

## 目录

一、 说明.....	1
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 .....	4
1. 初始化热云 SDK .....	4
2. 统计玩家服务器注册数据 .....	5
3. 统计玩家登入服务器数据 .....	7
4. 统计玩家的充值数据 .....	8
5. 统计玩家在游戏内的虚拟交易数据 .....	10
6. 统计玩家的任务、副本数据 .....	11
7. 统计玩家的自定义事件 .....	12
四、 技术支持 .....	13

## 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 **Objective C** 来开发游戏的产品，目前兼容 **iOS4.3** 以上操作系统的设备，如果您使用 **Cocos2D-X** 框架或者 **Unity** 引擎来开发，请下载专用的 **SDK** 和接入文档进行 **SDK** 的集成。

### 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

指某一台安装了游戏的终端，例如 iPhone、iPad、iTouch

✓ 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.h** 和 **reyun.a** 导入到您的工程中。

导入方法：在工程目录结构中，右键选择 Add->Existing Files...，选择 reyun.h 和 reyun.a 或者将这两个文件拖入 XCode 工程目录结构中，在弹出的界面中勾选 Copy items into destination group's folder(if needed), 并确保 Add To Targets 勾选相应的 target。

添加 SDK 所需要的依赖框架，热云-游戏运营支撑平台：

使用 Security.framework 来存储设备标识，

使用 CoreTelephony.framework 来获取运营商信息，

使用 AdSupport.framework 来获取 Advertising Identifier 信息，

使用 SystemConfiguration.framework 来检查当前网络环境，










使用 libsqlite3.dylib 用来存储数据。

添加方法：在工程目录中，选择

TARGETS-->Build Phases-->Link Binary With Libraries-->+ -->选择

Security.framework、CoreTelephony.framework、AdSupport.framework、

libsqlite3.dylib，如下图所示：

Name	Status
 Security.framework	Required ↕
 CoreTelephony.framework	Optional ↕
 AdSupport.framework	Required ↕
 SystemConfiguration.framework	Required ↕
 CFNetwork.framework	Required ↕
 libsqlite3.0.dylib	Required ↕
 CoreGraphics.framework	Required ↕
 UIKit.framework	Required ↕
 Foundation.framework	Required ↕

### 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. 初始化热云 SDK

方法用途：

- ✓ 用于在游戏启动后，初始化热云 SDK。

使用方法：

- ✓ 在初始化文件 `appdelegate.m` 中导入 `reyun.h`，在 `application:didFinishLaunchingWithOptions` 方法中调用 `reyun initWithAppkey` 方法进行初始化。
- ✓ 通过传入 `channelid` 参数方法来设置渠道 ID，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 并重新编译打包，默认的渠道 ID 是 `unknown`。

接口说明：

方法接口：

```
+(void)initWithAppkey(NSString*)appKey channelId(NSString*)channelID;
```

参数说明：

参数	类型	必填	描述
appKey	NSString	是	填写在创建游戏时获得的 32 个字符长度的 APP ID
channelID	NSString	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：

```
#import "reyun.h"

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary
*)launchOptions{

    [reyun initWithAppKey:@"475938c702f7451a88eaffb524962649"
channelID:@"QQ"];

}
```

## 2. 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 或者用户平台 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：

- ✓ 在玩家注册账号时调用 `reyun 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 `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 `getDeviceId` 方法即可。

接口说明：

方法接口：

```
+(void) setRegisterWithAccountID:(NSString *)account andGender:(gender)gender
andage:(NSString*)age andServerId:(NSString *)server andAccountType:(NSString
*)accountType;
```

参数说明：

参数	类型	必填	描述
account	NSString	是	设定账号的唯一标识 ID
accountType	NSString	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
			设定玩家性别：  男：m  女：f



			其他: o 默认: unknown
age	NSString	否	设定玩家年龄, 范围为 0-120, 默认: -1
serverId	NSString	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 456 服时调用

```
[reyun setRegedWithAccountID:@"hero" andGender:f andage:@"123"
andServerId:@"456" andAccountType:@"1231"];
```

### 3. 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、level、gender 和 age、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 reyun setLoginWithAccountid 方法。

接口说明:

方法接口:

```
+(void)setLoginWithAccountID:(NSString *)account andGender:(gender)gender
andage:(NSString*)age andServerId:(NSString *)server andlevel:(NSInteger)level;
```

参数说明:

参数	类型	必填	描述
accountId	NSString	是	
level	NSInteger	否	设定玩家的等级，范围为：0-1000，默认：-1
gender	Enum	否	设定玩家性别： 男：m 女：f 其他：o 默认：unknown
age	NSString	否	设定玩家年龄，范围为 0-120，默认：-1
serverId	NSString	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:@"1234" andGender:f andage:@"12"
andServerId:@"456" andlevel:12];
```

## 4. 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 `reyun setPayment` 方法。

接口说明:

方法接口:

```
+(void)setPayment:(NSString *)transactionId paymentType:(NSString*)paymentType  
currentType:(NSString*)currentType currencyAmount:(float)currencyAmount  
virtualCoinAmount:(float)virtualCoinAmount iapName:(NSString*)iapName  
iapAmount:(NSInteger)iapAmount andLevel:(NSInteger)level;
```

参数说明:

参数	类型	必填	描述
transactionId	NSString	是	交易的流水号，最长 64 个字符
paymentType	NSString	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	NSString	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， <a href="#">详情请点击</a>
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	NSString	是	游戏内购买道具的名称，最长 32 个字符
level	NSInteger	否	设定玩家的等级，范围为：0-1000，默认：-1
iapAmount	NSInteger	是	游戏内购买道具的数量，最长 16 个字符

示例代码:

```
[reyun setPayment:@"123" paymentType:@"apple" currentType:@"CNY"  
currencyAmount:12.5 virtualCoinAmount:12.5 iapName:@"coinPay" iapAmount:123  
andlevel:12];
```

## 5. 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 `reyun setEconomy` 方法。

接口说明: 调用时机: 虚拟交易发生之后调用

方法接口: `+(void)setEconomy:(NSString *)itemName  
andEconomyNumber:(NSInteger)itemAmount  
andEconomyTotalPrice:(float)itemTotalPrice andlevel:(NSInteger)level;`

参数说明:

参数	类型	必填	描述
itemName	NSString	是	游戏内虚拟物品的名称/ID, 最长 64 个字符
itemAmount	NSInteger	是	交易的数量, 最长 16 个字符
level	NSIntger	否	设定玩家的等级, 范围为: 0-1000, 默认: -1
itemTotalPrice	float	是	交易的总价, 最长 16 个字符

示例代码:

```
[reyun setEconomy:@"12" andEconomyNumber:22 andEconomyTotalPrice:12.5  
andlevel:12];
```

## 6. 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 `reyun setQuest` 方法。

接口说明: 调用时机: 用户接受任务/用户完成任务之后

方法接口: `+(void)setQuest:(NSString *)questId`

`andTaskState:(questStatus)questStatus andTaskType:(NSString *)questType;`

参数说明:

参数	类型	必填	描述
questId	NSString	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: start 完成: done 失败: fail
			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

示例代码:

```
[reyun setQuest:@"123" andTaskState:start andTaskType@"123"];
```

## 7. 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 **reyun 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: +

```
+(void)setEvent:(NSString *)eventName andExtra:(NSDictionary *)extra;
```

方法接口: 调用时机: 事件发生之后调用

参数说明:

参数	类型	必填	描述
eventName	NSString	是	自定义事件的名称
extra	NSDictionary	否	NSDictionary 的 key 只能是 NSString 类型，value 目前支持字符串、数字、日期和布尔类型。

示例代码:

```
NSMutableDictionary* dict = [[NSMutableDictionary alloc]
initWithObjectsAndKeys:@"value1",@"key1",nil];

[reyun setEvent:@"value" andExtra:dict];
```

示例 1:

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 2:

```
NSMutableDictionary *tempDic = [[NSMutableDictionary alloc]
initWithObjectsAndKeys:@"value1",@"key1", nil];

[reyun setEvent:@"startGame" andExtra: tempDic];
```

## 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](mailto:support@reyun.com)
- ✓ 技术支持 QQ: 8474881
- ✓ 技术支持电话: 18601925200

