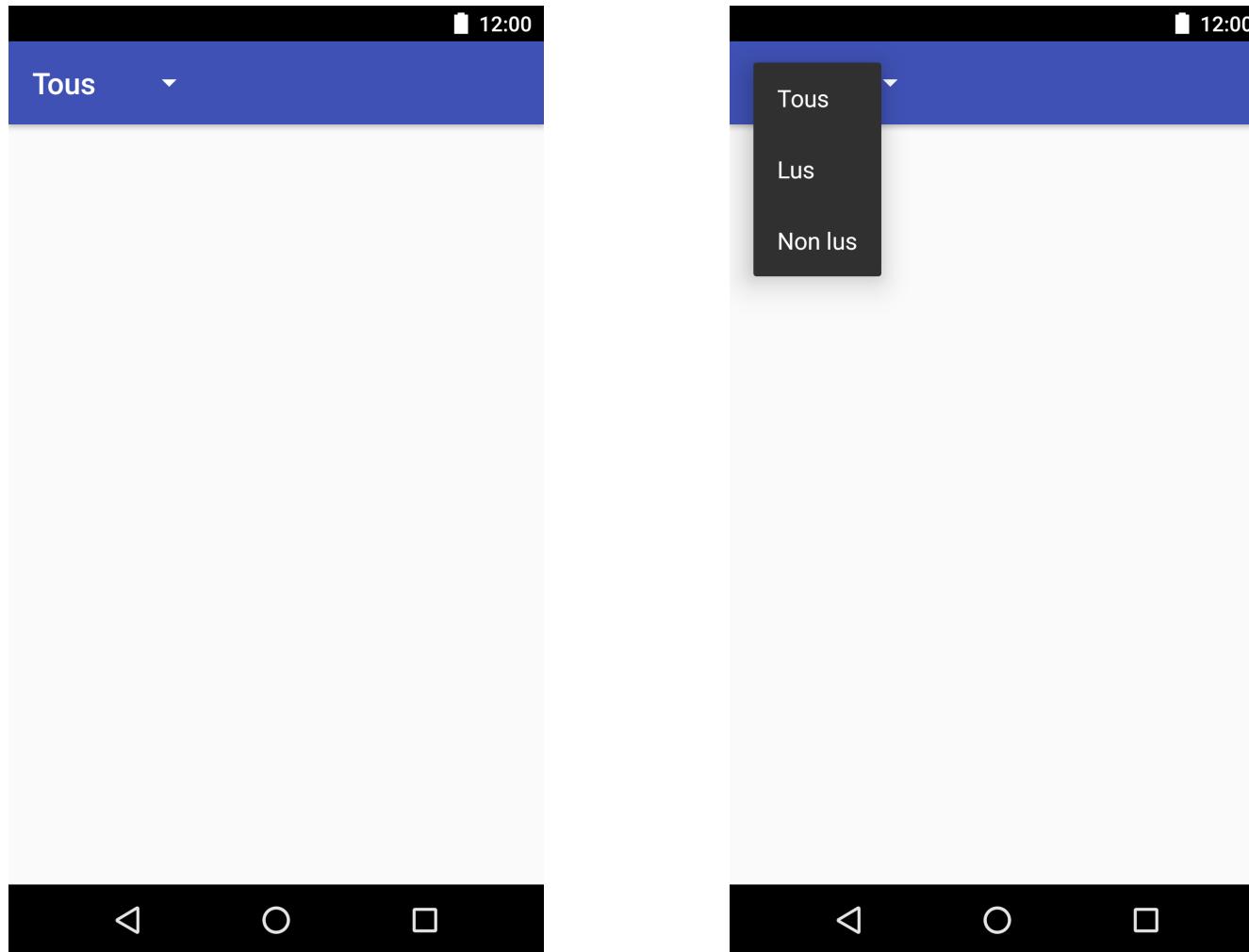
Floating action button

- Hérite d'un ImageButton
- Utilise la couleur colorAccent du thème par défaut
- Toujours affecter un minimum de margins pour l'affichage de l'élévation

```
<android.support.design.widget.FloatingActionButton  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentBottom="true"  
    android:layout_alignParentRight="true"  
    android:src="@drawable/ic_add"/>
```

Ajout d'un FAB dans un layout

Spinner dans l'App bar



```
my_layout.xml
<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

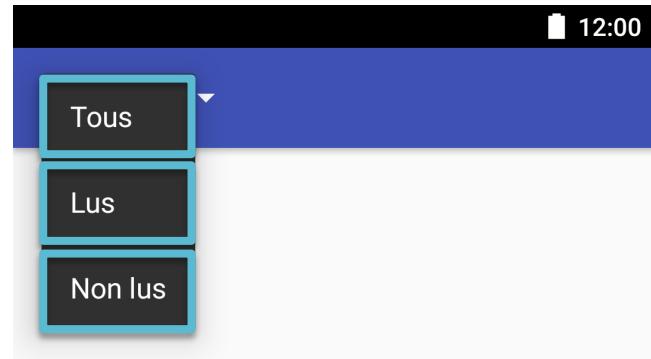
    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary">

        <Spinner
            android:id="@+id/spinner"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </android.support.v7.widget.Toolbar>

</android.support.design.widget.AppBarLayout>
```

```
spinner_title.xml
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="sans-serif-medium"
    android:textColor="@color/white"
    android:textSize="@dimen/app_bar_title_size" />
```

```
@Override  
protected void onCreate(@Nullable Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.my_layout);  
    ...  
  
    ArrayList<String> filters = new ArrayList<>();  
    filters.add("Tous");  
    filters.add("Lus");  
    filters.add("Non lus");  
  
    Spinner spinner = (Spinner) findViewById(R.id.spinner);  
  
    ArrayAdapter adapter = new ArrayAdapter<>(this, R.layout.spinner_title,  
        filters);  
    adapter.setDropDownViewResource(android.R.layout.simple_list_item_1);  
    spinner.setAdapter(adapter);  
}
```



```
@Override  
protected void onCreate(@Nullable Bundle savedInstanceState) {  
    ...  
  
    Spinner spinner = (Spinner) findViewById(R.id.spinner);  
  
    ArrayAdapter adapter = new ArrayAdapter<>(this, R.layout.spinner_title,  
        filters);  
    adapter.setDropDownViewResource(android.R.layout.simple_list_item_1);  
    spinner.setAdapter(adapter);  
    spinner.setOnItemSelectedListener(listener);  
}  
  
AdapterView.OnItemSelectedListener listener =  
    new AdapterView.OnItemSelectedListener() {  
        @Override  
        public void onItemSelected(AdapterView<?> parent, View view, int position,  
            long id) {  
            String selectedItem = (String) parent.getAdapter().getItem(position);  
        }  
  
        @Override  
        public void onNothingSelected(AdapterView<?> parent) { }  
    };
```

App bar collapsible



App bar collapsible

CoordinatorLayout

- Gére les différents contenus scrollables de l'écran



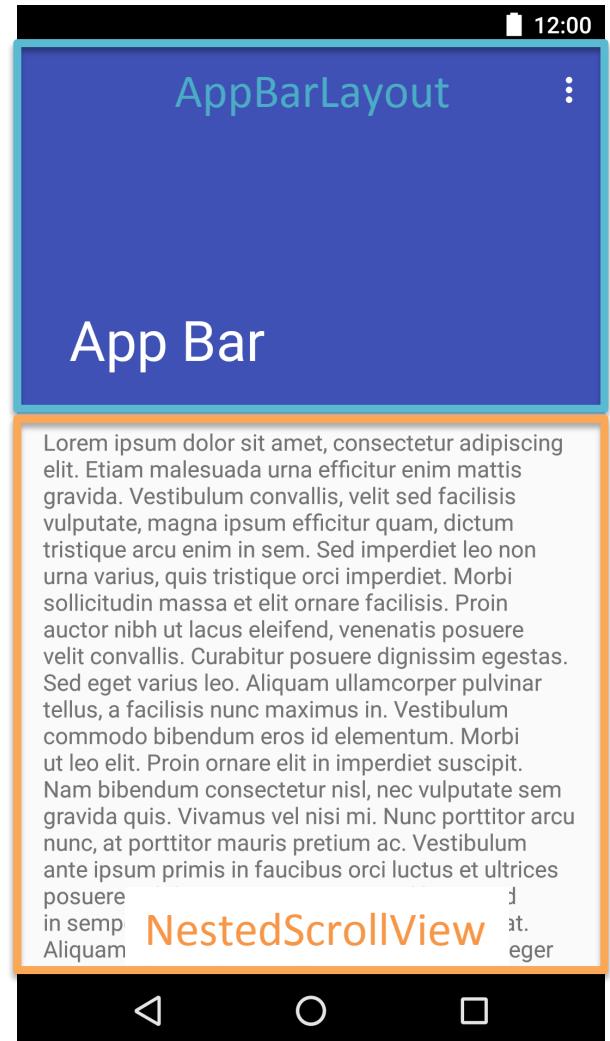
App bar collapsible

AppBarLayout

- Contient l'App bar
- Apporte les différentes particularités du Material Design

NestedScrollView

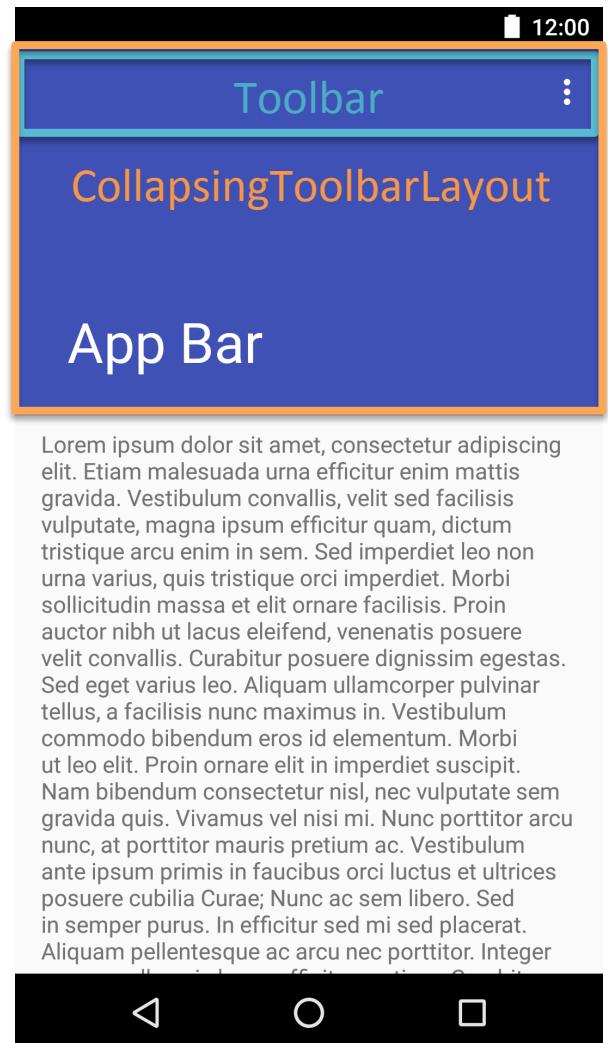
- Contient le contenu de l'écran



App bar collapsible

CollapsingToolbarLayout

- Contient une Toolbar
 - Layout qui s'expand/collapse



```
<android.support.design.widget.CoordinatorLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">
```

```
</android.support.design.widget.CoordinatorLayout>
```

```
<android.support.design.widget.CoordinatorLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        // Hauteur qu'on souhaite que l'App bar prenne une fois expanded
        android:layout_height="250dp"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

        </android.support.design.widget.AppBarLayout>

        <android.support.v4.widget.NestedScrollView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            // Flag la vue comme étant située à côté de l'AppBarLayout.
            // Permet à l'AppBarLayout de savoir quand scroller.
            app:layout_behavior="@string/appbar_scrolling_view_behavior">
        </android.support.v4.widget.NestedScrollView>

    </android.support.design.widget.CoordinatorLayout>
```

```
<android.support.design.widget.CoordinatorLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="250dp"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

        <android.support.design.widget.CollapsingToolbarLayout
            android:id="@+id/collapsing_toolbar"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            app:layout_scrollFlags="scroll|exitUntilCollapsed">

            <android.support.v7.widget.Toolbar
                android:id="@+id/toolbar"
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                android:background="?attr/colorPrimary"
                app:layout_collapseMode="pin"/>
        </android.support.design.widget.CollapsingToolbarLayout>
    </android.support.design.widget.AppBarLayout>

    <android.support.v4.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior">
    </android.support.v4.widget.NestedScrollView>

</android.support.design.widget.CoordinatorLayout>
```

```
@Override  
protected void onCreate(@Nullable Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.my_layout);  
  
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
    setSupportActionBar(toolbar);  
  
    // Le titre de l'App bar s'affecte au collapsingToolbarLayout  
    CollapsingToolbarLayout collapsingToolbarLayout =  
        (CollapsingToolbarLayout) findViewById(R.id.collapsing_toolbar);  
    collapsingToolbarLayout.setTitle("App bar");  
}
```

Combinaison ScrollFlags/CollapseMode

```
layout_scrollFlags="scroll  
|enterAlways|enterAlwaysCollapsed"  
layout_collapseMode="pin"
```

Comportement :

Quand on scrolle vers le haut, l'App bar et la ScrollView scrollent en même temps.

Quand on scrolle vers le bas, l'App bar commence à s'afficher dans sa taille collapse puis reste à cette taille jusqu'à ce que la ScrollView scroll entièrement puis l'App bar s'expand entièrement.



Combinaison ScrollFlags/CollapseMode

```
layout_scrollFlags="scroll  
|existUntilCollapsed"  
layout_collapseMode="pin"
```

Comportement :

Quand on scrolle vers le haut, l'App bar et la ScrollView scrollent en même temps. L'App bar ne scroll plus quand elle atteint sa taille collapse.

Quand on scrolle vers le bas, la ScrollView scroll entièrement puis l'App bar s'expand entièrement.



```
<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="250dp"
    android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

    <android.support.design.widget.CollapsingToolbarLayout
        android:id="@+id/collapsing_toolbar"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        // La couleur de l'App bar est transparente quand elle est expanded
        // Elle est colorPrimary quand elle est collapsed
        app:contentScrim="?attr/colorPrimary"
        app:layout_scrollFlags="scroll|exitUntilCollapsed">

        <ImageView
            android:id="@+id/image"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:scaleType="centerCrop"
            // Effet de parallax sur l'image quand l'App bar se collapse
            app:layout_collapseMode="parallax" />

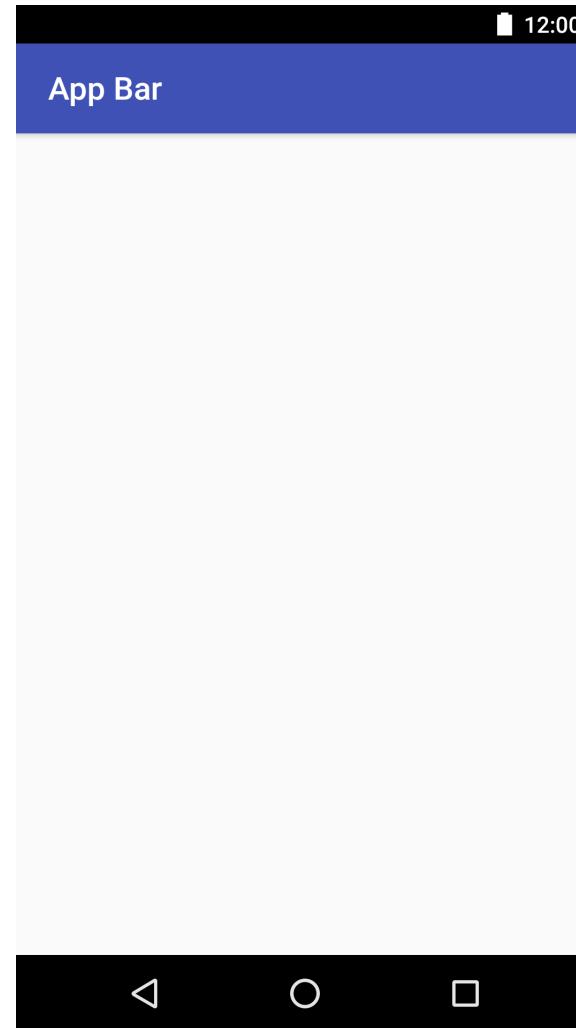
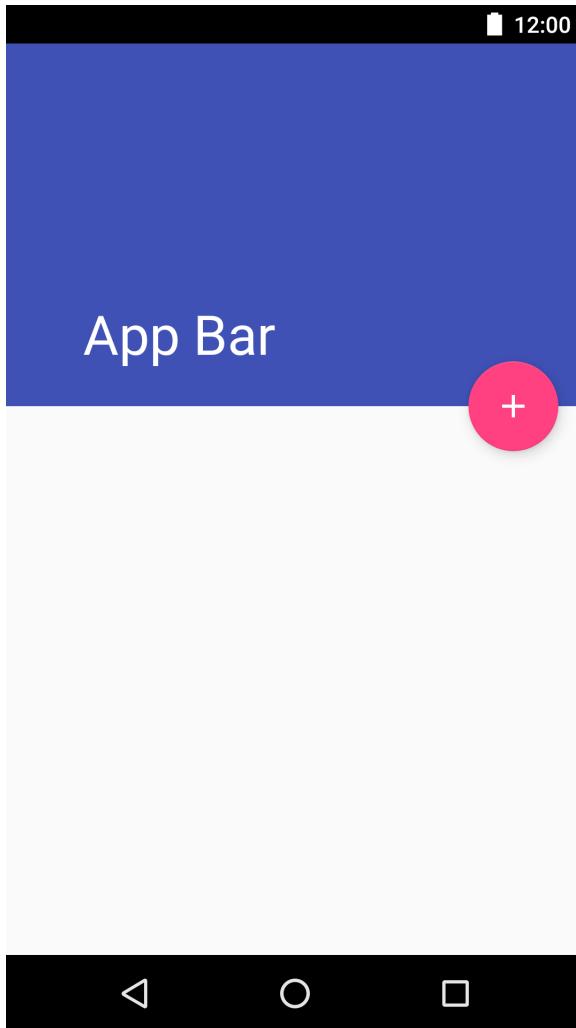
        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:layout_collapseMode="pin"/>
    </android.support.design.widget.CollapsingToolbarLayout>
</android.support.design.widget.AppBarLayout>
```

Combinaison ScrollFlags/CollapseMode

```
layout_scrollFlags="scroll  
|existUntilCollapsed"  
layout_collapseMode="parallax"
```



App bar collapsible avec un FAB



```
<android.support.design.widget.CoordinatorLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.design.widget.AppBarLayout
        android:id="@+id/app_bar"
        android:layout_width="match_parent"
        android:layout_height="250dp"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
    </android.support.design.widget.AppBarLayout>

    <android.support.v4.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior">
    </android.support.v4.widget.NestedScrollView>

</android.support.design.widget.CoordinatorLayout>
```

```
<android.support.design.widget.CoordinatorLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

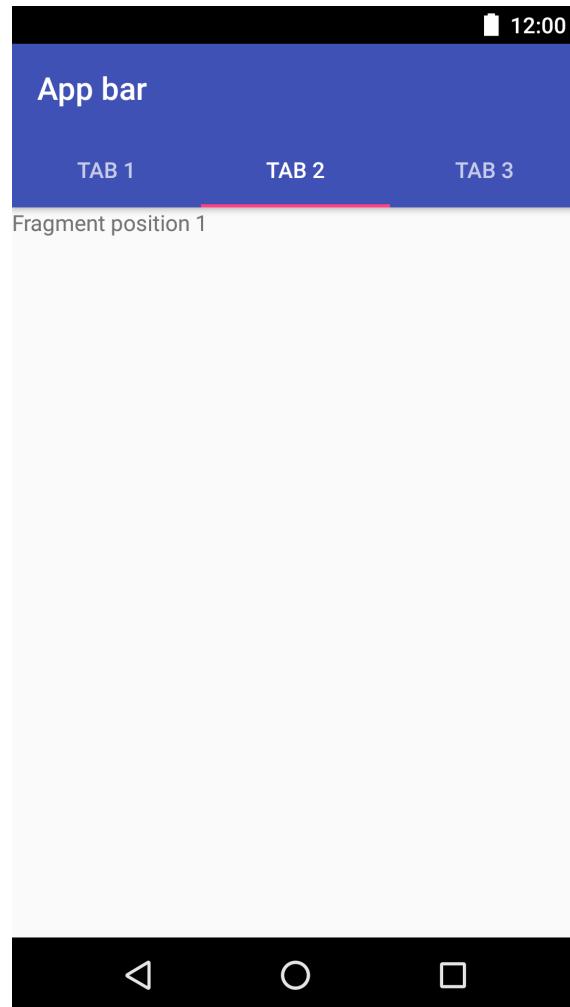
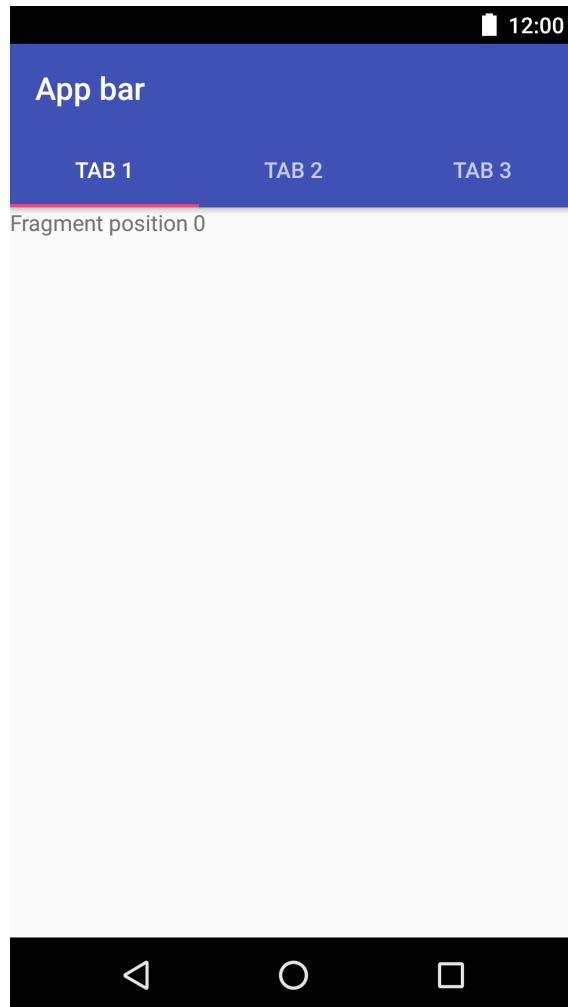
    <android.support.design.widget.AppBarLayout
        android:id="@+id/app_bar"
        android:layout_width="match_parent"
        android:layout_height="250dp"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
    </android.support.design.widget.AppBarLayout>

    <android.support.v4.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior">
    </android.support.v4.widget.NestedScrollView>

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/action"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:src="@drawable/ic_add"
        app:layout_anchor="@id/app_bar"
        app:layout_anchorGravity="bottom|right|end" />

</android.support.design.widget.CoordinatorLayout>
```

Tabs dans l'App bar



```
fragment_main.xml
<LinearLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
```

```
MyFragment.java
public class MyFragment extends Fragment {
    private int tab;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        tab = getArguments().getInt("tab");
    }

    // Appelé lors de la création de la vue du Fragment
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle b) {
        // Récupération du layout du fragment
        final View view = inflater.inflate(R.layout.fragment_main, null, false);
        final TextView textView = (TextView) view.findViewById(R.id.text);
        textView.setText("Fragment position " + tab);
        return view;
    }
}
```

```
<android.support.design.widget.CoordinatorLayout
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"/>

        <android.support.design.widget.TabLayout
            android:id="@+id/tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            // Les tabs prennent toute la place disponible en largeur
            app:tabGravity="fill"
            // Les tabs ne scroll pas
            app:tabMode="fixed" />
    </android.support.design.widget.AppBarLayout>

    <android.support.v4.view.ViewPager
        android:id="@+id/viewpager"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior" />
</android.support.design.widget.CoordinatorLayout>
```

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_tab);  
  
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
    ViewPager viewPager = (ViewPager) findViewById(R.id.viewpager);  
    TabLayout tabLayout = (TabLayout) findViewById(R.id.tabs);  
  
    setSupportActionBar(toolbar);  
    viewPager.setAdapter(new ViewPagerAdapter(getSupportFragmentManager()));  
    tabLayout.setupWithViewPager(viewPager);  
}
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    viewPager.setAdapter(new ViewPagerAdapter(getSupportFragmentManager()));
    tabLayout.setupWithViewPager(viewPager);
}

class ViewPagerAdapter extends FragmentPagerAdapter {

    public ViewPagerAdapter(FragmentManager manager) { super(manager); }

    // Appelé pour chaque tab avec en paramètre la position de la tab
    @Override
    public Fragment getItem(int position) {
        // Définition des données qu'on souhaite passer à un fragment
        Bundle argument = new Bundle();
        argument.putInt("tab", position);
        // Création du fragment
        MainFragment fragment = new MainFragment();
        fragment.setArguments(argument);
        return fragment;
    }

    @Override
    public int getCount() { return 3; } // Nombre de tabs

    @Override
    public CharSequence getPageTitle(int position) {
        // Titre de chaque tab
        return "TAB " + position;
    }
}
```