**Unity和Android、iOS交互说明**

* **一、友盟相关功能**

**（一） 友盟初始化**

**1.安卓端（android 不必调用initUmeng ，初始化已在sdk中处理）**

|  |
| --- |
| C#  /\*\*  参数说明：  appKey: 友盟的appkey  channelName: 渠道名称  *secret Push推送业务的secret 填充Umeng Message Secret对应信息*  \*/  void initUmeng(string appKey，string channelName，String *secret* ) |

**2.iOS端（使用友盟相关功能前，先调用这个方法)**

|  |
| --- |
| C#  /\*\*  参数说明：  appKey: 友盟的appkey  channelName: 渠道名称  \*/  void initUmeng(string appKey，string channelName) |

**（二）一键登录**

**1.Unity调用原生**

**（1） 初始化intLogin**

|  |
| --- |
| C#  /\*\*  参数说明：  secretKey: 登录方案密钥  \*/  void initLogin(string secretKey) |

**（2）检测一键登录环境是否可用checkEnv~~A~~vailable**

检测结果通过异步的方式回调，详见“原生调用Unity”中的envAvailableCallback方法

|  |
| --- |
| C#  void checkEnvAvailable() |

**（3）拉起一键登录界面，请求获取token authLoginCallBack**

获取token的结果通过异步的方式回调，详见“原生调用Unity”中的authLoginCallBack方法

|  |
| --- |
| C#  /\*\*  参数说明：  app\_type:1 语文 2 英语  ui\_type 1 非全屏（原来的那个），2全屏  agreements：各种协议，json字符串，格式参考如下：  {  "agreements": [  {  "name": "用户协议1",  "url": "url1"  },  {  "name": "用户协议2",  "url": "url2"  },  {  "name": "用户协议3",  "url": "url3"  }  ],  "app\_type": 1,  "ui\_type": 1  }  \*/  void authLogin(string agreements) |

**（4）关闭授权界面quitLoginPage**

|  |
| --- |
| C#  void quitLoginPage() |

**2.原生调用Unity**

**（1）envAvailableCallback(s1, s2, s3)一键登录环境是否可用的回调**

|  |
| --- |
| Java  参数说明  s1: (SDK\_CONNECT)  s2: envAvailableCallback  //code: 0 不可用 1 可用  /\*message: 不可用：不可用原因  可用：略  \*/  s3: {"code":0,"message":"失败"} |

**（2）startAuthPageFailedCallback(s1,s2,s3) 拉起登陆界面失败后的回调**

|  |
| --- |
| Java  参数说明  s1: (SDK\_CONNECT)  s2: startAuthPageFailedCallback  // 空字符串  s3: "" |

**（3）authLoginCallback(s1,s2,s3) 登录回调（只有成功的时候会回调）**

|  |
| --- |
| Java  参数说明  s1: (SDK\_CONNECT)  s2: authLoginCallback  // code: 1  // message: token  // verify\_id: 认证id  s3: {"code":1,"message":"[token]","verify\_id":"1232134214"} |

**（4）userCancelLoginCallback(s1,s2,s3)用户取消登录**

|  |
| --- |
| Plain Text  参数说明  s1: (SDK\_CONNECT)  s2: userCancelLoginCallback  /\* 空字符串 \*/  s3: "" |

**（5）userChangeLoginMethodCallback(s1,s2,s3)切换登录方式**

|  |
| --- |
| Plain Text  参数说明·  s1: (SDK\_CONNECT)  s2: userChangeLoginMethodCallback  /\* 空字符串 \*/  s3: "" |

**（三）分享(需要先初始化友盟)**

**1.Unity调用原生**

**（0）分享初始化**

|  |
| --- |
| Java      /\*\*  参数说明：  appid: 应用appid android/ios  appsecret: 应用appsecret android/ios  package\_name:应用包名 android  英语：com.qingguo.childEn  语文：com.houxiaoji.yw  universal\_links：通用链接 ios    例子：  {  "appid": "wx89be22cbf632f7ab",  "appsecret": "6ac6f88006b53f82e29c1755b4314b08",  "package\_name":"com.qingguo.childEn",  "universal\_links": "121"  }    \*/    void initShare(string json) |

**（1） 链接分享 shareWeb**

|  |
| --- |
| C#      /\*\*  参数说明：  platform:分享平台（微信：1；微信朋友圈：2） android/ios  title: 分享标题 android/ios  description: 分享描述 android/ios  img\_url: 缩略图地址(如果是本地图片需要传图片路径) android/ios  url: 跳转的链接 android/ios  image\_type: 图片类型（1网络图片2本地图片）android/ios （iOS下，如果图片路径不正确，微信SDK可能无法拉起分享）    例子：  {  "platform": "1",  "title": "title",  "description": "desc",  "img\_url": "https://111.png",  "url": "https://www.baidu.com.cn",  "image\_type":"1"  }    \*/    void shareWeb(string json) |

**（2）小程序分享shareMin**

|  |
| --- |
| Lua  /\*\*  【小程序只有分享微信好友，没有朋友圈】  参数说明：  title: 分享标题 android/ios  description: 分享描述 android/ios  img\_url: 缩略图地址(如果是本地图片需要传图片路径) android/ios  url: 兼容低版本的网页链接 android/ios  image\_type: 图片类型（1网络图片2本地图片）android/ios  min\_path:小程序路径  min\_id:原始id  例子：  {  "title": "title",  "description": "desc",  "img\_url": "https://111.png",  "url": "https://www.baidu.com.cn" ,  "image\_type":"1"，  "min\_path":"pages/page10007/xxxxxx",  "min\_id":"gh\_abded03f991d"  }    \*/    void shareMin(string json) |

**（3）单图分享shareImage**

|  |
| --- |
| Java  /\*\*  参数说明：  platform:分享平台（微信：1；微信朋友圈：2） android/ios  img\_url: 图片地址(如果是本地图片需要传图片路径) android/ios  image\_type: 图片类型（1网络图片2本地图片）android/ios    例子：  {  "platform": "1",  "img\_url": "https://111.png",  "image\_type":"1"  }    \*/  void shareImage(string json) |

**2.原生调用Unity**

**（1）shareCallback(s1,s2,s3) 链接分享结果回调**

|  |
| --- |
| C#  参数说明  s1: (SDK\_CONNECT)  s2: shareCallback  // code: 成功1 失败2  // message: 当失败的时候有值，其他情况为空  s3: {"code":1,"message":"信息"} |

**（四）推送**

**1.Unity调用iOS/Android**

Android 指定推送界面：英语 com.qingguo.childEn.UmengPushActivity，语文*com.houxiaoji.ywai.UmengPushActivity*

**（1）设置标签**

|  |
| --- |
| Java  */\*\**  *\* 设置标签*  *\* 标签可以给某一类人群推送消息*  *\* @param tag 设置标签（比如：3岁以上,男，女等）*  *\*/*  void setTag(String tag) |

**（2）删除标签(单个删除，所有删除)**

|  |
| --- |
| Java  */\*\**  *\* 删除标签*  *\* 删除标签和用户的关联*  *\* @param tag 标签*  *\*/*  void deleteTag(String tag)      /\*\*    \*删除所有标签  \*/  void deleteAllTag() |

**（3）设置别名**

|  |
| --- |
| Java    */\*\**  *\* 设置别名*  *\* 别名可以给指定用户推送消息。*  *\* @param Stringalias 设置别名 （如果 设置账号：131xxxxxxxx）*  *\* @param Stringtype 设置类型（比如：跟上边对应，类型为 phone ，或者uid）*  \* 别名和类型需要对应，手机号，或者uid  *\*/*    void setAlias(String alias,String type) |

**（4）删除别名**

|  |
| --- |
| Java    */\*\**  *\* 删除别名，与上边添加对应，用于退出账号之后调用，使退出的账号不在收到推送根据自己使用场景进行调用*  *\* @param Stringalias 设置别名 （如果 设置账号：131xxxxxxxx）*  *\* @param Stringtype 设置类型（比如：跟上边对应，类型为 phone ，或者uid）*  \* 别名和类型需要对应，手机号，或者uid  *\*/*  void removeAlias(String alias,String type) |

**（5）获取Umeng推送Token**

|  |
| --- |
| C#  /\*\*  \*获取Umeng推送Token，同步返回  如果为空的话，返回空字符串  \*/  string getUmengDeviceToken() |

**（6）获取推送信息**

|  |
| --- |
| C#  /\*\*  \* 获取离线推送自定义参数  \*/  void pushMessage() |

**2.Android/iOS回调Unity**

**（1）推送初始化成功的回调 initPushResultCallback**

注意：安卓端成功、失败均会回调；iOS端只有成功时有回调

|  |
| --- |
| C#  参数说明  s1: (SDK\_CONNECT)  s2: initPushResultCallback  // code: 1 成功, 0 失败  // message: message  s3: {"code":1,"message":"成功"} |

**（2）原生回调消息内容中的“自定义部分”给Unity：pushMessageCallback**

|  |
| --- |
| Java  参数说明  s1: (SDK\_CONNECT)  s2: pushMessageCallback    /\*\*  字典类型的json字符串  （后台传的什么，原样返回；Unity端和后台自行协商要传的内容、业务逻辑）  格式参考如下，action、type、url均为后台自定义的key，此处仅做示例用：  { "action":"doSomething","type":"customizeType","url":"urlName"}    ⚠️如果收到Unity端获取推送信息的请求后，但是没有推送消息，返回空字符串  \*/  s3: jsonString |

**（3）设置、删除标签或者别名的回调 modifyPushConfigCallback**

|  |
| --- |
| Java  参数说明  s1: (SDK\_CONNECT)  s2: modifyPushConfigCallback    /\*\* json类型的字符串  // code: 成功1 0失败  // message: 当失败的时候有值，其他情况为空  // type: 具体修改操作的类型：  1： 设置标签  2： 删除单个标签  3： 删除所有标签  4： 设置别名  5： 删除别名  // contents: 别名、标签的具体值（回调给Unity，供核对用）  \*/  s3: {"code":1,"message":"信息", "type": 1, "content": "content1, content2"} |

**二、腾讯相关功能**

**（一）一键登录**

**1.Unity调用原生**

**（1） 初始化initLogin**

|  |
| --- |
| C#  /\*\*  参数说明：  appId: 应用id  \*/  void initLogin(string appId) |

**（2）检测一键登录环境是否可用checkEnv~~A~~vailable**

检测结果通过异步的方式回调，详见“原生调用Unity”中的envAvailableCallback方法

|  |
| --- |
| C#  void checkEnvAvailable() |

**（3）拉起一键登录界面，请求获取token authLoginCallBack**

获取token的结果通过异步的方式回调，详见“原生调用Unity”中的authLoginCallBack方法

|  |
| --- |
| C#  /\*\*  参数说明：  agreements：各种协议，json字符串，格式参考如下：  [  {  "name": "用户协议1",  "url": "url1"  },  {  "name": "用户协议2",  "url": "url2"  },  {  "name": "用户协议3",  "url": "url3"  }  ]  \*/    void authLogin(string agreements) |

**（4）关闭授权界面quitLoginPage**

|  |
| --- |
| C#  void quitLoginPage() |

**2.原生调用Unity**

**（1）envAvailableCallback(s1, s2, s3)一键登录环境是否可用的回调**

|  |
| --- |
| Plain Text  参数说明  s1: (SDK\_CONNECT)  s2: envAvailableCallback  //code: 0 不可用 1 可用  /\*message: 不可用：不可用原因  可用：略  \*/  s3: {"code":0,"message":"失败"} |

**（2）startAuthPageFailedCallback(s1,s2,s3) 拉起登陆界面失败后的回调**

|  |
| --- |
| C#  参数说明  s1: (SDK\_CONNECT)  s2: startAuthPageFailedCallback  // 空字符串  s3: "" |

**（3）authLoginCallback(s1,s2,s3) 登录回调（只有成功的时候会回调）**

|  |
| --- |
| JSON  参数说明  s1: (SDK\_CONNECT)  s2: authLoginCallback  // code: 1  // message: token  // verify\_id: 腾讯api未提供该参数，暂时为空  s3: {"code":1,"message":"[token]","verify\_id":""} |

**（4）userCancelLoginCallback(s1,s2,s3)用户取消登录**

|  |
| --- |
| Plain Text  参数说明  s1: (SDK\_CONNECT)  s2: userCancelLoginCallback  /\* 空字符串 \*/  s3: "" |

**（5）userChangeLoginMethodCallback(s1,s2,s3)切换登录方式**

|  |
| --- |
| Plain Text  参数说明·  s1: (SDK\_CONNECT)  s2: userChangeLoginMethodCallback  /\* 空字符串 \*/  s3: "" |

**三、华为OBS**

**（一）Unity调用原生**

**1.初始化OBS**

|  |
| --- |
| C#  /// <summary>  /// 初始化OBS  /// ak: Access Key Id  /// sk： Secret Access Key  /// endpoint: endpoint  /// bucketName: 桶名  /// </summary>  void initOBSSDK(string ak,string sk,string endpoint, string bucketName) |

**2.上传文件**

|  |
| --- |
| C#  /// <summary>  /// 上传资源  /// fileName: 要上传的文件的绝对路径  /// urlName： 指定的url地址（OBS作为objectKey处理，在回调时原封不动返回）  /// customizeStr: Unity用来处理业务逻辑，在回调时原封不动回调给Unity  /// </summary>  void uploadResources(string fileName,string urlName,string customizeStr) |

**（二）原生回调给Unity：uploadResourcesCallback**

|  |
| --- |
| Java  参数说明  s1: (SDK\_CONNECT)  s2: uploadResourcesCallback  // code: 0 失败  1 成功  // customizeStr: 原路返回  // urlName: 原路返回  s3: { "code":1,"customizeStr":"customizeStr","urlName":"urlName"} |

**四、Sentry**

**初始化Sentry（ios专用，Android已在sdk内部配置）**

|  |
| --- |
| C#  /\*  参数说明：  dsn: 具体项目的dsn  \*/  void initSentry(string dsn) |

**五、其他功能**

**1.剪贴板读取功能**

**（1）复制文本到剪切板**

|  |
| --- |
| C#  /\*  参数说明：  content:需要复制的文本  \*/  void copyToClipboard(string content) |

**（2） 获取剪切板内容**

|  |
| --- |
| C#  /\*  获取剪切板上的文本内容  （如果为空的话，返回空字符串）  \*/  string getClipboardContent() |

**2.Unity调用原生WebView**

|  |
| --- |
| C#  /\*  参数：  url:  h5 url地址，必须传  title:  标题，可选。如果该参数有值，WebView界面的标题用该值，否则，动态显示WebView的标题  orientation:  朝向："1"为横屏，"2"为竖屏  \*/  void showWebView(string url, string title, string orientation)    /\*    和上边方法不通之处在于：下边这个调用的时候，点击返回按钮或者手势返回，直接finish 当前界面，不会判断webview 的cangoback  参数：  url:  h5 url地址，必须传  title:  标题，可选。如果该参数有值，WebView界面的标题用该值，否则，动态显示WebView的标题  orientation:  朝向："1"为横屏，"2"为竖屏    \*/  void showWebViewWithBack(string url, string title, string orientation) |

**2.1webview 的关闭回调： webviewFinishCallback(s1,s2,s3)**

|  |
| --- |
| Makefile    参数说明  s1: (SDK\_CONNECT)  s2: webviewFinishCallback  s3: "" |

**3.获取设备唯一标识**

**（1）Unity调用原生**

|  |
| --- |
| C#  /\*  获取设备唯一标识，同步返回  \*/  void getDeviceUniqueIdentifier() |

**（2）原生回调Unity**

|  |
| --- |
| Groovy  参数说明  s1: (SDK\_CONNECT)  s2: deviceUniqueIdentifierCallback  // code: 0 失败 1 成功  // message: 成功：设备唯一码  失败：错误信息（如果有的话）  \*/  s3: {"code":1,"message":""} |

**4.获取手机品牌 Android**

|  |
| --- |
| C#  /\*\*  \* 获取手机品牌（Android），同步返回  \*/  string getPhoneBrand() |

**六、支付**

**1.微信支付**

|  |
| --- |
| Java    */\*\**  *\* 微信支付*  *\**  *\* @param context*  *\* @param json 支付所需要的json，包含以下内容*  *\* 应用id appid*  *\* 商户号 partnerid*  *\* 预支付交易会话ID prepayid*  *\* 订单详情扩展字符串 package*  *\* 随机字符串 noncestr*  *\* 时间戳 timestamp*  *\*参数详细描述请查看微信官方文档：*  \* [*https://pay.weixin.qq.com/wiki/doc/apiv3/apis/chapter3\_2\_4.shtml*](https://pay.weixin.qq.com/wiki/doc/apiv3/apis/chapter3_2_4.shtml)  *\** *@param appId 微信appid 回调需要用*  *\*/*  void payWx(String json, String appId) |

**2.支付宝支付**

|  |
| --- |
| Java    */\*\**  *\* 支付宝支付*  *\**  *\* @param orderInfo 支付生成订单信息，需要服务端生成，如下*    *\* 服务端接入流程*  *https://opendocs.alipay.com/open/204/01dcc0*  *\*/*    void payAli(String orderInfo) |

**七.下载apk**

**1.下载**

|  |
| --- |
| Java  */\*\**  *\* 下载apk*  *\* @param path 下载到本地路径完整地址（xx/xx/xx/x.apk）*  *\* @param url apk 下载地址*  *\*/*  void downloadApk(String url,String path) |

**2.点击状态栏安装 noticeInstallApkCallBack**

|  |
| --- |
| Java  参数说明  s1: (SDK\_CONNECT)  s2: noticeInstallApkCallBack    s3: { "code":1,"msg":"install"} |

**3.apk 安装**

|  |
| --- |
| Java    */\*\**  *\* 下载apk*  *\* @param filePath下载到本地路径完整地址（xx/xx/xx/x.apk）*  *\*/*  void installApk(String filePath) |