

目录

一、 说明.....	2
1. 适用范围.....	2
2. 统计说明.....	2
二、 接入流程	2
第一步、申请 APP ID	2
第二步、下载 SDK，导入到项目	2
第三步、接入 SDK.....	3
第四步、测试	3
三、 接入方法说明	3
1. 配置应用权限到 AndroidManifest.xml 文件	3
2. 应用中所有 Activity 继承基类 BaseActivity。	4
3. 初始化热云 SDK	4
4. 统计玩家服务器注册数据	5
5. 统计玩家的账号登陆数据	7
6. 统计玩家的充值数据	9
7. 统计玩家在游戏内的虚拟交易数据	10
8. 统计玩家的任务、副本数据	11
9. 统计玩家的自定义事件	12
10. 退出 sdk	13
四、 技术支持	14

一、说明

1. 适用范围

热云-游戏运营支撑平台为游戏开发者提供全方位的数据分析服务，帮助开发者解决在游戏运营过程中的遇到的数据分析问题。

本文档适用于使用 Android 来开发游戏的产品，目前兼容 Android2.2 及以上操作系统的设备。

2. 统计说明

为了能够在接入过程中保持概念的统一，针对一些通用的概念做如下说明：

✓ 设备

指某一台安装了 Android 游戏的终端，例如 Samsung GALAXY Note II、HTC one 、Samsung GALAXY S3 等。

✓ 账号

指玩家在某一台设备上安装了游戏之后，在游戏中的唯一标识符，通常可以用作账号的标识符可以是用户注册游戏时的唯一用户名，例如邮件地址，也可以是游戏在用户注册时为玩家自动生成相对应的唯一字符串。

二、接入流程

第一步、申请 APP ID

打开<http://game.reyun.com>，使用您的热云账号登陆后在“我的游戏”管理后台，点击“添加游戏”，在弹出的面板中创建新游戏，完成后您将获得一个 32 位的 16 进制 APP ID，用来唯一标识您刚刚创建的游戏。

注：为了保证您的数据安全，请勿泄露您的 APP ID 给其他人。

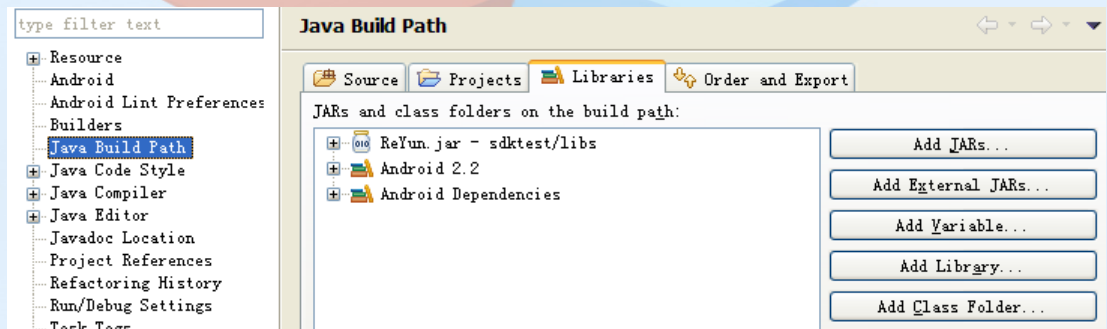
第二步、下载 SDK，导入到项目

打开<http://game.reyun.com/doc#download>，下载和您的游戏开发语言相对应的 SDK。

下载完成后解压缩到本地目录，将目录中的 reyun.jar 导入到您的工程目录下的 libs/ 文件夹中。

导入方法：在工程目录结构中，右键选择 Properties->Java Build Path, 点击右侧 Libraries 选项卡，点击 Add JARs 按钮在项目中 libs/目录选择 reyun.jar, 点击 OK 导入

操作方法如下图所示：



第三步、接入 SDK

详见：第三步的接入方法说明来完成 SDK 的接入。

第四步、测试

测试包括在 SDK 的接入过程中在开发环境中进行 Debug 测试，以及接入完成后，将游戏打包后进行测试。

三、接入方法说明

1. 配置应用权限到 AndroidManifest.xml 文件

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission
android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.READ_LOGS"/>
<uses-permission android:name="android.permission.GET_TASKS" />
```

2. 应用中所有 Activity 继承基类 BaseActivity。BaseActivity 类代码如下：

```
public class BaseActivity extends Activity {

    boolean isActive = true;

    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {  
    // TODO Auto-generated method stub  
    super.onCreate(savedInstanceState);  
}  
  
@Override  
protected void onResume() {  
    // TODO Auto-generated method stub  
    super.onResume();  
  
    if( !isActive){ // app 从后台唤醒到前台  
  
//        ReYun.postSessionData();  
        ReYun.startHeartBeat();  
        isActive =true;  
    }  
}  
  
@Override  
protected void onStop() {  
    // TODO Auto-generated method stub  
    super.onStop();  
    if (ReYun.isAppOnForeground() == false) {  
  
        //app 进入后台  
        isActive = false;  
    }  
}  
}
```

注：如果在程序不继承了BaseActivity，继承其他Activity也可以，仿照baseactivity 写就行，但是得保证都继承一个activity，代码里面所有 activity 都继承自一个基类 activity，比如他们的 UnityActivity

3. 初始化热云 SDK

方法用途：

- ✓ 用于在游戏启动后，初始化热云 SDK。报送游戏安装或启动事件

使用方法：

- ✓ 调用 `Reyun.initWithKeyAndChannelId(String appId,String channelId)` 方法进行初始化。
- ✓ 参数 `channelId` 为渠道 ID,如果传 `null` 或 `""`则默认向热云报送为 `unknown`, 如果设置了渠道 ID, 则可以在游戏运营支撑平台中查询到不同渠道的数据, 对于不同的渠道包, 需要分别设置不同的渠道 ID。需要重新编译打包 apk

接口说明:

方法接口: `Reyun.initWithKeyAndChannelId(Context appContext ,String appKey,String channelId);`

参数说明:

参数	类型	强制正确填写	描述
appContext	Context	是	需获取全局应用上下文
appKey	String	是	填写在创建游戏时获得的 32 个字符长度的 APP ID
channelId	String	否	填写用来标识推广渠道的字符, 支持中文、英文、数字、下划线, 最长 32 个字符, 默认 unknown

示例代码:

```
ReYun.initWithKeyAndChannelId(getApplicationContext(),  
"8283e21a7484c03edee3d61cc12e93ss", "moboge");
```

4. 统计玩家服务器注册数据

方法用途:

- ✓ 用于在玩家首次进入某个服务器, 用来统计这个服务器一个新的玩家账号 (账号需使用游戏的平台 id, 这样方便统计用户滚服的信息)。如果是单服游戏, 那么在游戏内只需调用一次即可。多个服务器的游戏需要在首次登服务器时调用一次且仅一次。

- ✓
- ✓ 玩家账号需要保证唯一性，否则会造成统计数据出错，作为统计使用的玩家账号一般为游戏内给玩家生成的唯一 ID。
- ✓ 当某个玩家用相同的账号在不同的设备上登录时，新增账号不会增加，但设备激活会增加，但在相同设备上有多不同账号登录时，新增账号会增加，但设备激活不增加。

使用方法：

- ✓ 在玩家首次进入某个服务器时调用 `setRegisterWithAccountID` 方法，具体注册账号的时机和位置由开发者自己来决定，但要确保参数 `accountId` 不能为空。
- ✓ 如果开发者没有自己的用户系统，希望使用用户的设备 ID 来作为 `accountId`，直接调用 `Reyun.getDeviceId()` 方法即可。

接口说明：

方法接口：`setRegisterWithAccountID(String accountId, String accountType, Gender gender, int age, String serverId);`

参数说明：

参数	类型	强制正确填写	描述
accountId	String	是	设定账号的唯一标识 ID
accountType	String	否	设定账号的类型，分匿名和游戏自有注册和其他第三方开放平台等： 匿名：anonymous 自有注册：registered 第三方开放平台如： 新浪微博：sinaWeibo 微信：wechat

			QQ: qq Facebook: facebook Twitter: twitter 开发者也可以自行传入此参数 默认: unknown
gender	Enum	否	设定玩家性别: 男: M 女: F 其他: 0 默认: unknown
age	int	否	设定玩家年龄, 范围为 0-120, 默认: -1
serverId	String	否	设定玩家登陆的区服, 最多 16 个字符, 默认: UNKNOWN

示例代码:

当用户首次进入 xiaobaojianghu 服务器时调用:

```
Reyun.setRegisterWithAccountID ("123456", ,AccountType.ANONYMOUS.name(),
Gender.UNKNOWN,10,"xiaobaojianghu");
```

5. 统计玩家的账号登陆服务器数据

方法用途:

- ✓ 统计玩家在游戏中的登陆、账号切换操作, 游戏的每日/周/月活跃人数、登陆次数/频次、留存数据等重要数据均通过这个方法统计。
- ✓ 每次登陆后 accountId、serverId 信息缓存至本地, 其他后续的方法调用使用到这几个参数时, 均使用本地缓存的值来报送。

使用方法:

- ✓ 当玩家有登陆、切换账号操作时调用 `setLoginWithAccountID` 方法。

接口说明:

方法接口: `setLoginWithAccountID(final String accountId, final int level, final String serverId);`

参数说明:

参数	类型	必填	描述
accountId	String	是	设定账号的唯一标识 ID
level	int	否	设定玩家的等级, 范围为: 0-1000, 默认: -1
serverId	String	否	设定玩家登陆的区服, 最多 16 个字符, 默认: unknown

示例代码:

```
Reyun.setLoginWithAccountID("654321", 26, "xiaobaojianghu");
```

6. 统计玩家的充值数据

方法用途:

- ✓ 在玩家在游戏中充值成功后, 进行统计充值数据, 所有付费相关分析的数据报表均依赖此方法。
- ✓ 当系统通过不同形式赠送虚拟货币给玩家时, 调用此方法来记录系统赠送的虚拟币数据, 当 `paymentType` 为 `FREE` 时, 系统认为是系统赠送的, 不会计算到当天的收入中。

使用方法:

- ✓ 在玩家充值成功后调用 `setPayment` 方法。

接口说明:

方法接口: `setPayment(String transactionId, String paymentType, String currencyType, float currencyAmount, float virtualCoinAmount, String iapName, long iapAmount, level);`

参数说明:

参数	类型	强制正确填写	描述
transactionId	String	是	交易的流水号, 最长 64 个字符
paymentType	String	是	支付类型, 例如支付宝, 银联, 苹果、谷歌官方等, 最多 16 个字符, 如果是系统赠送的, paymentType 为: FREE
currencyType	String	是	货币类型, 按照国际标准组织 ISO 4217 中规范的 3 位字母, 例如 CNY 人民币、USD 美金等, 详情请点击
currencyAmount	float	是	支付的真实货币的金额, 最长 16 个字符
virtualCoinAmount	float	是	通过充值获得的游戏内货币的数量, 最长 16 个字符
iapName	String	是	游戏内购买道具的名称, 最长 32 个字符
iapAmount	long	是	游戏内购买道具的数量, 最长 16 个字符
level	int	是	用户游戏等级

示例代码:

```
Reyun.setPayment("0062001242", PaymentType.FREE.name(), "USD", 1000.00f, 2000.00f, "diamond" 1000, 28);
```

7. 统计玩家在游戏内的虚拟交易数据

方法用途：

- ✓ 统计游戏内玩家产生的虚拟货币消费数据。

使用方法：

- ✓ 当玩家有虚拟货币消费时，调用 `setEconomy` 方法。

接口说明：

方法接口：`setEconomy(String itemName, long itemAmount, float itemTotalPrice, int level);`

参数说明：

参数	类型	必填	描述
itemName	String	是	游戏内虚拟物品的名称/ID，最长 64 个字符
itemAmount	long	是	交易的数量，最长 16 个字符
itemTotalPrice	float	是	交易的总价，最长 16 个字符
level	int	是	用户游戏等级

示例代码：

```
Reyun.setEconomy("diamond",1000,2000.0f,28);
```

8. 统计玩家的任务、副本数据

方法用途：

- ✓ 统计玩家在游戏内接受任务、完成任务和进入副本、结束副本等数据。

使用方法：

- ✓ 在玩家接受任务/关卡/副本、结束任务/关卡/副本，失败任务/关卡/副本时调用 `setQuest` 方法。

接口说明:

方法接口: `setQuery(String questId, QuestStatus status ,String questType, int level);`

参数说明:

参数	类型	必填	描述
questId	String	是	当前任务/关卡/副本的编号或名称, 最长 32 个字符
QuestStatus	Enum	是	当前任务/关卡/副本的状态, 有如下三种类型: 开始: a 完成: c 失败: f
questType	String	是	当前任务/关卡/副本的类型, 例如: 新手任务: new 主线任务: main 支线任务: branch 开发者也可以根据自己游戏的特点自定义类型, 最多 16 个字符
level	int	是	玩家游戏等级

示例代码:

```
Reyun.setQuest("Arena", QuestStatus.F, "branch",29);
```

9. 统计玩家的自定义事件

方法用途:

- ✓ 开发者可以自由的统计玩家在游戏内的任意用户行为, 例如打开某个面板, 点击某个 Button, 参与某个活动等。

- ✓ **eventName** 只能是合法的字符，包括中文、英文字母、数字和下划线。
- ✓ 开发者调用此方法后，即可在热云的游戏运营支撑平台使用多维分析的强大功能。

使用方法：

- ✓ 在您希望进行用户行为统计的地方，调用 **setEvent** 方法。
- ✓ 除了开发者可以自定义 **eventName** 以外，还可以通过 **extra** 参数以字典的格式来记录更多信息，例如一款德州 **Poker** 游戏，定义了 **eventName** 为一盘结束，那么您可以同时记录牌型、下注次数、同桌牌友、输赢筹码等数据。

接口说明：

方法接口：**setEvent**(final String eventName, Map extra);

参数说明：

参数	类型	必填	描述
eventName	String	是	自定义事件的名称
extra	Map	否	无

示例代码：

```
Map extra = new HashMap();  
  
extra.put("itemid", "798");  
  
extra.put("itemValue", "123");  Reyun.setEvent("click button", extra);
```

示例

大家都知道，对于一款刚刚上线的游戏、或者对于第一次接触游戏的用户而言，游戏开始的引导是非常重要的环节，甚至是决定了玩家是否能够留存的关键，因此我们专门做了一张报表来帮助开发者分析这部分的数据，如果开发者希望便捷的使用这个报表，可以按照如下的预留 **eventName** 来在相关的位置报送数据：

启动游戏/开始加载游戏->完成游戏加载->开始新手引导->结束新手引导->开始 FTUE (First Time User Experience) ->结束 FTUE, 分别对应:

startGame->loadComplete->startTutorial->endTutorial->startFTUE->endFTUE。

示例代码:

```
Reyun. setEvent("startGame", null);
```

```
Reyun. setEvent("loadComplete", null);
```

```
Reyun. setEvent("startTutorial", null);
```

```
Reyun. setEvent("endTutorial", null);
```

10. 退出 sdk

方法用途: 应用退出时释放 sdk 占用资源。

使用方法:

✓ 在程序退出时候调用 `exitSdk` 方法。

接口说明:

方法接口: `exitSdk ()`;

参数说明: 无参数

示例代码

```
Reyun. exitSdk ();
```

备注: 获取手机唯一识别号方法:

```
String deviceId = "";  
if (checkPhoneState(context)) {  
    TelephonyManager tm = (TelephonyManager)  
context
```

```
.getSystemService(Context.TELEPHONY_SERVICE);  
deviceId = tm.getDeviceId();  
  
if(deviceId == null){  
    deviceId =  
Settings.Secure.getString(context.getContentResolver(),  
Settings.Secure.ANDROID_ID);  
    }  
}  
if (deviceId != null) {  
  
    return deviceId;  
} else {  
  
    return "unknown";  
}
```

四、技术支持

如果您有任何问题，我们的技术支持工程师会第一时间给您提供帮助，
可以通过如下方式联系：

- ✓ 技术支持邮箱： support@reyun.com
- ✓ 技术支持 QQ： 8474881
- ✓ 技术支持电话： 18601925200