淘米数据平台接入指南-1.0.0

C01研发部-统计平台项目组

2013年11月

目录

[淘米数据平台接入指南-1.0.0 1](#_Toc381796357)

[C01研发部-统计平台项目组 1](#_Toc381796358)

[2013年11月 1](#_Toc381796359)

[一、综述 2](#_Toc381796360)

[1、适用范围 2](#_Toc381796361)

[2、统计标准 2](#_Toc381796362)

[3、问题联系人 2](#_Toc381796363)

[二、接入流程 2](#_Toc381796364)

[1、为游戏申请GameID 2](#_Toc381796365)

[2、向工程中添加logger库文件 3](#_Toc381796366)

[3、添加调用方法 3](#_Toc381796367)

[4、进行数据测试 3](#_Toc381796368)

[三、添加调用方法(C++) 3](#_Toc381796369)

[0、构造函数 3](#_Toc381796370)

[1、注册米米号 3](#_Toc381796371)

[2、登录游戏验证用户名和密码 4](#_Toc381796372)

[3、创建游戏中角色 5](#_Toc381796373)

[4、登录游戏服务器(online服务器) 6](#_Toc381796374)

[5、登录设备数(b03使用) 7](#_Toc381796375)

[6、用户退出游戏服务器(online服务器) 8](#_Toc381796376)

[7、统计当前在线人数 8](#_Toc381796377)

[8、用户升级 9](#_Toc381796378)

[9、付费 9](#_Toc381796379)

[10、获得游戏内一级币(这里的获得是指通过游戏内部赠送或完成任务奖励，而非用户购买) 10](#_Toc381796380)

[11、使用游戏内一级货币购买道具 11](#_Toc381796381)

[12、消耗游戏内一级货币 11](#_Toc381796382)

[13、接收任务 12](#_Toc381796383)

[14、完成任务 12](#_Toc381796384)

[15、放弃任务 12](#_Toc381796385)

[16、退订VIP服务(取消自动续费) 13](#_Toc381796386)

[17、销户(VIP时间到期) 13](#_Toc381796387)

[18、自定义统计 14](#_Toc381796388)

[18.1、概述 14](#_Toc381796389)

[18.2、示例 15](#_Toc381796390)

[四、添加调用方法(PHP) 16](#_Toc381796391)

[4.1、注册角色 17](#_Toc381796392)

[4.2 用户登录游戏(做活跃统计) 17](#_Toc381796393)

[4.3、付费(开通VIP、米币购买道具、人民币购买游戏内一级货币) 18](#_Toc381796394)

[4.4、自定义统计项接口(只支持统计人数人次) 19](#_Toc381796395)

[五、web端统计或AS统计 19](#_Toc381796396)

# 一、综述

## 1、适用范围

淘米数据平台帮助各游戏部门解决玩家数据收集到数据标准化分析的全部繁琐过程，以行业标准形式展现于报表中。

SDK适用于C++程序开发的后台程序调用。

## 2、统计标准

统计中我们采用游戏自身的账户(米米号)作为一个玩家单元。

付费: B01是指通过米币购买道具和开通VIP包月服务

B03是指通过人民币购买游戏内货币

## 3、问题联系人

3.1、系统不支持的需求以及系统中指标不明确的地方，请联系C01统计平台项目组berry

3.2、使用方式和显示方面的问题， 请联系C01研发部violet

3.3、接入方式有不明确的地方，请联系C01研发部henry

# 二、接入流程

## 1、为游戏申请GameID

申请地址：预先创建一款游戏，您将获得一个整型的gameid,用于唯一标识您的这款游戏

## 2、向工程中添加logger库文件

下载统计的logger压缩包并解压至本地,将其中的lib32bitstatlogger.a(32位)或lib64bitstatlogger.a(64位)和statlogger.h文件加入您的工程代码中。

下载地址:

http://10.1.1.60/dev/doku.php?id=10.1.1.104

## 3、添加调用方法

参考“三、添加调用方法”

## 4、进行数据测试

调用方法添加完毕后，应当进行数据测试，以确保方法调用正确。

测试方法：

# 三、添加调用方法(C++)

先初始化一个StatLogger类的实例,只需要每次后台服务启动时初始化一次即可，建议做成全局的。

StatLogger logger(1);

统计数据一律强制写入/opt/taomee/stat/data/目录下，需要提前建立该目录，并将该目录开通777的权限，否则程序无法启动

## 0、构造函数

StatLogger( int game\_id, //游戏ID,由数据分析平台统一分配

int16\_t zone\_id = -1, //区ID，默认-1表示全区或不分区

int16\_t svr\_id = -1, //服ID，默认-1标识全服或不分服

int16\_t site\_id = -1, //平台ID，默认为-1代表全平台，如果这款游戏将来或者现在可能放到其他外部平台运营，那么这里填上对应平台的ID，淘米平台用1表示

int isgame = 1 //标识日志是来自游戏后台(1)还是公共平台组(0);游戏后台无需传该参数，采用默认值即可

);

## 1、注册米米号

**接口定义**

void reg\_account(std::string acct\_id,

uint32\_t cli\_ip,

std::string ads\_id,

std::string browser = "",

std::string device = "",

std::string os = "",

std::string resolution = "",

std::string network = "",

std::string isp = "" );

**参数说明**

|  |  |
| --- | --- |
| acct\_id | 用户账户(米米号) |
| cli\_ip | 用户的IP地址,无法获取时取值0 |
| ads\_id | 用户是从哪个广告渠道跳转过来的 |
| browse | 用户使用的浏览器 |
| device | 用户使用的设备 |
| os | 用户使用的操作系统 |
| resolution | 用户屏幕的分辨率 |
| network | 用户使用的网络 |
| isp | 用户网络的服务提供商 |

**用途**

1、统计项名称(stid)：\_ newuid \_ 子统计项名称(sstid): \_ newuid \_

用于统计每日注册米米号的数量以及这些用户的地区分布

每天从各个广告渠道带来的注册米米号数量

使用不同的浏览器注册的米米号数量

使用不同的设备注册的米米号数量

使用不同的操作系统、分辨率、网络以及网络服务提供商注册的米米号数量

**用法**

logger.reg\_account(“47159775”, “102.78.34.112”,”4399.com”, ‘firefox”, “ipad2.0”, “ubuntu10”, “1024\*768”, “3G”, “电信”);

## 2、登录游戏验证用户名和密码

**接口定义**

void verify\_passwd (std::string acct\_id,

uint32\_t cli\_ip,

std::string ads\_id,

std::string browser = "",

std::string device = "",

std::string os = "",

std::string resolution = "",

std::string network = "",

std::string isp = "" );

**参数说明**

|  |  |
| --- | --- |
| acct\_id | 用户账户(米米号) |
| cli\_ip | 用户的IP地址, 无法获取时取值0 |
| ads\_id | 用户是从哪个广告渠道跳转过来的 |
| browse | 用户使用的浏览器 |
| device | 用户使用的设备 |
| os | 用户使用的操作系统 |
| resolution | 用户屏幕的分辨率 |
| network | 用户使用的网络 |
| isp | 用户网络的服务提供商 |

**用途**

1、统计项名称(stid)：\_veripass\_ 子统计项名称(sstid): \_veripass\_  
 统计每天验证用户名和密码的人数人次以及这些用户的地区分布

从各个广告位过来的用户验证用户名和密码的人数人次

使用不同的浏览器进行验证用户名和密码的人数人次

使用不同设备、操作系统、分辨率、网络以及网络服务提供商进行验证用户名和密码的人数人次

**用法**

logger.verify\_passwd(“47159775”, “106.235.12.11”, “innermedia.taomee.seer.topbar”);//不按照浏览器、设备等维度进行统计分析

## 3、创建游戏中角色

**接口定义**

void reg\_role(std::string acct\_id,

std::string player\_id,

std::string race,

uint32\_t cli\_ip,

std::string ads\_id,

std::string browser = "",

std::string device = "",

std::string os = "",

std::string resolution = "",

std::string network = "",

std::string isp = "");

**参数说明**

|  |  |
| --- | --- |
| acct\_id | 用户账户(米米号) |
| player\_id | 用户角色标识(对于单角色游戏以及不需要分角色查看数据的，该字段可赋空值),通常用用户创建角色的时间标识 |
| race | 用户选择的职业(对于不区分职业或不需要按职业进行统计，该字段可赋空值) |
| cli\_ip | 用户的IP地址, 无法获取时取值0 |
| ads\_id | 用户是从哪个广告渠道跳转过来的 |
| browse | 用户使用的浏览器 |
| device | 用户使用的设备 |
| os | 用户使用的操作系统 |
| resolution | 用户屏幕的分辨率 |
| network | 用户使用的网络 |
| isp | 用户网络的服务提供商 |

**用途**

1 、统计项名称(stid)：\_newac\_ 子统计项名称(sstid): \_newac\_

统计每天新注册的用户账户数以及这些账户的地区分布、使用的浏览器、设备、操作系统、分辨率、网络、网络服务提供商的人数人次

2、统计项名称(stid)：\_newpl\_ 子统计项名称(sstid): \_newpl\_

统计每天新注册的角色数，即根据acct\_id和player\_id来唯一标识一个用户

3、统计项名称(stid)：\_newrace\_ 子统计项名称(sstid): 具体的职业名称

统计每天各个职业的新增用户数,根据acct\_id来标识

**用法**

logger.reg\_role(“47159775”, “1383019208”, “魔法师”,” 61.155.182.56”, “innermedia.taomee.mole.banner”);//这里不按照浏览器、设备等指标统计；如果项目部需要,可自行带上相应参数即可

## 4、登录游戏服务器(online服务器)

**接口定义**

void login\_online(std::string acct\_id,

std::string player\_id,

std::string race,

bool isvip,

int lv,

uint32\_t cli\_ip,

std::string ads\_id,

std::string zone = “”,

std::string browser = "",

std::string device = "",

std::string os = “”,

std::string resolution = “”,

std::string network = “”,

std::string isp = “”);

**参数说明**

|  |  |
| --- | --- |
| acct\_id | 用户账户(米米号) |
| player\_id | 用户角色标识(对于单角色游戏以及不需要分角色查看数据的，该字段可赋空值),通常用用户创建角色的时间标识 |
| race | 用户选择的职业(对于不区分职业或不需要按职业进行统计，该字段可赋空值) |
| isvip | 是否VIP用户（true:是 false：否） |
| lv | 用户当前等级 |
| cli\_ip | 用户的IP地址, 无法获取时取值0 |
| ads\_id | 用户是从哪个广告渠道跳转过来的 |
| zone | 默认表示总体登录；主要是提供给页游这边不区分区服的游戏使用，用于使用电信或网通登录人数用的 |
| browse | 用户使用的浏览器 |
| device | 用户使用的设备 |
| os | 用户使用的操作系统 |
| resolution | 用户屏幕的分辨率 |
| network | 用户使用的网络 |
| isp | 用户网络的服务提供商 |

**用途**

1、统计项名称(stid): \_lgac\_ 子统计项名称(sstid): \_lgac\_

统计帐号登录人数人次，也就是活跃用户以及活跃用户的地区分布

VIP和非VIP用户的登录人数人次

每个等级的用户登录人数人次

使用不同的浏览器登录人数人次，不同设备登录人数人次 不同操作系统、分辨率、网络以及网路服务提供商登录人数人次

2、统计项名称(stid): \_lgpl\_ 子统计项名称(sstid)：\_lgpl\_

根据角色统计登录人数人次，已acct\_id和player\_id唯一标识用户

3、统计项名称(stid): \_lgrace\_ 子统计项名称(sstid): 各职业名称

统计各个职业每天的登录人数人次

**用法**

logger.login\_online(“47159775”, “1383019208”, “魔法师”, true, 15, “115.12.116.57”, “”, “ie8”, “iphone4”, “debian”,”1280\*768”, “wifi”, “网通”);

logger.login\_online(“47159775”, “1383019208”, “”, true, 15, “115.12.116.57”, “网通一区”, “ie8”, “iphone4”, “debian”,”1280\*768”, “wifi”, “网通”);

## 5、登录设备数(b03使用)

**接口定义**

void start\_device(std::string device\_id);

**参数说明**

|  |  |
| --- | --- |
| Device\_id | 设备ID |

**用途**

统计项名称(stid): \_startdev\_ 子统计项(sstid): \_startdev\_

可统计每日登录设备数，经过加工可以统计出每日新增设备数

**用法:**

logger.start\_device(“asdfdg23432q52345234534”);

## 6、用户退出游戏服务器(online服务器)

**接口定义**

void logout(std::string acct\_id, bool isvip, int lv, int oltime);

**参数说明**

|  |  |
| --- | --- |
| acct\_id | 用户账户(米米号) |
| isvip | 是否VIP用户(true:是 false:否) |
| lv | 用户退出时的等级 |
| oltime | 本次用户总共的在线时长 |

**用途**

1. 统计项名称(stid): \_logout\_ 子统计项名称(sstid): \_logout\_

可统计每天登出游戏的人数人次、总的在线时长(单位秒)以及时长区间分布,默认的时长区间划分为：

0-10

11-60

61-300

301-600

601-1200

1201-1800

1801-2400

2401-3000

3001-3600

3601-7200

7201以上

**用法**

logger.logout(“47159775”, true, 13,3204);

## 7、统计当前在线人数

**接口定义**

void online\_count(int cnt, std::string zone=””);

**参数说明**

|  |  |
| --- | --- |
| cnt | 当前在线人数 |
| zone | 默认表示总体在线人数；该字段主要是提供给页游不区分区服类的游戏使用的，用于统计电信或网通的在线 |

**用途**

统计项名称(stid): \_olcnt\_ 子统计项名称(sstid): \_olcnt\_

统计游戏当前在线人数

**用法**

logger.online\_count(103476);//每分钟至少调用一次

## 8、用户升级

**接口定义**

void level\_up(std::string acct\_id, std::string race, int lv);

**参数说明**

|  |  |
| --- | --- |
| acct\_id | 用户帐号(米米号) |
| race | 用户职业(如果不需要区分职业，赋值为空即可””) |
| lv | 升之后的等级 |

**用途**

1、统计项名称(stid) : \_aclvup\_ 子统计项名称(sstid): \_aclvup\_

统计每天的用户等级分布

1. 统计项名称(stid) : \_racelvup\_ 子统计项名称(sstid): \_racelvup\_

分职业统计用户的等级分布

**用法**

每次用户升级时

logger.level\_up(“47159775”, “”, 20);

## 9、付费

接口定义

void pay(std::string acct\_id,

bool isvip,

float pay\_amount,

CurrencyType currency,

PayReason pay\_reason,

std::string outcome,

int outcnt，

std::string pay\_channel = "米币帐户");

参数说明

|  |  |
| --- | --- |
| acct\_id | 用户账户(米米号) |
| isvip | 是否VIP用户(true:是 false:否) |
| pay\_amount | 付费额度,**单位统一为”分”** |
| currency | 货币类型(ccy\_mibi:米币 ccy\_cny:人民币),如是其他货币类型，请做转换，一个游戏中不可交替使用米币和人民币 |
| pay\_reason | 支付原因(pay\_vip:开通VIP包月 pay\_buy:购买道具 pay\_charge:充值游戏币 pay\_free:赠送) |
| outcome | 产出的物品(pay\_vip:可以是包月的类型 pay\_buy:可以是道具的名称或ID pay\_charge: 填写游戏币名称即可) |
| outcnt | 购买道具或游戏币的数量(pay\_vip:填写包月时长) |
| pay\_channel | 支付渠道(“支付宝”、”财付通”等) |

用途

1、统计项名称(stid) : \_acpay\_ 子统计项名称(sstid): \_acpay\_

可统计付费总额、VIP和非VIP用户分别的付费总额、各个支付渠道付费总额、每种货币单位的付费总额

2、统计项名称(stid) : \_acpay\_ 子统计项名称(sstid): \_buycoins\_ vipmonth buyitem

可统计三个子统计项的付费总额、VIP和非VIP用户分别的付费总额、各个支付渠道付费总额、每种货币单位的付费总额

1. 统计项名称(stid): \_getgolds\_ 子统计项名称(sstid): \_userbuy\_

用户购买产出的金币总量

4、统计项名称(stid): \_buyvip\_ 子统计项名称(sstid): \_buyvip\_

各个包月时长的人数人次以及付费总额

5、统计项名称(stid): \_buyitem\_ 子统计项名称(sstid): \_mibiitem\_

可统计通过米币购买的道具人数人次、销售数量、销售金额

用法

B01:

logger.pay(“47159775”, true, 1000,ccy\_mibi,pay\_vip,”1个月VIP”, 1, “短信”);//包月

logger.pay(“47159775”, true, 2000, ccy\_mibi, pay\_buy, “圣诞套装”, 20);//米币购买道具 这里不需要在调用一次购买道具的接口了

B03:

Logger.pay(“47159775”, true, 300, ccy\_cny, pay\_charge,”游戏金币”,30);//购买游戏金币;这里不需要在调用一次获得游戏金币的接口了

## 10、获得游戏内一级币(这里的获得是指通过游戏内部赠送或完成任务奖励，而非用户购买)

**接口定义**

void obtain\_golds(std::string acct\_id, int amt);

**参数说明**

|  |  |
| --- | --- |
| acct\_id | 用户账户(米米号) |
| amt | 获得金币的数量,**单位统一为”分”**(1~1千万) |

**用途**

1. 统计项名称(stid): \_getgolds\_ 子统计项名称(sstid): \_systemsend\_

统计通过各种系统赠送途径获得金币的人数人次以及总数量

**用法**

比如在IMOLE游戏里面通过完成某个任务奖励了**1000**个贝壳

logger.obtain\_golds(“47159876”, **100000**); //接口中的数量需要乘以100

## 11、使用游戏内一级货币购买道具

**接口定义**

void buy\_item(std::string acct\_id, bool isvip, int lv, float pay\_amount, std::string outcome, int outcnt);

**参数说明**

|  |  |
| --- | --- |
| acct\_id | 用户账户(米米号) |
| is\_vip | 是否VIP用户(true:是 false：否) |
| lv | 用户购买道具的时的等级 |
| pay\_amount | 支付数量,**单位统一为”分”** |
| outcome | 购买的道具 |
| outcnt | 购买的道具数量 |

**用途**

1. 统计项名称(stid): \_buyitem\_

子统计项名称(sstid)：\_coinsbuyitem\_

可统计通过非米币渠道购买的道具人数人次、销售数量、销售金额

**用法**

Logger.buy\_item(“34159876”, true, 13, **2000**, “元旦礼包”, 10);//用户花了20游戏币购买10个元旦礼包

通常这里只有B03手游这边会调用，B01的米币购买道具在发送pay信息的时候，已经包含该信息了。

## 12、消耗游戏内一级货币

**接口定义**

void use\_golds(std::string acct\_id, bool isvip, std::string reason, int amt, int lv);

**参数说明**

|  |  |
| --- | --- |
| acct\_id | 用户帐号(米米号) |
| isvip | 是否VIP用户(true:是 false:否) |
| reason | 原因（购买道具、开启新功能等） |
| amt | 支付数量,**单位统一为”分”** |
| lv | 用户等级 |

**用途**

1. 统计项名称(stid): \_userglods\_ 子统计项(sstid): reason

可统计各种原因消耗金币的数量，VIP和非VIP用户消耗的金币数量以及各等级用户消耗的数量

**用法**

logger.user\_golds(“47169879”, true, “\_开启新功能\_”, **1800**, 17);//用户花18游戏币开启新功能

//用户购买道具时不需要调用该接口哦

## 13、接收任务

**接口定义**

void accept\_task(TaskType type, std::string acct\_id, std::string task\_name, int lv);

**参数说明**

|  |  |
| --- | --- |
| type | 任务类型(新手任务：task\_newbie 主线任务：task\_story 支线任务:task\_supplement ) |
| acct\_id | 用户(米米号) |
| task\_name | 任务名称 |
| lv | 接收任务时的等级 |

**用途**

1. 统计项名称(stid): \_getnbtsk\_ \_getmaintsk\_ \_getauxtsk\_ \_getetctsk\_

子统计项名称(sstid): 任务名称

可统计各个任务的接收人数人次

**用法**

Logger.accept\_task(task\_newbie, “3781654”, “打开背包”,20);

## 14、完成任务

**接口定义**

void finish\_task(TaskType type, std::string acct\_id, std::string task\_name, int lv);

**参数说明**

|  |  |
| --- | --- |
| type | 任务类型(新手任务：task\_newbie 主线任务：task\_story 支线任务:task\_supplement ) |
| acct\_id | 用户(米米号) |
| task\_name | 任务名称 |
| lv | 完成任务时的用户等级 |

**用途**

1. 统计项名称(stid): \_donenbtsk\_ \_donemaintsk\_ \_doneauxtsk\_ \_doneetctsk\_

子统计项名称(sstid): 任务名称

可统计各个任务的完成人数人次以及完成任务时的用户等级分布

**用法**

logger.accept\_task(task\_newbie, “3781654”, “打开背包”, 18);

## 15、放弃任务

**接口说明**

void abort\_task(TaskType type, std::string acct\_id, std::string task\_name, int lv);

参数说明

|  |  |
| --- | --- |
| type | 任务类型(新手任务：task\_newbie 主线任务：task\_story 支线任务:task\_supplement ) |
| acct\_id | 用户(米米号) |
| task\_name | 任务名称 |
| lv | 放弃任务时的等级 |

**用途**

1. 统计项名称(stid): \_abrtnbtsk\_ \_abrtmaintsk\_ \_abrtauxtsk\_ \_abrtetctsk\_

子统计项名称(sstid): 任务名称

可统计各个任务的放弃人数人次,以及放弃任务时的用户等级分布

**用法**

logger.accept\_task(task\_newbie, “3781654”, “打开背包”, 19);

## 16、退订VIP服务(取消自动续费)

**接口说明**

void unsubscribe(std::string acct\_id, UnsubscribeChannel uc);

**参数说明**

|  |  |
| --- | --- |
| acct\_id | 用户(米米号) |
| uc | 渠道，目前只有短信和米币两个渠道(StatLogger::uc\_duanxin和StatLogger::uc\_mibi) |

**用途**

统计项名称(stid) : \_unsub\_

子统计项名称(sstid): \_unsub\_

对UC字段做item运算

**用法**

Logger.unsubscribe(“342352345”, StatLogger::uc\_duanxin);

Logger.unsubscribe(“342352345”, StatLogger::uc\_mibi);

## 17、销户(VIP时间到期)

**接口说明**

void cancel\_acct(std::string acct\_id, string channel);

参数说明

|  |  |
| --- | --- |
| acct\_id | 用户(米米号) |
| channel | 销户渠道，如支付宝、财付通等 |

**用途**

统计项名称(stid) : \_ccacct\_

子统计项名称(sstid): \_ccacct\_

对channel字段做item运算

**用法**

Logger.cancel\_acct(“342352345”, “zhifubao”);

## 18、自定义统计

### 18.1、概述

**接口定义**

void log(std::string stat\_name, std::string sub\_stat\_name, std::string acct\_id,

std::string player\_id, const StatInfo& info = StatInfo());

**参数说明**

|  |  |
| --- | --- |
| stat\_name | 统计项名称 |
| sub\_stat\_name | 子统计项名称 |
| acct\_id | 用户账户(米米号) |
| palyer\_id | 用户角色标识(如果不需要分角色查看，只能传空字符串即可) |
| info | 附加信息 |

**用途**

默认用于统计子统计项的人数人次

**用法**

StatLogger logger(1);

StatInfo info;

Info.add\_info(“赛尔豆”, 843);

Info.add\_op(StatInfo:op\_sum, “赛尔豆”);

Logger.log(“游戏输出”, “保护导航仪”, “47189678”, “”, info);

//可统计出参与保护导航仪这个小游戏的人数人次以及输出的赛尔豆总量

其中add\_info和add\_op接口声明如下

void add\_info(std::string key, float value);

void add\_info(std::string key, std::string value);

void add\_op(OpCode op, std::string key1, std::string key2 = "");

上述接口中的key由各项目部自定义，注意需要保证同一个项目内部不冲突，一次调用最多允许30个key-value对;

对key的取值不能以 ”\_” 开头或结束，StatInfo会自动过滤，比如\_KEY\_，会自动被修改成KEY; 不能够有“= : , ; . | \t”字符中的任何一个，否则程序运行会跳过该统计项 ;

对value的取值不能有“= | \t”字符中的任何一个

Key1 key2必须是已经通过add\_info方法添加好了的

OpCode的取值以及说明如下:

|  |  |
| --- | --- |
| StatInfo::op\_sum(key) | 对指定key的value求和,组合键statname substatname key |
| StatInfo::op\_max(key) | 对指定key的value求最大值; 组合键statname substatname key |
| StatInfo::op\_set(key) | 对指定key的value做set做操作; 组合键statname substatname key |
| StatInfo::op\_ucount(key) | 对指定key的value做去重操作; 组合键statname substatname key |
| StatInfo::op\_item(key) | 对指定key的value计算出对应的人数人次; 组合键statname substatname acctid playerid key |
| StatInfo::op\_item\_sum(key1,key2) | 按照key1分类，对key2的value字段求和; 组合键statname substatname acctid playerid key1 key2 |
| StatInfo::op\_item\_max(key1, key2) | 按照key1分类，对key2的value字段求最大值; 组合键statname substatname acctid playerid key1 key2 |
| StatInfo::op\_item\_set(key1, key2) | 按照key1分类，对key2的value字段做set操作，即取每天最后一个值; 组合键statname substatname acctid playerid key1 key |
| StatInfo::op\_sum\_distr(key) | 需要依赖于区间分布，然后计算出对key的value字段求和后属于哪个区间的分布; 组合键statname substatname acctid playerid key |
| StatInfo::op\_max\_distr(key) | 需要依赖于区间分布，然后计算出对key的value字段求最大值后属于哪个区间的分布; ; 组合键statname substatname acctid playerid key |
| StatInfo::op\_min\_distr(key) | 需要依赖于区间分布，然后计算出对key的value字段求最小值后属于哪个区间的分布; ; 组合键statname substatname acctid playerid key |
| StatInfo::op\_set\_distr(key) | 对key的 value字段做set操作后做分布; 组合键statname substatname acctid playerid key1 |
| StatInfo::op\_ip\_distr(key) | ip分布，需要根据IP地址查出对应的地区 |
|  |  |

### 18.2、示例

1. 默认类型(人数人次)

比如赛尔号->用户->玩家掉线统计 人数人次



赛尔号游戏ID为2

StatLogger logger(2);

logger.log(“用户”, “玩家掉线统计”, “48197896”, “”);

1. StatInfo::op\_sum(key) StatInfo::op\_max(key) StatInfo::op\_set(key)

比如赛尔号->游戏输出->撞球对抗赛(人数、人次、产生的赛尔豆)

StatLogger logger(2);

StatInfo info;

info.add\_info(“赛尔豆”, 100);

info.add\_op(StatInfo::op\_sum, “赛尔豆”);

logger.log(“游戏输出”, “保护导航仪”, “37896574”, “”, info);  
op\_max:是对发过来的同一天或同一分钟的数据求最大值

op\_set:是对发过来的同一天或同一分钟的数据做set操作,即覆盖操作

1. StatInfo::op\_item(key)

StatInfo::op\_item\_sum(key1,key2)

StatInfo::op\_item\_max(key1, key2)

StatInfo::op\_item\_set(key1, key2)

赛尔号->触达率->太空站

 也可以如下方式：

StatLogger logger(2);

StatInfo info;

Info.add\_info(“item”, “进入英佩恩堡垒”);

Info.add\_info(“item”, “进入英佩恩堡垒|场景”);

Info.add\_info(“item”, “进入英佩恩堡垒二层|场景”);

Info.add\_op(StatInfo::op\_item, “item”);

Logger.log(触达率“, “太空站”, “324234534”, “”, info);

1. StatInfo::op\_sum\_distr(key) StatInfo::op\_max\_distr(key) Statino::op\_min\_distr(key)

这一项也是用于统计区间分布的，只是这里的区间需要策划或相关开发人员去配置，比如统计今日产出的金豆的区间分布，例如

产出金豆数为1-100的人数

产出金豆数为101-500的人数

产出金豆数为501-1000的人数

产出金豆数大于1000的人数

那么就需要在用户产出金豆的时候，按照如下格式发送日志：

StatLogger logger(1);

StatInfo info;

Info.add\_info(“产出金豆”, 56);

Info.add\_op(StatInfo::op\_sum\_distr, “产出金豆”);

logger.log(“小游戏产出”， “产出金豆”, “34478323”, “”, info);

就会将以用户为单位，计算每个人今日产出的金豆总量，然后看属于哪个区间，做分布

Op\_max\_distr和op\_min\_distr 类似,只是分别去每个用户的最大值或最小值而已.

1. StatInfo::op\_set\_distr(key)

这一项用于做等级分布

比如iseer用户最高精灵等级分布

StatLogger logger(78);

StatInfo info;

Info.add(“max\_level”, 19);

Info.add\_op(StatInfo::op\_set\_distr, “max\_level”);

Logger.log(“等级分布”, “用户最高精灵等级分布”, “2341343”, “”, info);

1. StatInfo::op\_ip\_distr(key)

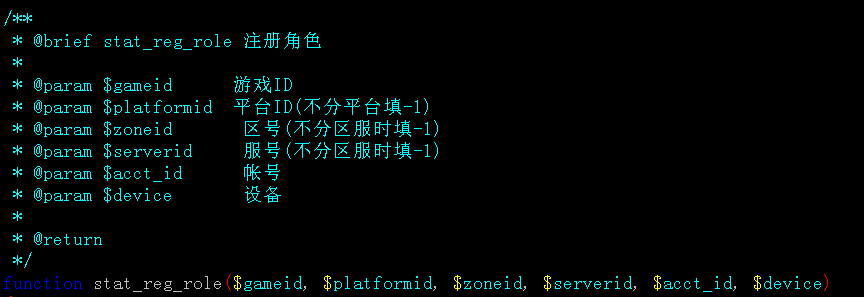
地区分布统计各个地区的人数人次，精确到省份

# 四、添加调用方法(PHP)

Php这里只提供了注册角色、登录游戏 、付费和自定义的接口

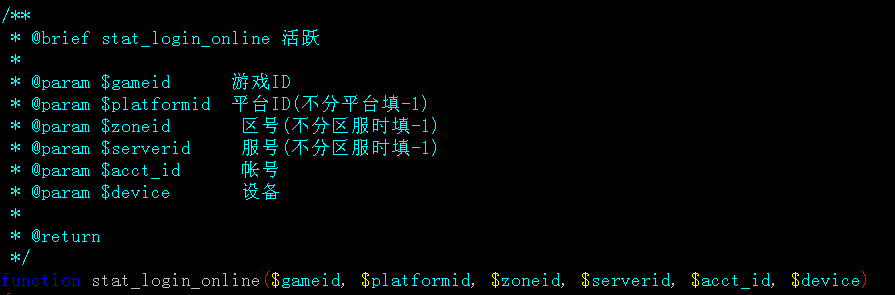
### 4.1、注册角色

接口定义:



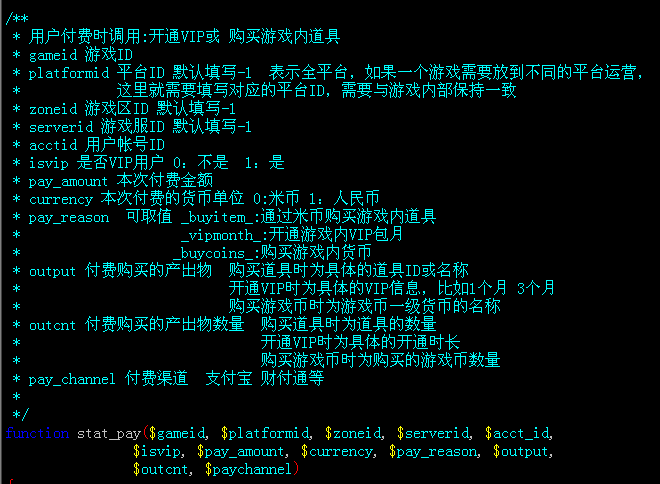
### 4.2 用户登录游戏(做活跃统计)

接口定义:

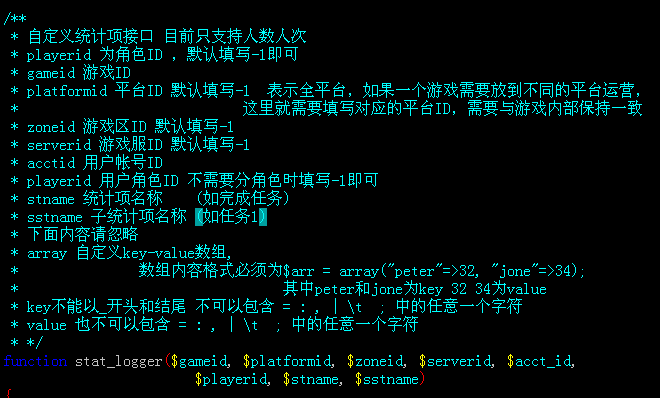


### 4.3、付费(开通VIP、米币购买道具、人民币购买游戏内一级货币)

接口定义：(**注意付费金额的单位统一为 “分”)**



### 4.4、自定义统计项接口(只支持统计人数人次)

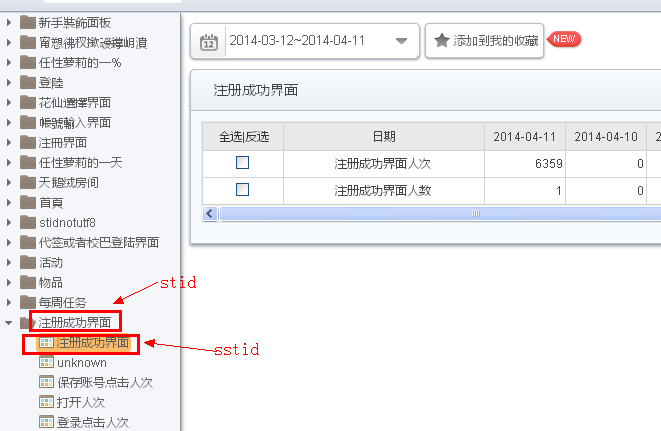


# 五、web端统计或AS统计

通过发送http请求的形式，只支持人数人次的统计，提供二级和三级两种统计方式

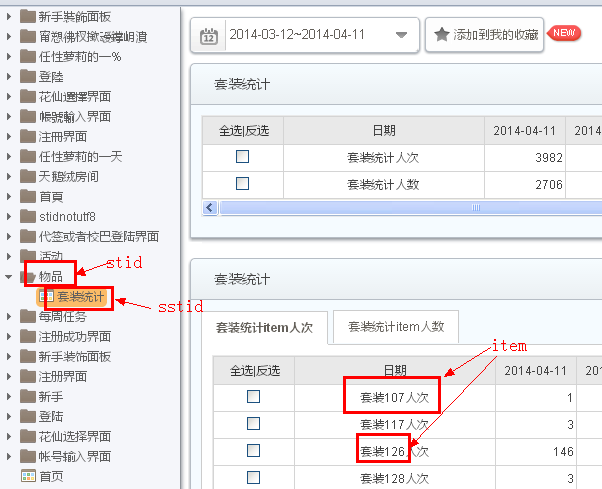
http请求地址：

二级统计项：<http://misc.taomee.com/weblogger.php?gameid=$gameid&stid=$stid&sstid=$sstid&uid=$uid>



三级统计项：

<http://misc.taomee.com/weblogger.php?gameid=$gameid&stid=$stid&sstid=$sstid&uid=$uid&item=$item>



gameid:游戏ID(**必须**为数字)

stid: 统计项名称(为避免乱码问题，这里**必须**填写英文或数字字符串)

sstid: 子统计项名称(为避免乱码问题，这里**必须**填写英文或数字字符串)

uid: 用户米米号(获取不到米米号时可不填写,如果填写**必须**为英文或数字字符串)

item：第3级统计项名称