

友盟 Android 分享组件开发手册

友盟社会化组件，可以让移动应用快速具备社会化分享、登录、评论、喜欢等功能，并提供实时、全面的社会化数据统计分析服务。

前提说明

假设在你的Activity中声明了如下成员变量,文档中将使用该成员变量进行函数调用.

- 1、UMSocialService mController = UMServiceFactory.getUMSocialService("com.umeng.share");
- 2、文档中出现YourActivity.this或者mContext 的地方开发者替换成要调用函数时用户所在的Activity即可；
- 3、文档中出现的mUmengSdkButton假设为你的布局文件中的某个按钮;
- 4、与摇一摇相关的功能请参考[线上摇一摇集成文档](#)。

目录

友盟 Android 分享组件开发手册.....	1
一、设置分享内容接口.....	4
1.1、设置要分享的文字内容 (setShareContent).....	4
1.2、设置通用的多媒体分享内容 (setShareMedia).....	4
1.3、设置平台独立的分享内容 (setShareMedia).....	5
二、分享接口.....	6
2.1、打开分享面板 (openShare).....	6
2.2、指定平台分享, 内容可编辑 (postShare).....	7
2.3、直接分享接口 (directShare).....	8
2.4、分享到多个已授权的平台 (postShareMulti).....	9
2.5、分享到短信 (shareSms).....	10
2.6、分享到邮件 (shareEmail).....	10
2.7、注册回调接口 (registerListener).....	11
三、授权相关接口.....	12
3.1、授权接口 (doOAuthVerify).....	12
3.2、用户登录 (Login).....	13
3.3、用户注销 (loginout).....	14
3.4、删除授权 (deleteOAuth).....	15
3.5、获取用户个人信息 (getUserInfo).....	16
3.6、获取平台相关信息 (getPlatformInfo).....	17
3.7、获取好友列表 (getFriends).....	18
四、评论相关接口.....	19
4.1、发表评论 (postComment).....	19
4.2、获取评论列表 (getComments).....	20
4.3、打开评论列表页面 (openComment).....	21
五、用户中心相关接口.....	22
5.1、打开用户中心 (openUserCenter).....	22
六、喜欢相关接口.....	23
6.1、发布 Like (postLike).....	23
6.2、发表 UnLike (postUnLike).....	23
七、SDK 平台配置.....	25
7.1、设置 SSO 授权 (setSsoHandler).....	25
7.2、分享面板上的平台排序 (setPlatformOrder).....	25
7.3、移除分享面板上的平台 (removePlatform).....	26
7.4、获取用户选中的平台 (getSelectedPlatform).....	26

7.5、关注某平台的官方微博 (addFollow).....	26
7.6、设置分享面板显示的平台 (setPlatforms).....	27
7.7、检查是否授权且 Token 没有过期 (isAuthenticatedAndTokenNotExpired).....	28
7.8 设置 sina SSO 授权回调地址 (setSinaCallbackUrl).....	28
7.9、检查授权平台 Token 是否过期 (checkTokenExpired).....	29
7.10、设置是否启用图片缓存(setCacheValidStatus).....	30
7.11、设置短信分享时是否检查 SIM 卡(enableSIMCheck).....	30
7.12 配置 Android 系统 App 分享(supportAppPlatform).....	31
7.13 添加开发者自定义平台(addCustomPlatform).....	31
八、注册摇一摇相关功能.....	33
8.1、注册摇一摇截图分享 (registerShakeListender).....	33
8.2、注册摇一摇打开分享面板 (registerShakeToOpenShare).....	34
8.3、注册摇一摇截图 (registerShakeToScrShot).....	35
8.4、直接截图 (不需要摇一摇) (takeScrShot).....	36
九、线上文档.....	37

一、设置分享内容接口

1.1、设置要分享的文字内容（ setShareContent）

接口原型:

```
public void setShareContent(String content);
```

参数说明：

参数	参数说明
content	要分享的文字内容

调用 setShareContent 接口的示例代码如下：

```
mController.setShareContent("这些文字我要分享到新浪微博");
```

1.2、设置通用的多媒体分享内容（ setShareMedia）

接口原型:

```
public void setShareMedia(UMediaObject media);
```

参数说明：

参数	参数说明
media	要分享的多媒体内容，可传递 UMImage, Umusic, UMVideo对象

要将多媒体内容，例如图片、音乐等分享到社交平台，则需使用该方法。设置的分享内容使用于所有社交平台（设置了平台独立的分享内容平台除外，见 1.24）。

调用 setShareMedia 接口的示例代码如下：

```
// 构建UMImage，参数1为当前的activity 参数2为图片的链接
UMImage shareImage = new UMImage(getActivity(),
    "http://www.umeng.com/images/pic/banner_module_social.png");
// 设置图片分享内容，
```

```
mController.setShareMedia( shareImage );
```

1.3、设置平台独立的分享内容 (setShareMedia)

接口原型:

```
public void setShareMedia(UMediaObject media);
```

参数说明 :

参数	参数说明
media	平台独立的分享内容，可传递例如 SinaShareContent, QzoneShareContent, WeiXinShareContent, WeiXinCircleShareContent, QQShareContent等类型的对象. (命名风格为：平台名 + ShareContent)

在某些情况下，开发者可能希望用户分享到各个平台的内容不一样。开发者可以通过设置不同平台的内容来实现该功能。单独设置了分享内容平台则使用该平台的分享内容，没有设置的则使用通用的分享内容。调用 setShareMedia 接口设置平台内容的示例代码如下：

```
// 构建UMImage，参数1为当前的activity 参数2为图片的链接
UMImage shareImage = new UMImage(getActivity(),
    "http://www.umeng.com/images/pic/banner_module_social.png");

// 设置微信平台的分享内容
// 微信平台的分享内容, 多媒体内容设置为图片, 且只能设置为一种类型
WeiXinShareContent weixinContent = new WeiXinShareContent(shareImage);
weixinContent.setShareContent("来自友盟社会化组件（SDK）让移动应用快速整合社交分享功能，微信");
weixinContent.setTitle("友盟社会化分享组件-微信");
mController.setShareMedia(weixinContent);

// 音乐类型
UMusic uMusic = new
    UMusic("http://sns.whalecloud.com/test_music.mp3");
uMusic.setAuthor("umeng");
uMusic.setTitle("天籁之音");
uMusic.setThumb(mUMImgBitmap);
//设置QQ平台的分享内容
QQShareContent qqShareContent = new QQShareContent( uMusic );
qqShareContent.setShareContent("来自友盟社会化组件（SDK）让移动应用快速整合社交分享功能 - QQ");
qqShareContent.setTitle("hello, qq");
qqShareContent.setTargetUrl("http://www.google.com.hk");
mController.setShareMedia(qqShareContent);
```

注：其他平台的类型请参考SDK中附带的文档。

二、分享接口

2.1、打开分享面板（openShare）

接口原型1：

```
public void openShare(final Activity activity, boolean forceLogin) ;
```

参数说明：

参数	参数说明
activity	调用者activity。使用Social SDK时，打开的分享平台选择面板为Dialog类型，依赖于开发者的Activity。
forceLogin	打开分享面板时是否强制用户登录，默认为false。

调用打开分享面板接口的示例代码如下：

```
public void onClick(View v) {
    // 点击相关按钮，打开分享面板
    if ( v == mUmengSdkButton ) {
        mController.openShare(YourActivity.this, false);
    }
}
```

接口原型2：

```
public void openShare(Activity activity, SnsPostListener listener);
```

参数说明：

参数	参数说明
activity	调用者activity。使用Social SDK时，打开的分享平台选择面板为Dialog类型，依赖于开发者的Activity。
listener	分享后的回调函数。分享后将回调此函数

调用打开分享面板接口的示例代码如下（含有回调）：

```
public void onClick(View v) {
    // 点击相关按钮，打开分享面板
    if ( v == mUmengSdkButton ) {
        mController.openShare(getActivity(), new SnsPostListener() {
```

```

@Override
public void onStart() {
    //分享开始
}

@Override
public void onComplete(SHARE_MEDIA platform, int eCode, SocializeEntity entity) {
    //分享结束，eCode==200代表分享成功，非200代表分享失败
}
});
}
}

```

2.2、指定平台分享，内容可编辑（postShare）

接口原型:

public void postShare(Context context, SHARE_MEDIA platform, SnsPostListener listener)

参数说明：

参数	参数说明
context	调用者 应用的上下文环境。
platform	用户要分享到的目标平台
listener	分享监听器

调用该接口时如果平台已经授权，则直接跳转至分享编辑页面；否则跳转到授权页面完成授权，再跳转到分享编辑页面。目前不支持授权的平台有：微信、微信朋友圈、Facebook、Twitter、G+、易信、来往，这些平台调用postShare不会跳转分享编辑页，直接打开客户端进行分享。调用postShare接口的示例代码如下：

```

public void onClick(View v) {
    // 点击相关按钮进行分享
    if ( v == mUmengSdkButton ) {
        mController.postShare(mContext, SHARE_MEDIA.QZONE, new SnsPostListener() {

            @Override
            public void onComplete(SHARE_MEDIA arg0, int arg1,
                SocializeEntity arg2) {
                Toast.makeText(mContext, "分享完成", Toast.LENGTH_SHORT).show();
            }

            @Override
            public void onStart() {
                Toast.makeText(mContext, "开始分享", Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

2.3、直接分享接口 (directShare)

接口原型：

```
public void directShare(final Context context, final SHARE_MEDIA platform,
    final SnsPostListener listener) {
```

参数说明：

参数	参数说明
context	调用者 应用的上下文环境。
platform	调用者要分享到的目标平台
listener	分享的监听接口，分享开始、完成的回调

如果用户还没有授权目标平台，则需要先进行授权，授权完成后则直接分享，不弹出内容编辑页面。此接口适合开发者自定义UI，然后调用directShare接口进行分享。调用直接分享接口的示例代码如下：

```
public void onClick(View v) {
    // 点击相关按钮直接分享， 如果用户没有授权则需要先授权。
    if ( v == mUmengSdkButton ) {
        mController.directShare(YourActivity.this, SHARE_MEDIA.SINA,
            new SnsPostListener() {
                @Override
                public void onStart() {}

                @Override
                public void onComplete(SHARE_MEDIA platform, int eCode,
                    SocializeEntity entity) {
                    Toast.makeText(YourActivity.this,
                        "分享完成", Toast.LENGTH_SHORT).show();
                }
            });
    }
} // end if
}
```


2.4、分享到多个已授权的平台（postShareMulti）

接口原型:

```
public void postShareMulti(final Context context,  
                           final MulStatusListener listener, SHARE_MEDIA... medias);
```

参数说明：

参数	参数说明
context	开发者Activity运行的上下文环境。
listener	分享监听器
medias	开发者指定的可变数量的分享平台，平台授权的情况下则可直接分享。

开发者指定多个平台，调用该接口时，会将内容分享到meidas中所有已授权的平台。postShareMulti目前支持的平台有：sina微博，腾讯微博，豆瓣，人人网。注意：由于QQ互联已关闭add_share权限,postShareMulti接口对QZone将不再生效。调用postShareMulti接口的示例代码如下：

```
public void onClick(View v) {  
    // 点击相关按钮进行分享  
    if ( v == mUmengSdkButton ) {  
        mController.postShareMulti(mContext, new MulStatusListener() {  
            @Override  
            public void onStart() {  
                Toast.makeText(mContext, "开始分享.", Toast.LENGTH_SHORT).show();  
            }  
  
            @Override  
            public void onComplete(MultiStatus snsSt, int st,  
                                   SocializeEntity entity) {  
                Toast.makeText(mContext, snsSt.toString(), Toast.LENGTH_LONG)  
                    .show();  
            }  
        }, SHARE_MEDIA.SINA, SHARE_MEDIA.TENCENT);  
    } // end if  
}
```

2.5、分享到短信 (shareSms)

接口原型:

```
public void shareSms(Context context);
```

参数说明 :

参数	参数说明
context	调用者 应用的上下文环境。

通过mController设置要分享的内容后，直接调用该方法即可分享到短信。该接口不建议开发者使用，其openShare、postShare、directShare都包含短信分享功能。调用 shareSms 接口的示例代码如下：

```
public void onClick(View v) {  
    // 点击相关按钮进行分享  
    if ( v == mUmengSdkButton ) {  
        // 分享到短信  
        mController.shareSms( mContext ) ;  
    } // end if  
}
```

2.6、分享到邮件 (shareEmail)

接口原型:

```
public void shareEmail(Context context);
```

参数说明 :

参数	参数说明
context	调用者 应用的上下文环境。

通过mController设置要分享的内容后，直接调用该方法即可分享到email。该接口不建议开发者使用，其openShare、postShare、directShare都包含邮件分享功能。

调用 shareEmail 接口的示例代码如下：

```

public void onClick(View v) {
    // 点击相关按钮进行分享
    if ( v == mUmengSdkButton ) {
        // 分享到email
        mController.shareEmail ( mContext ) ;
    } // end if
}

```

2.7、注册回调接口（ registerListener ）

接口原型:

```
public boolean registerListener(ICallbackListener listener) ;
```

参数说明：

参数	参数说明
listener	回调接口,子类有SnsPostListener, UMAuthListener, MulStatusListener等

调用手动注册回调接口 registerListener 接口的示例代码如下：

```

ICallbackListener listener = new SnsPostListener() {

    @Override
    public void onStart() {
        Toast.makeText(mContext, "onStart", 0).show();
    }

    @Override
    public void onComplete(SHARE_MEDIA platform, int eCode,
        SocializeEntity entity){
        Toast.makeText(mContext, "分享成功", 0).show();
    }

};
// 注册分享监听器
mController.registerListener( listener );

```

比如在分享到微信时，sdk默认没有提供分享完成的回调，开发者可以通过该方法注册回调接口，当分享完成时则会回调onComplete方法。用户也可以调用unregisterListener(ICallbackListener listener)来注销监听器。

三、授权相关接口

3.1、授权接口 (doOAuthVerify)

接口原型:

```
public void doOAuthVerify(final Context context,  
                          final SHARE_MEDIA platform, UMAuthListener listener)
```

参数说明：

参数	参数说明
context	调用者 应用的上下文环境。
platform	用户要进行授权的目标平台。目前不支持 Facebook, Twitter, GooglePlus, 易信, 易信朋友圈, 来往, 来往动态, 因为这些平台不提供授权接口。
listener	授权监听接口,包括授权错误、授权完成、授权取消、授权开始的回调函数

调用授权接口的示例代码如下：

```
public void onClick(View v) {  
    // 点击相关按钮进行授权  
    if ( v == mUmengSdkButton ) {  
        mController.doOAuthVerify(mContext, SHARE_MEDIA.SINA,  
                                   new UMAuthListener() {  
                                       @Override  
                                       public void onError(SocializeException e,  
                                           SHARE_MEDIA platform) {  
                                           }  
                                         
                                       @Override  
                                       public void onComplete(Bundle value,  
                                           SHARE_MEDIA platform) {  
                                           String uid = value.getString("uid");  
                                           // 判断是否成功  
                                           if (value != null && !TextUtils.isEmpty( uid ) ) {  
                                               Toast.makeText(mContext, "授权成功",  
                                                           Toast.LENGTH_SHORT).show();  
                                           } else {  
                                               Toast.makeText(mContext, "授权失败",  
                                                           Toast.LENGTH_SHORT).show();  
                                           }  
                                       }  
                                   }  
    }  
}
```

```
        }

        @Override
        public void onCancel(SHARE_MEDIA arg0) {
        }

        @Override
        public void onStart(SHARE_MEDIA arg0) {
        }

    });
}
}
```

3.2、用户登录（Login）

接口原型:

```
public void login(Context context, SHARE_MEDIA platform, SocializeClientListener listener) ;
```

参数说明：

参数	参数说明
context	调用者 应用的上下文环境。
platform	要登录的平台
listener	监听器，包含开始登录、完成登录两个回调函数

登录成功后会返回用户昵称、评论数量等用户信息，可在entity对象上通过get方法获取。该接口不建议开发者调用，其相关功能可以通过其它接口实现。调用login 接口的示例代码如下：

```
public void onClick(View v) {
    // 点击相关按钮进行分享
    if ( v == mUmengSdkButton ) {
        mController.login(mContext, SHARE_MEDIA.SINA,
            new SocializeClientListener() {
                @Override
                public void onStart() {

                }

                @Override
```



```
        Toast.LENGTH_SHORT).show();
        } else
            Toast.makeText(mContext, "注销失败",
        Toast.LENGTH_SHORT).show();
        }
    });

} // end if
}
```

3.4、删除授权 (deleteOauth)

接口原型:

```
public void deleteOauth(Context context, SHARE_MEDIA platform,
                        SocializeClientListener listener);
```

参数说明 :

参数	参数说明
context	调用者 应用的上下文环境。
platform	要删除授权的平台
listener	监听器

调用 deleteOauth接口的示例代码如下：

```
mController.deleteOauth(mContext, SHARE_MEDIA.SINA,
    new SocializeClientListener() {
        @Override
        public void onStart() {}

        @Override
        public void onComplete(int status, SocializeEntity entity) {
            if (status == 200) {
                Toast.makeText(mContext, "删除成功.", Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(mContext, "删除失败", Toast.LENGTH_SHORT).show();
            }
        }
    });
```

3.5、获取用户个人信息（getUserInfo）

接口原型:

```
public void getUserInfo(Context context, FetchUserListener listener);
```

参数说明：

参数	参数说明
context	调用者 应用的上下文环境。
listener	回调函数

获取多个已授权平台的授权信息。调用 getUserInfo 接口的示例代码如下：

```
mController.getUserInfo(this, new FetchUserListener() {  
    @Override  
    public void onStart() {  
    }  
  
    @Override  
    public void onComplete(int status, SocializeUser user) {  
        if( user != null &&  
            !TextUtils.isEmpty(user.mLoginAccount.getAccountIconUrl()) ){  
            List<SnsAccount> lists = user.mAccounts;  
            for (SnsAccount snsAccount : lists) {  
                String userName = snsAccount.getUserName();  
            }  
        }  
    }  
});
```


3.6、获取平台相关信息（getPlatformInfo）

接口原型:

```
public void getPlatformInfo(Context context, SHARE_MEDIA media,
                           UMDataListener listener);
```

参数说明：

参数	参数说明
context	调用者 应用的上下文环境。
media	要获取信息的平台
listener	回调函数

获取已授权用户信息，例如姓名、头像等信息。调用getPlatformInfo接口的示例代码如下：

```
mController.getPlatformInfo(mContext,
    SHARE_MEDIA.SINA, new UMDataListener() {
    @Override
    public void onStart() {}

    @Override
    public void onComplete(int status,
        Map<String, Object> info) {
        // 成功获取平台信息
        if (status == 200 && info != null) {
            StringBuilder sb = new StringBuilder();
            Set<String> keys = info.keySet();
            for (String kStr : keys) {
                sb.append(kStr + "=" +
                    info.get(kStr).toString() + "\r\n");
            }
            textInfo.setText(sb.toString());
            android.util.Log.d("TestData", sb.toString());
        } else {
            textInfo.setText("发生错误： " + status);
        }
    }
});
```

3.7、获取好友列表（getFriends）

接口原型:

```
public void getFriends(Context context, FetchFriendsListener listener,  
                        SHARE_MEDIA platform);
```

参数说明：

参数	参数说明
context	调用者 应用的上下文环境。
listener	监听器
platform	目标平台，比如新浪微博，腾讯微博等

调用 getFriends 接口的示例代码如下：

```
mController.getFriends(mContext, new FetchFriendsListener() {  
    @Override  
    public void onStart() {}  
  
    @Override  
    public void onComplete(int status, List<UMFriend> friends) {  
        StringBuilder sb = new StringBuilder();  
        if (status == 200 && friends != null) {  
            for (UMFriend friend : friends) {  
                sb.append(friend.getName() + "\r\n");  
            }  
        } else {  
            sb.append("status_code=" + status);  
        }  
        textInfo.setText(sb.toString());  
    }  
}, SHARE_MEDIA.TENCENT);
```

四、评论相关接口

4.1、发表评论（postComment）

接口原型:

```
public void postComment(Context context, UMComment comment,
                        MulStatusListener listener, SHARE_MEDIA... platforms);
```

参数说明：

参数	参数说明
context	调用者 应用的上下文环境。
comment	要发表的评论内容
listener	监听器
platforms	评论要发布到的平台

调用 postComment 接口的示例代码如下：

```
//socializeComment为 UMComment类型
mController.postComment(mContext, socializeComment, new MulStatusListener() {
    @Override
    public void onStart() {
    }

    @Override
    public void onComplete(MultiStatus multiStatus,int status,
        SocializeEntity entity) {
        if(status == 200) {
            Toast.makeText(mContext, "发送成功",
                Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(mContext, "发送失败",
                Toast.LENGTH_SHORT).show();
        }
    }
});
```

4.2、获取评论列表 (getComments)

接口原型:

```
public void getComments(Context context, FetchCommentsListener listener, long sinceTime);
```

参数说明 :

参数	参数说明
context	调用者 应用的上下文环境。
listener	监听器
sinceTime	若指定此参数，则返回此指定时间以前的10条评论

调用该接口将返回调用 getComments接口的示例代码如下：

```
mController.getComments(YourActivity.this, new FetchCommentsListener() {
    @Override
    public void onStart() {}

    @Override
    public void onComplete(int status, List<UMComment>
        comments, SocializeEntity entity) {
        if(status == 200 && comments != null){
            mComments.addAll(comments);
            Toast.makeText(mContext, "已获取评论
                count="+comments.size(), 1).show();
        }
    }
},
-1);
```

4.3、打开评论列表页面 (openComment)

接口原型:

```
public void openComment(Context context, boolean forceLogin);
```

参数说明 :

参数	参数说明
context	调用者 应用的上下文环境。
forceLogin	是否强制登录，一般情况下为false

调用 openComment 接口的示例代码如下：

```
mController.openComment(mContext, false);
```

五、用户中心相关接口

5.1、打开用户中心（openUserCenter）

接口原型:

```
public void openUserCenter(Context context, int... config);
```

参数说明：

参数	参数说明
context	调用者 应用的上下文环境。
config	config 登录页面配置参数,可使用 flag_a flag_b 方式，仅第一个可变参数起作用

调用该接口将展示一个友盟分享组件默认的个人中心页面。开发者也可以自定义个人中心，使用“授权+getPlatformInfo”即可。调用 openUserCenter接口的示例代码如下：

```
mController.openUserCenter( YourActivity.this);
```

六、喜欢相关接口

6.1、发布 Like (postLike)

接口原型:

```
public void postLike(Context context, SocializeClientListener listener);
```

参数说明 :

参数	参数说明
context	调用者 应用的上下文环境。
listener	监听器

调用 postLike 接口的示例代码如下:

```
mController.postLike(mContext, new SocializeClientListener() {  
    @Override  
    public void onStart() { }  
  
    @Override  
    public void onComplete(int status, SocializeEntity entity) {  
        if (status == 200) {  
            Toast.makeText(mContext, "发布Like成功.", Toast.LENGTH_SHORT).show();  
        } else {  
            Toast.makeText(mContext, "发布like失败", Toast.LENGTH_SHORT).show();  
        }  
    }  
});
```

6.2、发表 Unlike (postUnlike)

接口原型:

```
public void postUnlike(Context context, SocializeClientListener listener);
```

参数说明 :

参数	参数说明
context	调用者 应用的上下文环境。
listener	监听器

调用 postUnLike 接口的示例代码如下：

```
mController.postUnLike(mContext, new SocializeClientListener() {  
    @Override  
    public void onStart() {}  
  
    @Override  
    public void onComplete(int status, SocializeEntity entity) {  
        if (status == 200) {  
            Toast.makeText(mContext, "发布unLike成功.", Toast.LENGTH_SHORT).show();  
        } else {  
            Toast.makeText(mContext, "发布unlike失败", Toast.LENGTH_SHORT).show();  
        }  
    }  
});
```


七、SDK 平台配置

7.1、设置 SSO 授权 (setSsoHandler)

接口原型:

```
public void setSsoHandler(UMSsoHandler ssoHandler);
```

参数说明 :

参数	参数说明
ssoHandler	SSO授权的处理对象,目前有 SinaSsoHandler, TencenWBSsoHandler, QZoneSsoHandler, RenrenSsoHandler

设置sso授权,调用 setSsoHandler接口的示例代码如下:

```
// 设置新浪为sso授权,当用户安装了新浪微博客户端后就能够使用sso授权.  
// 其他平台设置方法类似  
mController.getConfig().setSsoHandler( new SinaSsoHandler() );
```

7.2、分享面板上的平台排序 (setPlatformOrder)

7.2.1 接口原型:

```
public void setPlatformOrder (SHARE_MEDIA... platforms);
```

参数说明 :

参数	参数说明
platforms	设置平台顺序的可变参数,平台顺序会按照platforms 的出现顺序

调用 setPlatformOrder 接口的示例代码如下:

```
// 分享面板中的平台将按照如下顺序进行排序,微信朋友圈、人人网、新浪、QQ.....  
mController.getConfig().setPlatformOrder( SHARE_MEDIA.WEIXIN_CIRCLE,  
    SHARE_MEDIA.RENREN, SHARE_MEDIA.SINA,  
    SHARE_MEDIA.QQ, SHARE_MEDIA.QZONE,  
    SHARE_MEDIA.WEIXIN, SHARE_MEDIA.GOOGLEPLUS, SHARE_MEDIA.EMAIL,  
    SHARE_MEDIA.TENCENT);
```

7.2.2 接口原型（支持自定义平台排序）:

```
public void setPlatformOrder (String... platforms);
```

参数说明：

参数	参数说明
platforms	设置平台顺序的字符串可变参数，平台顺序会按照platforms 的出现顺序

调用 setPlatformOrder 接口的示例代码如下：

```
// 分享面板中的平台将按照如下顺序进行排序, 微信朋友圈、人人网、新浪、QQ.....
mController.getConfig().setPlatformOrder( SHARE_MEDIA.WEIXIN_CIRCLE.toString(),
    SHARE_MEDIA.RENREN.toString(), SHARE_MEDIA.SINA.toString(),
    SHARE_MEDIA.QQ.toString(), SHARE_MEDIA.QZONE.toString());
```

7.3、移除分享面板上的平台（removePlatform）

接口原型:

```
public void removePlatform(SHARE_MEDIA... platforms);
```

参数说明：

参数	参数说明
platforms	设置要移除的平台

调用 removePlatform 接口的示例代码如下：

```
// 分享面板中的人人网、新浪将会被移除
mController.getConfig().removePlatform( SHARE_MEDIA.RENREN, SHARE_MEDIA.SINA);
```

7.4、获取用户选中的平台 (**getSelectedPlatform**)

接口原型:

```
public static SHARE_MEDIA getSelectedPlatform();
```

例如开发者想知道用户当前在分享面板中选择的分享平台，则可以通过 `getSelectedPlatform` 来获取。比如用户点击的是新浪平台，则会返回 `SHARE_MEDIA.SINA`。
示例代码如下：

```
// 获取用户点击的平台  
mController.getConfig().getSelectedPlatform();
```

7.5、关注某平台的官方微博 (**addFollow**)

接口原型1:

```
public void addFollow (SHARE_MEDIA platform, String fid);
```

参数说明：

参数	参数说明
platform	指定平台
fid	官方微博的fid

设置在授权或者分享时关注官方微博，只支持新浪微博、腾讯微博两个平台。
示例代码如下：

```
// 关注新浪微博的官方微博  
mController.getConfig().addFollow(SHARE_MEDIA.SINA, "1914100420");
```

接口原型2:

```
public void addFollow(SHARE_MEDIA platform,String[] fids)
```

参数说明：

参数	参数说明
platform	指定平台
fids	多个官方微博的fid

设置在授权或者分享时关注多个官方微博，目前只支持新浪微博、腾讯微博两个平台。调用示例代码如下：

```
// 关注新浪微博的官方微博
mController.getConfig().addFollow(SHARE_MEDIA.SINA,
    new String[] {“1914100420”, “1914100421”});
```

7.6、设置分享面板显示的平台 (setPlatforms)

接口原型1:

```
public void setPlatforms (SHARE_MEDIA... platforms);
```

参数说明：

参数	参数说明
platforms	需要显示的平台。如果该平台未添加，则将不显示。

设置分享面板中显示的平台列表,如果该平台未添加，则该平台将不显示。需要手动添加的平台有：QQ，信，微信朋友圈，易信，易信朋友圈，来往，来往动态，Twitter，Facebook，Instagram，google plus (g+) 。示例代码如下：

```
// 设置分享面板上显示的平台
mController.getConfig().setPlatforms(SHARE_MEDIA.SINA,SHARE_MEDIA.TENCENT,
    SHARE_MEDIA.DOUBAN)
```

7.7、检查是否授权且 Token 没有过期 (isAuthenticatedAndTokenNotExpired)

接口原型:

```
public static boolean isAuthenticatedAndTokenNotExpired (Context context, SHARE_MEDIA platform)
```

参数列表:

参数	参数说明
context	应用上下文环境
platform	平台。SHARE_MEDIA对象

该接口检测是否授权且Token没有过期，如果该平台已经授权且Token没有过期，则返回True，否则返回false。该接口检查的Token过期只根据本地缓存的过期时间来判断，不是实时的从平台做验证。如果需要实时的检测，请参考1.25 检查授权平台Token是否过期 (checkTokenExpired)。调用示例代码如下:

```
//检查sina微博是否授权且token没有过期
OauthHelper. isAuthenticatedAndTokenNotExpired(this, SHARE_MEDIA. SINA)
```

7.8 设置 sina SSO 授权回调地址 (setSinaCallbackUrl)

接口原型:

```
public void setSinaCallbackUrl (String redirectUrl)
```

参数说明:

参数	参数说明
redirectUrl	Sina SSO时的回调地址

设置sina SSO的回调地址。当开发者在最先使用自己的回调地址，现在接入友盟分享组件，由于此时友盟分享组件默认使用http://sns.whalecloud.com, 因此出现SSO授权回调地址不匹配的情况，此时开发者只需要调用此接口设置成自己的回调地址即可解决。对于开发者自己没有设置回调地址的情况，请开发者不要调用此接口设置回调地址，默认会匹配到友盟分享组件的回调地址。调用示例代码:

```
mController.getConfig().setSinaCallbackUrl( "your callback url" );
```

7.9、检查授权平台 Token 是否过期 (checkTokenExpired)

接口原型:

```
public void checkTokenExpired(final Context context , SHARE_MEDIA[] platforms,
                             final UMDDataListener listener)
```

参数说明:

参数	参数说明
context	开发者调用的上下文环境
platforms	需要检测的平台。目前只支持新浪，腾讯微博，豆瓣，人人，QQ，QQ空间。
listener	回调函数

该方法用于实时检测Token是否过期，由于该平台需要实时的去第三方平台做过期检测，因此会有比较大的网络延时，请开发者酌情考虑使用。目前支持Token实时检查的平台有：新浪微博、腾讯微博、QQ、QQ空间、豆瓣、人人。检测平台Token是否过期的示例如下：

```
SHARE_MEDIA[] platforms = new SHARE_MEDIA[] { SHARE_MEDIA.SINA,
        SHARE_MEDIA.TENCENT, SHARE_MEDIA.QQ, SHARE_MEDIA.QZONE,
        SHARE_MEDIA.DOUBAN, SHARE_MEDIA.RENREN };

mController.checkTokenExpired(this, platforms, new UMDDataListener() {

    @Override
    public void onStart() {}

    @Override
    public void onComplete(int status, Map<String, Object> info) {
        if ( info != null) {
            Set<String> keys = info.keySet();
            for ( String key : keys ) {
                System.out.println( key + " = " + info.get(key) );
            }
        }
    }
});
```

7.10、设置是否启用图片缓存(setCacheValidStatus)

接口原型：

```
setCacheValidStatus (boolean valid)
```

参数说明：

参数	参数说明
valid	是否启用图片缓存

设置是否开启图片缓存，默认为开启状态。该设置只对url类型的图片生效，如果设置为false，则每次都需要根据url下载图片。调用示例代码如下：

```
//关闭图片缓存
mController.getConfig().setCacheValidStatus(false);
```

7.11、设置短信分享时是否检查 SIM 卡(enableSIMCheck)

接口原型：

```
public void enableSIMCheck(flag);
```

参数说明：

参数	参数说明
flag	分享短信前是否检测SIM卡。True表示分享前需要检查，false表示分享前不检查。默认情况短信分享会检查SIM卡

短信分享前是否检查SIM卡存在。如果不存在则不会进行短信分享，默认情况下会检查SIM是否存在。由于Android原生API不支持双卡检测，因此对于双卡手机的检查不是很准确，其不准确性跟手机厂商相关。如果开发者需要非常严格的检查是否存在主卡跟副卡，可以在分享前根据芯片信息来做检查（主流的是高通跟MTK芯片）调用示例代码如下：

```
//短信分享前不检查是否安装SIM卡  
mController.getConfig().enableSIMCheck(false);
```

7.12 配置 Android 系统 App 分享(supportAppPlatform)

接口原型：

```
public void supportAppPlatform(Context context, SHARE_MEDIA platform,  
    String descriptor, boolean isSupport)
```

参数说明：

参数	参数说明
context	Activity上下文环境
platform	平台
descriptor	平台描述符
isSupport	是否支持该平台

该方法将制定App分享，目前该方法仅支持Google Plus (g+) 跟Twitter。调用示例代码如下：

```
mController.getConfig().supportAppPlatform(getActivity(), SHARE_MEDIA.TWITTER,  
    "com.umeng.twitter", true);
```

7.13 添加开发者自定义平台(addCustomPlatform)

接口原型：

```
public void addCustomPlatform(CustomPlatform cPlatform);
```

参数说明：

参数	参数说明
cPlatform	CustomPlatform对象

调用该接口将添加一个开发者子定义的平台，开发者只需要创建CustomPlatform对象，然后自己实现OnSnsPlatformClickListener接口完成自己的逻辑处理即可。步骤跟示例代码如下：

步骤一：创建CustomPlatform对象

```
CustomPlatform customPlatform = new CustomPlatform("copy_link","复制链接", R.drawable.ic_launcher);
```

步骤二：给该平台添加点击事件

```
customPlatform.mClickListener = new OnSnsPlatformClickListener() {  
  
    @Override  
  
    public void onClick(Context context, SocializeEntity entity,  
SnsPostListener listener) {  
  
        Toast.makeText(getActivity(), "复制链接成功",  
Toast.LENGTH_SHORT). show();  
  
        SocializeUtils.sendAnalytic(getActivity(),  
  
            "com.umeng.custopm.copy_likn","分享内容", null,
```



```
        "copy_link");  
  
    }  
  
};
```

步骤三：添加该平台到平台集合中

```
mController.getConfig().addCustomPlatform(customPlatform);
```

如果你是自定义分享面板，需要调用该自定义平台分享时，可以直接调用customPlatform.performClick(Context context,

SocializeEntity entity,SnsPostListener listener)方法即可实现。

八、注册摇一摇相关功能

摇一摇功能独立于友盟分享sdk，因此该章节中的所有功能都需要依赖 SocialSDK_Shake_Lib.jar、 SocialSDK_ScrShot_Lib.jar以及相关的资源。开发者需要定义一个摇一摇Controller来注册摇一摇相关功能。示例如下：

```
定义摇一摇控制器  
IShakeService mShakeController = UMSHakeServiceFactory  
    .getShakeService("com.umeng.share");
```

注意：

如果您已经使用了UMServiceFactory.getUMSocialService(“com.umeng.share”， RequestType.SOCIAL)函数来生成用于整个SDK的控制器，例如该文档中的mController，则需保证构建两个控制器的字符串参数一致，例如这里为“com.umeng.share”。

8.1、注册摇一摇截图分享（registerShakeListender）

接口原型：

public void registerShakeListender(Activity activity,UMBaseAdapter adapter,
List<SHARE_MEDIA> platforms,
OnSensorListener listener) ;
参数说明：

参数	参数说明
activity	用户所在的activity
adapter	截图适配器，不同的应用有不同的实现，具体参考摇一摇集成文档
platforms	摇一摇分享时用户可选的社交平台，最多支持5个平台
listener	传感器监听器，包含摇一摇结束、分享完成等回调接口

用户摇一摇，然后截取当前屏幕图像进行分享；注册 registerShakeListender 接口的示例代码如下：

```
// 普通应用截屏适配器，具体参考摇一摇集成文档
MAppAdapter appAdapter = new UAppAdapter( YourActivity.this );
/ 配置平台
List<SHARE_MEDIA> platforms = new ArrayList<SHARE_MEDIA>();
platforms.add(SHARE_MEDIA.QZONE);
platforms.add(SHARE_MEDIA.SINA);

/ 注册摇一摇截图分享功能, mShakeController在上文定义
mShakeController.registerShakeListender(YourActivity.this,
appAdapter, platforms, mSensorListener);

/**
 * 传感器监听器
 */
private OnSensorListener mSensorListener = new OnSensorListener() {

    @Override
    public void onStart() {
    }

    /**
     * 分享完成后回调 (non-Javadoc)
     */
    @Override
    public void onComplete(SHARE_MEDIA platform, int eCode,
        SocializeEntity entity) {
        Toast.makeText(getActivity(), "分享完成", Toast.LENGTH_SHORT).show();
    }

    /**

     * @Description: 摇一摇动作完成后回调 (non-Javadoc)
     */
    @Override
    public void onActionComplete(SensorEvent event) {
        Toast.makeText(getActivity(), "用户摇一摇", Toast.LENGTH_SHORT).show();
    }
}
```

```

/**
 * @Description: 用户点击分享窗口的取消和分享按钮触发的回调
 */
@Override
public void onButtonClick(WhitchButton button) {
    if (button == WhitchButton.BUTTON_CANCEL) {
        Toast.makeText(getActivity(), "取消分享",
            Toast.LENGTH_SHORT).show();
    } else {
        // 点击的是分享按钮的回调函数
    }
}
};

```

8.2、注册摇一摇打开分享面板（registerShakeToOpenShare）

接口原型:

public void registerShakeToOpenShare(final Activity activity, boolean sound) ;

参数说明：

参数	参数说明
activity	用户所在的activity
sound	摇一摇时是否开启音效，音效资源在res/raw/shake_sound.mp3

用户摇一摇，然后打开分享面板。调用registerShakeToOpenShare 接口的示例代码如下：

注册摇一摇打开分享面板，开启摇一摇音效，mShakeController在上文定义
 shakeController.registerShakeToOpenShare(YourActivity.this, true);

8.3、注册摇一摇截图（registerShakeToScrShot）

接口原型:

public void registerShakeToScrShot(Activity activity, UMBaseAdapter adapter, OnScreenshotListener listener) ;

参数说明：

参数	参数说明
activity	用户所在的activity
adapter	截图适配器，不同的应用有不同的实现，具体参考摇一摇集成文档
listener	截图监听器，将截图回调给开发者

用户摇一摇，然后sdk会截取当前平台的图像，并且回调给开发者。
调用registerShakeToScrShot 接口的示例代码如下：

```
UMAppAdapter appAdapter = new UMAppAdapter( YourActivity.this );
// 注册摇一摇截图功能, mShakeController在上文定义
mShakeController.registerShakeToScrShot(YourActivity.this,
appAdapter , new OnScreenshotListener() {

    @Override
    public void onComplete(Bitmap bmp) {
        if (bmp != null ) {
            // 对图片的处理
        }
    }
});
```

8.4、直接截图（不需要摇一摇）（ takeScrShot ）

接口原型:

```
public void takeScrShot(Activity activity, UMBaseAdapter adapter,
OnScreenshotListener listener);
```

参数说明：

参数	参数说明
activity	用户所在的activity
adapter	截图适配器，不同的应用有不同的实现，具体参考摇一摇集成文档
listener	截图监听器，将截图回调给开发者

调用 takeScrShot进行截图的示例代码如下：

```
UAppAdapter appAdapter = new UAppAdapter( YourActivity.this );
// 注册摇一摇截图功能, mShakeController在上文定义
ShakeController.takeScrShot(YourActivity.this, appAdapter ,
    new OnScreenshotListener() {

        @Override
        public void onComplete(Bitmap bmp) {
            if (bmp != null ) {
                // 对图片的处理
            }
        }
    } );
```

九、线上文档

更详细的集成文档，请移步 <http://dev.umeng.com/social/android/share/quick-integration> 。