淘米数据平台接入指南-2.0.0(手游)

C01统计平台项目组

2014年9月

目录

[淘米数据平台接入指南-2.0.0(手游) 1](#_Toc397430263)

[C01统计平台项目组 1](#_Toc397430264)

[2014年7月 1](#_Toc397430265)

[一、综述 3](#_Toc397430266)

[1、适用范围 3](#_Toc397430267)

[2、统计标准 4](#_Toc397430268)

[3、问题联系人 4](#_Toc397430269)

[二、接入流程 4](#_Toc397430270)

[1、为游戏申请GameID 4](#_Toc397430271)

[2、向工程中添加logger库文件 4](#_Toc397430272)

[3、添加调用方法 4](#_Toc397430273)

[4、进行数据测试 5](#_Toc397430274)

[三、添加调用方法(C++) 5](#_Toc397430275)

[阅读指引（适用于接口1~15） 5](#_Toc397430276)

[0、构造函数 6](#_Toc397430277)

[1、登录游戏验证用户名和密码 6](#_Toc397430278)

[2、创建游戏中角色 8](#_Toc397430279)

[3、登录游戏服务器(online服务器) 10](#_Toc397430280)

[4、退出游戏服务器(online服务器) 12](#_Toc397430281)

[5、统计当前在线人数 13](#_Toc397430282)

[6、用户升级 14](#_Toc397430283)

[7、付费 14](#_Toc397430284)

[8、免费获得游戏币 16](#_Toc397430285)

[9、使用游戏币购买道具 17](#_Toc397430286)

[10、消耗游戏币（除购买游戏币道具） 17](#_Toc397430287)

[11、接收任务 18](#_Toc397430288)

[12、完成任务 19](#_Toc397430289)

[13、放弃任务 20](#_Toc397430290)

[14、获得精灵 21](#_Toc397430291)

[15、失去精灵 21](#_Toc397430292)

[16、新用户注册转化 22](#_Toc397430293)

[17、自定义统计 23](#_Toc397430294)

[17.1、概述 23](#_Toc397430295)

[特别注意：stat\_name和sub\_stat\_name均不能含%(中英文输入的百分号)、/（右斜划线）、?（英文输入的问号）、;（英文输入的分号）、-1（负1），否则会当成乱码屏蔽掉。 26](#_Toc397430296)

[17.2、示例 26](#_Toc397430297)

[四、web端统计或AS统计 30](#_Toc397430298)

**修订历史记录**

|  |  |  |
| --- | --- | --- |
| **日期** | **作者** | **修订内容** |
| 2013.11 | Henry | 创建文档 |
| 2014.4.10 | Lynn | “五、web端统计或AS统计“增加三级统计项调用方法 |
| 2014.4.15 | Ping | “三、添加调用方法”，增加9、获得精灵；10、失去精灵 |
| 2014.04.23 | lynn | “三、添加调用方法”，修改1~4接口中的os字段说明，页游、手游传不同值 |
| 2014.05.07 | Lynn | 1、在“三、添加调用方法”中增加接口调用责任方说明；  2、在“二、接入流程”中完善数据测试方法”； |
| 2014.05.08 | Lynn | 增加接口调用的**阅读指引**，增加每个接口的**用途说明** |
| 2014.06.05 | Berry | 制定手游页游两套接入标准，去除不用的接口，更改传输数量标准 |
| 2014.07.02 | Ping | 16、新用户注册转化 |
| 2014.09.02 | Ping | 删除php接口，修改as请求连接，增加长度字段说明 |

# 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

用户数：以账号（米米号）计数。

付费: 指购买游戏币或开通VIP包月服务。

## 3、问题联系人

数据分析平台的任何问题，请联系berry

# 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

后台调用请参考“三、添加调用方法（C++）”；

PHP调用请参考“四、添加调用方法（PHP）”；

WEB或AS请参考“五、web端统计或AS统计”

## 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。

**1、内网测试方法（推荐）：**

在内网测试机上，建立如下目录：

/opt/taomee/stat/data/inbox 及 /opt/taomee/stat/data/log

落的数据将会在/opt/taomee/stat/data/inbox里面，是文本，可直接打开查看。

2、外网测试方法

向开发方申请一个测试的gameid，一般为本项目gameid+100000，调用方法并登录系统查看。

# 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。

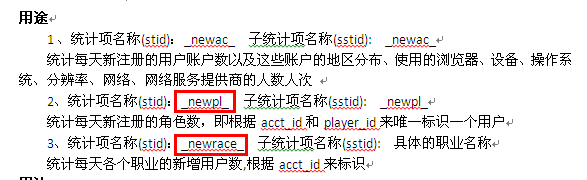
StatLogger logger(1);

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。

## 阅读指引（适用于接口1~15）

1、接口定义中“**用途”**是指调用此接口会生成的日志，其中stid和sstid都是由sdk自动生成，可以在测试时根据日志确认是否正确调用，其它参数值是否正确。

2、“**用途”**含有多条说明的，是指**如果参数全部填写完整**，将会在一次接口调用中，生成多条日志，例如第3个接口，若传的player\_id, race**不为空**，sdk也会生成第2条，第3条日志。

。

## 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、登录游戏验证用户名和密码

**接口定义**

void verify\_passwd (int 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版本 |
| 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

## 2、创建游戏中角色

**接口定义**

void reg\_role(int 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，该字段可赋空值，多角色游戏，填写角色id) |
| race | 用户选择的职业(对无职业游戏，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 | 设备型号 |
| os | 用户使用的操作系统版本 |
| 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

## 3、登录游戏服务器(online服务器)

**接口定义**

void login\_online(int 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，该字段可赋空值，多角色游戏，填写角色id) |
| race | 用户选择的职业(对无职业游戏，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 | 用户登陆区服（不区分区服的，填空值） |
| browse | 手游填空 |
| device | 设备型号； |
| os | 用户使用的操作系统版本， |
| 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”);

## 4、退出游戏服务器(online服务器)

**接口定义**

void logout(int 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);

## 5、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 | 1、默认值为空是统计总在线人数；  2填写‘\*服’，则分别统计各服的在线人数。 |

**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；

**手游需要调用总在线人数，也需要调用各服的在线人数。**

## 6、用户升级

**接口定义**

void level\_up(int 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) |
| 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);

## 7、付费

**接口定义**

void pay(int 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 |
| currency | 货币类型(ccy\_mibi:米币 ccy\_cny:人民币) |
| pay\_reason | 支付原因  pay\_charge: 手游，用米币（或人民币）兑换游戏币 |
| outcome | 产出的物品  pay\_charge: 手游，游戏币包对应的ID |
| outcnt | 获得数量  pay\_charge: 手游，游戏币包的数量 |
| pay\_channel | 支付渠道号，如，支付宝、财付通等渠道号 |

**用途**

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

统计付费总额

2、统计项名称(stid) : \_acpay\_ 子统计项名称(sstid): \_vipmonth\_，\_buyitem\_

统计按条或包月付费总额

3、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

4、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：**

手游：Logger.pay(“47159775”, true, 300, ccy\_cny, pay\_charge, ”游戏金币”,30);

## 8、免费获得游戏币

(指通过游戏内赠送或完成任务奖励，而非用户购买)

**接口定义**

void obtain\_golds(int 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 (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**

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”, **1000**); //

## 9、使用游戏币购买道具

**接口定义**

void buy\_item(int 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 | 用户购买道具时的等级 |
| pay\_amount | 支付数量 |
| 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\_

可统计通过游戏币购买道具的人数人次、销售数量、销售金额

**用法**

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

## 10、消耗游戏币（除购买游戏币道具）

**接口定义**

void use\_golds(int 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（开启新功能、跳过关卡等） |
| amt | 支付数量 |
| lv | 用户等级 |

**用途**

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

可统计各原因消耗游戏币的数量、VIP和非VIP用户消耗的游戏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\_”, **18**, 17);//用户花18游戏币开启新功能

## 11、接收任务

**接口定义**

void accept\_task(TaskType type, int 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);

## 12、完成任务

**接口定义**

void finish\_task(TaskType type, int 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);

## 13、放弃任务

**接口说明**

void abort\_task(TaskType type, int 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);

## 14、获得精灵

**接口定义**

void obtain\_spirit(int acct\_id, bool isvip, int lv, int spirit);

**参数说明**

|  |  |
| --- | --- |
| acct\_id | 用户帐号(米米号) |
| isvip | 是否VIP用户(true:是 false:否) |
| lv | 获得精灵时的用户等级 |
| spirit | 获得的精灵id |

**用途**

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

**用法**

每次用户获得精灵时

logger. logger.obtain\_spirit(stat\_itstr("47159775"),false, 20, "小火猴");

## 15、失去精灵

**接口定义**

void lose\_spirit(int acct\_id, bool isvip, int lv, int spirit);

**参数说明**

|  |  |
| --- | --- |
| acct\_id | 用户帐号(米米号) |
| isvip | 是否VIP用户(true:是 false:否) |
| lv | 失去精灵时的用户等级 |
| spirit | 失去的精灵id |

**用途**

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

**用法**

每次用户失去精灵时

logger. logger.lose\_spirit(stat\_itstr("47159775"),false, 25, "小火猴");

## 16、新用户注册转化

**接口说明**

void new\_trans(NewTransStep step, std::string acct\_id);

参数说明

|  |  |
| --- | --- |
| acct\_id | 用户(米米号) |
| step | 新用户注册转化步骤 |

**用途**

统计项名称(stid) : \_newtrans\_

子统计项名称(sstid): 新用户注册转化步骤

对每个新用户注册转化步骤求人数人次

**用法**

logger. new\_trans (bGetLoginReq, “342352345”);

## 17、自定义统计

### 17.1、概述

**接口定义**

void log(std::string stat\_name, std::string sub\_stat\_name, int 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(如不需要分角色查看，传空字符串) |
| 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 |
|  |  |

### 特别注意：stat\_name和sub\_stat\_name均不能含%(中英文输入的百分号)、/（右斜划线）、?（英文输入的问号）、;（英文输入的分号）、-1（负1），否则会当成乱码屏蔽掉。

### 17.2、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

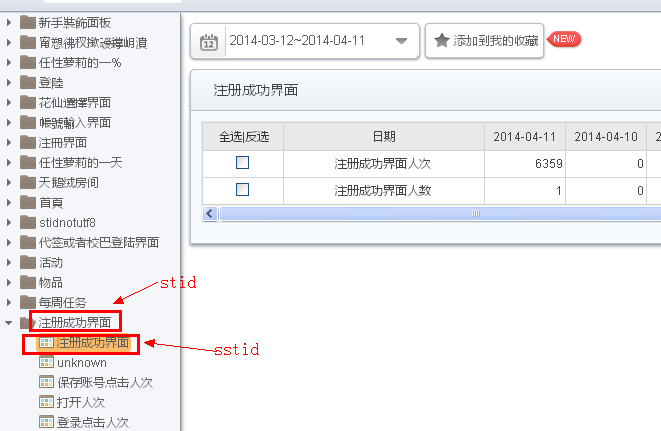
# 四、web端统计或AS统计

通过发送http请求的形式，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：

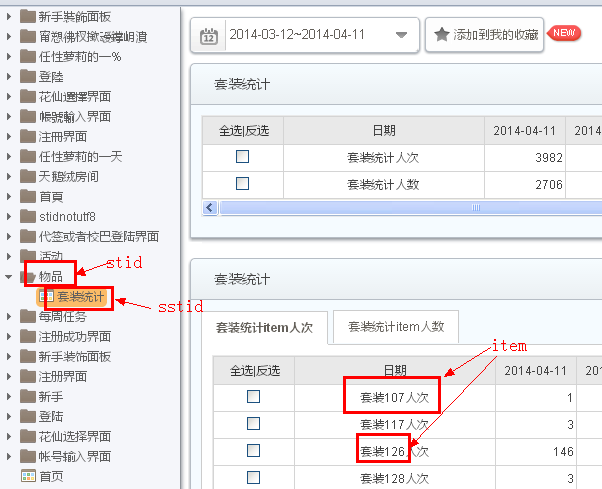
**二级统计项**：

<http://newmisc.taomee.com/misc.js?gameid=gameid&stid=stid&sstid=sstid&uid=uid>&stidlen=$stidlen&sstidlen=$sstidlen



**三级统计项**：

[http://newmisc.taomee.com/misc.js?gameid=gameid&stid=stid&sstid=sstid&uid=uid&item=item](http://misc.taomee.com/weblogger.php?gameid=gameid&stid=stid&sstid=sstid&uid=uid&item=item)&stidlen=$stidlen&sstidlen=$sstidlen&itemlen=$itemlen



gameid:游戏ID(**必须**为数字)

stid: 统计项名称(可填写中文、英文或数字字符串，但若不能避免乱码问题，可选择填写英文或数字字符串)

sstid: 子统计项名称(可填写中文、英文或数字字符串，但若不能避免乱码问题，可选择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（可填写中文、英文或数字字符串，但若不能避免乱码问题，可选择填写英文或数字字符串）

stidlen, sstidlen, itemlen: 对应字段的urlencode后的长度，如“淘米”,urlencode后为%e6%b7%98%e7%b1%b3，则长度为18（stelen(%e6%b7%98%e7%b1%b3)）