**从登陆页开始TextInputLayout**

**前言**

[源码传送门](https://github.com/shiquanliao/TextInputLayout)



效果图

**添加依赖**

TextInputLayout是在Material Design中的,如果我们要使用的话,必须在gradle文件中配置

dependencies {

compile 'com.android.support:appcompat-v7:25.0.1'

compile 'com.android.support:design:25.0.1'

}

**依赖完成之后开始使用**

在[TextInputLayout官方文档API](https://developer.android.com/reference/android/support/design/widget/TextInputLayout.html)中描述了它是一种新的继承自LinearLayout的布局,让我们使用时里面只能包含一个EditText或者其子类的控件,该布局可以通过设置hint和Error显示浮动标签.接下来我们看看布局文件

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.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.support.design.widget.TextInputLayout

android:layout\_width="368dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="16dp"

app:layout\_constraintLeft\_toLeftOf="@+id/til\_phone"

app:layout\_constraintRight\_toRightOf="@+id/til\_phone"

app:layout\_constraintTop\_toBottomOf="@+id/til\_phone"

android:id="@+id/textInputLayout">

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_password"

android:inputType="textPassword"/>

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

<android.support.design.widget.TextInputLayout

android:id="@+id/til\_phone"

android:layout\_width="368dp"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="16dp"

android:layout\_marginLeft="16dp"

android:layout\_marginRight="16dp"

android:layout\_marginStart="16dp"

android:layout\_marginTop="16dp"

app:counterEnabled="true"

app:counterMaxLength="11"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_phone"

android:inputType="phone"/>

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

<Button

android:text="@string/btn\_text"

android:layout\_width="0dp"

android:textColor="@android:color/white"

android:layout\_height="wrap\_content"

android:id="@+id/button"

android:layout\_marginStart="16dp"

app:layout\_constraintLeft\_toLeftOf="parent"

android:layout\_marginLeft="16dp"

android:layout\_marginEnd="16dp"

app:layout\_constraintRight\_toRightOf="parent"

android:layout\_marginRight="16dp"

android:background="@color/colorPrimary"

android:layout\_marginTop="16dp"

app:layout\_constraintTop\_toBottomOf="@+id/textInputLayout"/>

</android.support.constraint.ConstraintLayout>

使用TextInputLayout其实很简单,只需要我们使用TextInputLayout容器并且在其中加入一个EditText,需要我们特别注意的是:**一个TextInputLayout有且只有一个EditText**,  
如果我们想要浮动标签的效果只需要在TextInputLayout或者在EditText中设置hint,但是不能在两者中同时设置hint属性,假如我们在TextInputLayout跟EditText都设置了hint属性时,当EditText没有获取到焦点时,这时hint会显示(但是要注意,这时是两个hint重叠显示)  
如果我们想要关闭浮动标签(默认是开启的),可以加入下面一句话

app:hintEnable="false"

浮动标签显示与隐藏之间有一个过渡动画(默认是开启的),可以通过以下方法设置

app:hintAnimationEnabled="boolean"

如果我们想要在账号那栏后面加个字数统计,例如当我们的账号是固定位数时(例如手机号码,),我们可以让用户看到自己当前输入的长度,我们只需要在TextInputLayout加入

app:counterEnabled="true"

默认是关闭的,我们也能设置一个输入的最大长度

app:counterMaxLength="11"



这个设置不是说用户输入了11位账号之后就不能输入了,而是会用一种颜色提示用户你的输入长度超过了设置长度.  
TextInputLayout也很友好,为我们提供了设置错误提醒的功能(又为我们争取了一点品咖啡的时间),

textInputEditText.setError(CharSequence error);

当我们调用了这个方法之后提示是会一直在的,所以我们要当用户输入正确之后我们手动调用textInputEditText,setErro(null)去取消错误提示.我们可以通过addTextChangedListener监听,在合适的时候调用textInputEditText,setErro(null)

我们可能看到过其他的app在密码那栏的最右边有个👁的图标,TextInputLayout也为我们提供了这个功能

app:passwordToggleEnabled="true" //默认是关闭的

我们可能用我们自己设计的图标

app:passwordToggleDrawable="@mipmap/ic\_launcher"

passwordToggleTint给图标加颜色

app:passwordToggleTint="@color/colorAccent"

passwordToggleTintMode设置模式

//screen,src\_in,src\_atop,src\_over,multiply等几种设计模式

app:passwordToggleTintMode="screen" //screen

到此,你花了多少时间,是不是很简单.

**自定义浮动标签**

在TextInputLayout中加入

app:hintTextAppearance="@style/hintAppearance"

<style name="hintAppearance" parent="TextAppearance.AppCompat">

<item name="android:textSize">14sp</item>

<item name="android:textColor">#ffee00</item>

</style>

**自定义错误提示信息样式**

在TextInputLayout中加入

app:errorTextAppearance="@style/errorAppearance"

<style name="errorAppearance" parent="TextAppearance.AppCompat">

<item name="android:textSize">14sp</item>

<item name="android:textColor">@color/red</item>

</style>

**监听虚拟键盘**

有时候我们在用虚拟键盘的时候,在输入账号之后,我们点击下一项,焦点会跑到输入密码这一栏,当我们密码输入完成之后,点击确定按钮就能登录了,这个怎么实现的呢.  
账号EditText中加入

android:imeActionId="@+id/password"

android:imeOptions="actionNext"

密码EditText中加入

android:imeActionId="@+id/button"

android:imeOptions="actionDone"

mPasswordView.setOnEditorActionListener(new TextView.OnEditorActionListener() {

@Override

public boolean onEditorAction(TextView textView, int id, KeyEvent keyEvent) {

if ( id == EditorInfo.IME\_ACTION\_DONE) {

InputMethodManager inputMethodManager=(InputMethodManager)getSystemService(Context.INPUT\_METHOD\_SERVICE);

//inputMethodManager.showSoftInput(getWindow().getDecorView(),InputMethodManager.SHOW\_FORCED);//显示

inputMethodManager.hideSoftInputFromWindow(getWindow().getDecorView().getWindowToken(),InputMethodManager.RESULT\_UNCHANGED\_SHOWN);

//attemptLogin();

startLogin();

return true;

}

return false;

}

});

对于账号EditText的优化我们可能通过AutoCompleteTextView实现自动提示功能.