

# 淘米数据平台接入指南-3.0.1（页游）

淘米游戏-数据平台部

## 目录

淘米数据平台接入指南-3.0.1（页游） .....	1
淘米游戏-数据平台部 .....	1
一、综述 .....	4
1、适用范围 .....	4
2、统计标准 .....	4
3、问题联系人 .....	4
二、接入流程 .....	4
1、为游戏申请 GameID .....	4
2、向工程中添加 logger 库文件 .....	5
3、添加调用方法 .....	5
4、进行数据测试 .....	5
三、添加调用方法(C++) .....	6
调用前必读 .....	6
接口用途及调用方归属 .....	6
基础接口生成的日志通用字段说明 .....	7
0、构造函数（StatLogger） .....	8
1、登录游戏验证密码（verify_passwd） .....	8

2、创建游戏中角色 ( reg_role ) .....	10
3、登录游戏 online 服务器(login_online).....	12
4、退出游戏 online 服务器(logout) .....	15
5、统计当前在线人数(online_count) .....	16
6、用户升级(level_up) .....	17
7、付费 ( pay ) .....	18
8、免费获得游戏币(obtain_golds) .....	21
9、使用游戏币购买道具(buy_item) .....	22
10、消耗游戏币 ( use_golds ) .....	23
11、接收任务 ( accept_task ) .....	24
12、完成任务 ( finish_task ) .....	25
13、放弃任务 ( abort_task ) .....	26
14、获得精灵 ( obtain_spirit ) .....	27
15、失去精灵 ( lose_spirit ) .....	28
16、退订 VIP 服务 ( unsubscribe ) .....	28
17、销户 VIP ( cancel_acct).....	29
18、新用户注册转化 ( new_trans ) .....	30
19、自定义统计 ( log ) .....	31
19.1、概述.....	31
19.2、示例.....	35
20、根据统计项 ID 落自定义统计数据 ( custom_log ) .....	39
四、添加调用方法(PHP) .....	40

1、注册角色 .....	40
2 用户登录游戏(做活跃统计).....	40
3、付费(开通 VIP、米币购买道具、人民币购买游戏内一级货币 .....	41
4、自定义统计项接口(只支持统计人数人次).....	41
五、web 端统计或 AS 统计.....	41

## 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 <b>阅读指引</b> ，增加每个接口的 <b>用途说明</b>
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	17、新用户注册转化
2015.03.16	Kendy	调整文档，添加用户自定义统计新接口
2015.06.01	Lynn	游戏币相关接口兼容性更改说明，涉及 obtain_golds、buy_item、use_golds

## 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。

平台提供 2 种形式的调用：1) SDK 库的调用,落到本地服务器,目前提供 3 种语言的 SDK,分别是 C++程序、PHP 程序、C 程序。2) 发送 http 请求,直接发送相关统计。

### 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@B01-数据平台部

## 二、接入流程

温馨提示：您也可以阅读《【淘米游戏】数据分析平台接入流程—必读!.xlsx》来获知更详尽的接入流程。

### 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 SDK 的 logger 压缩包并解压至本地，将 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 C++：请参考【三、添加调用方法 ( C++ )】
- PHP：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** ):

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

向数据平台部申请一个测试的 gameid，一般为本项目 gameid+100000，生成测试数据并登录系统页面查看。

### 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。

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。

### 调用前必读

接口分为**基础接口**和**自定义接口**。**基础接口**用来落基础统计项,字段定义相对比较固定。

**自定义接口**用来落自定义事件等,相对更为灵活和自定义化。

基础接口中的 stid 和 sstid 都是由 SDK 自动生成。

**用途中的表格**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 login\_online,若传的 player\_id, race **不为空**,sdk 也会生成第 2 条,第 3 条日志。

#### 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

### 接口用途及调用方归属

统计数据由多方调用,形成完整的统计项体系。包括游戏方(项目前端和后台开发)、账户支付平台(@B01 运营开发部)等,请各方明确各自需调用的接口。

## 1、游戏方

游戏方是用户的所有行为数据的来源，涉及**活跃、等级、任务、付费**等，对于 VIP 系统在平台维护的游戏，除了第 1、16、17 个接口外，其余均需要调用。各接口的用途参见目录。

## 2、账户支付平台

**VIP 系统在平台维护的游戏，调用接口 7、16、17。**

7、付费(pay)接口落 vip 相关的付费信息

16、退订 VIP 服务 ( unsubscribe ) 落取消自动续费和短信退订的 VIP 信息

17、销户 VIP ( cancel\_acct ) 落因过期而被系统自动清理的 VIP 信息

## 基础接口生成的日志，其中的通用字段说明

字段	说明	必填?	默认值
hip	源数据发送 IP	No	接口自动落
gid	游戏 id	Yes	
zid	区 id，例如：1 区为 1，全区为-1	No ,由构造函数生成	-1
sid	服 id，例如：1 服为 1，全服为-1	No ,由构造函数生成	-1
pid	平台 id，例如：某游戏在多个平台运营时，需要区分平台，平台 id 由接入方制定	No ,由构造函数生成	-1

ts	落统计项时的时间戳	No	系统自动落
acid	帐号 id , 例如 : 米米号	Yes	
plid	角色 id , 可填 id 或字符串	No	-1

## 0、构造函数 ( StatLogger )

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、登录游戏验证密码 ( verify\_passwd )

### 用途

验证密码的人数人次以及这些用户的地区分布

各个广告位来源的用户验证密码的人数人次

使用不同的浏览器进行验证密码的人数人次

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

统计项 ( stid )	子统计项 ( sstid )
--------------	----------------



_veripass_	_veripass_
------------	------------

## 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	用户浏览器的 flash 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

## 示例

```
logger.verify_passwd( "47159775" , "106.235.12.11" , "innermedia.taomee.se
er.topbar" );//不填浏览器、设备等参数，则无法按此维度统计分析
```

### 生成的日志（待补充）

```
_hip_ =192.168.30.98      _stid_ = veripass_      _sstd_ = veripass_      _gid_ =16      _zid_ =-1      _sid_ =-1
_pid_ =-1      _ts_ =1427817601      _acid_ =671525710      _plid_ =-1      _ad_ =#http://news.4399.com/jlp/      _isp_ =网通      _cip_
=1211289199      _op_ =item: isp_ip_distr: cip_
```

## 2、创建游戏中角色（reg\_role）

### 用途

统计项（stid）	子统计项（sstd）	用途
_newac_	_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
_newpl_	_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
_newrace_	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

### 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, 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	页游, 填写空值
os	用户浏览器的 flash 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

## 示例

`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  
需要,可带上相应参数

生成的日志 ( 待补充 )

## 3、登录游戏 online 服务器(login\_online)

### 用途

统计项 ( stid )	子统计项 ( sstid )	用途
_lgac_	_lgac_	1、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  2、VIP 和非 VIP 用户的登录 人数人次  3、每个等级的用户登录人 数人次  4、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

_lgpl_	_lgpl_	根据角色统计登录人数人次， 以 acct_id 和 player_id 唯一 标识用户
_lgrace_	各职业名称	统计各个职业每天的登录人 数人次

### 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,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	页游填用户使用的浏览器类型 ( 如 firefox, IE )
device	页游填写空值
os	用户浏览器的 flash 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

### 示例

```
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" );
```

**生成的日志（待补充）**

## 4、退出游戏 online 服务器(loginout)

**用途**

统计项 ( std )	子统计项 ( sstid )	用途
_logout_	_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
```

**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

#### 示例

```
logger.logout( "47159775" , true, 13,3204);
```

生成的日志 ( 待补充 )

## 5、统计当前在线人数(online\_count)

#### 用途

统计项 ( std )	子统计项 ( sstid )	用途
_olcnt_	_olcnt_	统计游戏当前在线人数 , 每分钟至少调用一次

#### 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 “电信” 或 “网通” , 则分别统计电信或网通的在线人数。

#### 示例



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

生成的日志（待补充）

## 6、用户升级(level\_up)

用途

统计项 ( stdid )	子统计项 ( sstid )	用途
_aclvup_	_aclvup_	统计每天的用户等级分布
_racelvup_	_racelvup_	统计各职业用户的等级分布

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)
lv	升之后的等级

示例

每次用户升级时

```
logger.level_up( "47159775" , "" , 20);
```

生成的日志（待补充）

## 7、付费 ( pay )

**此统计关系到付费数据的准确性，请重点关注。**

该接口统计的是指米币级别的收入。是指用人民币或米币兑换游戏币或购买游戏内 VIP 服务。

该接口涉及到多种类型的付费统计，需要**游戏后台**、**BOSS 平台**、**PAY 网站后台**同时落取。

### 一、兑换游戏币

- 1、没有从游戏跳转到 pay 页面的兑换（直接在游戏内部完成的兑换），由**游戏后台**接入统计。
- 2、从游戏跳转到 pay 页面的兑换，由 **PAY 网站后台**接入统计。

### 二、购买 VIP 服务

- 1、VIP 系统由平台维护的，由 BOSS 平台接入统计
- 2、VIP 系统不在平台维护的，如果是用人民币或米币购买 VIP，由游戏后台接入统计。（如果是用游戏币购买 VIP，则不需要接入）

**以下是产生的统计项日志：**

统计项 ( stid )	子统计项 ( sstid )	用途
_acpay_	_acpay_	统计米币付费总额
_acpay_	_buyitem_ , _vipmonth_	统计按条或包月付费总额
_buyvip_	_buyvip_	统计各个包月时长的人数人次以及付费总额
_buyitem_	_mibiitem_	统计通过米币购买的道具人数人次、销售数量、销售金额

**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” , 例如花 10 米币 , 数量落 1000						
currency	货币类型 , 枚举类型 页游落 “ccy_mibi” ( 米币 ) 手游落 ccy_cny ( 人民币 )						
pay_reason	<b>支付类型</b> <table border="1"> <thead> <tr> <th>类型</th><th>落的内容</th></tr> </thead> <tbody> <tr> <td>兑换游戏币</td><td>“pay_buy”</td></tr> <tr> <td>VIP 包月</td><td>“pay_vip”</td></tr> </tbody> </table>	类型	落的内容	兑换游戏币	“pay_buy”	VIP 包月	“pay_vip”
类型	落的内容						
兑换游戏币	“pay_buy”						
VIP 包月	“pay_vip”						
outcome	<b>支付产生的物品</b> <table border="1"> <thead> <tr> <th>类型</th><th>落的内容</th></tr> </thead> <tbody> </tbody> </table>	类型	落的内容				
类型	落的内容						

	兑换游戏币	商品 id , 由账户支付平台 boss 分配
	VIP 包月	"1 个月 VIP" 、 "3 个月 VIP" 等
outcnt	获得数量	
	类型	落的内容
	兑换游戏币	兑换率是 1:10 的填 <b>10*米币数</b> ; 兑换率是其它的 , 填钻石包的数量 , 如兑换 2 个 100 钻石礼包 , 填 <b>2</b> 。
	VIP 包月	1,3,6,12 等
pay_channel	支付渠道号 , 如支付宝、财付通等 , 由账户支付平台定的渠道号。  <b>页游的游戏后台请注意 : 游戏内兑换 , 渠道号填 1 , 指米币渠道。</b>	

示例 :

#### 1、页游 :

1 ) 游戏内兑换游戏币 ,由**游戏后台调用**( 花了 20 米币购买 2 个 320001 对应的游戏币包 )

`logger.pay( "47159775" , true, 2000, ccy_mibi, pay_buy, 320001, 2 , 1);`

米米号      VIP      花了20米币      货币类型是米币      商品id      2个包      兑换游戏币      米币渠道号: 1

2) VIP 包月，由账户支付平台调用（花 10 米币购买了 1 个月的 VIP 时长）

`logger.pay( "47159775" , true, 1000,ccy_mibi,pay_vip," 1 个月 VIP" , 1, "短信" );`

## 2、手游：

`logger.pay( "47159775" , true, 300, ccy_cny, pay_charge, " 游戏金币" ,30);`

生成的日志：

## 8、免费获得游戏币(obtain\_golds)

指通过游戏内赠送或完成任务奖励获得，而非通过玩家购买获得

### 用途

统计项 ( stdid )	子统计项 ( sstid )	用途
_getgolds_	_systemsend_	统计通过各种系统赠送途径 获得游戏币的人数人次以及 总数量

### 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
-----	---------

### 示例

在 IMOLE 游戏里面通过完成某个任务奖励了 **10** 个贝壳

```
logger.obtain_golds( "47159876" , 10);
```

生成的日志 ( 待补充 )

## 9、使用游戏币购买道具(buy\_item)

### 用途

统计项 ( stid )	子统计项 ( sstid )	用途
_buyitem_	_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

### 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	用户购买道具时的等级
pay_amount	支付的金币数量
outcome	购买的道具

outcnt	购买的道具数量
--------	---------

#### 示例

用户花了 **20** 游戏币购买 10 个元旦礼包

```
Logger.buy_item( "34159876" , true, 13, 20, "元旦礼包" , 10);
```

生成的日志（待补充）

## 10、消耗游戏币（use\_golds）

是指除了购买游戏币道具之外的消耗方式

#### 用途

统计项（stid）	子统计项（sstid）	用途
_userglods_	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

#### 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（开启新功能、跳过关卡等）
amt	支付的金币数量

lv	用户等级
----	------

### 示例

用户花 **18** 游戏币开启新功能

```
logger.user_golds( "47169879" , true, "_开启新功能_" , 18, 17);
```

生成的日志 ( 待补充 )

## 11、接收任务 ( accept\_task )

### 用途

统计项 ( stdid )	子统计项 ( sstid )	用途
_getnbtsk_ ( 新手任务 )	任务名称	统计相应类型任务的接收人数人次
_getmaintsk_ ( 主线任务 )		
_getauxtsk_ ( 支线任务 )		
_getetctsk_ ( 其它任务 )		

### 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
----	----------

### 示例

```
Logger.accept_task(task_newbie, "3781654", "打开背包",20);
```

生成的日志（待补充）

## 12、完成任务（finish\_task）

### 用途

统计项（std）	子统计项（sstid）	用途
_donenbtsk_（新手任务）	任务名称	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
_donemaintsk_（主线任务）		
_doneauxtsk_（支线任务）		
_doneetctsk_（其它任务）		

### 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

## 示例

```
logger.accept_task(task_newbie, "3781654", "打开背包", 18);
```

生成的日志（待补充）

## 13、放弃任务（abort\_task）

### 用途

统计项（stid）	子统计项（sstid）	用途
_abrtnbtsk_（新手任务） _abrtmaintsk_（主线任务） _abrtauxtsk_（支线任务） _abrtetctsk_（其它任务）	任务名称	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

### 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

## 示例

```
logger.accept_task(task_newbie, "3781654", "打开背包", 19);
```

生成的日志（待补充）

## 14、获得精灵（obtain\_spirit）

### 用途

统计项（stid）	子统计项（sstid）	用途
_obtainspirit_	_obtainspirit_	统计 <b>获得</b> 精灵的数量、VIP 或非 VIP 获得精灵的数量、各等级获得精灵的数量

### 接口定义

```
void obtain_spirit(std::string acct_id, bool isvip, int lv, std::string 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	获得的精灵名字

### 用法

每次用户获得精灵时

```
logger.logger.obtain_spirit(stat_itstr("47159775"),false, 20, "小火猴");
```

生成的日志（待补充）

## 15、失去精灵 ( lose\_spirit )

### 用途

统计项 ( stid )	子统计项 ( sstid )	用途
_losespirit_	_losespirit_	统计 <b>失去</b> 精灵的数量、VIP 或非 VIP 获得精灵的数量、各等级获得精灵的数量

### 接口定义

```
void lose_spirit(std::string acct_id, bool isvip, int lv, std::string 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	失去的精灵名字

### 示例

每次用户失去精灵时

```
logger. logger.lose_spirit(stat_itstr("47159775"),false, 25, "小火猴");
```

### 生成的日志 ( 待补充 )

## 16、退订 VIP 服务 ( unsubscribe )

取消米币自动续费和退订短信 VIP 服务

### 用途

统计项 ( stid )	子统计项 ( sstid )	用途
_unsub_	_unsub_	统计每天指定渠道退订 VIP 服务的人数人次 对 uc ( 即渠道 ) 做 item 运算

对米币渠道而言，是取消自动续费，对短信渠道而言，是退订 VIP 服务，下月不再自动续费。

### 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)

### 示例

```
Logger.unsubscribe( "342352345" , StatLogger::uc_duanxin);
```

```
Logger.unsubscribe( "342352345" , StatLogger::uc_mibi);
```

### 生成的日志 ( 待补充 )

## 17、销户 VIP ( cancel\_acct)

是指 VIP 时间到期后，系统定时销户 VIP，淘米内部由平台 BOSS 系统调用

### 用途

统计项 ( stdid )	子统计项 ( sstid )	用途
_ccacct_	_ccacct_	统计各渠道销户 VIP 的人数 人次 对 channel 字段做 item 运算

### 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。Channel 是字符串，由调用方和平台共同商定

### 示例

```
Logger.cancel_acct( "342352345" , "zhifubao" );
```

### 生成的日志 ( 待补充 )

## 18、新用户注册转化 ( new\_trans )

### 用途

某游戏从注册米米号到进入游戏的所有步骤落统计项，以第一步为基数，层层做过滤计算

统计项 ( stdid )	子统计项 ( sstid )	用途
_newtrans_	新用户注册转化步骤 ( 步骤名称见该专题文档 )	对新用户注册转化的每个步骤求人数人次

## 接口说明

```
void new_trans(NewTransStep step, std::string acct_id);
```

### 参数说明

step	新用户注册转化步骤
acct_id	用户(米米号)

### 示例

```
Logger.new_trans (bGetLoginReq, "342352345" );
```

### 生成的日志（待补充）

## 19、自定义统计（log）

### 19.1、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

需统计 :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 ,代码如下 :

```
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);
```

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 = "");
```

## OpCode 操作类型说明表

操作类型	说明
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：**

**( 1 ) stat\_name 和 sub\_stat\_name 均不能含以下字符 , 否则会被当成乱码屏蔽掉。**

字符	释义
%	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
	竖线

**( 2 ) stat\_name 和 sub\_stat\_name 统计项名称不宜过长 , 最大为 64 个字节 , 否则会被屏蔽掉。**

**( 3 ) 接口中的 key 由各项目部自行定义 , 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**

- (4) 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
- (5) 对 value 的取值不能有 “=|\t” 字符中的任何一个
- (6) key1 、key2 必须是已经通过 add\_info 方法添加好了的

## 19.2、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



The screenshot shows a file explorer on the left with a tree structure. Under '赛尔号' (Sai Er Hao), there is a '用户' (User) folder, which contains '玩家掉线统计' (Player Disconnection Statistics). This item is highlighted with a red box. To the right, a table titled '玩家掉线的人次' (Player Disconnection Count) displays data for two dates: 2014-03-18 and 2014-03-17. The table has two rows: '玩家掉线的人次' (Player Disconnection Count) and '玩家掉线的人数' (Player Disconnection Number). The values for 2014-03-18 are 1450 and 1120 respectively, while for 2014-03-17 they are 29181 and 24375. The first row is highlighted with a red box.

玩家掉线的人次		2014-03-18	2014-03-17	2014-03-16
<input type="checkbox"/>	玩家掉线的人次	1450	29181	2014-03-16
<input type="checkbox"/>	玩家掉线的人数	1120	24375	

赛尔号游戏 ID 为 2

```
StatLogger logger(2);
```

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

## 2、StatInfo::op\_sum(key) StatInfo::op\_max(key) StatInfo::op\_set(key)

比如赛尔号-> 游戏输出->保护导航仪(人数、人次、产生的赛尔豆)



保护导航仪		2014-03-18	2014-03-17	20
全选   反选				
<input type="checkbox"/>	保护导航仪赛尔豆输出	416	17736	
<input type="checkbox"/>	保护导航仪人次	8	274	
<input type="checkbox"/>	保护导航仪人数	4	231	

```
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

## 3、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

赛尔号

2014运营活动

基础指标

2013运营活动

触达率

商城触达率

飞船

太空站

游戏场景

新加统计

新手相关

回流用户数(每天)

进入英佩恩堡垒人次

全选   反选		2014-03-18	2014-03-17
<input type="checkbox"/>	进入英佩恩堡垒人次	158	4285
<input type="checkbox"/>	进入英佩恩堡垒人数	75	2524
<input type="checkbox"/>	进入英佩恩堡垒[场景]人次	158	4285
<input type="checkbox"/>	进入英佩恩堡垒[场景]人数	75	2524
<input type="checkbox"/>	进入英佩恩堡垒二层[场景]人次	0	0
<input type="checkbox"/>	进入英佩恩堡垒二层[场景]人数	0	0

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);
```

4、StatInfo::op\_sum\_distr(key)

StatInfo::op\_max\_distr(key)

StatInfo::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.

#### 5、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);
```

#### 6、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

## 20、根据统计项 ID 落自定义统计数据 ( custom\_log )

### 用途

统计项 ( stid )	子统计项 ( sstid )	用途
_msgid_	stat_id	根据统计项内容对应的统计项 ID 落数据，得到自定义统计项数据。

前提是统计项 id 对应的信息已经先行上传到平台的统计项信息库里，平台有专门提供统计项的 xml 表上传的页面，并有相应的校验机制，保证上传的统计项信息无误。

此法简化了落自定义统计的代码，将规则写在一条统计项信息记录里，通过统计项 id 与统计项信息映射的关系来实现统计项的落取。

统计项信息包含，统计项 id 对应的 stat\_name、sub\_stat\_name、item，操作类型等。

### 接口说明

```
void custom_log(std::string acct_id, uint32_t stat_id, float value=0);
```

### 参数说明

acct_id	用户(米米号)
stat_id	统计项 ID
value	统计项的值，例如：道具数量

### 用法

#### 1. 人数人次

```
Logger.custom_log ( "558955495" , 156453783);
```

#### 2. 数量和或最大值

```
Logger.custom_log ( "558955495" , 156453783 , 100);
```

## 四、添加调用方法(PHP)

PHP 提供的接口，除了第 20 个接口 custom\_log 不提供，其它接口与【三、添加调用方法(C++)】完全一致，添加方法请参照第三点。以下挑选几个接口示例如下。

### 0、初始化

```
/**
 * @brief StatLogger 构造函数
 *
 * @param game_id 应用ID 即游戏ID
 * @param zone_id 区ID
 * @param svr_id 服ID
 * @param site_id 平台ID 默认 -1表示该游戏不会拿出来放在不同的平台上运营 1: 表示淘米平台
 * @param is_game 是否游戏后台络的数据
 */
public function __construct($game, $zone = -1, $svr = -1, $site = -1, $isgame = 1) {
```

### 1、注册角色

接口定义:

```
// race: 职业 isp: 运营商
/**
 * @brief 用户在游戏中创建角色时调用
 */
public function reg_role($acct_id, $player_id = "", $race = "", $cli_ip = "", $ads_id = "", $browser = "", $device = "", $os = "", $resolution = "", $network = "", $isp = "") {
```

### 2 用户登录游戏(做活跃统计)

接口定义:

```
/**
 * @brief 用户登录游戏时(登录online)调用
 * @param lv 取值范围1~5000
 * @param zone 为空表示总体，该字段主要时给公司老的单区服游戏使用，用于区分电信登录和网通登录
 */
public function login_online($acct_id, $player_id, $race, $isvip, $lv, $cli_ip = "", $ads_id = "", $zone="", $browser = "", $device = "", $os = "", $resolution = "", $network = "", $isp = "") {
```



### 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” )

```
/**
 * @brief 玩家每次在游戏内使用米币购买道具时调用
 * @param outcome 付费获得的道具名称。如果pay_reason选择的是pay_buy
，则outcome赋值为相应的“道具名字”；
 *          如果pay_reason的值不是pay_buy，则本参数可以赋空字符串。
 * @param pay_channel 支付类型，如米币卡、米币帐户、支付宝、苹果官方、
网银、手机付费等等，米币渠道传"1"。
 */
public function pay($acct_id, $isvip, $pay_amount, $currency, $pay_reason, $outcome, $outcnt, $pay_channel) {
```

### 4、自定义统计项接口(只支持统计人数人次)

```
/**
 * @brief 自定义统计项
 * @param stat_name 主统计名称，不能以_开头或结束，StatLogger会自动把头尾的_去掉。
 * @param sub_stat_name 子主统计名称，一个主统计名称下可以有多个子统计项。
 *          不能以_开头或结束，StatLogger会自动把头尾的_去掉。
 */
public function log($stat_name, $sub_stat_name, $acct_id, $player_id = "-1", $info = null)
```

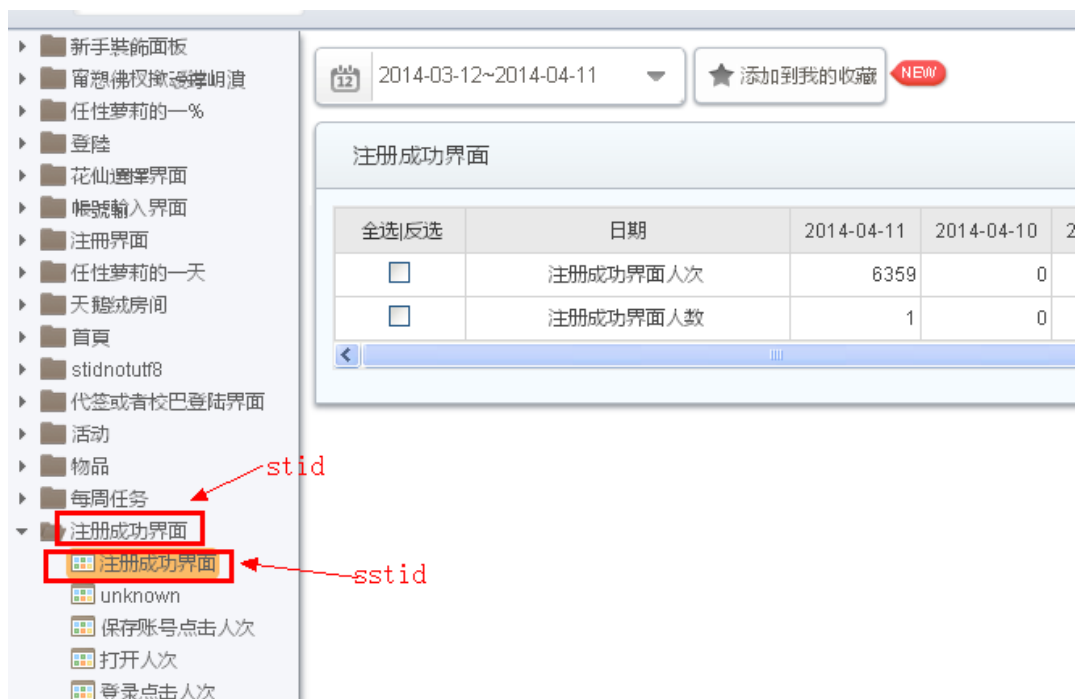
## 五、web 端统计或 AS 统计

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>



gameid:游戏 ID(**必须**为数字)

stid: 统计项名称

sstad: 子统计项名称

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

其中, stid、sstad 可填写中文、英文或数字字符串, 但若不能避免乱码问题, 请尽量选择填写英文或数字字符串

**三级统计项 :**

<http://newmisc.taomee.com/misc.js?gameid=gameid&stid=stid&sstad=sstad&uid=uid&item=item>

例如 :

[http://newmisc.taomee.com/misc.js?gameid=2&stid=物品 &sstad= 套装统计 &uid=500020&item= “套装107”](http://newmisc.taomee.com/misc.js?gameid=2&stid=物品&sstad=套装统计&uid=500020&item=套装107)

std

sstd

item

全选/反选	日期	2014-04-11	2014-04-12
<input type="checkbox"/>	套装统计人次	3982	
<input type="checkbox"/>	套装统计人数	2706	

全选/反选	日期	2014-04-11	2014-04-12
<input type="checkbox"/>	套装107人次	1	
<input type="checkbox"/>	套装117人次	3	
<input type="checkbox"/>	套装126人次	146	
<input type="checkbox"/>	套装128人次	3	

gameid:游戏 ID(**必须**为数字)

std: 统计项名称

sstd: 子统计项名称

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

item : 第 3 级统计项名称

其中，std、sstd、item 可填写中文、英文或数字字符串，但若不能避免乱码问题，

请尽量选择填写英文或数字字符串