接口规范说明文档

# 简介

本文档作为我方与数据调用方之间进行数据交互的桥梁，文档中规定了相关技术要求，接口方式，数据格式，数据质量要求等内容，旨在减少双方在数据共享方面的繁琐流程，加快有效信息的共享。

# 数据质量要求

**实时性：**我方在自有平台查询相关数据时，确保数据的及时性，若达不到及时性要求，则尽量缩短数据的间隔时间，例如缩短时间差在10分钟左右。添加运营数据接口时间在10分钟一次，添加玩家个人充值接口时间差为1分钟一次。其他接口均为每天凌晨三点统计之前X天数据。

**完整性：**务必按照本规范格式和内容上传相关数据；

**真实性：**不得对本规范规定的上传内容进行修改；

**正确性：**保证所查数据的正确性，所查的相关数据即为游戏log处理后的相关数据，符合运营的相关需求；

# 接口规范

## 相关说明

**WEB服务：**接口以webservice作为双方通讯规范，由我方提供相关数据接口地址，由贵方调用，调用过程中，需要依照接口规范进行相关数据的上传；

**userToken：**由我方提供给贵方的数据接入的唯一标识，待双方正式接入数据时，我方将提供Token，贵方调用接口传入即可。

**编码格式：**接口中所插入信息，无论中文或者英文字符，数字等，均为utf-8编码；

**服务器IP：**添加游戏接口中的gameServerIp规则为: 服务器ip:端口号

除添加游戏接口之外的所有其他接口的gameServerIp规则为：服务器ip：端口号：渠道号

**批量接口：**批量接口是为了方便数据的批量插入增加的，并非新的接口功能，所以当含有批量接口时，以批量接口调用为第一考虑。

## 相关规范

### 添加游戏( addGames )

**说明：**在调用其他方法之前，确保游戏接口中，已经存在游戏的相关数据。游戏和渠道为基础数据。每次调用添加游戏接口，必须调用一次添加渠道接口。

方法【addGames】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | gameName | 游戏名称 | string | 50 |  |
| gameServerIp | 服务器IP | string | 50 | 规则：  服务器IP：端口号 |
| gameServerName | 服务器名称 | string | 50 | 中文名称 |
| userToken | 用户标识 | string | 50 | 接口的唯一标识，我方提供 |
| gameCode | 游戏编码 | string | 50 | 游戏编码由我方分配，具有唯一性 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 添加渠道 ( addChannels )

**说明：**在调用其他方法之前，确保渠道接口中，已经存在渠道的相关数据。游戏和渠道为基础数据。

方法【addChannels】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | channelName | 渠道名称 | string | 50 | 依据渠道文档的渠道名称传递 |
| gameServerIp | 服务器IP | string | 50 | **规则：**  服务器IP：端口号：渠道号 |
| userToken | 用户标识 | string | 50 | 接口接入唯一标识 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 添加运营数据 ( addOperationData )

**要求：**10分钟要进行一次相关数据的汇总，通过调用本接口插入统计数据；接口中设计规则，同一日期，同一IP，每天只会保留一条数据，例如，凌晨1点，统计了前一小时的数据，调用接口已经插入数据库，当凌晨1点10分，那么就统计前1小时10分钟的数据，调用接口再进行插入，此时会将凌晨1点时的数据进行删除，只保留1点10分刚插入的数据；

**注意事项：**

**在凌晨0点之后，需要调用一次接口，避免因为10分钟更新一次数据，最后10分钟卡在23:50-24:00之间，导致最后几分钟的数据丢失，所以凌晨0点以后，要更新一下完整的数据。（当然也可以和其他接口一样在凌晨3点进行一次统计调用）**

此接口每次需要调用两次：

第一次只统计接口中有关设备的数据，其他数据可以忽略。原因是：新设备在未选择服务器之前，服务器的ip是没有的，所以就无法将涉及到服务器ip的设备数统计到对应服务器下，因此，采取措施为：服务器ip为：0.0.0.0:渠道编号；这时候提交一次设备的数据；

第二次统计所有的数据，因为这时候服务器ip已经可以得到，可以将相关设备统计到对应服务器。此时的服务器ip规则：服务器ip：端口号：渠道号

提交两次接口原因：想得到从新设备到用户注册或者用户登录的一个转化率。

方法【addOperationData】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | opDateTime | 分析时间 | datetime |  | 当前数据的分析日期 |
| gameServerIp | 服务器IP | string | 50 | **规则：**  服务器IP：端口号：渠道号 |
| userToken | 用户标识 | string | 50 |  |
| opPlatform | 运营平台 | string | 50 | 安卓或者苹果； |
| opNewStartDeviceNum | 新启动设备数 | int |  | 当天，从未安装过的游戏的全新设备并且启动游戏的设备数量（同设备一天仅记录一次） |
| opStartDeviceNum | 启动设备数 | int |  | 当天，新，旧设备并且启动游戏的设备数量（同设备一天仅记录一次） |
| opAddDeviceNum | 新增设备数 | int |  | 当天，从未安装过游戏的新设备并且登录账号的设备数（同设备一天仅记录一次） |
| opLoginDeviceNum | 登录设备数 | int |  | 当天，新旧设备并且登录账号的设备数（同设备一天仅记录一次） |
| opLoginAccountNum | 登录账号数 | int |  | 今日登陆游戏的账号数 |
| opFirstLoginAccountNum | 新登账号数 | int |  | 今日新登录游戏的账号数 |
| opPayAccountNum | 付费账号数 | int |  | 今日付费账号数 |
| opFirstPayAccountNum | 新付费账号数 | int |  | 今日付费账号数且本账号第一次进行付费 |
| opIncome | 收入 | decimal | (10,2) | 今日充值金额总额 |
| opFirstLoginIncome | 新登账号收入 | decimal | (10,2) | 今日新登录账号的充值金额 |
| opFirstPayIncome | 新付费账号收入 | decimal | (10,2) | 今日付费账号数且本账号第一次进行付费的玩家总充值金额 |
| opPayArpu | 付费ARPU | decimal | (10,2) | 今日收入/今日付费玩家数 |
| opDauArpu | DAU ARPU | decimal | (10,2) | 今日收入/今日登陆账号数 |
| opAccountPayPercent | 账号付费率 | decimal | (10,2) | 今日付费玩家数/今日登陆账号数 |
| opFirstLoginPayPercent | 新登账号付费率 | decimal | (10,2) | 今日新增付费账号数/今日新增账号数 |
| opYesterdayAccountRetention | 次日账号留存率 | decimal | (10,2) | 昨日新增在今日仍然登陆的账号/昨日新增账号数 |
| opThreeAccountRetention | 三日账号留存率 | decimal | (10,2) | 当日新增在第3日仍然登陆的账号/当日新增账号数 |
| opFourAccountRetention | 四日账号留存率 | decimal | (10,2) | 当日新增在第4日仍然登陆的账号/当日新增账号数 |
| opFiveAccountRetention | 五日账号留存率 | decimal | (10,2) | 当日新增在第5日仍然登陆的账号/当日新增账号数 |
| opSixAccountRetention | 六日账号留存率 | decimal | (10,2) | 当日新增在第6日仍然登陆的账号/当日新增账号数 |
| opSevenAccountRetention | 七日账号留存率 | decimal | (10,2) | 当日新增在第7日仍然登陆的账号/当日新增账号数 |
| opFifteenAccountRetention | 15日账号留存率 | decimal | (10,2) | 当日新增在第15日仍然登陆的账号/当日新增账号数 |
| opThirtyAccountRetention | 30日账号留存率 | decimal | (10,2) | 当日新增在第30日仍然登陆的账号/当日新增账号数 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 添加账号留存数据 ( addGamersRetention )

**说明：**

方法【addGamersRetention】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | gaDateTime | 分析时间 | datetime |  | 当前数据的分析日期 |
| gameServerIp | 服务器IP | string | 50 | **规则：**  服务器IP：端口号：渠道号 |
| userToken | 用户标识 | string | 50 |  |
| gaPlatform | 运营平台 | string | 50 | 安卓或者苹果； |
| gaRegisterAccountNum | 注册账号数 | int |  | 今日注册的账号数 |
| gaLoginAccountNum | 登录账号数 | int |  | 今日登陆账号数 |
| gaFirstLoginAccountNum | 新登账号数 | int |  | 今日新注册账号数 |
| gaPayAccountNum | 付费账号数 | int |  | 今日付费的账号数 |
| ga\_income | 收入 | decimal | (10,2) | 今日账号充值金额总额 |
| gaFirstLoginPayAccountIncome | 新登付费账号收入 | decimal | (10,2) | 今日付费账号且本账号第一次进行付费的收入 |
| gaFirstLoginPayAccountNum | 新登付费账号数 | int |  | 今日付费账号数且本账号第一次进行付费 |
| gaAccountPayPercent | 账号付费率 | decimal | (10,2) | 今日付费账号数/今日登陆账号数 |
| gaFirstLoginAccountPayPercent | 新登账号付费率 | decimal | (10,2) | 今日新增付费账号数/今日新增账号数 |
| gaFirstLoginAccountPayArpu | 新增账号付费ARPU | decimal | (10,2) | 今日收入/今日新增付费账号数 |
| gaAcu | 平均同时在线账号数 | int |  | 今日平均同时在线玩家账号数。（在线账号数5分钟统计一次） |
| gaPcu | 最高同时在线账号数 | int |  | 今日最高同时在线账号数。（在线账号数5分钟统计一次） |
| gaPcuTimeInterval | pcu时段 | string | 50 | 今日最高在线账号时间段（5分钟统计一次账号数，然后以一小时为单位，统计一小时内5分钟账号数总和，比较24个小时段里面，哪个最高，即为最高PCU时段，比如：12-13） |
| gaAccountAverageOnlineTime | 账号平均在线时长 | decimal | (10,2) | 所有账号在线时间/所有账号数 |
| gaYesterdayAccountRetention | 次日账号留存率 | decimal | (10,2) | 昨日新增在今日仍然登陆的账号/昨日新增账号数 |
| gaThreeAccountRetention | 三日账号留存率 | decimal | (10,2) | 当日新增在第3日仍然登陆的账号/当日新增账号数 |
| gaFourAccountRetention | 四日账号留存率 | decimal | (10,2) | 当日新增在第4日仍然登陆的账号/当日新增账号数 |
| gaFiveAccountRetention | 五日账号留存率 | decimal | (10,2) | 当日新增在第5日仍然登陆的账号/当日新增账号数 |
| gaSixAccountRetention | 六日账号留存率 | decimal | (10,2) | 当日新增在第6日仍然登陆的账号/当日新增账号数 |
| gaSevenAccountRetention | 七日账号留存率 | decimal | (10,2) | 当日新增在第7日仍然登陆的账号/当日新增账号数 |
| gaFifteenAccountRetention | 15日账号留存率 | decimal | (10,2) | 当日新增在第15日仍然登陆的账号/当日新增账号数 |
| gaThirtyAccountRetention | 30日账号留存率 | decimal | (10,2) | 当日新增在第30日仍然登陆的账号/当日新增账号数 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 添加角色留存数据 (addGamersRoleRetention)

方法【addGamersRoleRetention】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | gaDateTime | 分析时间 | datetime |  | 当前数据的分析日期 |
| gameServerIp | 服务器IP | string | 50 | **规则：**  服务器IP：端口号：渠道号 |
| userToken | 用户标识 | string | 50 |  |
| gaPlatform | 运营平台 | string | 50 | 安卓或者苹果； |
| gaRegisterRoleNum | 注册角色数 | int |  | 今日建立的角色数 |
| gaLoginRoleNum | 登录角色数 | int |  | 今日登陆角色数 |
| gaFirstLoginRoleNum | 新登角色数 | int |  | 今日新建立角色数 |
| gaPayRoleNum | 付费角色数 | int |  | 今日付费的角色数 |
| ga\_income | 收入 | decimal | (10,2) | 今日角色充值金额总额 |
| gaFirstLoginPayRoleIncome | 新登付费角色收入 | decimal | (10,2) | 今日新登付费的角色收入 |
| gaFirstLoginPayRoleNum | 新登付费角色数 | int |  | 今日付费角色数且本角色第一次进行付费 |
| gaRolePayPercent | 角色付费率 | decimal | (10,2) | 今日付费角色数/今日登陆角色数 |
| gaFirstLoginRolePayPercent | 新登角色付费率 | decimal | (10,2) | 今日新增付费角色数/今日新增角色数 |
| gaFirstLoginRolePayArpu | 新增角色付费ARPU | decimal | (10,2) | 今日收入/今日新增付费角色数 |
| gaAcu | 平均同时在线角色数 | int |  | 今日平均同时在线角色数（5分钟统计一次在线角色数）。 |
| gaPcu | 最高同时在线角色数 | int |  | 今日最高同时在线角色数（5分钟统计一次在线角色数）。 |
| gaPcuTimeInterval | pcu时段 | string | 50 | 今日最高在线时间段（5分钟统计一次角色数，然后以一小时为单位，统计一小时内5分钟角色数总和，比较24个小时段里面，哪个最高，即为最高PCU时段，比如：12-13） |
| gaRoleAverageOnlineTime | 角色平均在线时长 | decimal | (10,2) | 所有角色在线时间/所有角色数 |
| gaYesterdayRoleRetention | 次日角色留存率 | decimal | (10,2) | 昨日新增在今日仍然登陆的角色/昨日新增角色数 |
| gaThreeRoleRetention | 三日角色留存率 | decimal | (10,2) | 当日新增在第3日仍然登陆的角色/当日新增角色数 |
| gaFourRoleRetention | 四日角色留存率 | decimal | (10,2) | 当日新增在第4日仍然登陆的角色/当日新增角色数 |
| gaFiveRoleRetention | 五日角色留存率 | decimal | (10,2) | 当日新增在第5日仍然登陆的角色/当日新增角色数 |
| gaSixRoleRetention | 六日角色留存率 | decimal | (10,2) | 当日新增在第6日仍然登陆的角色/当日新增角色数 |
| gaSevenRoleRetention | 七日角色留存率 | decimal | (10,2) | 当日新增在第7日仍然登陆的角色/当日新增角色数 |
| gaFifteenRoleRetention | 15日角色留存率 | decimal | (10,2) | 当日新增在第15日仍然登陆的角色/当日新增角色数 |
| gaThirtyRoleRetention | 30日角色留存率 | decimal | (10,2) | 当日新增在第30日仍然登陆的角色/当日新增角色数 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 添加设备留存数据 (addGamersEquipRetention)

方法【addGamersEquipRetention】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | gaDateTime | 分析时间 | datetime |  | 当前数据的分析日期 |
| gameServerIp | 服务器IP | string | 50 | **规则：**  服务器IP：端口号：渠道号 |
| userToken | 用户标识 | string | 50 |  |
| gaPlatform | 运营平台 | string | 50 | 安卓或者苹果； |
| gaRegisterEquipNum | 注册设备数 | int |  | 今日安装设备数 |
| gaLoginEquipNum | 登录设备数 | int |  | 今日登陆设备数 |
| gaFirstLoginEquipNum | 新登设备数 | int |  | 今日新启动设备数 |
| gaPayEquipNum | 付费设备数 | int |  | 今日付费的设备数 |
| ga\_income | 收入 | decimal | (10,2) | 今日设备充值金额总额 |
| gaFirstLoginPayEquipIncome | 新登付费设备收入 | decimal | (10,2) | 今日新登付费的设备收入 |
| gaFirstLoginPayEquipNum | 新登付费设备数 | int |  | 今日付费设备数且本设备第一次进行付费 |
| gaEquipPayPercent | 设备付费率 | decimal | (10,2) | 今日付费设备数/今日登陆设备数 |
| gaFirstLoginEquipPayPercent | 新登设备付费率 | decimal | (10,2) | 今日新增付费设备数/今日新增设备数 |
| gaFirstLoginEquipPayArpu | 新增设备付费ARPU | decimal | (10,2) | 今日收入/今日新增付费设备数 |
| gaAcu | 平均同时在线设备数 | int |  | 今日平均同时在线设备数（在线设备数5分钟统计一次）。 |
| gaPcu | 最高同时在线设备数 | int |  | 今日最高同时在线设备数（在线设备数5分钟统计一次）。 |
| gaPcuTimeInterval | pcu时段 | string | 50 | 今日最高在线时间段（5分钟统计一次设备数，然后以一小时为单位，统计一小时内5分钟设备数总和，比较24个小时段里面，哪个最高，即为最高PCU时段，比如：12-13） |
| gaEquipAverageOnlineTime | 设备平均在线时长 | decimal | (10,2) | 所有设备在线时间/所有设备数 |
| gaYesterdayEquipRetention | 次日设备留存率 | decimal | (10,2) | 昨日新增在今日仍然登陆的设备/昨日新增设备数 |
| gaThreeEquipRetention | 三日设备留存率 | decimal | (10,2) | 当日新增在第3日仍然登陆的设备/当日新增设备数 |
| gaFourEquipRetention | 四日设备留存率 | decimal | (10,2) | 当日新增在第4日仍然登陆的设备/当日新增设备数 |
| gaFiveEquipRetention | 五日设备留存率 | decimal | (10,2) | 当日新增在第5日仍然登陆的设备/当日新增设备数 |
| gaSixEquipRetention | 六日设备留存率 | decimal | (10,2) | 当日新增在第6日仍然登陆的设备/当日新增设备数 |
| gaSevenEquipRetention | 七日设备留存率 | decimal | (10,2) | 当日新增在第7日仍然登陆的设备/当日新增设备数 |
| gaFifteenEquipRetention | 15日设备留存率 | decimal | (10,2) | 当日新增在第15日仍然登陆的设备/当日新增设备数 |
| gaThirtyEquipRetention | 30日设备留存率 | decimal | (10,2) | 当日新增在第30日仍然登陆的设备/当日新增设备数 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 添加常规LTV价值数据 ( addCommonLtvWorth)

方法【addCommonLtvWorth】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | coDateTime | 分析时间 | datetime |  | 当前数据的分析日期 |
| gameServerIp | 服务器IP | string | 50 | **规则：**  服务器IP：端口号：渠道号 |
| userToken | 用户标识 | string | 50 |  |
| coPlatform | 运营平台 | string | 50 | 安卓或者苹果 |
| coFirstLoginAccountNum | 新登账号数 | int |  | 今日新登录游戏的账号数 |
| coThreeDayLtv | 三日LTV | decimal | (10,2) | 3天内第一日新增账号付费总额/第一日新增账号数 |
| coSevenDayLtv | 七日LTV | decimal | (10,2) | 7天内第一日新增账号付费总额/第一日新增账号数 |
| coFifteenDayLtv | 15日LTV | decimal | (10,2) | 15天内第一日新增账号付费总额/第一日新增账号数 |
| coThirtyDayLtv | 30日LTV | decimal | (10,2) | 30天内第一日新增账号付费总额/第一日新增账号数 |
| coThreeDayIncome | 3日收入 | decimal | (10,2) | 3天内第一日新增账号付费总额 |
| coSevenDayIncome | 7日收入 | decimal | (10,2) | 7天内第一日新增账号付费总额 |
| coFifteenDayIncome | 15日收入 | decimal | (10,2) | 15天内第一日新增账号付费总额 |
| coThirtyDayIncome | 30日收入 | decimal | (10,2) | 30天内第一日新增账号付费总额 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 添加玩家日价值数据 ( addGamersDayWorth)

方法【addGamersDayWorth】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | gaDateTime | 分析时间 | datetime |  | 当前数据的分析日期 |
| gameServerIp | 服务器IP | string | 50 | **规则：**  服务器IP：端口号：渠道号 |
| userToken | 用户标识 | string | 50 |  |
| gaPlatform | 运营平台 | string | 50 | 安卓或者苹果 |
| gaPayAccountNum | 付费玩家账号数 | int |  | 今天有付费的账号数 |
| gaThreeDayRepeatPayAccountNum | 3日内重复付费账号数 | int |  | 3日内有两次以及两次以上付费的账号数 |
| gaSevenDayRepeatPayAccountNum | 7日内重复付费账号数 | int |  | 7日内有两次以及两次以上付费的账号数 |
| gaFifteenDayRepeatPayAccountNum | 15日内重复付费账号数 | int |  | 15日内有两次以及两次以上付费的账号数 |
| gaThirtyDayRepeatPayAccountNum | 30日内重复付费账号数 | int |  | 30日内有两次以及两次以上付费的账号数 |
| gaAddThreeDaySumIncome | 当日新增3日内累计收入 | decimal | (10,2) | 3天内第一日新增账号付费总额 |
| gaAddSevenDaySumIncome | 当日新增7日内累计收入 | decimal | (10,2) | 7天内第一日新增账号付费总额 |
| gaAddFifteenDaySumIncome | 当日新增15日内累计收入 | decimal | (10,2) | 15天内第一日新增账号付费总额 |
| gaAddThirtyDaySumIncome | 当日新增30日内累计收入 | decimal | (10,2) | 30天内第一日新增账号付费总额 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 添加玩家个人充值数据 ( addUserRecharges)

方法【addUserRecharges】

注意事项：

本接口一分钟调用一次，将一分钟之内的新增玩家充值信息提交一次，保证数据完整，无遗漏，同时具有良好的实时性。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | gameOrderNo | 游戏订单号 | string | 50 | 游戏订单号（游戏服务器本身生成的订单号） |
| channelOrderNo | 渠道订单号 | string | 50 | 渠道订单号，渠道返回的订单号 |
| gameServerIp | 服务器IP | string | 50 | **规则：**  服务器IP：端口号：渠道号 |
| userToken | CP用户标识 | string | 50 | CP用户唯一标识 |
| gamerAccount | 玩家账号 | string | 50 | 玩家账号（最好是原始注册账号，而不是加密后的账号） |
| roleName | 角色名称 | string | 50 | 角色名称 |
| roleId | 角色Id | string | 50 | 角色id |
| mainChannelName | 主渠道名称 | string | 50 | 比如棱镜对接了N个渠道，那么棱镜代表主渠道，子渠道就代表那N个渠道。默认为空 |
| equipId | 设备ID | string | 50 | 设备ID号（直接从设备拿到的设备号，而不是经过处理的） |
| platform | 联运方 | string | 50 | 默认中文在线 |
| gamerServerName | 玩家游戏区服 | string | 50 | 玩家游戏区服 |
|  | accountCreateTime | 账号创建时间 | datetime |  | 账号创建时间 |
|  | gamerPhone | 玩家手机号 | string | 20 | 玩家手机号 |
|  | orderCreateTime | 订单生成时间 | datetime |  | 订单生成时间 |
|  | orderMoney | 订单金额 | decimal | (10,2) | 订单金额 |
|  | rechargeMoney | 充值金额 | decimal | (10,2) | 充值金额 |
|  | rechargeFunc | 充值方式 | string | 50 | 比如微信，支付宝，银行卡等 |
|  | rechargetState | 支付状态 | int |  | 0．未支付（玩家未付款）；  1．已经支付（玩家已经付款） |
|  | orderState | 订单状态 | int |  | 1．下单失败（玩家点击充值，没有弹出渠道支付页面）；  2.下单成功（玩家点击充值进入充值页面）；  3.支付成功（玩家进入付款页面且完成支付）；  4.支付失败（玩家进入付款页面未进行支付）；  5.通知成功（游戏得到通知，发放钻石）；  6.通知失败（游戏未得到通知，没有发放钻石）； |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 添加每日新增玩家等级分布 ( addEverydayAddGamerLevelFenbu)

方法【addEverydayAddGamerLevelFenbu】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | evDateTime | 日期 | datetime |  | 分析的日期 |
| gameServerIp | 服务器IP | string | 50 | **规则：**  服务器IP：端口号：渠道号 |
| userToken | 用户标识 | string | 50 |  |
| evGamerLevel | 玩家等级 | int |  | 当日新增玩家的等级 |
| evGamerNumber | 玩家数量 | int |  | 当日新增玩家处于该等级的数量 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 批量添加每日新增玩家等级分布(addEverydayAddGamerLevelFenbuBatch)

方法【addEverydayAddGamerLevelFenbuBatch】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | userToken | 用户标识 | string | 50 |  |
| jsonBatch | 虚拟币消耗json字符串 | string |  | 泛型List<EverydayAddGamerLevelFenbuInfo>经过序列化的json字符串 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

说明：

参数名称jsonBatch （泛型List<EverydayAddGamerLevelFenbuInfo>经过序列化的json字符串）

需要贵方定义类，类名称为EverydayAddGamerLevelFenbuInfo含有以下属性：

evDateTime datetime类型，

gameServerIp string类型，

evGamerLevel int类型，

evGamerNumber int类型

举例：

原始List< EverydayAddGamerLevelFenbuInfo>：

var test = new List<EverydayAddGamerLevelFenbuInfo>() {

new EverydayAddGamerLevelFenbuInfo() { gameServerIp = "123.22.33.45", evDateTime = Convert.ToDateTime("2016-05-16 12:00:00"), evGamerLevel = 1, evGamerNumber = 112 },

new EverydayAddGamerLevelFenbuInfo() { gameServerIp = "122.21.33.45", evDateTime = Convert.ToDateTime("2016-05-16 12:00:00"), evGamerLevel = 2, evGamerNumber = 130 },

new EverydayAddGamerLevelFenbuInfo() { gameServerIp = "114.22.33.45", evDateTime = Convert.ToDateTime("2016-05-16 12:00:00"), evGamerLevel = 3, evGamerNumber = 192 }

};

序列化传递的json字符串：

[{"evDateTime":"\/Date(1463379737034)\/","gameServerIp":"123.22.33.45","evGamerLevel":1,"evGamerNumber":112},{"evDateTime":"\/Date(1463379737034)\/","gameServerIp":"122.21.33.45","evGamerLevel":2,"evGamerNumber":130},{"evDateTime":"\/Date(1463379737034)\/","gameServerIp":"114.22.33.45","evGamerLevel":3,"evGamerNumber":192}]

### 添加所有玩家等级分布 ( addAllGamerLevelFenbu)

方法【addAllGamerLevelFenbu】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | allDateTime | 日期 | datetime |  | 分析的日期 |
| gameServerIp | 服务器IP | string | 50 | **规则：**  服务器IP：端口号：渠道号 |
| userToken | 用户标识 | string | 50 |  |
| allGamerLevel | 玩家等级 | int |  | 当日所有玩家的等级 |
| allGamerNumber | 玩家数量 | int |  | 当日所有玩家处于该等级下的数量 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 批量添加所有玩家等级分布 ( addAllGamerLevelFenbuBatch)

方法【addAllGamerLevelFenbuBatch】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | userToken | 用户标识 | string | 50 |  |
| jsonBatch | 虚拟币消耗json字符串 | string |  | 泛型List<AllGamerLevelFenbuInfo>经过序列化的json字符串 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

说明：

参数名称jsonBatch （泛型List<AllGamerLevelFenbuInfo>经过序列化的json字符串）

需要贵方定义类，类名称为AllGamerLevelFenbuInfo含有以下属性：

allDateTime datetime类型，

gameServerIp string类型，

allGamerLevel int类型，

allGamerNumber int类型

举例：

原始List<AllGamerLevelFenbuInfo>：

var test = new List<AllGamerLevelFenbuInfo>() {

new AllGamerLevelFenbuInfo() { gameServerIp = "123.22.33.45", allDateTime = Convert.ToDateTime("2016-05-16 12:00:00"), allGamerLevel = 1, allGamerNumber = 112 },

new AllGamerLevelFenbuInfo() { gameServerIp = "122.21.33.45", allDateTime = Convert.ToDateTime("2016-05-16 12:00:00"), allGamerLevel = 2, allGamerNumber = 130 },

new AllGamerLevelFenbuInfo() { gameServerIp = "114.22.33.45", allDateTime = Convert.ToDateTime("2016-05-16 12:00:00"), allGamerLevel = 3, allGamerNumber = 192 }

};

序列化传递的json字符串：

[{"allDateTime":"\/Date(1463379737034)\/","gameServerIp":"123.22.33.45","allGamerLevel":1,"allGamerNumber":112},{"allDateTime":"\/Date(1463379737034)\/","gameServerIp":"122.21.33.45","allGamerLevel":2,"allGamerNumber":130},{"allDateTime":"\/Date(1463379737034)\/","gameServerIp":"114.22.33.45","allGamerLevel":3,"allGamerNumber":192}]

### 添加等级变更数据 ( addGamerLevelChanges)

方法【addGamerLevelChanges】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | chDateTime | 日期 | datetime |  | 分析log的日期 |
| gameServerIp | 服务器IP | string | 50 | **规则：**  服务器IP：端口号：渠道号 |
| userToken | 用户标识 | string | 50 |  |
| chGamerLevel | 玩家等级 | int |  |  |
| chGamerNumber | 玩家数量 | int |  | 第二日等级没有变化的玩家数量 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 批量添加等级变更数据 ( addGamerLevelChangesBatch)

方法【addGamerLevelChangesBatch】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | userToken | 用户标识 | string | 50 |  |
| jsonBatch | 虚拟币消耗json字符串 | string |  | 泛型List<GamerLevelChangeInfo>经过序列化的json字符串 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

说明：

参数名称jsonBatch （泛型List<GamerLevelChangeInfo>经过序列化的json字符串）

需要贵方定义类，类名称为GamerLevelChangeInfo含有以下属性：

chDateTime datetime类型，

gameServerIp string类型，

chGamerLevel int类型，

chGamerNumber int类型

举例：

原始List<GamerLevelChangeInfo>：

var test = new List<GamerLevelChangeInfo>() {

new GamerLevelChangeInfo() { gameServerIp = "123.22.33.45", chDateTime = Convert.ToDateTime("2016-05-16 12:00:00"), chGamerLevel = 1, chGamerNumber = 112 },

new GamerLevelChangeInfo() { gameServerIp = "122.21.33.45", chDateTime = Convert.ToDateTime("2016-05-16 12:00:00"), chGamerLevel = 2, chGamerNumber = 130 },

new GamerLevelChangeInfo() { gameServerIp = "114.22.33.45", chDateTime = Convert.ToDateTime("2016-05-16 12:00:00"), chGamerLevel = 3, chGamerNumber = 192 }

};

序列化传递的json字符串：

[{"chDateTime":"\/Date(1463379737034)\/","gameServerIp":"123.22.33.45","chGamerLevel":1,"chGamerNumber":112},{"chDateTime":"\/Date(1463379737034)\/","gameServerIp":"122.21.33.45","chGamerLevel":2,"chGamerNumber":130},{"chDateTime":"\/Date(1463379737034)\/","gameServerIp":"114.22.33.45","chGamerLevel":3,"chGamerNumber":192}]

### 添加等级流失数据 ( addGamerLevelLeft) --玩家7日未登录判定为流失

方法【addGamerLevelLeft】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | leDateTime | 日期 | datetime |  | 分析log的日期 |
| gameServerIp | 服务器IP | string | 50 | **规则：**  服务器IP：端口号：渠道号 |
| userToken | 用户标识 | string | 50 |  |
| leGamerLevel | 玩家等级 | int |  |  |
| leGamerNumber | 玩家数量 | int |  | 流失玩家在该等级下的数量 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 批量添加等级流失数据 (addGamerLevelLeftBatch) --玩家7日未登录判定为流失

方法【addGamerLevelLeftBatch】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | userToken | 用户标识 | string | 50 |  |
| jsonBatch | 等级流失json字符串 | string |  | 泛型List< GamerLevelLeftInfo>经过序列化的json字符串 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

说明：

参数名称jsonBatch（泛型List< GamerLevelLeftInfo>经过序列化的json字符串）

需要贵方定义类，类名称为GamerLevelLeftInfo含有以下属性：

leDateTime datetime类型，

gameServerIp string类型，

leGamerLevel int类型，

leGamerNumber int类型

举例：

原始List<GamerLevelLeftInfo>：

var test = new List<GamerLevelLeftInfo>() {

new GamerLevelLeftInfo() { gameServerIp = "123.22.33.45", leDateTime = DateTime.Now, leGamerLevel = 1, leGamerNumber = 112 },

new GamerLevelLeftInfo() { gameServerIp = "122.21.33.45", leDateTime = DateTime.Now, leGamerLevel = 2, leGamerNumber = 130 },

new GamerLevelLeftInfo() { gameServerIp = "114.22.33.45", leDateTime = DateTime.Now, leGamerLevel = 3, leGamerNumber = 192 }

};

序列化传递的json字符串：

[{"leDateTime":"\/Date(1463379737034)\/","gameServerIp":"123.22.33.45","leGamerLevel":1,"leGamerNumber":112},{"leDateTime":"\/Date(1463379737034)\/","gameServerIp":"122.21.33.45","leGamerLevel":2,"leGamerNumber":130},{"leDateTime":"\/Date(1463379737034)\/","gameServerIp":"114.22.33.45","leGamerLevel":3,"leGamerNumber":192}]

### 添加虚拟币消耗数据 (addVirtualCornCost)

方法【addVirtualCornCost】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | coDateTime | 日期 | datetime |  | 分析log的日期 |
| gameServerIp | 服务器IP | string | 50 | **规则：**  服务器IP：端口号：渠道号 |
| userToken | 用户标识 | string | 50 |  |
| coCostCategoryName | 虚拟币消耗类型名称 | string | 50 | 详细的虚拟币消耗名称，比如每日登陆，副本通关，购买钻石，赠送钻石等 |
| coCostNumber | 玩家数量 | int |  | 该虚拟币消耗下的玩家数量 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 批量添加虚拟币消耗数据 (addVirtualCornCostBatch)

方法【addVirtualCornCostBatch】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | userToken | 用户标识 | string | 50 |  |
| jsonBatch | 虚拟币消耗json字符串 | string |  | 泛型List<VirtualCornCostInfo>经过序列化的json字符串 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

说明：

参数名称jsonBatch （泛型List<VirtualCornCostInfo>经过序列化的json字符串）

需要贵方定义类，类名称为VirtualCornCostInfo含有以下属性：

coDateTime datetime类型，

gameServerIp string类型，

coCostCategoryName string类型，

coCostNumber int类型

举例：

原始List<GamerLevelLeftInfo>：

var test = new List<VirtualCornCostInfo>() {

new VirtualCornCostInfo() { gameServerIp = "123.22.33.45", coDateTime = Convert.ToDateTime("2016-05-16 12:00:00"), coCostCategoryName = "金币", coCostNumber = 112 },

new VirtualCornCostInfo() { gameServerIp = "122.21.33.45", coDateTime = Convert.ToDateTime("2016-05-16 12:00:00"), coCostCategoryName = "钻石", coCostNumber = 130 },

new VirtualCornCostInfo() { gameServerIp = "114.22.33.45", coDateTime = Convert.ToDateTime("2016-05-16 12:00:00"), coCostCategoryName = "铜钱", coCostNumber = 192 }

};

序列化传递的json字符串：

[{"coDateTime":"\/Date(1463379737034)\/","gameServerIp":"123.22.33.45","coCostCategoryName":"金币","coCostNumber":112},{"coDateTime":"\/Date(1463379737034)\/","gameServerIp":"122.21.33.45","coCostCategoryName":"钻石","coCostNumber":112},{"coDateTime":"\/Date(1463379737034)\/","gameServerIp":"114.22.33.45","coCostCategoryName":"铜钱","coCostNumber":113}]

### 添加鲸鱼用户数据(addRechargeGamerInfo)

方法【addRechargeGamerInfo】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | caculateDateTime | 分析log日期 | datetime |  | 原始数据的日期 |
| inRoleName | 角色名称 | string | 50 | 角色名称 |
| gameServerIp | 服务器IP | string | 50 | **规则：**  服务器IP：端口号：渠道号 |
| userToken | 用户标识 | string | 50 |  |
| inRoleId | 角色id | string | 50 |  |
| inEquip | 设备 | string | 50 |  |
| inRechargeMoney | 充值金额 | decimal | (10,2) |  |
| inRechargeLevel | 充值等级 | int |  |  |
| inDateTime | 最后一次充值时间 | datetime |  |  |
| inVirtualCornOwnNumber | 虚拟币拥有数量 | int |  |  |
| inVirtualCornSumCost | 虚拟币消耗总量 | int |  |  |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

### 批量添加鲸鱼用户数据 (addRechargeGamerInfoBatch)

方法【addRechargeGamerInfoBatch】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 参数名 | 中文说明 | 数据类型 | 最大长度 | 详细说明 |
| 传入参数 | userToken | 用户标识 | string | 50 |  |
| jsonBatch | 鲸鱼用户数据json字符串 | string |  | 泛型List<BigRechargeGamerInfo>经过序列化的json字符串 |
| 返回值 | Json字符串，ResultCode为1代表成功，0代表失败；ResultMsg：请求成功则为Success，请求失败则为异常信息内容； | | | | |

说明：

参数名称jsonBatch （泛型List<BigRechargeGamerInfo>经过序列化的json字符串）

需要贵方定义类，类名称为BigRechargeGamerInfo含有以下属性：

caculateDateTime datetime类型，

inRoleName string类型，

gameServerIp string类型，

inRoleIdstring类型

inEquip string 类型

inRechargeMoney decimal类型

inRechargeLevel int 类型

inDateTime datetime 类型

inVirtualCornOwnNumber int 类型

inVirtualCornSumCost int类型

举例：

原始List<GamerLevelLeftInfo>：

var test = new List<BigRechargeGamerInfo>() {

new BigRechargeGamerInfo(){ caculateDateTime=DateTime.Parse("2016-05-17 12:00:00").ToLocalTime(), gameServerIp="123.123.123.123:80", inDateTime=DateTime.Parse("2016-05-17 12:00:00").ToLocalTime(), inEquip="Ewwq", inRechargeLevel=12, inRechargeMoney=120, inRoleId = "11223", inRoleName="ceshi001", inVirtualCornOwnNumber=2100, inVirtualCornSumCost=1200},

new BigRechargeGamerInfo(){ caculateDateTime= DateTime.Parse("2016-05-17 12:00:00").ToLocalTime(), gameServerIp="123.123.123.123:8001", inDateTime=DateTime.Parse("2016-05-17 12:00:00").ToLocalTime(), inEquip="Xtte", inRechargeLevel=13, inRechargeMoney=150, inRoleId = "11000", inRoleName="ceshi002", inVirtualCornOwnNumber=1200, inVirtualCornSumCost=1100}

};

序列化传递的json字符串：

[{"caculateDateTime":"\/Date(1463486400000)\/","inRoleName":"ceshi001","gameServerIp":"123.123.123.123:80","inRoleId":"11223","inEquip":"Ewwq","inRechargeMoney":120,"inRechargeLevel":12,"inDateTime":"\/Date(1463486400000)\/","inVirtualCornOwnNumber":2100,"inVirtualCornSumCost":1200},{"caculateDateTime":"\/Date(1463486400000)\/","inRoleName":"ceshi002","gameServerIp":"123.123.123.123:8001","inRoleId":"11000","inEquip":"Xtte","inRechargeMoney":150,"inRechargeLevel":13,"inDateTime":"\/Date(1463486400000)\/","inVirtualCornOwnNumber":1200,"inVirtualCornSumCost":1100}]