服务器调用说明书

[一、Googlebuffer服务 2](#_Toc3421)

[（一）协议及解释 2](#_Toc8800)

[（二）具体接口说明 5](#_Toc10350)

[（三）心跳 6](#_Toc12653)

[（四）信令 6](#_Toc17832)

[1.用户登录 6](#_Toc14658)

[2.用户登出 8](#_Toc21440)

[3.获取直播列表 9](#_Toc20945)

[4.发布直播 10](#_Toc9903)

[5.关注/取消关注 12](#_Toc21457)

[6.获取短信验证码 13](#_Toc15976)

[7.开始/结束观看视频 13](#_Toc2332)

[8.上报位置 16](#_Toc10311)

[9.查询用户信息 18](#_Toc12509)

[10.修改好友 19](#_Toc30982)

[11.修改自身信息 20](#_Toc32318)

[12.获取粉丝（关注我的） 20](#_Toc2865)

[13.获取关注列表（我关注的） 21](#_Toc11639)

[14.为直播点赞 22](#_Toc20086)

[15.打赏/打赏通话 23](#_Toc27250)

[16.社交分享 28](#_Toc5705)

[17.获取直播中的用户 29](#_Toc23894)

[18.悬赏 29](#_Toc2725)

[19.获取资金变动日志 34](#_Toc5314)

[20.获取当前服务器列表 35](#_Toc13311)

[21.设置/获取媒体设备号 35](#_Toc634)

[二、Http服务 36](#_Toc9123)

[（一）上传用户头像 36](#_Toc9933)

[（二）下载用户头像 36](#_Toc1035)

[（三）上传客户端日志 37](#_Toc10254)

服务器调用说明书

# 一、Googlebuffer服务

当前Googlebuffer协议服务的端口号为：9997

## （一）协议及解释

syntax = "proto3"; //Googlebuffer协议版本

package bean; //生成文件所在的包名

option java\_package = "com.zrsg.bean"; //生成文件所在的包名（全路径）

option java\_outer\_classname = "WebPackage"; //生成文件的类名

message Packet{ //协议主体类

type packetType =1; //包类型：枚举。用于区分信令、心跳还是其他

string from = 2; //发送者

string to =3; //接收者

string id = 4; //包ID

string method =5; //方法名。用于区分不同的接口

string operateType =6; //操作类型。用户区分同一接口下的不同操作类型

Result result =7; //本次请求的操作结果，是否成功，失败原因

Data data =8; //接口返回的数据

string version=9; //当前版本号。iq类型包必填

enum type{ //枚举。包类型

iq = 0; //信令类型

msg = 1; //消息类型

buffer = 2; //流媒体文件类型

beat = 3; //心跳包

}

}

message Result{ //请求的处理结果类

bool result = 1; //是否成功

string error = 2; //错误信息

}

message Data{ //数据类

string normal = 1; //通用数据

bytes byteDate = 2; //二进制数据

int64 fromTime = 3; //筛选数据的起始时间

int64 toTime = 4; //筛选数据的截止时间

int32 sum = 5; //数据的总量

int32 from = 6; //当前传送数据的起始位置

int32 to = 7; //当前传送数据的结束位置

repeated Position position= 8; //（可重复）位置类

repeated User user = 9; //（可重复）用户类

repeated Video video = 10; //（可重复）直播视频类

repeated Gratuity gratuity = 11; //（可重复）打赏类

repeated Reward reward = 12; //（可重复）悬赏类

repeated UserState userState = 13;//（可重复）用户状态

repeated AssetLog assetLog = 14; //（可重复）资金变动的日志类 ServerList

repeated ServerBean serverBean = 15; //（可重复）当前可用的服务器列表

}

message User{ //用户类

int32 id = 1; //用户ID

string phone = 2; //电话号码

string name = 3; //用户名/昵称

string pwd = 4; //用户密码

string pwd2orCode = 5; //新密码或者验证码

string realName = 6; //用户真实姓名

string headurl = 7; //用户头像地址

string signText = 8; //个性化签名

int32 sex= 9; //性别

string address= 10; //注册地址

int32 experience= 11; //用户经验值

int32 fansCount= 12; //粉丝数量

int32 followCount= 13; //关注的用户数量

Position position= 14; //用户所在的位置

string deviceID = 15; //登陆所用的设备ID

int32 FollowType = 16; //关注（其他用户）的类型

string descName = 17; //备注名称

string v2ID = 18; //会议系统用户ID

string v2UserName = 19;//会议系统用户名

string v2Pwd = 20; //会议系统用户密码

int32 videoCount = 21; //用户发布的直播总量

string v2DeviceID1 = 22; //前置摄像头ID

string v2DeviceID2 = 23; //后置摄像头ID

}

message Position{ //位置类

double longitude = 1; //地图经度

double latitude = 2; //地图维度

int32 radius = 3; //查询半径,以米为单位

}

message Video{ //直播视频类

int32 id = 1; //直播的ID

string videoNum = 2; //直播的会议号

int32 userId = 3; //发起直播的用户ID

Position position= 4; //该直播所在的位置

int32 sum = 5; //当前观看该直播的总人数

int32 userCount = 6; //累计观看人次

int32 likeCount = 7; //当前的点赞数

float gratuityCount = 8; //当前直播的打赏总额

int32 degree = 9; //与该用户的相关度，排序依据

int32 resultType = 10; //结果类型：0搜索内容；1推荐内容

bool HLSMode = 11; //是否为HLS格式

string videoPwd = 12; //直播的密码

}

message Gratuity{ //打赏类

int32 giftType = 1; //打赏物的类型

int32 toUserID = 2; //打赏的对象

int32 fromUserID = 3; //发起打赏的用户

float amount = 4; //打赏金额

float sumOrRemain = 5; //累计打赏总额或者打赏后的剩余金额

Answer answer = 6; //枚举类。响应结果

int32 videoId = 7; //打赏行为发生的那个直播

enum Answer{ //枚举。响应结果

receive = 0; //收到信息

accept = 1; //接受

refuse = 2; //拒绝

cancel = 3; //取消

}

}

message Reward{ //悬赏类

int32 id = 1; //此次悬赏的数据库ID

int32 powerTime = 2; //悬赏的有效时间

int64 releaseTime = 3; //悬赏的开始时间

repeated Gift gift = 4; /悬赏的物品及数量，可以是多种

int32 fromUserID = 5; //发起悬赏用户的ID

int32 answerUserID = 6; //接受悬赏用户的ID

Operate operate = 7; //枚举类。操作类型

string desc = 8; //文字描述

Position position = 9; //位置信息

int32 pushSum= 10; //推送的用户数量

string appealDesc = 11; //申诉的内容

int64 appealTime = 12; //申诉的时间

enum Operate{ //枚举。操作类型

release = 0; //发布悬赏

cancle = 1; //取消悬赏

pay = 2; //支付

answer = 3; //接受

appeal = 4; //申诉

}

}

message Gift{ //礼物类

int32 giftType = 1; //类别

float amount = 2; //数量

float sumOrRemain = 3; //悬赏/打赏量，或者剩余量

}

message UserState{ //用户状态通报类

int32 userId = 1; //用户ID

int32 operateId = 2; //操作ID（悬赏或者其他ID）

State state = 3; //当前状态

string desc = 4; //说明（报告原由）

enum State{ //枚举。用户状态

online = 0; //在线

offline = 1; //离线

p2pVideo = 2; //视频通话中

MultiVideo = 3; //多人视频中

other = 4; //其他状态

}

}

message AssetLog{ //资金变动的日志类

int32 fromUserID = 1; //资金流出的用户ID

int32 toUserID = 2; //资金流入的用户ID

repeated Gift gift = 3; //（可重复）涉及的资产类

string desc = 4; //文本说明

OperateType operateType = 5; //枚举，查询类型

int64 operateTime = 6; //操作发生的时间

int32 soType = 7; //后台处理类型（支付、冻结、解冻）

int32 payType = 8; //支付类型（通道）

int64 payTime = 9; //支付时间

int32 videoId = 10; //所在的直播ID

int32 rewardId = 11; //所在的悬赏的ID

string appealDesc = 12; //申诉的内容

int64 appealTime = 13; //申诉发生的时间

enum OperateType{ //枚举。查询类型

recharge = 0; //充值

cash = 1; //提现

gratuity = 2; //打赏

reward = 3; //悬赏

all = 4; //所有

income = 5; //入账

payout = 6; //出账

}

}

message ServerBean{ //服务器数量变化的通知类

int32 serverId = 1; //zk的服务器编号,客户端无需处理

string addr\_zk = 2; //ZK服务器的IP:端口,客户端无需处理

string addr\_buffer = 3; //buffer服务器的IP:端口

string addr\_web = 4; //Web服务器的IP:端口

bool add = 5; //新增服务器还是删除服务器

}

## （二）具体接口说明

概要介绍：

每一个包至少要有一个packetType，声明是什么类型的包；信令（Iq）必须有id，method，声明是调用哪一个接口。必须有一个版本号，用来验证当前客户端的协议版本号是否过低；服务器返回的信令类型包必然有一个Result，用来声明客户端的请求是否响应成功，若失败给出原因；如果有数据如要返回，则用Data嵌套各种数据类返回给客户端。

3.0.0版Googlebuffer协议，不再传递字段的默认值（等于0的int类型等）。

## （三）心跳

协议说明：心跳包只有一个包类型，没有其他数据。

协议内容：packetType: beat

服务器返回：服务器返回心跳包。

其他说明：客户端需检测TCP链接闲置事件，如果超过10秒没有成功的收发动作，则需向服务器发送心跳包。如果连续超过三个心跳周期仍没有任何成功的收发动作，服务器会判定该客户端已经离线，强制关闭该TCP链接。

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.*newBuilder*();

packet.setPacketType(WebPackage.Packet.type.***beat***);

## （四）信令

说明：类型为iq，返回值中 packetType、from（如果有）、to（如果有）、id、method、operateType等信息保持不变；返回值的Data中加入Result信息来标示服务器处理结果。用Data中的sum来记录数据总量，用from来记录当前的起始位置，用to来记录当前的结束位置。用fromTime和toTime来记录数据的开始时间及截止时间（用于筛选数据）。

### 1.用户登录

协议说明：登录方式有三种，所必须的参数有所不同。其中手机号+短信验证码方式也可以用作注册（若该手机号尚未注册，则自动注册为用户）

1. 手机号+密码方式：必须参数为operateType（normal：普通方式）、phone（注册的手机号）、pwd（用户密码），选填参数deviceID（登录所用的设备ID）。

协议内容：

id: "2"

method: "login"

operateType: "normal"

data {

user {

phone: "15911004092"

pwd: "123456"

deviceID : "1232323242433"

}

}

version: "1.3.0"

B.手机号+短信验证码：必须参数为operateType（smscode：短信验证码方式）、Phone（注册的手机号）、pwd（用户密码），选填参数deviceID（登录所用的设备ID）。

协议内容：

id: "2"

method: "login"

operateType: "smscode"

data {

user {

phone: "15911004092"

pwd2orCode: "111111"

deviceID : "1232323242433"

}

}

version: "1.3.0"

C.匿名登录：必须参数为operateType（visitor：匿名方式）、deviceID（登录所用的设备ID）。

协议内容：

id: "2"

method: "login"

operateType: "visitor"

data {

user {

deviceID : "1232323242433"

}

}

version: "1.3.0"

服务器返回（以短息验证码方式登录为例）：id、method、operateType不变，result标记登录是否成功（若不成功，则用error来说明失败原因）。user返回登录用户的基本信息。serverBean返回当前可用的服务器列表（以备服务器宕机时使用）。

id: "2"

method: "login"

operateType: "smscode"

result {

result: true

}

data {

user {

id: 10

phone: "15911004092"

headurl: "\\upload\\user\_image\\image\_head1\\1472303197136.jpg"

v2UserName: "null"

v2Pwd: "null"

}

serverBean {

serverId: 1

addr\_buffer: "192.168.31.161:9997"

addr\_web: "192.168.31.161:9998"

add: true

}

}

version: "1.3.0"

服务器推送（登录成功后，自动推送内容）：系统推荐给该登录用户的直播信息。

from: "pushServer"

id: "0000"

method: "queryVideoList"

result {

result: true

}

data {

sum: 3

from: 1

to:3

video {

id: 1

videoNum: "123456"

userId: 1

position {

longitude: 39.970423

latitude: 116.43732

}

userCount: 12

degree: 13

}

video {

id: 2

videoNum: "123456"

userId: 1

position {

longitude: 39.966884

latitude: 116.322912

}

userCount: 21

degree: 13

}

video {

id: 3

videoNum: "123456"

userId: 1

position {

longitude: 39.890308

latitude: 116.33211

}

userCount: 232

degree: 13

}

}

其他说明：若用户不存在，服务器会直接注册该用户，然后正常返回信息。

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("2");

packet.setVersion(version);

packet.setMethod("login");

packet.setOperateType("smscode");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.User.Builder user = WebPackage.User.newBuilder();

user.setPhone("15911004092");

user.setPwd2OrCode("111111");

user.setDeviceID("");

data.addUser(user);

packet.setData(data);

### 2.用户登出

协议说明：登出方式有两种，直接登出与注销匿名登录。method均为logout，如果是直接登出，需增加operateType为logout（此时服务器会强行关闭该TCP链接）。

协议内容：

id: "2"

method: "logout"

operateType: "logout"

version: "1.3.0"

服务器返回：若是直接登出（增加operateType为logout），则无返回。

method: "logout"

result {

result: true

}

version: "1.3.0"

其他说明：如果是直接登出，需增加operateType为logout（此时服务器会强行关闭该TCP链接）。

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("logout");

### 3.获取直播列表

协议说明：method为queryVideoList。有两种查询方式（operateType），分别为map（地图经纬度）和label（地图标签）。目前使用的是map方式。data中的from、to分别为欲获取数据的起始位置（必须指定）。如果是要获取第一页的数据，则指定为第一页的起始位置。Position为欲查询的地图位置（三要素：经度、纬度、半径）。

协议内容：

id: "IQ111"

method: "queryVideoList"

operateType: "map"

data {

from: 1

to: 20

position {

longitude: 113.612288

latitude: 34.770904

radius: 5000

}

}

version: "1.3.0"

服务器返回：

id: "IQ111"

method: "queryVideoList"

operateType: "map"

result {

result: true

}

data {

sum: 2

from: 1

to: 2

video {

id: 1

videoNum: "123456"

userId: 1

position {

longitude: 113.612288

latitude: 34.770904

}

userCount: 12

degree: 13

}

video {

id: 2

videoNum: "123456"

userId: 1

position {

longitude: 113.612298

latitude: 34.770914

}

userCount: 21

degree: 13

}

}

version: "1.3.0"其他说明：如果查询到的结果集为0，则仅显示from：1，其余项不再显示（int型等于默认值0的不显示）。其他没有显示的直播属性sum（当前观看该直播的人数）、usercount（累计观看该直播的人数）、likeCount（当前的点赞数）、degree（与当前用的相关度）、resultType（搜索内容还是推荐内容）、HLSMode（是否为HLS模式）、videoPwd（直播密码）。

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("queryVideoList");

packet.setOperateType("map");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

data.setFrom(1);

data.setTo(20);

WebPackage.Position.Builder position = WebPackage.Position.newBuilder();

position.setLongitude(longitude);

position.setLatitude(latitude);

position.setRadius(5000);

data.addPosition(position);

packet.setData(data);

### 4.发布直播

协议说明：method为publicVideo。必须参数videoNum（会议号）、position（地图经纬度），选填参数videoPwd（观看密码）。

协议内容：

id: "IQ111"

method: "publicVideo"

data {

video {

videoNum: "123456789633"

position {

longitude: 113.612288

latitude: 34.770904

}

videoPwd: "1234"

}

}

version: "1.3.0"

服务器返回：data中嵌套的video内容不变，用normal来传递当前直播的ID。

id: "IQ111"

method: "publicVideo"

result {

result: true

}

data {

normal: "48"

video {

videoNum: "123456789633"

position {

longitude: 113.612288

latitude: 34.770904

}

videoPwd: "1234"

}

}

version: "1.3.0"

服务器推送：发布直播可以获得经验值，所以发布直播后会收到经验值变化的推送。Sum传递当前总的经验值，normal传递当前操作获得的经验值。

from: "pushServer"

id: "0000"

method: "experience"

operateType: "publicVideo"

result {

result: true

}

data {

normal: "2"

sum: 10

}

服务器推送（给其他用户）：增加嵌套的User来传递创建直播者的信息。

from: "pushServer"

id: "IQ111"

method: "publicVideo"

result {

result: true

}

data {

normal: "49"

user {

id: 10

phone: "15911004092"

headurl: "\\upload\\user\_image\\image\_head1\\1472303197136.jpg"

}

video {

videoNum: "123456789633"

position {

longitude: 113.612288

latitude: 34.770904

}

videoPwd: "1234"

}

}

version: "1.3.0"

其他说明：为尽可能的缩短处理路径，这里用data的normal的传递直播ID，不再修改原video内容；若此次发布直播是接受悬赏后发起的直播，则需在data中再加入Reward信息（悬赏ID）。

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("publicVideo");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.Video.Builder video = WebPackage.Video.newBuilder();

video.setVideoNum("123456789633");

WebPackage.Position.Builder position = WebPackage.Position.newBuilder();

position.setLongitude(longitude);

position.setLatitude(latitude);

video.setPosition(position);

video.setVideoPwd("1234");

data.addVideo(video);

// WebPackage.Reward.Builder reward = WebPackage.Reward.newBuilder();

// reward.setId(1);

// data.addReward(reward);

packet.setData(data);

### 5.关注/取消关注

协议说明：method为followUser。operateType为add（添加关注）、delete（取消关注）；必须参数User（ID必填，descName选填）。

协议内容：

id: "IQ111"

method: "followUser"

operateType: "add"

data {

user {

id: 10

descName: "92"

}

}

version: "1.3.0"

服务器返回：仅增加处理结果，其余信息不变。

id: "IQ111"

method: "followUser"

operateType: "add"

result {

result: true

}

data {

user {

id: 10

descName: "92"

}

}

version: "1.3.0"

服务器推送（给其他用户）：暂时没有推送。

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("followUser");

packet.setOperateType("add");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.User.Builder user = WebPackage.User.newBuilder();

user.setId(10);

user.setDescName("92");

data.addUser(user);

packet.setData(data);

### 6.获取短信验证码

协议说明：method为getSMScode。operateType为login（登陆用）等。

协议内容：

id: "IQ111"

method: "getSMScode"

operateType: "login"

data {

normal: "15811084491"

}

version: "1.3.0"

服务器返回：仅增加处理结果，其余信息不变。

id: "IQ111"

method: "getSMScode"

operateType: "login"

result {

result: true

}

data {

normal: "15811084491"

}

version: "1.3.0"

服务器推送（给其他用户）：暂时没有推送。

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("getSMScode");

packet.setOperateType("login");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

data.setNormal("15811084491");

packet.setData(data);

### 7.开始/结束观看视频

协议说明：method为watchVideo。operateType为enter（开始观看）、leave（结束观看）。

协议内容：

开始观看：必须参数video的ID，videoPwd（默认为0）。

from: "1"

id: "IQ111"

method: "watchVideo"

operateType: "enter"

data {

video {

id: 50

videoPwd: "1234"

}

}

version: "1.3.0"

服务器返回：data中嵌套的video内容不变，用user来传递当前直播中的用户（默认仅传递第一页的用户，总人数用sum传递）。

from: "1"

id: "IQ111"

method: "watchVideo"

operateType: "enter"

result {

result: true

}

data {

sum: 1

from: 1

to: 1

user {

id: 10

phone: "15911004092"

headurl: "\\upload\\user\_image\\image\_head1\\1472303197136.jpg"

experience: 14

position {

longitude: 113.612288

latitude: 34.770904

}

}

video {

id: 50

videoPwd: "1234"

}

}

version: "1.3.0"

服务器推送：发布直播可以获得经验值，所以发布直播后会收到经验值变化的推送。Sum传递当前总的经验值，normal传递当前操作获得的经验值。

from: "pushServer"

id: "0000"

method: "experience"

operateType: "watchVideo"

result {

result: true

}

data {

normal: "1"

sum: 35

}

服务器推送（给其他用户）：增加嵌套的User来传递观看直播者的信息。

from: "pushServer"

id: "IQ111"

method: "watchVideo"

operateType: "enter"

result {

result: true

}

data {

user {

id: 8

phone: "15911004090"

experience: 34

position {

}

}

video {

id: 50

videoPwd: "1234"

}

}

version: "1.3.0"

结束观看：必须参数为normal（用来传递直播的ID）。

from: "1"

id: "IQ111"

method: "watchVideo"

operateType: "leave"

data {

normal: "50"

}

version: "1.3.0"

服务器返回：data中嵌套的video内容不变，用result来传递处理结果。

from: "1"

id: "IQ111"

method: "watchVideo"

operateType: "leave"

result {

result: true

}

data {

normal: "50"

}

version: "1.3.0"

服务器推送（给其他用户）：增加嵌套的User来传递离开直播者的信息。

from: "pushServer"

id: "IQ111"

method: "watchVideo"

operateType: "leave"

result {

result: true

}

data {

normal: "50"

user {

id: 8

phone: "15911004090"

experience: 34

position {

}

}

}

version: "1.3.0"

其他说明：为尽可能的缩短处理路径，离开直播时，不再用video来传递直播ID，而直接用data的normal传递。

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("watchVideo");

packet.setFrom("1");

packet.setOperateType("enter");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.Video.Builder video = WebPackage.Video.newBuilder();

video.setId(50);

video.setVideoPwd("1234");

data.addVideo(video);

packet.setData(data);

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("watchVideo");

packet.setFrom("1");

packet.setOperateType("leave");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

data.setNormal("50");

packet.setData(data);

### 8.上报位置

协议说明：method为mapPosition。operateType为normal（正常上报）、report（用户接受悬赏后的行程记录等）。

协议内容：

普通上报：

id: "IQ111"

method: "mapPosition"

data {

position {

longitude: 113.612288

latitude: 34.770904

}

}

version: "1.3.0"

行程上报

to: "1"

id: "IQ111"

method: "mapPosition"

operateType: "report"

data {

position {

longitude: 113.612288

latitude: 34.770904

}

}

version: "1.3.0"

服务器返回：增加处理结果，增加当前位置的悬赏信息（没有就不增加），其余信息不变。

to: "1"

id: "IQ111"

method: "mapPosition"

result {

result: true

}

data {

position {

longitude: 113.612288

latitude: 34.770904

}

reward {

id: 20

powerTime: 30

releaseTime: 1473220508908

gift {

giftType: 1

amount: 58.88

}

fromUserID: 9

desc: "go to hosptial"

pushSum: 3

}

}

version: "1.3.0"

服务器推送（给其他用户）：行程上报的推送，增加from信息（上报者的用户ID）。

from:"3"

to: "1"

id: "IQ111"

method: "mapPosition"

result {

result: true

}

data {

position {

longitude: 113.612288

latitude: 34.770904

}

}

version: "1.3.0"

服务器推送（给其他用户）：若用户在直播中，则把位置信息推送给直播中的其他用户，增加from信息（pushServer）。

from:"pushServer"

to: "1"

id: "IQ111"

method: "mapPosition"

result {

result: true

}

data {

user{

phone:15811084400

position {

longitude: 113.612288

latitude: 34.770904

}

}

}

version: "1.3.0"

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("mapPosition");

// packet.setOperateType("report");

// packet.setTo("1");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.Position.Builder position = WebPackage.Position.newBuilder();

position.setLongitude(longitude);

position.setLatitude(latitude);

data.addPosition(position);

packet.setData(data);

### 9.查询用户信息

协议说明：method为queryUser。支持多用户同时查询，若查询的用户不存在，则返回信息中的name为“error”。若用户在线，则加上位置信息。

协议内容：

id: "IQ111"

method: "queryUser"

data {

user {

phone: "15811080000"

}

user {

phone: "15811080001"

}

user {

phone: "15811080002"

}

}

version: "1.3.0"

服务器返回：增加处理结果，增加所查询的用户信息（id,phone,name,signText,headurl,sex,experience,address,Position），其余信息不变。

id: "IQ111"

method: "queryUser"

result {

result: true

}

data {

user {

phone: "15811080000"

name: "error"

}

user {

phone: "15811080001"

name: "error"

}

user {

phone: "15811080002"

name: "error"

}

}

version: "1.3.0"

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("queryUser");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.User.Builder user = WebPackage.User.newBuilder();

user.setPhone("15811080000");

data.addUser(user);

user = WebPackage.User.newBuilder();

user.setPhone("15811080001");

data.addUser(user);

user = WebPackage.User.newBuilder();

user.setPhone("15811080002");

data.addUser(user);

packet.setData(data);

### 10.修改好友

协议说明：method为editFollow。必须参数为user的id及备注名称descName。

协议内容：

id: "IQ111"

method: "editFollow"

data {

user {

id: 8

descName: "25Desc"

}

}

version: "1.3.0"

服务器返回：增加处理结果（示例结果为数据库操作错误。触发原因为不是好友）。

id: "IQ111"

method: "editFollow"

result {

error: "6:\346\225\260\346\215\256\345\272\223\351\224\231\350\257\257"

}

data {

user {

id: 8

descName: "25Desc"

}

}

version: "1.3.0"

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("editFollow");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.User.Builder user = WebPackage.User.newBuilder();

user.setId(8);

user.setDescName("25Desc");

data.addUser(user);

packet.setData(data);

### 11.修改自身信息

协议说明：method为editUser。支持多字段同时修改，也可以仅修改一个字段。

协议内容：

id: "IQ111"

method: "editUser"

data {

user {

name: "sunny"

realName: "wang\*\*"

sex: 1

address: "\345\214\227\344\272\254\345\270\202\346\265\267\346\267\200\345\214\272\347\237\245\346\230\245\350\267\257"

}

}

version: "1.3.0"

服务器返回：增加处理结果，其余信息不变。

id: "IQ111"

method: "editUser"

result {

result: true

}

data {

user {

name: "sunny"

realName: "wang\*\*"

sex: 1

address: "\345\214\227\344\272\254\345\270\202\346\265\267\346\267\200\345\214\272\347\237\245\346\230\245\350\267\257"

}

}

version: "1.3.0"

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("editUser");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.User.Builder user = WebPackage.User.newBuilder();

user.setRealName("wang\*\*");

user.setName("sunny");

user.setAddress("北京市海淀区知春路");

user.setSex(1);

data.addUser(user);

packet.setData(data);

### 12.获取粉丝（关注我的）

协议说明：method为getFansList。必选参数为数据范围，一般用1至20来加载第一页数据。

协议内容：

id: "IQ111"

method: "getFansList"

data {

from: 1

to: 20

}

version: "1.3.0"

服务器返回：增加处理结果，增加所查询的粉丝信息，其余信息不变。示例信息为无粉丝时的情景（为0的int型字段不显示），有数据时，用多个User类型来传递粉丝信息，返回信息包括（id,phone,nickName,headUrl,signText,sex,address,experience,followType）。

id: "IQ111"

method: "getFansList"

result {

result: true

}

data {

from: 1

}

version: "1.3.0"

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("getFansList");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

data.setFrom(1);

data.setTo(20);

packet.setData(data);

### 13.获取关注列表（我关注的）

协议说明：method为getFollowList。必选参数为数据范围，一般用1至20来加载第一页数据。

协议内容：

id: "IQ111"

method: "getFollowList"

data {

from: 1

to: 20

}

version: "1.3.0"

服务器返回：增加处理结果，增加所查询的关注用户信息，其余信息不变。示例信息为无关注用户时的情景（为0的int型字段不显示），有数据时，用多个User类型来传递关注用户的信息，返回信息包括（id,phone,nickName,headUrl,signText,sex,address,experience,followType）。

id: "IQ111"

method: "getFollowList"

result {

result: true

}

data {

from: 1

}

version: "1.3.0"

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("getFollowList");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

data.setFrom(1);

data.setTo(20);

packet.setData(data);

### 14.为直播点赞

协议说明：method为likeVideo。必选参数为normal（点赞直播的id）。

协议内容：

id: "IQ111"

method: "likeVideo"

data {

normal: "1"

}

version: "1.3.0"

服务器返回：增加处理结果，增加video来传递所点赞直播当前的点赞数（likeCount）。

id: "IQ111"

method: "likeVideo"

result {

result: true

}

data {

video {

id: 1

likeCount: 1

}

}

version: "1.3.0"

服务器推送（给直播中的其他用户）：

id: "IQ111"

method: "likeVideo"

result {

result: true

}

data {

video {

id: 1

likeCount: 1

}

}

version: "1.3.0"

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("likeVideo");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

data.setNormal("1");

packet.setData(data);

### 15.打赏/打赏通话

协议说明：method为gratuity。operateType为normal（直接打赏）或者askTalk（打赏通话）。

其他说明： 超时事件由发和收两个客户端各自计算。收端用户15秒未处理视为超时，发端用户20秒未收到接收端的信令回复视为超时；

有效等待时间内，发起端可以取消此次打赏通话；同一操作周期内，同一打赏对象，用户仅可打赏通话一次；被打赏对象有回复（拒绝、接受、超时）后，可以继续发起打赏通话。

协议内容：

普通打赏：必选参数为gratuity（对方ID toUserID，打赏金额amount）。可选参数为videoId（如果是在直播中打赏，则推送给直播中的其他成员）。若对方不在线则打赏失败。

id: "IQ111"

method: "gratuity"

operateType: "normal"

data {

gratuity {

toUserID: 2

amount: 40.0

}

}

version: "1.3.0"

服务器返回：增加处理结果，增加打赏后，用户剩余的资产信息。

id: "IQ111"

method: "gratuity"

operateType: "normal"

result {

result: true

}

data {

gratuity {

toUserID: 2

amount: 40.0

sumOrRemain: 49601.12

}

}

version: "1.3.0"

服务器推送（经验值变化）：normal为当前的增加值，sum为变化后的经验值。

from: "pushServer"

id: "0000"

method: "experience"

operateType: "putGratuity"

result {

result: true

}

data {

normal: "40"

sum: 50

}

服务器推送（给收到打赏的用户）：

from: "pushServer"

id: "IQ111"

method: "gratuity"

operateType: "normal"

result {

result: true

}

data {

gratuity {

toUserID: 2

fromUserID: 1

amount: 40.0

}

}

version: "1.3.0"

打赏通话：

协议内容：

id: "IQ111"

method: "gratuity"

operateType: "askTalk"

data {

gratuity {

toUserID: 3

amount: 40.0

videoId: 222

}

}

version: "1.3.0"

服务器返回：仅增加处理结果。

id: "IQ111"

method: "gratuity"

operateType: "askTalk"

result {

result: true

}

data {

gratuity {

toUserID: 3

amount: 40.0

videoId: 222

}

}

version: "1.3.0"

服务器推送（给打赏通话的另一方）：

from: "pushServer"

id: "IQ111"

method: "gratuity"

operateType: "askTalk"

result {

result: true

}

data {

gratuity {

toUserID: 3

fromUserID: 1

amount: 40.0

videoId: 222

}

gratuity {

toUserID: 3

amount: 40.0

videoId: 222

}

}

version: "1.3.0"

打赏通话的另一方，收到服务器推送后，需自动回复信令，如下（receive = 0不显示）：

id: "IQ111"

method: "gratuity"

operateType: "answerTalk"

data {

gratuity {

toUserID: 3

fromUserID: 1

videoId: 222

}

}

version: "1.3.0"

服务器返回：

id: "IQ111"

method: "gratuity"

operateType: "answerTalk"

result {

result: true

}

data {

gratuity {

toUserID: 3

fromUserID: 1

videoId: 222

}

}

version: "1.3.0"

服务器推送（给发起者）：

from: "pushServer"

id: "IQ111"

method: "gratuity"

operateType: "answerTalk"

result {

result: true

}

data {

gratuity {

toUserID: 3

fromUserID: 1

videoId: 222

}

}

version: "1.3.0"

对方接受：

id: "IQ111"

method: "gratuity"

operateType: "answerTalk"

data {

gratuity {

toUserID: 3

fromUserID: 1

answer: accept

videoId: 222

}

}

version: "1.3.0"

服务器返回：

id: "IQ111"

method: "gratuity"

operateType: "answerTalk"

result {

result: true

}

data {

gratuity {

toUserID: 3

fromUserID: 1

answer: accept

videoId: 222

}

}

version: "1.3.0"

服务器推送（给发起打赏通话者）：

from: "pushServer"

id: "IQ111"

method: "gratuity"

operateType: "answerTalk"

result {

result: true

}

data {

gratuity {

toUserID: 3

fromUserID: 1

videoId: 222

}

}

version: "1.3.0"

取消/拒绝过程类似，不再重复。

构造示例（java版）：

发送请求

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("gratuity");

packet.setOperateType("askTalk");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.Gratuity.Builder gratuity = WebPackage.Gratuity.newBuilder();

gratuity.setAmount(40.00F);

gratuity.setToUserID(3);

gratuity.setVideoId(222);

data.addGratuity(gratuity);

packet.setData(data);

对方客户端自动回复收到请求

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("gratuity");

packet.setOperateType("answerTalk");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.Gratuity.Builder gratuity = WebPackage.Gratuity.newBuilder();

gratuity.setFromUserID(1);

gratuity.setToUserID(3);

gratuity.setVideoId(222);

gratuity.setAnswer(WebPackage.Gratuity.Answer.receive);

data.addGratuity(gratuity);

packet.setData(data);

对方拒绝了本次打赏通话

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("gratuity");

packet.setOperateType("answerTalk");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.Gratuity.Builder gratuity = WebPackage.Gratuity.newBuilder();

gratuity.setFromUserID(1);

gratuity.setToUserID(3);

gratuity.setVideoId(222);

gratuity.setAnswer(WebPackage.Gratuity.Answer.refuse);

data.addGratuity(gratuity);

packet.setData(data);

取消打赏通话（在对方接受前，发起方可取消）

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("gratuity");

packet.setOperateType("answerTalk");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.Gratuity.Builder gratuity = WebPackage.Gratuity.newBuilder();

gratuity.setFromUserID(1);

gratuity.setToUserID(3);

gratuity.setVideoId(222);

gratuity.setAnswer(WebPackage.Gratuity.Answer.cancel);

data.addGratuity(gratuity);

packet.setData(data);

对方接受打赏通话

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("gratuity");

packet.setOperateType("answerTalk");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.Gratuity.Builder gratuity = WebPackage.Gratuity.newBuilder();

gratuity.setFromUserID(1);

gratuity.setToUserID(3);

gratuity.setVideoId(222);

gratuity.setAnswer(WebPackage.Gratuity.Answer.accept);

data.addGratuity(gratuity);

packet.setData(data);

### 16.社交分享

协议说明：method为shareVideo。必选参数为operateType（分享的社交媒体类别，具体待定），normal（所分享的直播ID）。

协议内容：

id: "IQ111"

method: "shareVideo"

operateType: "1"

data {

normal: "1"

}

version: "1.3.0"

服务器返回：增加处理结果，其余信息不变。

id: "IQ111"

method: "shareVideo"

operateType: "1"

result {

result: true

}

data {

normal: "1"

}

version: "1.3.0"

服务器推送：经验值变动

from: "pushServer"

id: "0000"

method: "experience"

operateType: "shareVideo"

result {

result: true

}

data {

normal: "2"

sum: 10

}

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("getFollowList");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

data.setFrom(1);

data.setTo(20);

packet.setData(data);

### 17.获取直播中的用户

协议说明：method为queryUserInVideo。必选参数为normal（直播的id）、数据的起始位置。

协议内容：

id: "IQ111"

method: "getUserInVideo"

data {

normal: "1"

from: 1

to: 20

}

version: "1.3.0"

服务器返回：增加处理结果，增加user信息来传递所查询到的用户信息（id,phone,name,headurl,sex,experience,position）。示例结果为直播中没有用户的情景。

id: "IQ111"

method: "getUserInVideo"

result {

result: true

}

data {

normal: "1"

from: 1

}

version: "1.3.0"

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("getUserInVideo");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

data.setNormal("1");

data.setFrom(1);

data.setTo(20);

packet.setData(data);

### 18.悬赏

协议说明：method为reward。必选参数为资产类型（1为货币资产）和资产数量。operateType为release（发布）、cancel（取消）、pay（支付）、answer（应答）及appeal（申诉）。默认悬赏的有效期是30秒，超出有效期没有人应答则自动失效。

协议内容：

发起悬赏：

id: "IQ111"

method: "reward"

operateType: "release"

data {

reward {

gift {

giftType: 1

amount: 58.88

}

desc: "go to hosptial"

position {

longitude: 113.612288

latitude: 34.770904

radius: 500

}

}

}

version: "1.3.0"

服务器返回：增加处理结果及当前推送的人数（若为0则不显示）。

id: "IQ111"

method: "reward"

operateType: "release"

result {

result: true

}

data {

reward {

id: 21

gift {

giftType: 1

amount: 58.88

}

desc: "go to hosptial"

position {

longitude: 113.612288

latitude: 34.770904

radius: 500

}

pushSum: 1

}

}

version: "1.3.0"

服务器推送（给其他地图范围内的在线用户）：

from: "pushServer"

id: "IQ111"

method: "reward"

operateType: "release"

result {

result: true

}

data {

reward {

id: 22

gift {

giftType: 1

amount: 58.88

}

desc: "go to hosptial"

position {

longitude: 113.612288

latitude: 34.770904

radius: 500

}

pushSum: 1

}

}

version: "1.3.0"

有用户接受了此次悬赏：

id: "IQ111"

method: "reward"

operateType: "answer"

data {

reward {

id: 21

gift {

giftType: 1

amount: 58.88

}

desc: "go to hosptial"

position {

longitude: 113.612288

latitude: 34.770904

radius: 500

}

}

}

version: "1.3.0"

服务器返回：

id: "IQ111"

method: "reward"

operateType: "answer"

result {

result: true

}

data {

reward {

id: 21

gift {

giftType: 1

amount: 58.88

}

desc: "go to hosptial"

position {

longitude: 113.612288

latitude: 34.770904

radius: 500

}

}

}

version: "1.3.0"

服务器推送（给其他用户：发起者及其他收到此次悬赏推送的用户）：

from: "pushServer"

id: "IQ111"

method: "reward"

operateType: "answer"

result {

result: true

}

data {

reward {

id: 21

gift {

giftType: 1

amount: 58.88

}

desc: "go to hosptial"

position {

longitude: 113.612288

latitude: 34.770904

radius: 500

}

}

}

其他过程略。

构造示例（java版）：

发布悬赏

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("reward");

packet.setOperateType("release");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.Reward.Builder reward = WebPackage.Reward.newBuilder();

WebPackage.Gift.Builder gift = WebPackage.Gift.newBuilder();

gift.setAmount(58.88F);

gift.setGiftType(1);

reward.addGift(gift);

WebPackage.Position.Builder position = WebPackage.Position.newBuilder();

position.setLongitude(longitude);

position.setLatitude(latitude);

position.setRadius(500);

reward.setPosition(position);

reward.setDesc("go to hosptial");

data.addReward(reward);

packet.setData(data);

取消悬赏

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("reward");

packet.setOperateType("cancle");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.Reward.Builder reward = WebPackage.Reward.newBuilder();

WebPackage.Gift.Builder gift = WebPackage.Gift.newBuilder();

gift.setAmount(88.88F);

gift.setGiftType(1);

reward.addGift(gift);

WebPackage.Position.Builder position = WebPackage.Position.newBuilder();

position.setLongitude(longitude);

position.setLatitude(latitude);

position.setRadius(500);

reward.setPosition(position);

reward.setId(21);

reward.setDesc("go to hosptial");

data.addReward(reward);

packet.setData(data);

支付悬赏

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("reward");

packet.setOperateType("pay");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.Reward.Builder reward = WebPackage.Reward.newBuilder();

WebPackage.Gift.Builder gift = WebPackage.Gift.newBuilder();

gift.setAmount(58.88F);

gift.setGiftType(1);

reward.addGift(gift);

WebPackage.Position.Builder position = WebPackage.Position.newBuilder();

position.setLongitude(longitude);

position.setLatitude(latitude);

position.setRadius(500);

reward.setPosition(position);

reward.setId(21);

reward.setDesc("go to hosptial");

data.addReward(reward);

packet.setData(data);

应答悬赏

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("reward");

packet.setOperateType("answer");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.Reward.Builder reward = WebPackage.Reward.newBuilder();

WebPackage.Gift.Builder gift = WebPackage.Gift.newBuilder();

gift.setAmount(58.88F);

gift.setGiftType(1);

reward.addGift(gift);

WebPackage.Position.Builder position = WebPackage.Position.newBuilder();

position.setLongitude(longitude);

position.setLatitude(latitude);

position.setRadius(500);

reward.setPosition(position);

reward.setId(22);

reward.setDesc("go to hosptial");

data.addReward(reward);

packet.setData(data);

发起申诉

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("reward");

packet.setOperateType("appeal");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.Reward.Builder reward = WebPackage.Reward.newBuilder();

reward.setId(21);

reward.setAppealDesc("uuufuBUBUBUBUBUB");

data.addReward(reward);

packet.setData(data);

### 19.获取资金变动日志

协议说明：method为assetLog。必选参数为数据范围（一般用1至20来加载第一页数据），assetLog的operateType（筛选的日志类型，具体见协议文件）。

协议内容：

id: "IQ111"

method: "assetLog"

data {

from: 1

to: 20

assetLog {

operateType: income

}

}

version: "1.3.0"

服务器返回：增加处理结果，第一个查询的assetLog类不变。增加assetLog 来传递所查询到的日志信息。

id: "IQ111"

method: "assetLog"

result {

result: true

}

data {

sum: 1

from: 1

to: 1

assetLog {

operateType: income

}

assetLog {

fromUserID: 2

toUserID: 1

gift {

giftType: 1

amount: 58.0

}

operateType: reward

operateTime: 1467894001000

soType: 1

}

}

version: "1.3.0"

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("assetLog");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

data.setFrom(1);

data.setTo(20);

WebPackage.AssetLog.Builder assetLog = WebPackage.AssetLog.newBuilder();

assetLog.setOperateType(WebPackage.AssetLog.OperateType.income);

data.addAssetLog(assetLog);

packet.setData(data);

### 20.获取当前服务器列表

协议说明：method为serverList。返回当前所有的业务服务器信息，无分页。

id: "IQ111"

method: "serverList"

version: "1.3.0"

服务器返回：增加处理结果，增加serverBean 来传递查询结果。

id: "IQ111"

method: "serverList"

result {

result: true

}

data {

serverBean {

serverId: 1

addr\_buffer: "192.168.31.161:9997"

addr\_web: "192.168.31.161:9998"

add: true

}

}

version: "1.3.0"

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("serverList");

### 21.设置/获取媒体设备号

协议说明：method为V2DeviceID。operateType为set（设置设备号）、get（获取设备号）。

id: "IQ111"

method: "V2DeviceID"

operateType: "set"

data {

user {

v2DeviceID1: "124@#d111"

v2DeviceID2: "124@#d222"

}

}

version: "1.3.0"

服务器返回：增加处理结果。

id: "IQ111"

method: "V2DeviceID"

operateType: "set"

result {

result: true

}

data {

user {

v2DeviceID1: "124@#d111"

v2DeviceID2: "124@#d222"

}

}

version: "1.3.0"

构造示例（java版）：

WebPackage.Packet.Builder packet = WebPackage.Packet.newBuilder();

packet.setPacketType(WebPackage.Packet.type.iq);

packet.setId("IQ111");

packet.setVersion(version);

packet.setMethod("V2DeviceID");

packet.setOperateType("set");

WebPackage.Data.Builder data = WebPackage.Data.newBuilder();

WebPackage.User.Builder user = WebPackage.User.newBuilder();

user.setV2DeviceID1("124@#d111");

user.setV2DeviceID2("124@#d222");

data.addUser(user);

packet.setData(data);

# 二、Http服务

（暂定，优化后尽量不用netty的http客户端服务）

## （一）上传用户头像

在http请求头中增加"info1"、"info2"两个参数，分别用来传递用户ID及手机号，如果两个参数匹配且当前用户在线，则允许上传。

客户端上传一个大尺寸的头像，服务器收到后，再裁出两张小尺寸的图片（240\*240、60\*60）统一存储。然后将头像的保存地址传送给客户端（如果客户端离线，则无法完成推送）。

注：推送的文件保存路径示例 \upload\user\_image\image\_head1\1472303197136.jpg。前面为相对路径，后面的数字为文件名。下载头像时，需要指定头型类型方可正常下载。下载时需指明的文件名示例：

1472303197136\_l.jpg 大尺寸

1472303197136\_m.jpg 中间尺寸

1472303197136\_s.jpg 小尺寸

## （二）下载用户头像

请求路径：Web服务器IP+“：”+端口+图片路径+图片名称+“\_”+图片尺寸。

例：http://127.0.0.1:9998/upload/user\_image/image\_head1/1472303197136\_l.jpg

## （三）上传客户端日志

如果服务器识别不是头像上传，则作为日志上传处理。