

收入数据服务器传输接入 1.0

最后修正：2013-10-30

目录

一、综述	2
1.用途	2
2.限制	2
二、接口说明	2
1.调用方式	2
2.参数说明	3
3.接口返回信息	4
4.调用注意事项	5
三、FAQ	5
技术支持联络方式	6

一、 综述

1. 用途

允许游戏开发者直接从服务器侧将收入类数据传输至 TalkingGame，以使总体数据更加精准可靠，服务器收入接口可帮助具体解决以下问题：

- 对于只在服务器存储了哪笔订单充值成功的游戏，可以简化开发，无需再去通知客户端订单充值成功；
- 解决单纯从客户端发送数据由于网络原因造成的数据丢量问题；

- 可以传入一定量的历史数据，会以发送数据中的时间戳计算重新计算历史日期收入数据量；但注意最早只能导入应用创建日期之后的收入事件。

2. 限制

出于安全性考虑，我们会对传入数据进行 IP 校验，只有从可信的 IP 地址传来的数据才可游戏处理。

如果您需要使用服务器收入接口，请向 TalkingGame 技术支持人员申请接口使用权，提交您的服务器 IP，即可进行调用。

注意：如果您的服务器使用云服务或没有固定的 IP，也请联络我们技术支持人员寻求解决办法。

二、接口说明

1. 调用方式

接口类型	HTTP
编码和格式	UTF-8 编码，数据请使用 json 格式
支持方式	单条传入、批量传入均可。 批量传入时一次最多 1000 条。
调用方法	以 HTTP 的 post 方式，请求接口 URL 传入数据。 数据需要通过 gzip 方式压缩放在 HTTP 的 body 中；您在 TalkingGame 获得的 APP ID 放入请求的 URL 中。

- **URL 格式：**

`http://api.talkinggame.com/api/charge/appId`

其中，AppID 是您在 TalkingGame 中创建游戏时获得的 AppID，用来唯一标识您的游戏。与您客户端集成时使用的 AppID 保持一致。

- **调用示例：**以 post 方式访问地址

`http://api.talkinggame.com/api/charge/1A9214AC9ED73185****ED026DA8E7BA`

Body 中传入数据；

数据为 json 格式的数组，经过 gzip 方式压缩。

```
[[
  {
    "msgID": "1001",
    "gameVersion": "2.0",
    "OS": "android",
    "accountID": "xxx",
    "level": 15,
    "gameServer": "金戈铁马",
    "orderID": "xxx",
    "iapID": "充值包类型",
    "currencyAmount": 99,
    "currencyType": "CNY",
    "virtualCurrencyAmount": 990,
    "paymentType": "支付宝",
    "status": "request",
    "chargeTime": 1374732001321,
    "mission": "新手任务 1"
  },
  {
    "msgID": "1002",
    "gameVersion": "2.0",
    "OS": "android",
    "accountID": "6411b18c-5394-44f1-b826-2e12589619e8",
    "level": 15,
    "gameServer": "金戈铁马",
    "orderID": "eff88347-b3e6-45ba-9216-51ae767c1d5f",
    "iapID": "充值包类型",
    "currencyAmount": 99,
    "currencyType": "CNY",
    "virtualCurrencyAmount": 990,
    "paymentType": "支付宝",
    "status": "success",
    "chargeTime": 1374732001321,
    "mission": "新手任务 1"
  }
]]
```

2. 参数说明 (请务必注意：参数字段大小写敏感，且参数的值严格按照要求填写，否则会造成数据统计偏差)

参数字段	类型	是否必填	格式要求	描述
msgID	String	必须	64 字符内	该条数据的 ID，由开发者自行定义； 唯一标识您发来的一条数据。 如果接口对某条数据反馈错误代码，可帮助您找出发送数据的问题，并在修改后对该条数据进行重发。
status	String	必须	request success	充值状态。标识这是一次充值请求，还是已经成功完成支付。 玩家请求某充值包时填写 request； 确认玩家完成支付时候填写 success； 收入数据只按 success 的数据计数。
OS	String	必须	ios android	玩家此次充值时所用的设备系统平台。 只支持格式要求限定的字符串。
accountID	String	必须	64 字符内	玩家的账户 ID； 与客户端调用 SDK 的 setAccount 需传入玩家的 accountId 类似。
orderID	String	必须	200 字符内	订单 ID，唯一标识一次交易。 注意： 1、如果您同时在客户端和服务端都有发送收入数据，请确保两边发送的 orderID 定义方式是一致的，否则可能造成数据多

				算； 2、同 orderID 的充值成功如多次收到，只记录最早一条成功的数据，重复数据不计数。
currencyAmount	double	必须		充值金额； 现金金额或现金等价物的额度。
currencyType	String	必须		充值货币类型； 请使用 ISO 4217 中规范的 3 位字母代码标记货币类型。 点击查看参考 例：人民币 CNY；美元 USD；欧元 EUR。 如果传入了 IOS 标准中不存在的类型，则会转成 CNY 的人民币类型。
virtualCurrencyAmount	double	必须		充值获得的虚拟币额度。
chargeTime	Long	建议		玩家充值发生的时间(13 位的毫秒数，例如:1390356610000)。 如果您未传入时间，则根据接口接收到数据的时间来记录时间。 注意：如果批量调用接口或传入历史数据，请务必填写时间，否则数据会集中计算到同一时期。
iapID	String	否	64 字符内	玩家购买的充值包类型。 如：6000 元宝大宝箱； 当您未传入此字段时，平台会根据货币金额+虚拟币数量来命名一个充值包。 例如：人民币 15 元 600 虚拟币包
paymentType	String	否	16 字符内	支付方式。 如：支付宝、苹果 iap、银联支付、爱贝支付聚合等。
gameServer	String	建议	16 字符内	玩家充值的区服。 如未填入，则数据计入默认的

				未知区服。
gameVersion	String	否	如： 1.0.0	充值时游戏客户端的版本号。 如果未传送此字段，充值会记入到该 accountID 近期使用的客户端版本上。
level	Int	建议	如：22	玩家充值时的等级。 如果您未传此字段，平台会按照对应 accountID 在发生充值时的最近期等级填充一个等级。
mission	string	否		玩家充值时所在的关卡或任务。亦可传入一个玩家打到的最高关卡。 如果不传，则不会记录充值关卡数据。

3. 接口返回信息

每次调用接口时您都可以在 HTTP response 的内容中得到详细的响应结果，response 内容同样是采用 UTF-8 编码，json 格式并以 gzip 方式压缩。

response 内容中包含两种响应码，整体数据包是否有效受理，数据包中每条数据消息是否成功受理。

1、对单次请求中的数据包的响应码：

Code	描述
100	成功——整体数据包成功接收，数据包验证通过。
101	有误——发送数据的 IP 被允许。
102	有误——请求 url 中携带的 appId 在平台中不存在。
103	有误——数据不是 json 格式。
104	有误——数据没有经过 gzip 压缩，解包失败。
105	服务器内部错误
106	有误——数据包中数量条数超过 1000 条限制。

2、对数据包中每条充值数据的响应码：

Code	描述
1	有效——数据验证通过。

2	错误——必须填写的字段没有完全填写。
3	错误——字段格式有误，无法处理。

➤ 返回信息示例：

例 1：某次的请求，数据包整体正确。数据包中包含了 3 条数据，其中 1 条数据有必填字段为空的情况，1 条正确，1 条数据字段类型有错误。

```
{
  "code": 100,
  "msg": "success",
  "dataStatus": [
    {
      "msgId": "101",
      "code": 1,
      "msg": "success"
    },
    {
      "msgId": "102",
      "code": 2,
      "msg": "Field [userID] is required, cannot be empty."
    },
    {
      "msgId": "103",
      "code": 3,
      "msg": "Field [currencyAmount] type is wrong, this type should be [Double], current value is: [abc]"
    }
  ]
}
```

4. 调用注意事项

1、您需要申请了服务器接口的使用权，并提供了您的服务器 IP 地址，方

可成功调用；

- 2、**确认 accountID 和 orderID 与客户端调用时候的策略是一致的，否则会影响数据准确性；**
- 3、**确保调用时必须填入的字段都已经携带，否则数据不会受理；**
- 4、**注意 body 中的数据包需要经过 gzip 方式压缩；**
- 5、**注意批量传数据时，一次请求做多可以发 1000 条；**
- 6、**注意只有 success 状态的数据才计算到收入数据中；**
- 7、**留意每次请求的返回码，对未成功受理的数据修正后做重发。**
- 8、**注意服务器接口只接受时间戳在平台中创建游戏日期之后发生的充值数据，过早的历史数据请勿发送，不会有效处理。**

三、FAQ

1. 客户端不做任何集成，我只用服务器收入接口是否可以？

答：不可以。收入数据只是整套游戏数据中的一部分，如果客户端未接入，设备量和用户帐户没有处理情况下，收入也无法正常计算；

但是，您可以在客户端中省略掉收入部分的集成，收入数据的传输上只使用服务器接口。

2. 在客户端中进行了收入部分的集成，还需要用服务器收入接口吗？

答：单纯从客户端收集收入数据无法保证绝对的准确，如果您觉得偏离度可以接收可不进行服务器的对接，如果对精度有更高要求则建议两侧都发送数据。

3. 如果我在客户端和服务器接口都发送收入数据，是否会造成收入偏高呢？

答：不会。我们会根据订单 ID (orderID) 来确认每一条收入数据的有效性，如果重复发送了 orderID 相同的收入数据，不会重复计数，不影响正常数据。

4. 收入服务器接口中有两种状态，我是否可以只发送成功状态的数据呢？

答：可以。只发送成功的充值数据可以有效的记录收入数据。

但是，请注意，是否发送每笔订单的请求数据，会影响到一项指标“支付成功率”的计算，如果不发送请求，则支付成功率无法查阅。

5. 如果调用服务器收入接口时只填入了必须字段，其他不填会有什么影响呢？

答：收入总体数据不会有所影响。付费率、ARPU 值等宏观数据也不会有影响。

但是在您对收入数据进行筛选和展开维度分析时无法得到有效的结果，不能了解到获得收入的原因。

四、附录

Demo(Java):

```
package com.tendcloud.game.client;

import java.io.ByteArrayOutputStream;
import java.io.DataOutputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.zip.GZIPInputStream;
import java.util.zip.GZIPOutputStream;
import com.fasterxml.jackson.core.JsonProcessingException;

/**
 * 收入接口调用示例代码（仅供参考）
 */
```

```
* @author whl
*
*/
public class JustDemo {

    public static void main(String[] args) throws JsonProcessingException {

        String jsonData =
        "[{\"msgID\":\"1001\",\"accountID\":\"whl\",\"gameVersion\":\"1.0\",\"OS\":\"ios\",\"level\":"
        + "21\",\"gameServer\":\"金戈铁马\",\"orderID\":\"e46bd2bb-89b6-4d52-860e-07d1b7b9ce95\","
        + "\"iapID\":\"6000 元宝大宝箱\",\"currencyAmount\":\"99.01\",\"currencyType\":\"CNY\","
        + "\"virtualCurrencyAmount\":\"990.01\",\"paymentType\":\"支付宝\","
        + "\"status\":\"success\",\"chargeTime\":\"1382687435864\",\"mission\":\"新手任务 1\"}]]";

        System.out.println("post data : "+jsonData);

        //压缩字符串数据为 gzip 字节流

        byte[] dataByte = gzip(jsonData);

        HttpClient clinet = new HttpClient("api.talkinggame.com", "80",
        "/api/charge/8AE2354AFA1E879EC048408A44FD4C65");

        System.out.println("result data : "+clinet.doPost(dataByte));

    }

    /**
     * 將字符串压缩为 gzip 流
     * @param content
     * @return
     */
    private static byte[] gzip(String content) {

        ByteArrayOutputStream baos = null;

        GZIPOutputStream out = null;

        byte[] ret = null;

        try {
```

```
        baos = new ByteArrayOutputStream();

        out = new GZIPOutputStream(baos);

        out.write(content.getBytes());

        out.close();

        baos.close();

        ret = baos.toByteArray();

    } catch (FileNotFoundException e) {

        e.printStackTrace();

    } catch (IOException e) {

        e.printStackTrace();

    } finally {

        if(out != null){

            try {

                baos.close();

            } catch (IOException e) {

                e.printStackTrace();

            }

        }

        if(out != null){

            try {

                out.close();

            } catch (IOException e) {

                e.printStackTrace();

            }

        }

    }

    return ret;
```

```
}

public static class HttpClient{

    HttpURLConnection _HttpURLConnection = null;

    URL url = null;

    private String DEFAULT_PROTOCOL = "http";

    private String SLASH = "/";

    private String COLON = ":";

    public String DEFAULT_NET_ERROR = "NetError";

    public String POST = "POST";

    public String doPost(byte[] Message) {

        String result = "";

        try {

            _HttpURLConnection = (HttpURLConnection) url.openConnection();

            _HttpURLConnection.setRequestMethod(POST);

            _HttpURLConnection.setDoOutput(true);

            _HttpURLConnection.setRequestProperty("Content-Type",

                "application/msgpack");

            _HttpURLConnection.setRequestProperty("Content-Length",

                String.valueOf(Message.length));

            DataOutputStream ds = new DataOutputStream(

                _HttpURLConnection.getOutputStream());

            ds.write(Message);

            ds.flush();

            ds.close();

            result = _gzipStream2Str(_HttpURLConnection.getInputStream());

            _HttpURLConnection.disconnect();

        }

    }

}
```

```
        } catch (Exception e) {

            _URLConnection.disconnect();

            e.printStackTrace();

        }

        return result;

    }

    private String _gzipStream2Str(InputStream inputStream) throws IOException{

        GZIPInputStream gzipInputStream = new GZIPInputStream(inputStream);

        byte[] buf = new byte[1024];

        int num = -1;

        ByteArrayOutputStream baos = new ByteArrayOutputStream();

        while ((num = gzipInputStream.read(buf, 0, buf.length)) != -1) {

            baos.write(buf, 0, num);

        }

        return new String(baos.toByteArray(), "utf-8");

    }

    public HttpClient(String ServerName, String ServerPort, String QuestPath) {

        try {

            String ServerURL = "";

            ServerURL += DEFAULT_PROTOCOL;

            ServerURL += COLON;

            ServerURL += SLASH;

            ServerURL += SLASH;

            ServerURL += ServerName;

            if ((ServerPort != null) && (ServerPort.trim().length() > 0)) {

                ServerURL += COLON;

                ServerURL += ServerPort.trim();

            }

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}
```

```
        }

        if (QuestPath.charAt(0) != '/') {

            ServerURL += SLASH;

        }

        ServerURL += QuestPath;

        url = new URL(ServerURL);

    } catch (Exception e) {

        e.printStackTrace();

    }

}

}
```

技术支持联络方式

如果您在集成和阅读数据时遇到任何问题，请及时与我们取得联系：

- 技术支持邮箱：support@tendcloud.com
- 企业 QQ：4008701230
- 热线：4008701230