

简介

本文档是调用ipipv，开通代理的说明。

开通代理先向我方获取appKey 和 appSecret,在沙盒环境下调试。

沙盒环境地址: <https://sandbox.ipipv.com/>

沙盒调试成功后，可以切换至生产环境。

注意：生产环境测试会正式开通代理，产生费用。

生产环境地址: <https://api.ipipv.com/>

sdk

golang

<https://github.com/clpublic/ipv-sdk>

java（部分完成）

<https://github.com/clpublic/ipv-java-sdk>

php

<https://github.com/clpublic/ipv-php-sdk.git>

版本更新

全新对接，直接跳过当前，往后看，如果已经对接，看一下最新更新的内容

版本	发布时间	更新内容
v2.0.8	2024-09-21	增加ip段的asn返回，时长开通增加cycleTimes字段
v2.0.7	2024-08-08	动态流量记录改为int64类型
v2.0.6	2024-08-05	更具产品号返回动态的流量和使用记录
v2.0.5	2024-07-27	实例信息返回（包含订单中的实例返回）增加产品编号信息
v2.0.4	2024-07-26	动态提取增加产品编号，根据产品编号提取
v2.0.3	2024-07-24	增加app信息返回，增加php sdk
v2.0.2	2024-07-11	增加动态流量使用记录
v2.0.1	2024-07-09	修改统一请求加密方式，补充java语言的 aes cbc 加密及实现的 demo
v2.0.0	2024-07-08	app开放接口第一个版本发布

通用

请求统一加密传输

请求参数

参数	二级参数	类型	必填	说明
version	-	string	是	版本 v2
encrypt	-	string	是	加密方式 aes,rsa（版本2以上提供，默认aes,老接口rsa） aes cbc模式
appKey	-	string	是	appKey 渠道商号 找我方获取
reqId	-	string	是	请求id，每次生成，如果失败重复请求，保持不变
params	-	string	是	根据加密方式密文 转base64

返回数据

参数	二级参数	类型	必填	说明
reqId	-	string	是	请求id，每次生成，如果失败重复请求，保持不变
code	-	int	是	状态码，详见状态码
msg	-	string	是	消息内容
data	-	string	是	密文转base64

加密方式

接口通过aes 加密 cbc 模式 请求时序图如下



备注：

- 请求加密串最终存在params，返回数据加密串存在data参数

- 如果返回数据只有状态码，没有data数据，就不需要做加密返回。

数据加密

- 消息内容为： TestMessage （前后无空格）
- 密钥为： qwertyuiop123456 （前后无空格）
- 通过aes_cbc模式，并Base64后： 5cfJMwX0w65sEQVBxu9zHw==
- 初始向量参数为密钥前16位

请求json示例

```
{
  "version":"2.0",
  "reqId":"xxxxxx",
  "encrypt":"aes",
  "appKey":"testappid",
  "params":"5cfJMwX0w65sEQVBxu9zHw=="
}
```

返回json示例

```
{
  "reqId":"xxxxxx",
  "code":200,
  "msg":"ok",
  "data":"5cfJMwX0w65sEQVBxu9zHw=="
}
```

go aes cbc加解密函数

```
package cryptos

import (
    "bytes"
    "crypto/aes"
    "crypto/cipher"
)
```

```
// ===== CBC =====

// key的长度必须为16, 24或者32
func AesEncryptCBC(origData []byte, key []byte) (encrypted []byte, err error) {
    // 分组密钥

    block, _err := aes.NewCipher(key)
    if _err != nil {
        err = _err
        return
    }
    blockSize := block.BlockSize() // 获取密钥块的长度

    origData = pkcs5Padding(origData, blockSize) // 补全码
    blockMode := cipher.NewCBCEncrypter(block, key[:blockSize]) // 加密模式
    encrypted = make([]byte, len(origData)) // 创建数组
    blockMode.CryptBlocks(encrypted, origData) // 加密
    return encrypted, nil
}

// key的长度必须为16, 24或者32
func AesDecryptCBC(encrypted []byte, key []byte) (decrypted []byte, err error)
{
    block, _err := aes.NewCipher(key) // 分组密钥
    if _err != nil {
        err = _err
        return
    }
    blockSize := block.BlockSize() // 获取密钥块的长度

    blockMode := cipher.NewCBCDecrypter(block, key[:blockSize]) // 加密模式
    decrypted = make([]byte, len(encrypted)) // 创建数组
    blockMode.CryptBlocks(decrypted, encrypted) // 解密
    decrypted = pkcs5UnPadding(decrypted) // 去除补全码
    return decrypted, nil
}

func pkcs5Padding(ciphertext []byte, blockSize int) []byte {
    padding := blockSize - len(ciphertext)%blockSize
    padtext := bytes.Repeat([]byte{byte(padding)}, padding)
    return append(ciphertext, padtext...)
}

func pkcs5UnPadding(origData []byte) []byte {
    length := len(origData)
    unpadding := int(origData[length-1])
}
```

```
        return origData[: (length - unpadding)]
    }
```

java aes cbc加解密函数

```
package com.ipipv.open.utils;

import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.util.Base64;

public class AESCBC {
    public static byte[] encryptCBC(byte[] data, byte[] key, byte[] iv) throws
NoSuchPaddingException, NoSuchAlgorithmException, InvalidKeyException,
BadPaddingException, IllegalBlockSizeException,
InvalidAlgorithmParameterException {
        Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
        cipher.init(Cipher.ENCRYPT_MODE, new SecretKeySpec(key, "AES"), new
IvParameterSpec(iv));
        byte[] result = cipher.doFinal(data);
        return result;
    }

    public static byte[] decryptCBC(byte[] data, byte[] key, byte[] iv) throws
NoSuchPaddingException, NoSuchAlgorithmException, InvalidKeyException,
BadPaddingException, IllegalBlockSizeException,
InvalidAlgorithmParameterException {
        Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
        cipher.init(Cipher.DECRYPT_MODE, new SecretKeySpec(key, "AES"), new
IvParameterSpec(iv));
        byte[] result = cipher.doFinal(data);
        return result;
    }

    public static void main(String[] args) throws IllegalBlockSizeException,
InvalidKeyException, BadPaddingException, NoSuchAlgorithmException,
```

```

NoSuchPaddingException, InvalidAlgorithmParameterException {
    String data = "TestMessage"; // 待加密的原文
    String key = "qwertyuiop123456asdfghjk"; // key 长度只能是 16、24 或 32 字
节
    String iv = key.substring(0,16); // CBC 模式需要用到初始向量参数

    byte[] ciphertext = encryptCBC(data.getBytes(), key.getBytes(),
iv.getBytes());
    System.out.println("CBC 模式加密结果 (Base64) : " +
Base64.getEncoder().encodeToString(ciphertext));

    byte[] plaintext = decryptCBC(ciphertext, key.getBytes(),
iv.getBytes());
    System.out.println("解密结果: " + new String(plaintext));
}
}

```


同步地域

请求路径

/api/open/app/area/v2

请求参数

参数	二级参数	类型	必填	说明
codes	-	[]string	否	获取地域代码对应列表，为null获取全部

返回参数

参数	二级参数	类型	必填	说明
code	-	string	是	地域代码
name	-	string	否	地域名称
cname	-	string	是	地域中文 名
children	-	array	否	下级地域
-	code	string	是	地域代码
-	name	string	否	地域名称
-	cname	string	是	地域中文 名
-	children	array	否	下级地域

同步地域

请求路径

/api/open/app/city/list/v2

请求参数

参数	二级参数	类型	必填	说明
codes	-	[]string	否	城市代码列表，为null获取全部

返回参数

参数	二级参数	类型	必填	说明
cityCode	-	string	是	城市代码
cityName	-	string	否	城市中文名称
stateCode	-	string	是	省、州代码
stateName	-	string	否	省、州中文名称
countryCode	-	string	是	国家代码
countryName	-	string	否	国家中文名称
areaCode	-	string	是	洲代码
areaName	-	string	否	洲中文名称
status	-	int	否	状态 1=正常

获取产品库存

请求路径

/api/open/app/product/query/v2

请求参数

当主接口中的method为GetOrder时，该参数是主接口中的params参数

参数	二级参数	类型	必填	说明
proxyType	-	[]int	是	代理类型，详见字典
productNo	-	string	否	产品Id 如果传了产品Id，则只返回该Id对应的产品信息
countryCode	-	string	否	国家代码 可选
cityCode	-	string	否	城市代码 可选
supplierCode	-	string	否	供应商代码 可选
unit	-	int	否	时长单位，详见字典 可选
ispType	-	int	否	isp类型，详见字典 可选
duration	-	int	否	相对于时长单位的最小购买时长 可选

eg

```
{
  "proxyType": [103],
  "productNo": "productNo"
}
```

返回数据

参数	二级参数	类型	必填	说明
productNo	-	string	是	产品Id 保持唯一
productName	-	string	是	商品名
proxyType	-	int16	是	代理类型，详见字段定义
useType	-	string	是	以(,)分割 1=账密 2=白名单 3=uuid
protocol	-	string	是	1=socks5 2=http 3=https 4=ssh
useLimit	-	int8	是	1=出口ip国外 2=出口ip国内 3=无限制
sellLimit	-	int8	是	1=大陆可售 2=海外可售 3=无限制
areaCode	-	string	否	区域code 动态代理没有区域，提取的时候查看动态代理支持的区域
countryCode	-	string	是	国家代码 3位 iso3
stateCode	-	string	是	州省代码 6位 000结尾要看citycode 是否有值或者结尾，（有些小国家是没有州省级别的，但有城市区分）
cityCode	-	string	是	城市代码 9位 为空或者000
detail	-	string	否	商品描述
costPrice	-	string	是	价格
inventory	-	int	是	库存
ipType	-	int	否	ip类型 1=ipv4 2=ipv6 3=随机 默认1
ispType	-	int	否	isp类型： 1=单isp 2=双isp 0=未知 默认
netType	-	int	否	网络类型： 1=原生 2=广播 0=未知 默认
duration	-	int	是	时长 0无限制
unit	-	int	是	单位 1=天 2=周(7天) 3=月(自然月) 4=年(自然年365, 366)
bandWidth	-	int	否	带宽 流量时必要 单位 MB
bandWidthPrice	-	string	否	额外带宽价格

maxBandWidth	-	int	否	可设置最大带宽
flow	-	int	否	动态代理按照流量方式 最小购买和计价流量包 单位MB
cpu	-	int	否	cpu数
memory	-	float64	否	内存容量
enable	-	int8	否	是否可以购买 1可以
supplierCode	-	string	否	供应商代码
ipCount	-	int	否	动态代理按照ip数方式 最小购买和计价ip数 单位个
ipDuration	-	int	否	动态代理按照ip数方式 时长 单位分钟
assignIp	-	int	否	是否支持指定ip开通静态代理 1=是 -1=否 默认为-1
parentNo	-	string	否	父产品编号
cidrStatus	-	int	否	是否支持ip网段功能 1=是 -1=否 默认为-1
cidrBlocks	-	array	否	支持网段及数量
-	cidr	string	否	网段 比如 192.168.0.0/24
-	count	int	否	该网段数量
-	asn	string	否	该网段属于哪个asn
-	isp	string	否	该网段属于哪个运营商 比如ATT

备注：

- 只有enable = 1 且 inventory 大于 0 的产品可以购买
- 产品列表的时长：duration 和单位：unit的结合，代表这款产品的最小时长

eg：

- duration=1 unit=1 表示最少购买1天
- duration=1 unit=3 表示最少购买1个月
- duration=30 unit=1 表示最少购买30天

开通代理

请求路径

/api/open/app/instance/open/v2

请求参数

主接口中的params参数

参数	二级参数	三级参数	类型	必填	说明
appOrderNo	-	-	string(32)	是	购买者订单号(渠道商订单号), 同一个订单保持唯一, 我方或作幂等性检查
params	-	-	array	是	购买代理产品列表
-	productNo	-	string	是	商品编号, 推荐👍 (按商品编号购买的时候, 后面7项商品筛选条件无意义) →着重看下面红色部分备注
-	proxyType	-	uint16	否	(1) 代理类型, 详见字典
-	countryCode	-	string	否	(2) 国家代码
-	cityCode	-	string	否	(3) 城市代码
-	supplierCode	-	string	否	(4) 供应商代码 (可为null,随机分配)
					(5) 单位 1=天

-	unit	-	int8	否	2=周(7天) 3=月 (自然月) 4=年 (自然年365, 366) 10=无限制
-	ispType	-	int	否	(6) isp类型 1= 单isp 2=双isp
-	duration	-	int32	是	(7) 产品定义的 时长单位
-	count	-	int	是	购买数量 (实 例个数) 静态必 填 默认1 一次最 大20
-	cycleTimes	-	int32	是	购买时长周期 数, 此字段对有 时长的产品有意 义, 默认1表示 产品的duration 个unit的时长,详 见 →备注
-	renew	-	bool	否	是否续费 1续费 默认0
-	extBandWidth	-	int32	否	额外增加带宽 单位Mbps
-	appUsername	-	string	否	主账号, 开通动 态代理的时候必 填(必须在平台 上注册过)
-	flow	-	int	是	动态流量, 静态 的字段无意义 动态必填 单位 MB
-	useBridge	-	uint8	否	1=不使用桥 2= 使用桥 不传跟 随app设置 默认
					购买项目id,保

-	projectId	-	string	否	留字段，后续会支持
-	cidrBlocks	-	array	否	支持网段及数量
-	-	cidr	string	否	网段 192.168.0.0/24 172.16.0.0/16 10.0.0.0/8
-	-	count	int	否	该网段数量

eg

```
{
  "appOrderNo": "TEST20240726094927",
  "params": [
    {
      "productNo": "ipideash_598",
      "count": 20,
      "cycleTimes": 1
    },
    {
      "productNo": "mb_gmhd5exp2",
      "count": 20,
      "cycleTimes": 12
    }
  ]
}
```

备注：

cycleTimes 为新增字段，以区分产品的duration字段，老版本使用的duration，会自动兼容，新的对接方式请使用cycleTimes字段

产品购买分为按产品编号（productNo）唯一指定的产品 和 按产品筛选条件（proxyType,countryCode,cityCode,supplierCode,ispType,duration,unit）购买。当产品编号存在时，产品筛选条件不起作用（我们推荐按产品编号购买，这是唯一的，价格确定的，如果存在号段，也可以指定购买的号段）。

按筛选条件购买时，筛选条件越多，产品会越少，因为找不到产品，而购买失败。而且不能号段筛

选（所以不建议用这种形式）

购买数量（count）和购买周期数（cycleTimes）必填，不填默认为1，购买一条一个时间周期

备注：产品列表的时长：duration 和单位：unit的结合，代表这款产品的最小时长

eg:

- duration=1 unit=1 表示最少按1天卖 如果购买参数cycleTimes=1 表示购买1天 cycleTimes=30代表30天
- duration=1 unit=3 表示最少按1月卖 如果购买参数cycleTimes=1 表示购买1月 cycleTimes=12代表12月，3代表自然月，根据购买当月的天数计算，比如2月可能28天或者29，3月31天，4月30天
- duration=30 unit=1 表示最少按30天卖 如果购买参数cycleTimes=1 表示购买30天 cycleTimes=12代表360天
- duration=1 unit=4 表示最少按1年卖 如果购买参数cycleTimes=1 表示购买1年，即365天或者366天

费用计算

静态产品的费用 = 产品价格 * cycleTimes * count+ 带宽价格 * extBandWidth * cycleTimes * count+使用桥费用 * cycleTimes * count

- 带宽价格
 - 必须产品支持，extBandWidth为0，表示为默认带宽,无费用
- 桥（高速通道）
 - 必须实名认证企业可用
 - 如果useBridge = 1 不会产生桥费用，如果不传，跟随app，app配置为使用桥，购买海外产品（包含香港，台湾）就会使用桥
 - 桥使用需要先获取桥地址 动态

流量产品的费用 = 产品价格（MB）* flow

- 流量产品count 不起作用
- 流量产品只对flow起作用

返回数据

参数	二级参	类型	必	说明
----	-----	----	---	----

	数		填	
orderNo	-	string	是	平台订单号
appOrderNo	-	string	是	购买者订单号(渠道商订单号) 原样返回号
amount	-	string	否	花费金额（总共花费金额）

备注： 订单为异步开通， 具体查看[回调章节](#)

获取订单信息

请求路径

/api/open/app/order/v2

请求参数

参数	二级参数	类型	必填	说明
orderNo	-	string(32)	是	平台订单编号
page	-	int	否	页码 默认1
pageSize	-	int	是	每页显示数量 默认10 最大100

返回参数

参数	二级参数	类型	必填	说明
orderNo	-	string	是	平台订单号
appOrderNo	-	string	是	渠道商（购买订单）订单号
type	-	int8	是	订单类型 1=新建 2=续费 3=释放
status	-	int8	否	订单状态 1=待处理 2=处理中 3=处理成功 4=处理失败 5=部分完成
count	-	int	否	购买数量
amount	-	string	否	总价
refund	-	int	是	是否有退费 1存在退费
page	-	int	是	页码 原样返回
pageSize	-	int	否	每页显示数量 原样返回
total	-	int64	是	订单对应实例总数量

instances	-	array	是	订单对应实例列表
-	instanceNo	string	是	平台实例编号（渠道商续费和释放操作使用该编号）
-	proxyType	uint	是	代理类型 101=静态云平台 102=静态国内家庭 103=静态国外家庭 104=动态国外 105=动态国内 201=whatsapp
-	protocol	string	否	i协议类型 多个用英文逗号分隔 1=socks5 2=http 3=https 4=ssh
-	ip	string	是	代理ip地址
-	port	uint	否	代理端口
-	regionId	string	否	区域地址
-	countryCode	string	是	国家代码
-	cityCode	string	是	城市代码
-	useType	string	是	使用方式 多个用英文逗号分隔 1=账密 2=ip白名单 3=uuid（uuid写password内）
-	username	string	是	账户名或uuid 动态为平台主账号
-	pwd	string	是	密码
-	orderNo	string	是	创建该实例的平台订单号
-	userExpired	int64	否	到期时间
-	flowTotal	float64	是	总流量 单位MB 如果是ip数方式 表示ip个数
-	flowBalance	float64	否	剩余流量
-	status	int8	否	1=待创建 2=创建中 3=运行中 6=已停止 10=关闭 11=释放
-	renew	int8	是	1 自动续费
-	bridges	[]string	是	桥地址列表

-	openAt	time.Time	是	开通时间
-	renewAt	time.Time	是	最后成功续费时间
-	releaseAt	time.Time	是	释放成功时间
-	productNo	string	是	产品编号

获取实例

获取实例的

请求路径

/api/open/app/instance/v2

请求参数

参数	二级参数	类型	必填	说明
instances	-	[]string	是	供应商实例编号(供应商系统内部唯一)

返回数据

参数	二级参数	类型	必填	说明
instanceNo	-	string	是	平台实例编号（渠道商续费和释放操作使用该编号）
proxyType	-	uint	是	代理类型 101=静态云平台 102=静态国内家庭 103=静态国外家庭 104=动态国外 105=动态国内 201=whatsapp
protocol	-	string	否	i协议类型 多个用英文逗号分隔 1=socks5 2=http 3=https 4=ssh
ip	-	string	是	代理ip地址
port	-	uint	否	代理端口
regionId	-	string	否	区域地址
countryCode	-	string	是	国家代码

cityCode	-	string	是	城市代码
useType	-	string	是	使用方式 多个用英文逗号分隔 1=账密 2=ip白名单
username	-	string	是	账户名或uuid 动态为平台主账号
pwd	-	string	是	密码
orderNo	-	string	是	创建该实例的平台订单号
userExpired	-	int64	否	到期时间
flowTotal	-	float64	是	总流量
flowBalance	-	float64	否	剩余流量
status	-	int8	否	1=待创建 2=创建中 3=运行中 6=已停止 10=关闭 11=释放
renew	-	int8	是	1 自动续费
bridges	-	[]string	是	桥地址列表
openAt	-	time.Time	是	开通时间
renewAt	-	time.Time	是	最后成功续费时间
releaseAt	-	time.Time	是	释放成功时间
productNo	-	string	是	产品编号

续费代理

请求路径

/api/open/app/instance/renew/v2

请求参数

当主接口中的method为RewProxy时，该参数是主接口中的params参数

参数	二级参数	类型	必填	说明
appOrderNo	-	string	是	购买者订单号(渠道商订单号)
instances	-	array	是	实例列表
-	instanceNo	string	是	平台实例编号
-	duration	int32	是	可选 时长 默认1

返回数据

参数	二级参数	类型	必填	说明
orderNo	-	string	是	平台订单号
appOrderNo	-	string	是	购买者订单号(渠道商订单号) 原样返回号
amount	-	string	否	花费金额（总共花费金额）

备注：订单为异步开通，具体查看[回调章节](#)

释放代理

请求路径

/api/open/app/instance/release/v2

请求参数

当主接口中的method为DelProxy时，该参数是主接口中的params参数

参数	二级参数	类型	必填	说明
orderNo	-	string	是	购买者订单号(渠道商订单号)
instances	-	[]string	是	平台实例编号

返回数据

参数	二级参数	类型	必填	说明
orderNo	-	string	是	平台订单号
appOrderNo	-	string	是	购买者订单号(渠道商订单号) 原样返回号
amount	-	string	否	花费金额（总共花费金额）

备注：订单为异步开通，具体查看[回调章节](#)

获取App信息

请求路径

/api/open/app/info/v2

请求参数

无

返回数据

参数	二级参数	类型	必填	说明
appName	-	string	是	app名称
coin	-	string	是	账户余额
useBridge	-	int	否	使用桥 1 不使用 2使用
callbackUrl	-	string	否	回调地址
status	-	int	否	1正常 -1禁用

指定ip开通代理

请求路径

/api/open/app/instance/open/assign/ip/v2

请求参数

主接口中的params参数

参数	二级参数	三级参数	类型	必填	说明
appOrderNo	-	-	string(32)	是	购买者订单号(渠道商订单号), 同一个订单保持唯一, 我方或作幂等性检查
productNo	-	-	string	是	商品编号, 最终开出来的ip地域和产品不一致
cycleTimes	-	-	int32	是	购买时长周期数, 此字段对有时长的产品有意义, 默认1表示产品的duration个unit的时长, 详见 →备注
renew	-	-	bool	否	是否续费 1续费 默认0
extBandWidth	-	-	int32	否	额外增加带宽 单位Mbps
useBridge	-	-	uint8	否	1=不使用桥 2=使用桥 不传跟随app设置 默认
assignIp	-	-	string	是	指定ip

eg

```
{
  "appOrderNo":"TEST20240726094927",
  "productNo":"ipideash_598",
  "assignIp":"127.0.0.1",
  "cycleTimes":1
}
```

返回数据

参数	二级参数	类型	必填	说明
orderNo	-	string	是	平台订单号
appOrderNo	-	string	是	购买者订单号(渠道商订单号) 原样返回号
amount	-	string	否	花费金额（总共花费金额）

备注：订单为异步开通，具体查看[回调章节](#)

查询指定Ip可用信息

请求路径

/api/open/app/assign/ip/info/v2

请求参数

主接口中的params参数

参数	二级参数	三级参数	类型	必填	说明
ip	-	-	string	是	指定ip

eg

```
{
  "ip": "127.0.0.1"
}
```

返回数据

参数	二级参数	类型	必填	说明
ip	-	string	是	指定ip
canBuyStatus	-	int	是	购买状态 (-1: 不可购买, 1: 可购买)

动态

创建或修改主账户

该用户为动态用户的主账户，所有的流量分配都在主账户下，该用户下子账户可以使用主账户的流量。

请求路径

/api/open/app/user/v2

请求参数

参数	二级参数	类型	必填	说明
appUsername	-	string	否	渠道商主账号 该渠道商唯一 不支持修改 (不传随机生成，建议不传)
password	-	string	否	主账号密码(不传随机生成，建议不传)
phone	-	string	是	主账号手机号
email	-	string	是	主账号邮箱
authType	-	int8	是	认证类型 1=未实名 2=个人实名 3=企业实名
authName	-	string	否	主账号实名认证的真实名字或者企业名
no	-	string	是	主账号实名认证的实名证件号码或者企业营业执照号码
vsp	-	uint8	否	vsp
status	-	int8	是	状态 1=正常 2=禁用

返回参数

参数	二级参数	类型	必填	说明
appUsername	-	string	是	渠道商子账号

username	-	string	是	平台子账号
password	-	string	是	子账号密码
status	-	int8	是	用户状态 1=正常 2=禁用
authStatus	-	int8	是	认证状态 1=未实名 2=个人实名 3=企业实名

同步实名

请求路径

/api/open/app/userAuth/v2

请求参数

参数	二级参数	类型	必填	说明
username	-	string	否	平台主账号 选填 平台主账号和渠道商主账号两个必填一个
appUsername	-	string	否	渠道商主账号 选填 平台主账号和渠道商主账号两个必填一个
authType	-	int8	是	认证类型 1 未实名 2 个人实名 3 企业实名
authName	-	string	否	真实姓名或者企业名
no	-	string	否	实名认证号码或者企业营业执照号码
vsp	-	string	否	vsp

返回参数

参数	二级参数	类型	必填	说明
username	-	string	是	平台账号
authStatus	-	int	是	认证状态 1=未实名 2=个人实名 3=企业实名

动态产品区域列表

请求路径

/api/open/app/product/area/v2

请求参数

参数	二级参数	类型	必填	说明
productNo	-	string	是	平台产品编号
proxyType	-	int16	是	代理类型 104=动态国外 105=动态国内

返回参数

参数	二级参数	类型	必填	说明
-	-	array	是	平台产品编号
-	productNo	string	否	平台产品编号 如果为空，则表示同一个上游供应商公用的区域列表
-	proxyType	int16	是	代理类型
-	areaCode	string	是	区域代码（洲）
-	countryCode	string	是	国家代码
-	stateCode	string	是	州省代码 为空或者后面 000结尾，说明当前地域只到上级国家，国家内随机
-	cityCode	string	是	城市代码 为空或者后面 000结尾，说明当前地域只到上级，上级随机

-	status	int8	是	状态 1=上架 -1=下架
-	region	string	是	上游供应商区域
-	supplierCode	string	是	上游供应商代码

```
[  
  
{"productNo":"out_dynamic_1","proxyType":104,"areaCode":"6","countryCode":"USA"  
,"stateCode":"","cityCode":"","status":1,"region":"us","supplierCode":"xxx"},  
  
{"productNo":"out_dynamic_1","proxyType":104,"areaCode":"2","countryCode":"FRA"  
,"stateCode":"","cityCode":"","status":1,"region":"fr","supplierCode":"xxx"},  
  
{"productNo":"out_dynamic_1","proxyType":104,"areaCode":"2","countryCode":"GBR"  
,"stateCode":"","cityCode":"","status":1,"region":"gb","supplierCode":"xxx"},  
  
{"productNo":"out_dynamic_1","proxyType":104,"areaCode":"1","countryCode":"JPN"  
,"stateCode":"","cityCode":"","status":1,"region":"jp","supplierCode":"xxx"}  
]
```

获取代理余额信息

通过接口获取代理商户

请求路径

/api/open/app/proxy/info/v2

请求参数

参数	二级参数	类型	必填	说明
username	-	string	否	平台主账号，选填 平台主账号和渠道商主账号两个必填一个
appUsername	-	string	否	渠道商主账号，选填 平台主账号和渠道商主账号两个必填一个
proxyType	-	uint16	是	代理类型 必填 104=动态国外 105=动态国内
productNo	-	string	是	产品编号

返回参数

参数	二级参数	类型	必填	说明
list	-	array	否	列表
-	used	string	否	已使用流量 MB
-	total	int	否	总流量 MB
-	balance	string	是	剩余流量 MB
-	productNo	string	否	产品编号
-	ipWhiteList	[]string	否	ip白名单
-	ipUsed	int	否	已使用ip数量

-	ipTotal	int	否	总ip数量
---	---------	-----	---	-------

创建或修改代理用户（子账号）

请求路径

/api/open/app/proxy/user/v2

请求参数

参数	二级参数	类型	必填	说明
appUsername	-	string(32)	否	渠道商子账号 该渠道商唯一 (不传随机生成) 不支持修改 (不传随机生成，建议不传)
password	-	string	否	密码(不传随机生成，建议不传)
limitFlow	-	int	是	动态流量上限 单位MB
mainUsername	-	string	否	平台主账号 选填 平台主账号和渠道商主账号两个必填一个
appMainUsername	-	string	否	渠道商主账号 选填 平台主账号和渠道商主账号两个必填一个
remark	-	string(255)	否	备注
status	-	int8	是	状态 1=正常 2=禁用

返回数据

参数	二级参数	类型	必填	说明
appUsername	-	string	是	渠道商子账号

username	-	string	是	平台子账号
password	-	string	是	子账号密码
status	-	int8	是	用户状态 1=正常 2=禁用
authStatus	-	int8	是	认证状态 1=未实名 2=个人实名 3=企业实名

账密提取

请求路径

/api/open/app/proxy/draw/pwd/v2

请求参数

参数	二级参数	类型	必填	说明
appUsername	-	string	是	渠道商子账号名
addressCode	-	string	否	地址代码 可以传 areaCode countryCode stateCode cityCode 四种之一
sessTime	-	string	否	有效时间 1-120分钟 默认5分钟
num	-	int	否	数量 默认1
proxyType	-	uint16	否	代理类型 104=动态国外 105=动态国内
maxFlowLimit	-	int	否	子账号最大流量限制 可选 大于0的时候生效
productNo	-	string	是	产品编号

返回参数

参数	二级参数	类型	必填	说明
list	-	array	是	
-	proxyUrl	string	否	代理地址

-	list	[]string	否	
---	------	----------	---	--

Api提取代理请求

请求路径

/api/open/app/proxy/draw/api/v2

请求参数

参数	二级参数	类型	必填	说明
appUsername	-	string	是	渠道商主账号
proxyType	-	uint16	是	代理类型 104=动态国外 105=动态国内
num	-	int	否	提取ip数量 可选 默认1
addressCode	-	string	否	地址代码 可选 取值 areaCode countryCode stateCode cityCode 四种之一
protocol	-	string	否	协议 可选 默认socks5 取值 socks5 http 之一
returnType	-	string	否	数据格式 可选 默认txt 取值 txt json 之一
delimiter	-	int	否	分隔符 可选 只有数据格式是txt的时候生效 默认1 (1=\r\n 2=/br 3=\r 4=\n 5=\t)
maxFlowLimit	-	int	否	最大流量限制 可选 大于0的时候生效
productNo	-	string	是	产品编号

返回数据

参数	二级参数	类型	必填	说明

list	-	array	是	
-	proxyUrl	string	是	提取代理Api地址

流量使用记录

请求路径

/api/open/app/proxy/flow/use/log/v2

请求参数

参数	二级参数	类型	必填	说明
appUsername	-	string	是	主账号 必要
startTime	-	string	否	开始时间 可选 默认7天前 格式 2021-01-01 00:00:00
endTime	-	string	否	结束时间 可选当天 格式 2021-01-01 00:00:00
productNo	-	string	是	产品编号
page	-	int	否	页码 可选 默认1
pageSize	-	int	否	每页数量 可选 默认10 最大100

返回参数

参数	二级参数	类型	必填	说明
total	-	int	是	总数量
curPage	-	int	是	当前页
list	-	array	是	使用详情数组
-	used	int64	是	已使用流量 单位B
-	total	int64	是	总流量 B

-	balance	int64	是	剩余流量 B
-	usedTime	uint64	是	使用时间 单位秒
-	productNo	string	否	产品编号

添加ip白名单

请求路径

/api/open/app/proxy/addIpWhiteList/v2

请求参数

参数	二级参数	类型	必填	说明
appUsername	-	string	是	渠道商主账号
ip	-	string	是	ip地址
proxyType	-	uint16	是	代理类型
productNo	-	string	是	产品编号

返回参数

参数	二级参数	类型	必填	说明
ipWhiteList	-	[]string	是	ip白名单

删除ip白名单

请求路径

/api/open/app/proxy/dellpWhiteList/v2

请求参数

参数	二级参数	类型	必填	说明
appUsername	-	string	是	渠道商主账号
ip	-	string	是	ip地址
proxyType	-	uint16	是	代理类型 可选 默认104 104=动态国外 105=动态国内
productNo	-	string	是	产品编号

返回参数

参数	二级参数	类型	必填	说明
ipWhiteList	-	[]string	是	ip白名单

动态代理流量回收

请求路径

/api/open/app/proxy/return/v2

请求参数

参数	二级参数	类型	必填	说明
appUsername	-	string	是	渠道商主账号 必要
proxyType	-	int	是	代理类型 必填 104=动态国外 105=动态国内
ipNum	-	int	否	回收ip数量 单位个 产品如果是按照ip数量购买 使用该字段
productNo	-	string	是	产品编号
flowNum	-	int	否	回收流量数量 单位M 产品如果是按照流量购买 使用该字段
remark	-	string	否	备注 最多250个字符

返回参数

参数	二级参数	类型	必填	说明
returnAmount	-	float64	否	回收代理退还金额 单位元

回调

IPV系统存在三种回调：订单回调，实例回调，产品回调。回调地址由客户提供，我们统一在后台配置。

假设用户设置的回调地址为 `https://api.callback.com/ipv/callback`

产生回调方式为,我方发起get请求

```
https://api.callback.com/ipv/callback?type={callbackType}&no={callbackNo}&op={opType}
```

名称	type	no	op	说明
订单	order	订单号	订单类型，1创建 2续费 3 释放	我方订单编号
实例	instance	实例编号	-	我方的实例编号，唯一
产品	product	产品编号	-	获取产品库存中的productNo 字段

用户收到回调处理后，要返回回调结果，否则会多次进行回调，因为网络波动等原因，用户对同一回调，做好幂等性操作。

回调结果定义

```
{
  "code": "",
  "msg": ""
}
```

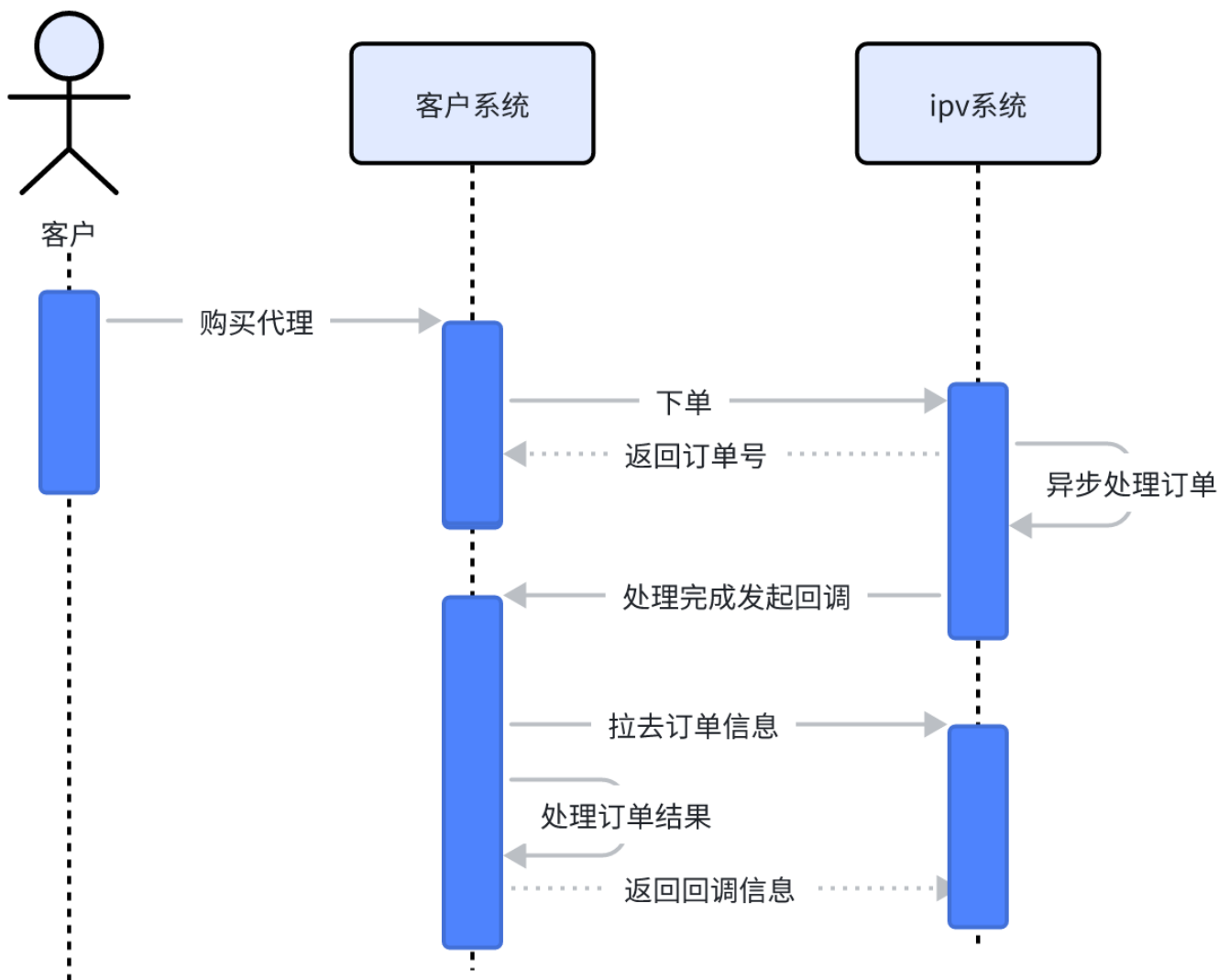
成功 code返回值 success 多次回调上一次已经成功，第二次还是返回success

订单回调

IPV系统的订单(开通, 续费, 释放)都是异步的, 即用户下单 -> ipv返回订单 同时开通代理 -> 代理开通完成回调客户回调地址 -> 客户通过回调订单返回信息并拉去订单信息

eg

<https://api.callback.com/ipv/callback?type=order&no=C20240429162339417081&op=1>



实例回调

当实例产生变化（比如网络不通, 停机等）的时候, 我方会主动回调

eg

https://api.callback.com/ipv/callback?type=instance&no=c_gz3h2igp6dz9cq3&op=

产品回调

当产品进行变动的时候，我方也会主动回调

eg

https://api.callback.com/ipv/callback?type=product&no=mb_gmfcq7blc&op=

数据字典

代理类型

编号	说明
101	静态云平台
102	静态国内家庭
103	静态国外家庭
104	动态国外
105	动态国内
201	whatsapp

代理协议

编号	说明
1	socks5
2	http
3	https
4	ssh

时长单位

编号	说明
1	天
2	周（7天）
3	自然月
4	自然年365， 366
10	无限制

ispType

编号	说明
0	未知（默认）
1	单isp
2	双isp
3	原生isp
4	机房数据中心

返回状态码

编号	说明
200	成功
500	系统错误
10001	解密失败，请检查密码
10002	创建代理失败，余额不足（返回当前值为未执行创建）
10003	创建代理失败，库存不足（返回当前值为未执行创建）
10004	创建代理失败，参数有误
10011	续费失败，实例不存在
10021	释放失败，实例不存在
10022	释放失败，实例已超过释放周期
10031	获取订单失败，订单不存在
10041	获取实例失败，实例不存在
10051	创建用户失败，用户名已存在
10052	获取用户失败，用户不存在
10061	创建子账户失败，用户名已存在
10062	获取子账户失败，用户不存在
10061	创建子账户失败，主账户不存在
10061	创建子账户失败，主账户已禁用
10071	分配流量失败，主账户不存在或者被禁用
10072	分配流量失败，余额不足
10081	回收流量失败，余额不足
10091	参数错误，region不存在
10092	参数错误，username子账户不存在
10093	参数错误，sessTime取值范围1-120（分钟）