

淘米数据平台接入指南-2.0.0 (页游)

C01 统计平台项目组

2014 年 7 月

目录

淘米数据平台接入指南-2.0.0 (页游)	1
C01 统计平台项目组	1
2014 年 6 月	1
一、综述.....	3
1、适用范围	3
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	4
3、添加调用方法	4
4、进行数据测试	5
三、添加调用方法(C++)	5
阅读指引 (适用于接口 1~16)	5
接口调用责任方说明.....	6
0、构造函数	6

1、登录游戏验证用户名和密码	7
2、创建游戏中角色	8
3、登录游戏服务器(online 服务器).....	10
4、退出游戏服务器(online 服务器).....	12
5、统计当前在线人数	13
6、用户升级	14
7、免费获得游戏币	15
8、使用游戏币购买道具	15
9、消耗游戏币（除购买游戏币道具）	16
10、接收任务	17
11、完成任务	18
12、放弃任务	19
13、获得精灵	19
14、失去精灵	20
15、退订 VIP 服务(取消自动续费)	21
16、销户(VIP 时间到期)	21
17、新用户注册转化	22
18、自定义统计	23
18.1、概述	23
18.2、示例	26
四、添加调用方法(PHP)	30
1、注册角色	30

2 用户登录游戏(做活跃统计).....	30
3、付费(开通 VIP、米币购买道具、人民币购买游戏内一级货币).....	31
4、自定义统计项接口(只支持统计人数人次).....	33
五、web 端统计或 AS 统计.....	33

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	17、新用户注册转化

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 1~16)

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。

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

接口调用责任方说明

调用方	接口	说明
游戏方（页游）	0-14，17	涵盖游戏内的登录、在线，等级、游戏币付费，任务等
账户支付平台	0，15，16	平台 vip 付费相关

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 (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
-----	------------

用途

1、统计项名称(stdid) : _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(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

用途

1、统计项名称(stdid) : _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(stdid) : _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(std) : _newrace_ 子统计项名称(sstd): 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  
需要,可带上相应参数
```

3、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,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

用途

1、统计项名称(stdid): _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(stdid): _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(stdid): _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(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(stdid): _logout_ 子统计项名称(sstdid): _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“电信”或“网通”，则分别统计电信或网通的在线人数。
--	--------------------------------

用途

统计项名称(stdid): _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

6、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)
lv	升之后的等级

用途

1、统计项名称(stdid): _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(stdid): _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、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(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”，原数量 需要乘以 100(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" , 100000); //接口中的数量需要乘以 100
```

8、使用游戏币购买道具

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	用户购买道具时的等级
pay_amount	支付数量,单位统一为“分”，原数量需要乘以 100
outcome	购买的道具
outcnt	购买的道具数量

用途

1、统计项名称(stdid): _buyitem_

子统计项名称(sstdid) : _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, 2000, "元旦礼包" , 10);//用户花了 20

游戏币购买 10 个元旦礼包

9、消耗游戏币（除购买游戏币道具）

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	支付数量,单位统一为“分”，原数量需要乘以 100
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_" , 1800, 17);//用户花 18
游戏币开启新功能
```

10、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(stdid): _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);
```

11、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(stdid): _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);
```

12、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);
```

13、获得精灵

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");
```

14、失去精灵

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

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");
```

15、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);
```

16、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(std) : _ccacct_

子统计项名称(sstid): _ccacct_

对 channel 字段做 item 运算

用法

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

17、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(std) : _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" );
```

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 key2
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。

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



The screenshot shows a game statistics interface. On the left is a tree view of categories, with 'Player Disconnection Statistics' (玩家掉线统计) selected and highlighted with a red box. On the right is a table titled 'Player Disconnection Statistics' (玩家掉线的人次). The table has columns for date and two data columns. The data rows are 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	20181
<input type="checkbox"/>	玩家掉线的人数	1120	24375	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->撞球对抗赛(人数、人次、产生的赛尔豆)

赛尔号				
2014运营活动				
基础指标				
2013运营活动				
触达率				
新加统计				
新手相关				
单一精灵				
渠道				
用户个性				
用户对战				
赛尔豆				
游戏输出				
撞球对抗赛				
套圈				
星座档案整理				
贝塔星勘探游戏				
保护导航仪				
黄金矿工				
双子阿尔法防空塔				

保护导航仪				
全选 反选		2014-03-18	2014-03-17	20
<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

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

1、注册角色

接口定义:

```
/**
 * @brief stat_reg_role 注册角色
 *
 * @param $gameid      游戏ID
 * @param $platformid  平台ID(不分平台填-1)
 * @param $zoneid      区号(不分区服时填-1)
 * @param $serverid    服号(不分区服时填-1)
 * @param $acct_id     帐号
 * @param $device      设备
 *
 * @return
 */
function stat_reg_role($gameid, $platformid, $zoneid, $serverid, $acct_id, $device)
```

2 用户登录游戏(做活跃统计)

接口定义:

```

/**
 * @brief stat_login_online 活跃
 *
 * @param $gameid      游戏ID
 * @param $platformid  平台ID(不分平台填-1)
 * @param $zoneid       区号(不分区服时填-1)
 * @param $serverid     服号(不分区服时填-1)
 * @param $acct_id      帐号
 * @param $device       设备
 *
 * @return
 */
function stat_login_online($gameid, $platformid, $zoneid, $serverid, $acct_id, $device)

```

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”)

```

/**
 * 用户付费时调用:开通VIP或 购买游戏内道具
 * gameid 游戏ID
 * platformid 平台ID 默认填写-1 表示全平台, 如果一个游戏需要放到不同的平台运营,
 * 这里就需要填写对应的平台ID, 需要与游戏内部保持一致
 * zoneid 游戏区ID 默认填写-1
 * serverid 游戏服ID 默认填写-1
 * acctid 用户帐号ID
 * isvip 是否VIP用户 0: 不是 1: 是
 * pay_amount 本次付费金额
 * currency 本次付费的货币单位 0:米币 1: 人民币
 * pay_reason 可取值 _buyitem_:通过米币购买游戏内道具
 * _vipmonth_:开通游戏内VIP包月
 * _buycoins_:购买游戏内货币
 * output 付费购买的产出物 购买道具时为具体的道具ID或名称
 * 开通VIP时为具体的VIP信息, 比如1个月 3个月
 * 购买游戏币时为游戏币一级货币的名称
 * outcnt 付费购买的产出物数量 购买道具时为道具的数量
 * 开通VIP时为具体的开通时长
 * 购买游戏币时为购买的游戏币数量
 * pay_channel 付费渠道 支付宝 财付通等
 *
 */
function stat_pay($gameid, $platformid, $zoneid, $serverid, $acct_id,
                 $isvip, $pay_amount, $currency, $pay_reason, $output,
                 $outcnt, $paychannel)

```


4、自定义统计项接口(只支持统计人数人次)

```
/**
 * 自定义统计项接口 目前只支持人数人次
 * playerid 为角色ID , 默认填写-1即可
 * gameid 游戏ID
 * platformid 平台ID 默认填写-1 表示全平台, 如果一个游戏需要放到不同的平台运营,
 * 这里就需要填写对应的平台ID, 需要与游戏内部保持一致
 * zoneid 游戏区ID 默认填写-1
 * serverid 游戏服ID 默认填写-1
 * acctid 用户帐号ID
 * playerid 用户角色ID 不需要分角色时填写-1即可
 * stname 统计项名称 (如完成任务)
 * sstname 子统计项名称 【如任务1】
 * 下面内容请忽略
 * array 自定义key-value数组,
 * 数组内容格式必须为$arr = array("peter"=>32, "jone"=>34);
 * 其中peter和jone为key 32 34为value
 * key不能以_开头和结尾 不可以包含 = : , | \t ; 中的任意一个字符
 * value 也不可以包含 = : , | \t ; 中的任意一个字符
 */
function stat_logger($gameid, $platformid, $zoneid, $serverid, $acct_id,
                    $playerid, $stname, $sstname)
{
```

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=u>

[id](#)

注册成功界面

全选/反选	日期	2014-04-11	2014-04-10	2
<input type="checkbox"/>	注册成功界面人次	6359	0	
<input type="checkbox"/>	注册成功界面人数	1	0	

三级统计项：

<http://newmisc.taomee.com/misc.js?gameid=gameid&stid=stid&sstid=sstid&uid=u>
[id&item=item](#)

套装统计

全选/反选	日期	2014-04-11	2014-04-10	2
<input type="checkbox"/>	套装统计人次	3982		
<input type="checkbox"/>	套装统计人数	2706		

套装统计item人次

全选/反选	日期	2014-04-11	2014-04-10	20
<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)

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)