

昨天读书：《MacTalk 人生元编程》，读到的一些信息：

写作即思考。写作其实就代表思考，你需要言之有物，需要架构、需要梳理，要有开端、有结尾、有结论，有主题，特别神奇的是你构思一片文章。写完后发现文章像具备了生命一样生长出了很多奇异的果实，他们就在那些文字中间微微颤动，闪烁着独特的光泽，仿佛被岁月冲刷过的鹅卵石一样，而这一切你可以完全没有想到过，而且不可复制。

优秀的写作者不仅能让事情变的容易理解，而且能够换位思考，沟通顺畅思维敏捷。与这样的程序员交流是赏心悦目的。遇到问题时他会抽丝剥茧，告诉你问题的前因后果，由表及里，并且能够反映问题的各种信息都提供给你，包括他自己尝试解决问题的措施和结果。所以，为自己写作！

那么，我为什么不为自己写作呢？从现在开始！

我在学习这个登录模块时学到的东西：

- 1、登录界面的布局；
- 2、登录流程，用户名和密码的加密、解密，存储和恢复显示；
- 3、输入法的调用、隐藏；
- 4、在输入法中点击“完成”和在界面点击“登录”后，不同的处理方式；
- 5、登录验证：登录成功、失败的不同提示。

下面将会按照这5个顺序来介绍程序中的登录模块：

一、登录界面的布局

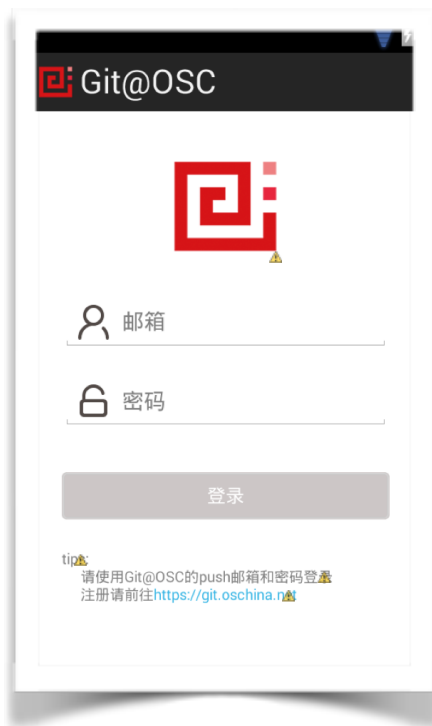
在Manifest文件中Activity的配置如下：

```
<activity
    android:name=".ui.LoginActivity"
    android:label="@string/login_title"
    android:launchMode="singleTask"
    android:screenOrientation="portrait"
    android:windowSoftInputMode="adjustUnspecified" >
```

需要配置：

- 1、`android:screenOrientation="portrait"` 这是为了登录界面始终保持为竖屏方向。
- 2、`android:windowSoftInputMode="adjustUnspecified"` 这是设置的输入法的一个属性，特点：`"adjustUnspecified"` 它不被指定是否该Activity主窗口调整大小以便留出软键盘的空间，或是否窗口上的内容得到屏幕上当前的焦点是可见的。系统将自动选择这些模式中一种主要依赖于是否窗口的内容有任何布局视图能够滚动他们的内容。如果有这样的一个视图，这个窗口将调整大小，这样的假设可以使滚动窗口的内容在一个较小的区域中可见的。这个是主窗口默认的行为设置。

布局代码如下：



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical" >

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/icon_login_logo" />

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:orientation="vertical"
        android:padding="25dip" >

        <AutoCompleteTextView
            android:id="@+id/login_account"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
```

```
        android:drawableLeft="@drawable/
icon_login_account"
        android:drawablePadding="10dip"
        android:ems="10"
        android:hint="@string/login_account_hint"
        android:imeOptions="actionNext"
        android:inputType="textEmailAddress"
        android:singleLine="true" />
```

<EditText

```
        android:id="@+id/login_password"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dip"
        android:drawableLeft="@drawable/icon_login_pass"
        android:drawablePadding="10dip"
        android:ems="10"
        android:hint="@string/login_password_hint"
        android:inputType="textPassword"
        android:singleLine="true" />
```

<Button

```
        android:id="@+id/login_btn_login"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="35dip"
        android:background="@drawable/btn_style_login"
        android:enabled="false"
        android:padding="10dip"
        android:text="@string/login_title"
        android:textColor="@color/white"
        android:textSize="@dimen/space_15" />
```

<LinearLayout

```
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_marginTop="25dip" >
```

<TextView

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
android:layout_gravity="left"
android:text="tips:"
android:textColor="@color/gray"
android:textSize="12sp" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="请使用Git@OSC的push邮箱和密码登录"
    android:layout_marginLeft="15dip"
    android:textColor="@color/gray"
    android:textSize="12sp" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="left"
    android:autoLink="web"
    android:layout_marginLeft="15dip"
    android:text="注册请前往https://
git.oschina.net"
    android:textColor="@color/gray"
    android:textSize="12sp" />
```

```
</LinearLayout>
</LinearLayout>
```

```
</LinearLayout>
```

可以看到整个界面就是一个线性布局，可以借鉴的地方是：

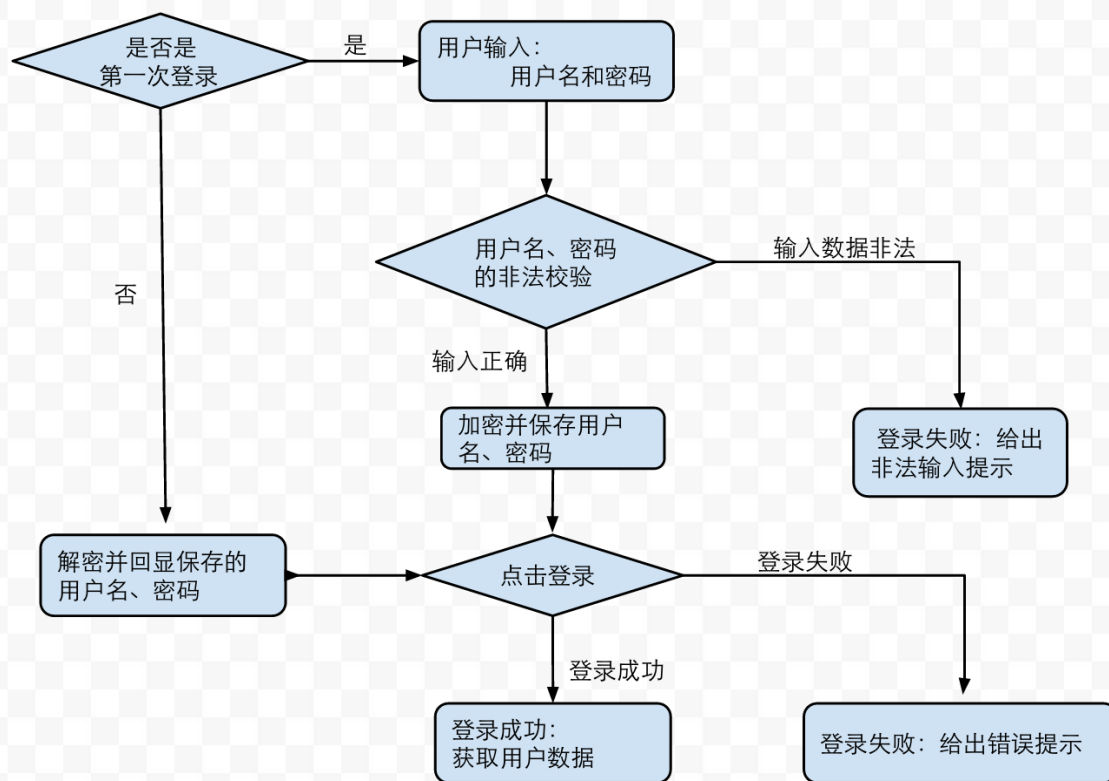
1、在用户名和密码输入框中都有一个属性：

```
android:drawableLeft="@drawable/icon_login_account"
```

可以设置出一个小图片来，替代丑陋的文字：用户名、密码，将这些文字改为hint值。

二、用户名和密码的加密、解密，存储和恢复显示

登录流程如下图所示：



用户名和密码的加密与解密：

用户名和密码属于敏感型数据，涉及到用户隐私和安全等重要信息，因此要进行加密存储，加密传输，源码中的加密是通过DES算法进行加密的，具体的加密过程，我想单独研究一下这个，另外总结一下！

加密并存储用户信息的代码：

// 保存用户名和密码

```
mAppContext.saveAccountInfo(CryptoUtils.encode(Contanst.ACCOUNT_EMAIL, email),  
CryptoUtils.encode(Contanst.ACCOUNT_PWD, passwd));
```

解密并回显用户信息的代码：

//解密回显用户名信息

```
String account = CryptoUtils.decode(Contanst.ACCOUNT_EMAIL,  
mAppContext.getProperty(Contanst.ACCOUNT_EMAIL));  
mAccountEditText.setText(account);
```

//解密回显用户密码信息

```
String pwd = CryptoUtils.decode(Contanst.ACCOUNT_PWD,  
mAppContext.getProperty(Contanst.ACCOUNT_PWD));  
mPasswordEditText.setText(pwd);
```

具体加密代码，在CryptoUtils.java中。

三、输入法的调用和显示：

1、要使用输入法，必须获取输入法的服务：

```
InputMethodManager imm = (InputMethodManager)
getSystemService(INPUT_METHOD_SERVICE);
```

2、输入法的隐藏，在输入法中点击了完成后，进行的响应操作：

//在输入法里点击了“完成”，则去登录

```
if(actionId == EditorInfo.IME_ACTION_DONE) {
    checkLogin();
    //将输入法隐藏
    InputMethodManager imm = (InputMethodManager)getSystemService(
        Context.INPUT_METHOD_SERVICE);
    imm.hideSoftInputFromWindow(mPasswordEditText.getWindowToken(), 0);
    return true;
}
```

总之，在使用输入法之前，需要获取输入法服务！

四、在输入法中点击“完成”和在界面点击“登录”后，不同的处理方式

在输入法中点击“完成”后登录的代码：

```
if(actionId == EditorInfo.IME_ACTION_DONE) {
    checkLogin();
    //将输入法隐藏
    InputMethodManager imm = (InputMethodManager)getSystemService(
        Context.INPUT_METHOD_SERVICE);
    imm.hideSoftInputFromWindow(mPasswordEditText.getWindowToken(), 0);
    return true;
}
```

在界面中点击“登录”按钮后登录的代码：

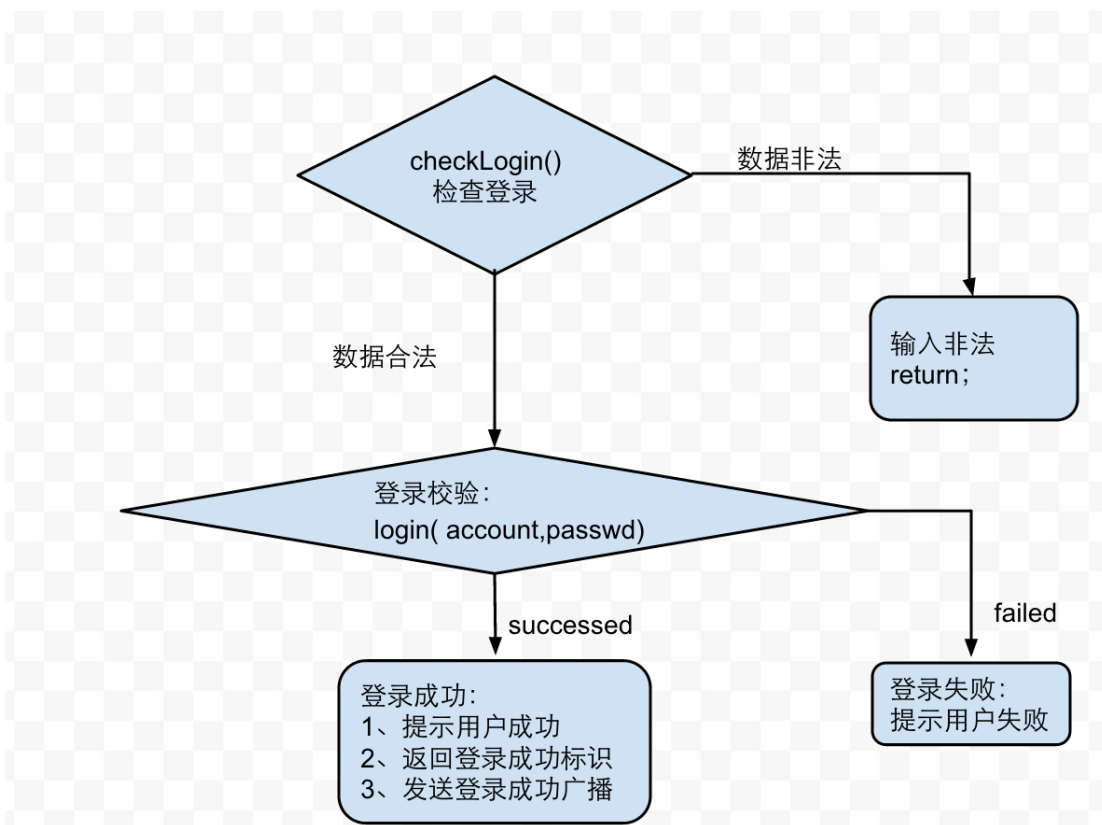
```
@Override
public void onClick(View v) {
    imm.hideSoftInputFromWindow(mPasswordEditText.getWindowToken(), 0);
    checkLogin();
}
```

两种方式都会完成如下流程：

- 1、隐藏输入法；
- 2、校验输入数据是否合法；
- 3、进行登录；

五、登录验证：登录成功、失败的不同提示：

如下流程图所示，即登录校验详情流程图：



这块儿的疑问：

- 1、登录成功发送这个广播干嘛的？这个广播发送给谁了？接受这个广播能干嘛？