畅言android平台sdk文档

**北京搜狐新媒体信息技术有限公司**

版本：CySDK V2.2.0

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 文档版本 | 修订人 | 备注 |
| 2014/01/07 | 2.1.0 |  | 新增评论框浮层，评论列表页，评论发表页和社交平台分享功能；原有api接口格式有变动 |
| 2014/03/20 | 2.1.1 |  | 增加评分功能，修改sdk初始化方式，防止泄露appsecret |
| 2014/04/11 | 2.1.2 |  | 优化单点登录功能流程 |
| 2014/08/11 | 2.2.0 |  | 优化评论框浮层、评论列表页、评论发表页和登陆页，新增分频道获取评论功能，优化顶踩接口 |
| 2014/12/16 | 2.2.1 |  | 1. 完善回复列表，增加消息推送功能 2. 增加“我做出的评论” 接口 |
| 2014/12/24 | 2.2.2 |  | 1.增加是否启用消息推送功能的开关 |
| 2014/12/30 | 2.2.3 |  | 1. 修复评论列表重复显示的bug 2. 新增api文档的详细说明 3. 增加消息推送中的注意事项 4. 修复列表中图片下载失败时有toast显示的bug |
| 2015/02/08 | 2.2.4 |  | 1. 修复图片缓存导致评论列表闪退的bug。 |

目录

[畅言android平台sdk文档 1](#_Toc13024)

[目录 2](#_Toc10942)

[一、 概述 4](#_Toc160)

[名词解释 4](#_Toc16631)

[二、运行demo程序 4](#_Toc2731)

[三、快速接入 4](#_Toc23375)

[1.导入依赖文件 4](#_Toc13908)

[2.添加应用权限 5](#_Toc31716)

[3.添加畅言配置 5](#_Toc12042)

[4.初始化畅言SDK 5](#_Toc13062)

[4.1.config——UI类 6](#_Toc5108)

[4.2.config——Login类 10](#_Toc6901)

[4.3.config——CommentSettings类 11](#_Toc30527)

[5.调用SDK接口插入评论框浮层。 11](#_Toc623)

[5.1 activity嵌入 11](#_Toc25836)

[5.2 fragement嵌入 11](#_Toc17593)

[6. SDK中单点登录功能的使用 12](#_Toc5427)

[7. SDK中消息推送的使用 12](#_Toc790)

[四、api调用示例及说明 14](#_Toc20133)

[1.畅言数据接口api 14](#_Toc21189)

[1.1 不登录可访问的api接口 14](#_Toc24243)

[1.2 需要登录的api 20](#_Toc19439)

[2.附带UI界面的api接口 23](#_Toc853)

[2.1 评论列表界面 23](#_Toc2368)

[2.2 评论发表界面 24](#_Toc14040)

[2.3 用户评论中心界面(包括发出的评论和接收到的评论) 24](#_Toc25845)

[3. 消息推送 25](#_Toc5915)

[配置见 三.7 25](#_Toc26533)

[设置notification的图标 25](#_Toc11207)

[如何自定义 处理接收到消息推送. 25](#_Toc15890)

[如果使用自定义的评论编辑页面 25](#_Toc11083)

[注意事项 26](#_Toc30404)

[附录 26](#_Toc7323)

[关于代码混淆 26](#_Toc3391)

# 一、 概述

畅言SDK为开发者提供app内评论功能的嵌入，通过使用畅言SDK，开发者可在app内拥有评论功能和分享功能。

本文档将对使用SDK时所用的配置、接口进行说明，并提供简单的实例，帮助开发者方便的使用SDK。

## 名词解释

|  |  |
| --- | --- |
| cyan\_app\_id | 畅言分配给第三方应用的appId，用来鉴别身份。登录changyan.sohu.com后，点击设置->高级，查看。 |
| cyan\_app\_key | 畅言分配给第三方应用的appkey，用来进行数据加密和签名校验。登录changyan.sohu.com后，点击设置->高级，查看。 |
| accesstoken | 表示用户身份的token，用于畅言api的调用。 |
| oauth\_redirect | 使用oauth方式登录畅言的回调页面地址，登录changyan.sohu.com后，点击设置->高级，进行修改。 |

# 二、运行demo程序

1.下载畅言SDK的zip包，解压。

2.将CySDK 2.2.0中的cyan\_sdk\_test项目导入到eclipse中。

4.右键点击cyan\_sdk\_test，run as->android application即可运行。

# 三、快速接入

## 1.导入依赖文件

将CySDK 2.2.0中lib文件夹下的所有jar包和CySDK中bin文件夹下的CySDK.jar复制到工程的libs目录下。

## 2.添加应用权限

在AndroidMainfest.xml中添加以下权限配置

<uses-permission android:name=*"android.permission.INTERNET"* />

<uses-permission android:name=*"android.permission.ACCESS\_NETWORK\_STATE"* />

<uses-permission android:name=*"android.permission.WRITE\_EXTERNAL\_STORAGE"* />

## 3.添加畅言配置

在AndroidMainfest.xml中添加如下配置项（可参考CySDK 2.1.0下的AndroidManifest.xml）

<!-- 评论列表界面 -->

<activity android:name=*"com.sohu.cyan.android.sdk.activity.CyanCommentActivity"*/>

<!-- 发表评论页面 -->

<activity android:name=*"com.sohu.cyan.android.sdk.activity.CyanPostCommentActivity"*/>

<!-- 社会化账号登陆界面-->

<activity android:name=*"com.sohu.cyan.android.sdk.activity.OAuthActivity"*></activity>

畅言的注册地址：<http://changyan.sohu.com/register>

## 4.初始化畅言SDK

Config config = **new** Config();

config.ui.toolbar\_bg = Color.WHITE;

config.comment.showScore = **true**;

config.comment.uploadFiles = **true**;

config.comment.anonymous\_token = "z6e0xTejZBmqP-dQcAGN2lWmTZu8\_yk9fW\_w7oUvJP8";

config.login.loginActivityClass = AppLoginActivity.**class**;

CyanSdk.*register*("cy2uLn0jfoQS", "16bd0e2533092b96e3e55958ef19d08a" , "http://10.2.58.251:8081/login-success.html", config);

cyanSdk = CyanSdk.*getInstance*(**this**.getActivity());

高亮部分修改为畅言授权的appid、appkey以及后台设置的callbackUri，登录changyan.sohu.com,点击设置->高级查看配置项，callbackUri在页面http://changyan.sohu.com/setting/common/further中设置，



如果之前没有设置过oauth的回调地址，可以将之设置为网站的域名。

下划线部分为自定义畅言显示选项，需要在config中进行设置，详细参数说明如下：

### 4.1.config——UI类

**1.颜色设置**

可设置评论框浮层的背景色、按钮色、文字颜色等，不修改则使用默认值。

toolbar\_bg ：toolbar背景色

toolbar\_border ：toolbar边框色

toolbar\_btn： toolbar按钮颜色

before\_clk ：顶按钮点击之前颜色

after\_clk：顶按钮点击之后颜色

list\_title：热门评论、最新评论标题的文字和边框颜色

login\_bg：登陆界面背景颜色

login\_btn：登陆界面按钮颜色

login\_btn\_text：登陆界面按钮文字颜色

login\_input ：登陆界面昵称显示框背景颜色

login\_input\_text： 登陆界面昵称显示框文字颜色

edit\_cmt\_bg：评论输入框背景颜色

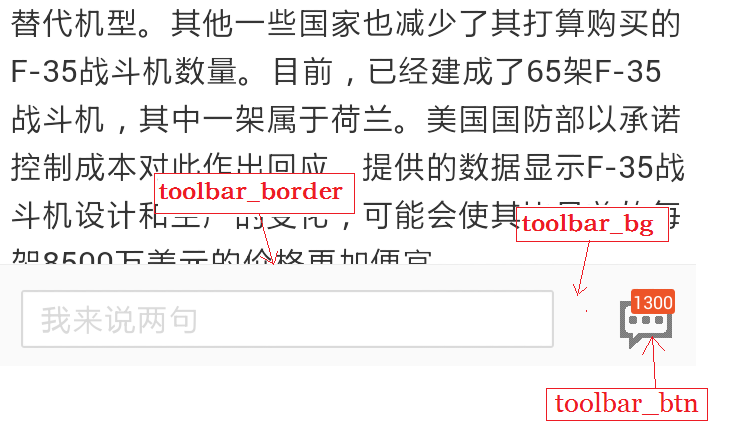


图1 bottomtoolbar颜色参数指示图



图2 评论展示界面颜色参数指示图



图3 评论界面颜色参数指示图



图4 登陆界面颜色参数指示图

**2.风格设置**

评论显示方式分为盖楼（floor）和缩进（indent）两种方式，通过 style 设置，默认值为”floor”，当选择indent风格时，需设置depth和sub\_size的值，depth表示缩进显示的深度，sub\_size 表示缩进显示的评论条数。

具体展示形式如下



图5 盖楼风格



图6 缩进风格(depth = 1，sub\_size=20)

设置方法：修改config

config.ui.style="indent";

config.ui.depth = 1;

config.ui.sub\_size = 20;

### 4.2.config——Login类

可设置登陆相关选项，不修改则使用默认值。

SSOLogin ：是否允许单点登录

SSO\_Assets\_Icon：单点登陆图标的名称(需将图片文件放在assets文件夹中)，SSOLogin为true时必须设置。

loginActivityClass ：单点登录的Activity，SSOLogin为true时必须设置。

Custom\_oauth\_login：第三方登录是否自己获得授权，默认设置为false，即授权给畅言。

QQ\_btn\_listerner/ SINA\_btn\_listerner ：(Custom\_oauth\_login为true时，必须设置。)

用户自己绑定在QQ/SINA登录按钮上的监听事件。

QQ/SINA/SOHU : Custom\_oauth\_login为false时，选取第三方登录方式。

### 4.3.config——CommentSettings类

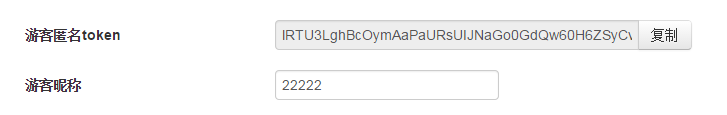
可设置评论相关选项，不修改则使用默认值。

showScore：是否允许用户评分

uploadFiles：是否允许上传附件

useFace：是否使用表情功能

anonymous\_token：匿名登录的access token，为null时，不允许匿名登录，在后台设置->wap设置中获取这个token。



hotsize：显示的热门评论数

latestsize：显示的最新评论数

pagesize：加载更多时显示的评论数

## 5.调用SDK接口插入评论框浮层。

目前SDK开发的评论框浮层支持嵌入到LinearLayout和RelativeLayout两种布局中，LinearLayout的嵌入需要将布局中的其他view整体包裹在一个布局中，并设置layout\_weight属性为1，layout\_height为0。请参照cyan\_sdk\_test项目下的res/toolbar\_demo.xml，是如何实现在LinearLayout中实现评论框浮层的嵌入的。RelativeLayout的嵌入不需要做特殊处理。

### 5.1 activity嵌入

如果在activity中直接嵌入评论框浮层，那么可以直接调用addCommentToolbar(final Activity activity, final String topicSourceId, String topicTitle, String topicUrl)接口，topicSourceId是文章在你自己系统的id，topicTitle是文章标题，topicUrl是文章链接。

### 5.2 fragement嵌入

com.example.cyan\_sdk\_test.CyActivity.java中的FixToolBarSectionFragment类展示了如何在Fragment中嵌入评论框浮层。先要在onCreate方法中出事化畅言SDK实例，畅言SDK采用单例模式，调用畅言SDK的api之前需要先初始化，然后在onCreateView中调用CyanSDk的addCommentToolbar(ViewGroup rootView, String topicSourceId, String topicTitle, String topicUrl)，需要将fragment的rootView转为ViewGroup类型传入，topicSourceId是文章在你自己系统的id，topicTitle是文章标题，topicUrl是文章链接。

@Override

**public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

View rootView = inflater.inflate(R.layout.*toolbar\_demo*, container, **false**);

WebView webview = (WebView) rootView.findViewById(R.id.*webview*);

webview.loadUrl("http://mil.sohu.com/20131103/n389458829.shtml");

cyanSdk.addCommentToolbar((ViewGroup) rootView, "20131125", "F-35战机",

"http://mil.sohu.com/20131103/n389458829.shtml");

**return** rootView;

}

## 6. SDK中单点登录功能的使用

**初始化SDK时需要设置单点登录相关配置项：**

Config config = **new** Config();

// 是否在发表评论时允许使用单点登录

config.login.SSOLogin = **true**;

// 登录图标名称(需将图片文件放在assets文件夹中)

config. login.SSO\_Assets\_Icon = "ico31.png";

// 登录界面activity的class

config.loginActivityClass = AppLoginActivity.**class**;

CyanSdk.*register*("cy2uLn0jfoQS", "16bd0e2533092b96e3e55958ef19d08a",

"http://10.2.58.251:8081/login-success.html", config);

**AdnroidMainfest.xml需要添加的配置项:**

<!—登录界面activity -->

<activityandroid:name=*"com.example.cyan\_sdk\_test.AppLoginActivity"*></activity>

**登录成功后需要调用的sdk接口，以实现畅言的同步登录：**

AccountInfo account = **new** AccountInfo();

//应用自己的用户id

account.isv\_refer\_id = "10001";

account.nickname = "testuser";

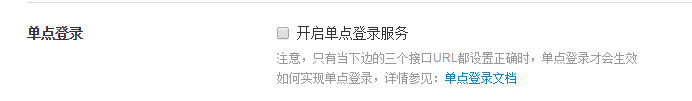
account.img\_url = "http://assets.changyan.sohu.com/upload/asset/scs/images/pic/pic42\_null.gif";

CyanSdk.*getInstance*(AppLoginActivity.**this**).setAccountInfo(account,callBack);

其中callBack接口是同步登陆畅言成功后需要做的回调操作。

**畅言后台开启单点登录开关**

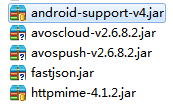
在后台设置->PC端设置->单点登录中，勾选“开启单点登录服务”，



以保证用户数据与站点的同步。

## SDK中消息推送的使用

**引入jar包**



**AndroidMainfest.xml中配置activity**

<activity android:name=*"com.sohu.cyan.android.sdk.activity.RepliesActivity"* android:launchMode=*"singleTask"*>

**AndroidMainfest.xml中配置服务**

<!-- 畅言消息推送 服务 -->

<service android:name=*"com.avos.avoscloud.PushService"* />

<receiver android:name=*"com.sohu.cyan.android.sdk.push.CyanPushReceiver"* >

<intent-filter>

<action android:name=*"android.intent.action.BOOT\_COMPLETED"* />

<action android:name=*"android.intent.action.USER\_PRESENT"* />

<action android:name=*"android.net.conn.CONNECTIVITY\_CHANGE"* />

<action android:name=*"action.com.sohu.cyan.push.receive"* />

</intent-filter>

</receiver>

**AndroidMainfest.xml中配置权限**

<uses-permission android:name=*"android.permission.INTERNET"* />

<uses-permission android:name=*"android.permission.READ\_PHONE\_STATE"*/>

<uses-permission android:name=*"android.permission.ACCESS\_WIFI\_STATE"* />

<uses-permission android:name=*"android.permission.CHANGE\_WIFI\_STATE"* />

<uses-permission android:name=*"android.permission.VIBRATE"* />

**AndroidMainfest.xml中<application>配置**

android:name=*"com.sohu.cyan.android.sdk.push.CyanPushApplication"*

*或者*

**public** **class** MyApplication **extends** CyanPushApplication {

@Override

**public** **void** onCreate() {

**super**.onCreate();

}

}

android:name=*"包名.*MyApplication*"*

**AndroidMainfest.xml中注册RepliesActivity**

<activity android:name=*"com.sohu.cyan.android.sdk.activity.RepliesActivity"* android:launchMode=*"singleTask"*>

Java代码中启用消息推送(需要写在CyanSdk.register(...)之后)

PushManager.*setUseService*(**Context**, boolean);

# 四、api调用示例及说明

## 1.畅言数据接口api

调用畅言api的数据接口需要先将CySDK 2.1.0目录中libs文件夹下的fastjson-1.1.36.jar和httpmime-4.1.2.jar复制到工程的libs目录下。

然后在项目的AndroidMainfest.xml中加入以下配置信息:

<uses-permission android:name=*"android.permission.INTERNET"* />

<uses-permission android:name=*"android.permission.ACCESS\_NETWORK\_STATE"* />

<uses-permission android:name=*"android.permission.WRITE\_EXTERNAL\_STORAGE"* />

调用api接口前需要初始化畅言sdk,sdk初始化配置祥见：

Config config = **new** Config();

CyanSdk.*register*("cy2uLn0jfoQS", "16bd0e2533092b96e3e55958ef19d08a", "http://10.2.58.251:8081/login-success.html", config);

cyanSdk = CyanSdk.*getInstance*(**this**.getActivity());

高亮部分修改为畅言授权的appid、appkey、callbackUri，登录changyan.sohu.com,点击设置->高级查看配置项，callbackUri在页面<http://changyan.sohu.com/setting/further>中设置。

SDK的api使用示例参见com.example.cyan\_sdk\_test.CyActivity.java中的ApiSectionFragment。

畅言SDK的网络数据请求采用异步方式，需要开发者自己实现CyanRequestListener接口处理返回数据，数据的反序列化采用泛型方式，需要开发者指定CyanRequestListener的泛型类型。CyanRequestListener中有onRequestSucceeded和onRequestFailed两个接口，前者处理正常响应，处理异常响应，下文中api参数介绍中出现该实现接口的类，不再详细说明。

### 1.1 不登录可访问的api接口

|  |  |  |
| --- | --- | --- |
| 接口名称 | loadTopic | |
| 描述 | 首次加载文章页面时调用，创建在畅言中的文章信息，加载最新最热列表。 | |
| 参数 | 名称 | 说明 |
|  | topicSourceId | 应用提供的文章唯一标识，与topicUrl二选一，优先使用topicSourceId |
|  | topicUrl | 文章url地址，与topicSourceId二选一 |
|  | topicTitle | 文章标题 |
|  | topicCategoryId | 新闻分类/频道id，可选参数，不使用传null |
|  | pageSize | 查询的最新评论数量 |
|  | hotSize | 查询的最热评论数量 |
|  | style | 评论展示风格，可选择盖楼模式（floor）或缩进模式（indent），传null时默认盖楼模式 |
|  | depth | 列表缩进深度，默认为1，即只缩进一次 |
|  | sub\_size | 子列表最大显示几条 |
|  | listener | 实现CyanRequestListener接口类的实例 |
| 返回值 | 无 | |
| listener回调方法onRequestSucceeded的参数值 | TopicLoadResp.java   |  |  |  | | --- | --- | --- | | 属性 | 类型 | 说明 | | topic\_id | **long** | 评论id | | total\_page\_no | **int** | 总页数 | | participation\_sum | **int** | 参与数 | | share\_sum | **int** | 分享数 | | cmt\_sum | **int** | 评论总数 | | hots | ArrayList<Comment> | 最热评论列表 | | comments | ArrayList<Comment> | 最新评论列表 | | error\_code | **int** | 错误码 | | error\_msg | String | 错误信息 |   Comment.java   |  |  |  | | --- | --- | --- | | 属性名 | 类型 | 说明 | | content | String | 内容 | | create\_time | **long** | 评论时间 | | comment\_id | **long** | 评论id | | support\_count | **int** | 顶数 | | oppose\_count | **int** | 参数 | | reply\_count | **int** | 回复数 | | top | **boolean** | 是否置顶 | | ip | String |  | | ip\_location | String | 地域信息 | | from | String |  | | passport | Passport | 评论用户信息 | | score | **float** | 文章评分 | | metadata | String |  | | childComment | Comment |  | | comments | ArrayList<Comment> | 盖楼列表 | | attachments | ArrayList<Comment> | 图片附件 | | reply\_id | **long** | 被回复评论id | | |

|  |  |  |
| --- | --- | --- |
| 接口名称 | getCommentCount | |
| 描述 | 查询单个文章评论数，可实时查询 | |
| 参数 | 名称 | 说明 |
|  | topicSourceIds | 应用提供的文章唯一标识，与topicUrl，topicIds三选一 |
|  | topicUrl | 文章url地址，与topicSourceId，topicIds三选一 |
|  | topicIds | 文章在畅言中的ID，通过loadTopic接口获取，与topicSourceId，topicIds三选一，优先使用topicIds，其次使用topicSourceIds |
|  | listener | 实现CyanRequestListener接口类的实例 |
| 返回值 | 无 | |
| listener回调方法onRequestSucceeded的参数值 | TopicCountResp.java   |  |  |  | | --- | --- | --- | | 属性值 | 类型 | 说明 | | count | **int** | 单个文章评论数 | | error\_code | **int** | 错误码 | | error\_msg | **String** | 错误信息 | | |

|  |  |  |
| --- | --- | --- |
| 接口名称 | getCommentCount | |
| 描述 | 批量获取文章评论数，批量接口无法实时更新缓存，所以此接口查询评论数无法实时 | |
| 参数 | 名称 | 说明 |
| List<String> | topicSourceIds | 应用提供的文章唯一标识，与topicUrl，topicIds三选一 |
| List<String> | topicUrl | 文章url地址，与topicSourceId，topicIds三选一 |
| List<Long> | topicIds | 文章在畅言中的ID，通过loadTopic接口获取，与topicSourceId，topicIds三选一，优先使用topicIds，其次使用topicSourceIds |
|  | listener | 实现CyanRequestListener接口类的实例 |
| 返回值 | 无 | |
| listener回调方法onRequestSucceeded的参数值 | TopicsCountResp.java   |  |  |  | | --- | --- | --- | | 属性值 | 类型 | 说明 | | result | Map<String, TopicCount> |  | | error\_code | **int** | 错误码 | | error\_msg | String | 错误信息 |   TopicCount.java   |  |  |  | | --- | --- | --- | | 属性名 | 类型 | 说明 | | id | **long** | 畅言中的新闻id | | sid | String | source id | | sum | **int** | 评论数，不推荐使用，请使用comments | | comments | **int** | 评论数 | | shares | **int** | 分享数 | | likes | **int** | 喜欢数 | | parts | **int** | 参与数 | | |

|  |  |  |
| --- | --- | --- |
| 接口名称 | getTopicComments | |
| 描述 | 分页查询文章评论数据 | |
| 参数 | 名称 | 说明 |
|  | topicId | 畅言分配的文章唯一ID，通过loadTopic接口获取 |
|  | pageSize | 分页大小 |
|  | pageNo | 页数 |
|  | style | 评论展示风格，可选择盖楼模式（floor）或缩进模式（indent），  传null时默认盖楼模式 |
|  | depth | 列表缩进深度，默认为1，即只缩进一次 |
|  | sub\_size | 子列表最大显示几条 |
|  | listener | 实现CyanRequestListener接口类的实例 |
| 返回值 | 无 | |
| listener回调方法onRequestSucceeded的参数值 | TopicCommentsResp.java   |  |  |  | | --- | --- | --- | | 属性名 | 类型 | 说明 | | topic\_id | **long** | 文章 id | | comments | List<Comment> | 评论 | | cmt\_sum | **int** | 评论总数 | | |

|  |  |  |
| --- | --- | --- |
| 接口名称 | commentAction | |
| 描述 | 评论顶踩接口 | |
| 参数 | 名称 | 说明 |
|  | topicId | 畅言分配的文章唯一ID，通过loadTopic接口获取 |
|  | commentId | 评论id |
|  | type | 操作类型，枚举类CommentActionType |
|  | listener | 实现CyanRequestListener接口类的实例 |
| 返回值 | 无 | |
| listener回调方法onRequestSucceeded的参数值 | CommentActionResp.java  服务器返回的JSON数据:  {} | |

|  |  |  |
| --- | --- | --- |
| 接口名称 | commentReplies | |
| 描述 | 获取一条评论的所有回复 | |
| 参数 | 名称 | 说明 |
|  | topicId | 畅言分配的文章唯一ID，通过loadTopic接口获取 |
|  | commentId | 评论id |
|  | pageSize | 分页大小 |
|  | pageNo | 页数 |
|  | listener | 实现CyanRequestListener接口类的实例 |
| 返回值 | 无 | |
| listener回调方法onRequestSucceeded的参数值 | CommentReplyResp.java   |  |  |  | | --- | --- | --- | | 属性名 | 类型 | 说明 | | comments | List<Comment> | 一条评论的所有回复 | | error\_code | **int** | 错误码 | | error\_msg | String | 错误信息 | | |

startAuthorize为登录接口，使用时需要在AndroidMainfest.xml中注册畅言的LoginActivity

<activity android:name=*"com.sohu.cyan.android.sdk.activity.OAuthActivity"*></activity>

调用startAuthorize接口的activity需要实现onActivityResult接口，接受登录是否成功的返回信息，参考com.example.cyan\_sdk\_test. LoginActivity。

|  |  |  |
| --- | --- | --- |
| 接口名称 | startAuthorize | |
| 描述 | 登录接口，获取用户信息，用户收到的回复，调用评论发表接口前需要先登录 | |
| 参数 | 名称 | 说明 |
|  | platform\_id | 通过oauth方式登录的第三方平台，见com.sohu.cyan.android.sdk.util. Constants. PlatFormId |
|  | activity | 调用登录接口的activity |
| 返回值 | 无 | |

|  |  |  |
| --- | --- | --- |
| 接口名称 | getAnonymousUserInfo | |
| 描述 | 根据Config中设置的anonymous\_token，获取匿名用户信息接口，包括匿名用户的昵称，头像地址等信息。 | |
| 参数 | 名称 | 说明 |
|  | listener | 实现CyanRequestListener接口类的实例 |
| 返回值 | 无 | |
| listener回调方法onRequestSucceeded的参数值 | UserInfoResp.java   |  |  |  | | --- | --- | --- | | 属性名 | 类型 | 说明 | | nickname | String | 昵称 | | from | String | 账号来源 | | img\_url | String | 头像 | | sign | String |  | | latest\_reply\_sum | **int** | 最近回复数 | | comment\_sum | **int** | 发表评论数 | | user\_id | **long** | 用户id | | |

|  |  |  |
| --- | --- | --- |
| 接口名称 | anonymousSubmitComment | |
| 描述 | 发表评论接口 | |
| 参数 | 名称 | 说明 |
|  | topicId | 畅言分配的文章ID，通过loadTopic接口获取 |
|  | content | 评论内容 |
|  | replyId | 回复的评论，默认为0 |
|  | attachUrl | 附件图片地址，通过attachUpload接口上传后获取 |
|  | score | 打分 |
|  | appType | 平台类型，40:ipone,41:ipad,43:android |
|  | accessToken | 匿名用户token，在站长后台wap设置中生成 |
|  | listener | 实现CyanRequestListener接口类的实例 |
| 返回值 | 无 | |
| listener回调方法onRequestSucceeded的参数值 | SubmitResp.java   |  |  |  | | --- | --- | --- | | 属性名 | 类型 | 说明 | | id | **long** |  |   服务器返回JSON样例  {  id: 1001  } | |

### 1.2 需要登录的api

|  |  |  |
| --- | --- | --- |
| 接口名称 | submitComment | |
| 描述 | 发表评论接口 | |
| 参数 | 名称 | 说明 |
|  | topicId | 畅言分配的文章ID，通过loadTopic接口获取 |
|  | content | 评论内容 |
|  | replyId | 回复的评论，默认为0 |
|  | attachUrl | 附件图片地址，通过attachUpload接口上传后获取 |
|  | score | 打分 |
|  | appType | 平台类型，40:ipone,41:ipad,43:android |
|  | listener | 实现CyanRequestListener接口类的实例 |
| 返回值 | 无 | |
| listener回调方法onRequestSucceeded的参数值 | SubmitResp.java   |  |  |  | | --- | --- | --- | | 属性名 | 类型 | 说明 | | id | **long** |  |   服务器返回JSON样例  {  id: 1001  } | |

|  |  |  |
| --- | --- | --- |
| 接口名称 | attachUpload | |
| 描述 | 发表评论接口 | |
| 参数 | 名称 | 说明 |
|  | file | 要上传的图片 |
|  | listener | 实现CyanRequestListener接口类的实例 |
| 返回值 | 无 | |
| listener回调方法onRequestSucceeded的参数值 | AttachementResp.java   |  |  |  | | --- | --- | --- | | 属性名 | 类型 | 说明 | | url | String | 附件的url |   服务器返回JSON,例如：  {  url : “http://img.itc.cn/comment/odYRJeX\_jpg”  } | |

|  |  |  |
| --- | --- | --- |
| 接口名称 | getUserInfo | |
| 描述 | 获取用户信息接口，包括用户的昵称，头像地址，最新回复数，发表评论总数等信息。 | |
| 参数 | 名称 | 说明 |
|  | listener | 实现CyanRequestListener接口类的实例 |
| 返回值 | 无 | |
| listener回调方法onRequestSucceeded的参数值 | UserInfoResp.java   |  |  |  | | --- | --- | --- | | 属性名 | 类型 | 说明 | | nickname | String | 昵称 | | from | String | 账号来源 | | img\_url | String | 头像 | | sign | String |  | | latest\_reply\_sum | **int** | 最近回复数 | | comment\_sum | **int** | 发表评论数 | | user\_id | **long** | 用户id | | |

|  |  |  |
| --- | --- | --- |
| 接口名称 | getUserComments | |
| 描述 | 获取用户做出的评论 | |
| 参数 | 名称 | 说明 |
|  | pageNo | 页数 |
|  | pageSize | 分页大小 |
|  | listener | 实现CyanRequestListener接口类的实例 |
| 返回值 | 无 | |
| listener回调方法onRequestSucceeded的参数值 | UserCommentResp.java   |  |  |  | | --- | --- | --- | | 属性名 | 类型 | 说明 | | total\_number | **int** | 做出的评论总数 | | comments | List<ReplySendComment> | 做出的评论信息 |   ReplySendComment.java参考Comment.java | |

|  |  |  |
| --- | --- | --- |
| 接口名称 | getUserNewReply | |
| 描述 | 获取用户收到的最新回复 | |
| 参数 | 名称 | 说明 |
|  | pageNo | 页数 |
|  | pageSize | 分页大小 |
|  | listener | 实现CyanRequestListener接口类的实例 |
| 返回值 | 无 | |
| listener回调方法onRequestSucceeded的参数值 | UserReplyResp.java   |  |  |  | | --- | --- | --- | | 属性名 | 类型 | 说明 | | total\_num | **int** | 收到的最新回复数 | | replies | List<ReplyRcvComment> | 最新回复列表 |   ReplyRcvComment.java参考Comment.java | |

|  |  |
| --- | --- |
| 接口名称 | logOut |
| 描述 | 退出登录 |
| 返回值 | 无 |

## 2.附带UI界面的api接口

### 2.1 评论列表界面

使用viewComment或getCommentBtn需要在项目的AndroidMainfest.xml中注册CyanCommentActivity

<activity android:name=*"com.sohu.cyan.android.sdk.activity.CyanCommentActivity"*/>

|  |  |  |
| --- | --- | --- |
| 接口名称 | viewComment | |
| 描述 | 调用畅言SDK的评论列表界面和发表评论界面 | |
| 参数 | 名称 | 说明 |
|  | topicSourceId | 文章在app中的唯一标识 |
|  | topicTitle | 文章标题 |
|  | context | 调用列表页接口的context |
| 返回值 | 无 | |

### 2.2 评论发表界面

使用editComment需要在项目的AndroidMainfest.xml中注册CyanCommentActivity

<activity android:name=*"com.sohu.cyan.android.sdk.activity.CyanPostCommentActivity"*/>

|  |  |  |
| --- | --- | --- |
| 接口名称 | editComment | |
| 描述 | 调用畅言SDK的发表评论界面 | |
| 参数 | 名称 | 说明 |
|  | context |  |
|  | topicId | 文章在畅言系统中的ID，通过loadTopic方法获取（必传） |
|  | replyId | 被回复的评论的Id，默认为0 |
|  | replyNick | 被回复的用户名称，默认为null |
| 返回值 | 无 | |

## 2.3 用户评论中心界面(包括发出的评论和接收到的评论)

需要在项目的AndroidMainfest.xml中注册RepliesActivity

<activity android:name=*"com.sohu.cyan.android.sdk.activity.RepliesActivity"*/>

|  |  |  |
| --- | --- | --- |
| 接口名称 | showUserSendAndRcvComments | |
| 描述 | 调用畅言SDK的用户评论中心界面(包括发出的评论和接收到的评论) | |
| 参数 | 名称 | 说明 |
|  | des | 0:跳转到评论中心的接收到的评论页面  1：跳转到评论中心的做出的评论页面 |
| 返回值 | 无 | |

## 消息推送

### 配置见 三.7

### 设置notification的图标

PushManager.setNotificationIcon(Context ctx,int DrawableId);

### 如何自定义 处理接收到消息推送.

**public** **class** YourReceiver **extends** BroadcastReceiver {

@Override

**public** **void** onReceive(Context context, Intent intent) {

**if** (intent.getAction().equals(PushConstants.*PUSH\_RECEIVE\_ACTION*)){

JSONObject json = **new** JSONObject(intent.getExtras().getString(

"com.avos.avoscloud.Data"));

**final** String message = json.getString("message");//收到的消息

...//自己的处理代码

}

}

}

在androidMenifest.xml中的

<receiver android:name=*"com.sohu.cyan.android.sdk.push.CyanPushReceiver"* >

改为

<receiver android:name=*"\*.YourReceiver"* >

### 如果使用自定义的评论编辑页面

回复别人的评论成功后，需要在new CyanRequestListener<SubmitResp>()的回调方法public void onRequestSucceeded(final SubmitResp SubmitData)

做如下操作：

PushManager.pushMessage(context , user\_id + "", yourStr);

user\_id:为被回复的用户user\_id.

yourStr:为你要推送的内容

### 注意事项

#### 1.需要在androidmanifest中配置application,receiver,service

#### 2.PushManager.setUseService(Context, Boolean);需要写在CyanSdk.register(...)之后

附录

# 关于代码混淆

畅言sdk中使用了java反射和泛型，因此在对项目进行代码混淆时需要保证sdk jar包中的某些类不被混淆。代码混淆配置文件请参考demo工程下的proguard-project.txt文件。