

TP: SnackBar

Introduction

Android Design Support Library a été présenté en même temps que la nouvelle version d'Android Marshmallow, mais elle peut être utilisée sur des appareils de la version Android 2.1. La bibliothèque a été conçue spécifiquement pour le nouveau design Material Design et permet l'utilisation de composants prêts à l'emploi dans le nouveau style.

Objectif du TP

Apprendre à maîtriser les outils de notification intégré: SnackBar & Toast. Et ainsi comprendre leurs différences.

Pour obtenir la version initial de ce TP:

<https://drive.google.com/file/d/0BzwmvSTBObi1NG1nWUhPNEJnekE/view?usp=sharing>

Pour obtenir la version finale de ce TP:

<https://drive.google.com/file/d/0BzwmvSTBObi1QTJDVkd0U2V0TkE/view?usp=sharing>

Courte description sur le SnackBar Android

SnackBar - widget de la bibliothèque de conception de matériel (Material Design) pour remplacer les messages pop-up Toast.

Le message indiqué pour un certain temps, en fonction de la durée du rétro-éclairage, puis se cachent automatiquement. En outre, les utilisateurs peuvent le cacher d'un geste sans attendre le délai. C'est presque similaire au comportement de l'ancien Toast.

SnackBar est un widget simple et fonctionnel, qui est utilisé pour afficher des messages dans le bas de l'application. En outre, ce widget peut contenir des boutons programmables supplémentaires pour différentes actions.



Les différences entre Toast et Snackbar

- Les toast-messages peuvent être ajustés partout sur l'écran, et Snackbar peut être affiché uniquement dans le bas de l'écran.
- Pour les toast-messages il n'est pas possible d'ajouter des contrôles. Et le widget Snackbar ne peut contenir qu'un seul bouton.
- Les toast-messages ne peuvent pas être fermés avant la fin du temps spécifié par le programmeur, tandis que Snackbar peut être fermé à n'importe quel moment.

Etape 1: Snackbar

Exemple de création d'un widget Snackbar

Avant la création d'un widget Snackbar, il faut déclarer un objet Button dans la classe MainActivity.

```
public class MainActivity extends AppCompatActivity {
    Button snackBarButton;
    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        snackBarButton = (Button) findViewById(R.id.snackBarButton);
        ...
    }
}
```

Et après on doit utiliser la méthode mise en page `Snackbar.make()` avec trois paramètres:

```
snackBarButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Snackbar snackbar = Snackbar.make(coordinatorLayout, "Download failed",
        Snackbar.LENGTH_LONG);
        snackbar.show();
    }
});
```

- Dans ce code le paramètre `coordinatorLayout` ci-dessus représente l'élément racine de l'Activité.
- Le paramètre «`Download failed`» est un message texte qui sera affiché dans le Snackbar
- `Snackbar.LENGTH_LONG` correspond à combien de temps le message sera affiché sur l'écran.

L'appel de méthode `show()` affichera sur l'écran message layed par la méthode `make()`.

Ajouter le bouton d'action

Sur le panneau de Snackbar, il peut être placé une bouton défini par la méthode d'action `setAction()`. Dans la méthode, on doit spécifier le texte du bouton et un gestionnaire pour le clic. Et dans cette méthode on peut ajouter une procédure `onClick` qui décrit l'apparition du message texte après avoir cliqué sur le bouton Snackbar.

```

snackBarButton.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {
        Snackbar snackbar = Snackbar.make(coordinatorLayout, "Download failed",
        Snackbar.LENGTH_LONG);
        snackbar.setAction("Retry", new View.OnClickListener() {

            @Override
            public void onClick(View view) {
                Snackbar snackbar = Snackbar.make(coordinatorLayout, "Download completed",
                Snackbar.LENGTH_LONG);
                snackbar.show();
            }
        });
        snackbar.show();
    }
});

```

Etape 2: Toast

Exemple de création d'un Toast

Pour créer un Toast, il faut ajouter un bouton via "activity_main.xml" puis ajouter `android:onClick="showToast"` dans la balise Button.

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Show Toast"
    android:id="@+id/showToast"
    android:nestedScrollingEnabled="true"
    android:onClick="showToast"/>

```

Il faut déclarer un objet de type Toast.

```

Toast toast;

```

Il doit être initialisé avec la méthode `Toast.makeText ()`, puis appeler la méthode `show()` pour afficher un message sur l'écran:

```

public void showToast(View view) {
    Toast toast = Toast.makeText(getApplicationContext(),
        "Some toast", Toast.LENGTH_LONG);
    toast.setGravity(Gravity.CENTER, 0, 0); //une notification apparaît dans le centre de l'écran
    toast.show();
}

```

Le méthode `makeText ()` a trois paramètres:

- Contexte d'application;
- Un message texte;
- La longueur des notifications d'affichage du temps.

Etape 3: Bonus

Toast avec image

En règle générale, Toast utilise des messages texte courts. Si nécessaire, vous pouvez ajouter une image au message. L'utilisation de la méthode `getView()` appartenant à l'objet Toast, vous permet d'afficher un Toast sur n'importe quelle vue.

Il est nécessaire d'ajouter l'image dans le dossier `app/src/main/res/drawable` du projet.

```
public void showToast(View view) {
    Toast toast = Toast.makeText(getApplicationContext(),
        "Some toast", Toast.LENGTH_LONG);
    //Add picture
    LinearLayout toastContainer = (LinearLayout) toast.getView();
    ImageView imV = new ImageView(getApplicationContext());
    imV.setImageResource(R.drawable.camera);
    toastContainer.addView(imV, 0);
    toast.show();
}
```

Changer la couleur du texte du bouton d'action

Dans la méthode `setActionTextColor()`, on peut changer la couleur du texte du bouton d'action.

```
@Override
public void onClick(View view) {
    Snackbar snackbar = Snackbar.make(coordinatorLayout, "Download failed",
    Snackbar.LENGTH_LONG);
    snackbar.setAction("Retry", new View.OnClickListener() {
        ... //répétition du code ci-dessus
    });

    snackbar.setActionTextColor(Color.RED);
    snackbar.show();
}
});
```

Définir du texte dans le Snackbar

Pour créer un widget Snackbar avec un texte personnalisé, il faut déclarer un objet Button et un objet EditText dans la classe MainActivity.

```
Button sendTextButton;
EditText editText;

@Override
protected void onCreate(Bundle savedInstanceState) {
    editText = (EditText) findViewById(R.id.editText);
    sendTextButton = (Button) findViewById(R.id.sendTextButton);
    ...
}
```

Créer une procédure permettant à l'utilisateur d'écrire son texte :

- Il faut créer un editText qui va contenir le texte;
- Il faut créer une variable `textMessage` de type string qui va récupérer le texte entré par l'utilisateur;
- Il faut enfin appeler la fonction `Snackbar.make` qui va appeler la variable `textMessage`.

```
sendTextButton.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        EditText editText = (EditText) findViewById(R.id.editText);  
        String textMessage = editText.getText().toString();  
        Snackbar snackbar =  
        Snackbar.make(coordinatorLayout, textMessage, Snackbar.LENGTH_LONG);  
        snackbar.show();  
    }  
});
```

Changer la couleur du texte

La classe de Snackbar contient une méthode `getView()` qui retourne une vue des composants. Après avoir reçu cette méthode, on peut changer la couleur de fond. Si vous voulez aussi changer la couleur du texte dans un message, vous devez avoir accès à la TextView:

```
sendTextButton.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
  
        ...//répétition du code ci-dessus  
        //Change the color of text sent  
        TextView textView = (TextView)  
        snackbar.getView().findViewById(android.support.design.R.id.snackbar_text);  
        textView.setTextColor(Color.GREEN);  
    }  
});
```

En pratique, il est rare de changer fréquemment la couleur, mais ça peut toujours être utile.

Fin du TP

C'est tout pour ce TP qui, je l'espère, vous a permis de comprendre comment créer un Snackbar simple, un Toast simple et ainsi voir les différences entre eux. Je vous invite alors à visiter ces liens pour en apprendre davantage sur les Snackbar et Toast :

<https://developer.android.com/reference/android/widget/Toast.html>

<https://developer.android.com/reference/android/support/design/widget/Snackbar.html>