# 格力账号客户端接口使用说明

## 一、启动界面：

说明：启动界面的方法主要在类：com.gree.greeaccountlib.util.LaunchActivityManager

若要启动这些界面，则必需要安装GreeAccount.apk

### 启动格力账号主界面：

说明：界面启动方式有两种，一种是调用jar提供的接口方法直接启动，另一种是采用给定的action和Scheme Uri来启动；

#### 1.1,接口方式启动：

public boolean launchAccountMainUI([LaunchActivityManager.LaunchActivityListener](file:///F:\mywork\MySVN\AppProject\GreeAccount\04_doc\GreeAccountLib%20Api\com\gree\greeaccountlib\util\LaunchActivityManager.LaunchActivityListener.html) ll, boolean isGuide)

说明：启动格力账号主界面

**参数:**

ll - ：界面状态的返回接口

isGuide - ：是否启动新机引导模式

**返回:**

是否安装格力账号界面APK

**例：**

LaunchActivityManager.*getInstance*(MainActivity.**this**).launchAccountMainUI(uiCall, **false**);

**private** LaunchActivityListener uiCall = **new** LaunchActivityListener() {

@Override

**public** **void** launchUICallBack(UILaunchBean arg0) {

**if** (**null** != arg0) {

**int** uiFlag = arg0.getUiFlag();

**boolean** isSucceed = arg0.isEventIsSucceed();

String msg = arg0.getEventMsg();

**switch** (uiFlag) {

**case** LaunchUIType.***LAUNCH\_UI\_ACCOUNT\_MAIN***:

**break**;

**default**:

**break**;

}

} **else** {

LogUtils.*e*(TAG, "launchUICallBack UILaunchBean is null");

}

}

};

#### 1.2，action方式启动：

Action: com.gree.greeaccountlib.common. Gconstants. LAUNCH\_ACTIVITY\_ACCOUNT\_MAIN\_URI

Scheme Uri: com.gree.greeaccountlib.common. Gconstants. SCHEME\_URI

**例：**

Intent intent = **new** Intent(

GConstants.***LAUNCH\_ACTIVITY\_ACCOUNT\_MAIN\_URI***, Uri

.*parse*(GConstants.***SCHEME\_URI***));

startActivity(intent);

### 启动登录界面：

#### 2.1,接口方式启动：

public boolean launchLoginUI([LaunchActivityManager.LaunchActivityListener](file:///F:\mywork\MySVN\AppProject\GreeAccount\04_doc\GreeAccountLib%20Api\com\gree\greeaccountlib\util\LaunchActivityManager.LaunchActivityListener.html) ll, boolean isGuide)

说明：启动登录界面

**参数:**

ll - ：界面状态的返回接口

isGuide - ：是否启动新机引导模式

**返回:**

是否已经安装格力账号界面APK

**例：**

LaunchActivityManager.*getInstance*(MainActivity.**this**)

.launchLoginUI(uiCall, **true**);

**private** LaunchActivityListener uiCall = **new** LaunchActivityListener() {

@Override

**public** **void** launchUICallBack(UILaunchBean arg0) {

**if** (**null** != arg0) {

**int** uiFlag = arg0.getUiFlag();

**boolean** isSucceed = arg0.isEventIsSucceed();

String msg = arg0.getEventMsg();

**switch** (uiFlag) {

**case** LaunchUIType.***LAUNCH\_UI\_LOGIN***:// 登录界面的返回

**break**;

**default**:

**break**;

}

} **else** {

LogUtils.*e*(TAG, "launchUICallBack UILaunchBean is null");

}

}

};

#### 2.2，action方式启动：

Action: com.gree.greeaccountlib.common. Gconstants.LAUNCH\_ACTIVITY\_LOGIN\_URI

Scheme Uri: com.gree.greeaccountlib.common. Gconstants. SCHEME\_URI

**例：**

Intent intent = **new** Intent(GConstants. ***LAUNCH\_ACTIVITY\_LOGIN\_URI***, Uri

.*parse*(GConstants.***SCHEME\_URI***));

startActivity(intent);

### 启动注册界面（通过手机注册和通过邮箱注册）：

#### 3.1, 接口方式启动：

* + - public boolean launchRegisterUI([LaunchActivityManager.LaunchActivityListener](file:///F:\mywork\MySVN\AppProject\GreeAccount\04_doc\GreeAccountLib%20Api\com\gree\greeaccountlib\util\LaunchActivityManager.LaunchActivityListener.html) ll, boolean isByPhone, boolean isGuide)

说明：启动注册界面，包括通过手机注册和通过邮箱注册界面

**参数:**

ll - ：界面状态的返回接口

isByPhone - ：是否通过手机号注册

isGuide - ：是否启动新机引导模式

**返回:**

是否已经安装格力账号界面APK

**例：**

LaunchActivityManager.*getInstance*(MainActivity.**this**)

. **launchRegisterUI** (uiCall, **true,true**);

**private** LaunchActivityListener uiCall = **new** LaunchActivityListener() {

@Override

**public** **void** launchUICallBack(UILaunchBean arg0) {

**if** (**null** != arg0) {

**int** uiFlag = arg0.getUiFlag();

**boolean** isSucceed = arg0.isEventIsSucceed();

String msg = arg0.getEventMsg();

**switch** (uiFlag) {

**case** LaunchUIType.***LAUNCH\_UI\_REGISTER***:// 注册界面的返回

**break**;

**default**:

**break**;

}

} **else** {

LogUtils.*e*(TAG, "launchUICallBack UILaunchBean is null");

}

}

};

#### 3.2, action方式启动：

Action1:

com.gree.greeaccountlib.common. Gconstants. LAUNCH\_ACTIVITY\_REGISTER\_URI(通过手机注册)

Action2:

com.gree.greeaccountlib.common. Gconstants. LAUNCH\_ACTIVITY\_REGISTER\_BYEMAIL\_URI (通过邮箱注册)

Scheme Uri: com.gree.greeaccountlib.common. Gconstants. SCHEME\_URI

**例：**

Intent intent = **new** Intent(GConstants. ***LAUNCH\_ACTIVITY\_REGISTER\_URI***, Uri

.*parse*(GConstants.***SCHEME\_URI***));

startActivity(intent);

## 二、业务接口：

* 说明：调用接口的方法主要在类：com.gree.greeaccountlib.util.CloudApiUtil

### 1、登录接口

* + - public void startLogin(Context ctx, java.lang.String userNamw, java.lang.String pwd, [ApiAsyncTask.CloudApiBackListener](file:///F:\mywork\MySVN\AppProject\GreeAccount\04_doc\GreeAccountLib%20Api\com\gree\greeaccountlib\asynctask\ApiAsyncTask.CloudApiBackListener.html) cl)
    - 说明：登录

**参数:**

ctx -

userNamw：手机号或者邮箱号 -

pwd：登录密码 -

cl：返回结果的接口 –

**例：**

CloudApiUtil.*getInstance*().startLogin(LoginActivity.**this**,

mPhoneOrEmailEt.getText().toString(),

mPasswordView.getText().toString(), apiCall);

### 2、通过手机号注册接口

* + - public void startRegisterByPhone(Context ctx, java.lang.String pwd, java.lang.String phone, [ApiAsyncTask.CloudApiBackListener](file:///F:\mywork\MySVN\AppProject\GreeAccount\04_doc\GreeAccountLib%20Api\com\gree\greeaccountlib\asynctask\ApiAsyncTask.CloudApiBackListener.html) cl)

说明：通过手机号注册

**参数:**

ctx -

pwd - ：密码

phone - ：手机号

cl - ：返回结果的接口

例：

CloudApiUtil.*getInstance*().startRegisterByPhone(RegisterByPhoneActivity.**this**, mPwdConfimEt.getText().toString(),mPhoneNumEt.getText().toString(), apiCall);

### 3、通过邮箱号注册接口

* + - public void startRegisterByEmail(Context ctx, java.lang.String pwd, java.lang.String email, [ApiAsyncTask.CloudApiBackListener](file:///F:\mywork\MySVN\AppProject\GreeAccount\04_doc\GreeAccountLib%20Api\com\gree\greeaccountlib\asynctask\ApiAsyncTask.CloudApiBackListener.html) cl)

说明：通过邮箱号注册

**参数:**

ctx -

pwd - ：密码

phone - ：邮箱号

cl - ：返回结果的接口

**例：**

CloudApiUtil.getInstance().startRegisterByEmail(RegisterByEmailActivity.this, mPwdConfimEt.getText().toString(), mEmailEt.getText().toString(),apiCall);

CloudApiUtil.getInstance().startRegisterByEmail(RegisterByEmailActivity.this, mPwdConfimEt.getText().toString(), mEmailEt.getText().toString(),apiCall);

### 4、获取当前登录账号的私钥

* + - public java.lang.String getCurrentGKey(Context context)

说明：获取当前私钥

**参数:**

ctx -

**返回:私钥**

**例：**

String gkey = CloudApiUtil.*getInstance*().getCurrentGKey(MainActivity.**this**);

### 5、获取当前登录账号的Token

* + - public java.lang.String getCurrentToken(Context context)

获取当前的token

**参数:**

ctx -

**返回:token**

**例：**

UserInfoBean f = CloudApiUtil.getInstance().readlocalUserInfo(MainActivity.this);

String token = CloudApiUtil.*getInstance*().getCurrentToken(MainActivity.**this**);

### 6、获取当前登录账号的用户信息（本地数据）：

* + - public [UserInfoBean](file:///F:\mywork\MySVN\AppProject\GreeAccount\04_doc\GreeAccountLib%20Api\com\gree\greeaccountlib\bean\UserInfoBean.html) readlocalUserInfo(Context context)

说明：获取当前登录的用户信息：从本地数据库获取用户信息

**参数:**

ctx -

**例：**

### 7、获取当前登录账号的用户信息（从服务器上获取）

* + - public void startGetCurUserInfo(Context ctx, [ApiAsyncTask.CloudApiBackListener](file:///F:\mywork\MySVN\AppProject\GreeAccount\04_doc\GreeAccountLib%20Api\com\gree\greeaccountlib\asynctask\ApiAsyncTask.CloudApiBackListener.html) cl)

说明：获取当前登录的用户信息（从服务器上获取最新的用户信息）

**参数:**

ctx -

cl - ：返回结果的接口

**例：**

**if** (CloudApiUtil.*getInstance*()

.isLoginCurUser(MainActivity.**this**)) {

CloudApiUtil.*getInstance*().startGetCurUserInfo(

MainActivity.**this**, apiCall);

} **else** {

Toast.*makeText*(MainActivity.**this**, "未登录", 2000).show();

}

### 8、更新当前账号的用户信息

* + - public void startUpdateCurUserInfo(Context ctx, java.lang.String email, [ApiAsyncTask.CloudApiBackListener](file:///F:\mywork\MySVN\AppProject\GreeAccount\04_doc\GreeAccountLib%20Api\com\gree\greeaccountlib\asynctask\ApiAsyncTask.CloudApiBackListener.html) cl, int requestType)

说明：更新当前账号的用户信息

### 9. RequestType介绍：

（详见GreeAccountLib\_Api\_doc）

