

Static IP Open API 接入文档

1. 概述

本文档基于当前生产代码实现整理，适用于客户通过 Open API 查询可售地区、创建新购订单、创建续费订单、查询订单和查询余额。

- 文档版本： 2026-08-19
- 基础路径： /api/v1
- 默认服务地址： https://{host}/api/v1
- 数据格式： application/json
- 字符编码： UTF-8

当前开放接口：

- GET /health
- GET /balance
- GET /regions
- POST /orders
- GET /orders
- GET /orders/{order_no}
- POST /orders/renewal

2. 认证方式

除 GET /health 外，其余接口都使用 HMAC-SHA256 签名认证。

2.1 请求头

每次请求必须携带以下请求头：

- X-API-Key
- X-Timestamp
- X-Nonce
- X-Signature

说明：

- X-API-Key：用户 API Key
- X-Timestamp：Unix 秒级时间戳
- X-Nonce：每次请求唯一随机串，推荐使用 UUID

- **X-Signature** : 签名结果, `Base64(HMAC_SHA256(StringToSign, api_secret))`

时间戳请控制在 5 分钟内, 避免因服务端时间窗口或防重放校验失败。

2.2 签名串

签名串格式:

```
HTTP_METHOD + "\n" +  
REQUEST_PATH + "\n" +  
CanonicalQueryString + "\n" +  
CanonicalHeaders + "\n" +  
HashedRequestBody
```

其中:

- **HTTP_METHOD** : 大写, 如 `GET`、`POST`
- **REQUEST_PATH** : 请求路径, 如 `/api/v1/orders`
- **CanonicalQueryString** : 按参数名升序拼接的查询字符串, 如 `page=1&page_size=10`
- **CanonicalHeaders** : 以下 3 个请求头转小写后按 key 升序拼接
 - `x-api-key`
 - `x-timestamp`
 - `x-nonce`
- **HashedRequestBody** : 请求体原文的 SHA256 十六进制小写值

2.3 CanonicalHeaders 示例

```
x-api-key:test_key  
x-nonce:550e8400-e29b-41d4-a716-446655440000  
x-timestamp:1760000000
```

2.4 空请求体哈希

GET 请求 body 为空时, **HashedRequestBody** 固定为:

```
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.5 Python 签名示例

```
import base64
import hashlib
import hmac
import json
import time
import uuid

def sha256_hex(text: str) -> str:
    return hashlib.sha256(text.encode("utf-8")).hexdigest()

def sign(method: str, path: str, query_string: str, api_key: str, timestamp: str, nonce: str, body: str, api_secret:
str) -> str:
    canonical_headers = "\n".join([
        f"x-api-key:{api_key}",
        f"x-nonce:{nonce}",
        f"x-timestamp:{timestamp}",
    ])
    string_to_sign = "\n".join([
        method.upper(),
        path,
        query_string,
        canonical_headers,
        sha256_hex(body),
    ])
    digest = hmac.new(
        api_secret.encode("utf-8"),
        string_to_sign.encode("utf-8"),
        hashlib.sha256,
    ).digest()
    return base64.b64encode(digest).decode("utf-8")

api_key = "your_api_key"
api_secret = "your_api_secret"
timestamp = str(int(time.time()))
nonce = str(uuid.uuid4())
body = json.dumps({
    "duration_days": 30,
    "merchant_order_no": "M202604110001",
    "items": [
        {"region_id": 1, "quantity": 2}
    ]
}, separators=(",", ":"))

signature = sign(
    method="POST",
    path="/api/v1/orders",
    query_string="",
    api_key=api_key,
    timestamp=timestamp,
    nonce=nonce,
    body=body,
    api_secret=api_secret,
)
```

```
print(signature)
```

3. 通用响应格式

成功响应：

```
{
  "code": 0,
  "message": "success",
  "data": {}
}
```

失败响应：

```
{
  "code": 40001,
  "message": "Invalid request parameters"
}
```

3.1 业务错误码

code	含义
0	成功
40001	请求参数错误
40101	未认证或认证失败
40401	资源不存在
41001	签名错误
41002	余额不足
41003	库存不足或不可履约
50001	服务器内部错误

说明：

- 订单接口的 HTTP 状态码和业务码并不完全一一对应，请以 code 为准
- POST /orders/renewal 在余额不足时会返回 HTTP 402

4. 地区与库存说明

4.1 接口视图

Open API 以平台售卖地区为单位提供查询、购买和续费能力。

4.2 库存口径

`GET /regions` 返回当前用户资源策略范围内有实际可下单库存的上架地区、定价计划和分租期可售库存。外部资源商白名单会同时限制可见地区和库存；仅由白名单外资源商提供、没有符合租期条件的资源商或当前库存为零的地区都不会返回。客户端应从 `pricing_plans` 读取可下单租期，从 `stocks_by_duration` 判断对应租期的可售库存。

当前正式订单租期支持 7、30、60、90 天。某个地区是否可按目标租期下单，以该地区返回的 `pricing_plans` 和 `stocks_by_duration` 为准。

4.3 价格口径

`regions[].price` 是展示价：

- 优先取 30 天定价
- 若没有 30 天定价，则取该地区第一个定价计划

精确价格请始终以 `pricing_plans` 为准。

4.4 语言

地区名称受 `Accept-Language` 影响，推荐值：

- `zh-CN`
- `en-US`
- `zh-TW`

5. 健康检查

5.1 `GET /health`

无需签名。

响应示例：

```
{
  "code": 0,
  "message": "success",
  "data": "Open API service is running"
}
```

6. 查询余额

6.1 GET /balance

响应示例：

```
{
  "code": 0,
  "message": "success",
  "data": {
    "balance": "100.0000",
    "currency": "USD"
  }
}
```

7. 查询可售地区

7.1 GET /regions

返回当前用户资源策略范围内的上架地区、价格计划和分租期库存。

响应字段说明：

- `region_id`：地区 ID，下单时使用
- `region_name`：地区名称
- `country_code`：ISO 3166-1 alpha-2 国家码
- `net_type`：地区 IP 类型，原始值为 `native` 或 `broadcast`
- `line_class`：线路类型，原始值为 `unclassified`、`ntt` 或 `dua_ntt`
- `price`：展示价
- `stock`：展示库存
- `subnets`：当前可直接指定的 CIDR 库存
- `pricing_plans`：该地区所有定价计划
- `stocks_by_duration`：分租期库存

响应示例：

```
{
  "code": 0,
  "message": "success",
  "data": [
    {
      "region_id": 1,
      "region_name": "香港",
      "country_code": "HK",
      "net_type": "native",
      "line_class": "unclassified",
      "price": "3",
      "stock": 4947,
      "subnets": [
        {
          "subnet_cidr": "86.53.92.0/24",
          "stock": 215
        }
      ],
      "pricing_plans": [
        {
          "duration_days": 7,
          "price": "1"
        },
        {
          "duration_days": 30,
          "price": "3"
        },
        {
          "duration_days": 60,
          "price": "6"
        },
        {
          "duration_days": 90,
          "price": "9"
        }
      ],
      "stocks_by_duration": [
        {
          "duration_days": 7,
          "supported": true,
          "total_stock": 4947,
          "subnets": [
            {
              "subnet_cidr": "86.53.92.0/24",
              "stock": 215
            }
          ]
        },
        {
          "duration_days": 30,
          "supported": true,
          "total_stock": 4947,
          "subnets": [
            {
              "subnet_cidr": "86.53.92.0/24",
              "stock": 215
            }
          ]
        }
      ]
    }
  ]
}
```

```
    ]
  }
]
}
```

说明：

- `subnets` 表示当前可直接指定的子网库存
- `pricing_plans[].duration_days` 表示该地区已配置并可报价的租期
- 不同租期下的可售数量请以 `stocks_by_duration` 为准
- 只有至少一个允许租期存在实际可下单库存时才返回该地区
- 下单前建议先取目标租期对应的 `stocks_by_duration` 记录，确认 `supported=true` 且 `total_stock` 满足购买数量

8. 创建新购订单

8.1 POST /orders

请求体：

```
{
  "duration_days": 7,
  "merchant_order_no": "M202604110001",
  "idempotency_key": "idem_202604110001",
  "providers": [1, 2],
  "items": [
    {
      "region_id": 1,
      "quantity": 2,
      "request_cidr": "86.53.92.0/24"
    }
  ]
}
```

字段说明：

字段	必填	说明
<code>duration_days</code>	是	租期天数，支持 <code>7</code> 、 <code>30</code> 、 <code>60</code> 、 <code>90</code>
<code>merchant_order_no</code>	否	商户侧订单号
<code>idempotency_key</code>	否	幂等键；不传则不启用幂等
<code>providers</code>	否	可选来源限制参数，仅在平台提供该能力时传入，值为平台约定的来源 ID
<code>items</code>	是	订单项数组，至少 1 项
<code>items[].region_id</code>	是	地区 ID

字段	必填	说明
items[].quantity	是	购买数量
items[].request_cidr	否	指定购买的目标 CIDR

当前实现约束：

- region_id 必须是上架地区
- request_cidr 为可选参数
- 同一订单可包含同地区的多个订单项
- providers 是候选来源范围；系统会与用户允许的资源商范围及目标租期能力取交集
- 候选范围中不支持目标租期或已停用的资源商会被忽略；交集为空时拒绝下单
- 不传 providers 时使用用户配置的资源商范围；用户未限制资源商时使用平台当前可用范围
- 若传 idempotency_key，同一用户下相同键且请求参数一致会返回原订单；请求参数不一致会返回错误

成功响应示例：

```
{
  "code": 0,
  "message": "success",
  "data": {
    "order_no": "20260411000123",
    "merchant_order_no": "M202604110001",
    "total_amount": "2",
    "status": "provisioning",
    "created_at": "2026-04-11T10:00:00+08:00",
    "updated_at": "2026-04-11T10:00:00+08:00"
  }
}
```

常见失败场景：

- 地区不存在或未上架
- 定价计划不存在
- 余额不足
- 库存不足
- 指定来源不支持该租期

9. 创建续费订单

9.1 POST /orders/renewal

请求体：

```
{
  "duration_days": 7,
  "merchant_order_no": "M202604110002",
  "idempotency_key": "idem_renewal_202604110002",
  "order_proxy_account_ids": [1001, 1002]
}
```

字段说明：

字段	必填	说明
duration_days	是	续费天数，支持 7、30、60、90
merchant_order_no	否	商户侧续费单号
idempotency_key	否	幂等键；不传则不启用幂等
order_proxy_account_ids	是	待续费账号对应的订单关联账号 ID 列表

说明：

- 续费使用的是 order_proxy_account_id，不是 proxy_account_id
- 续费价格按账号当前所属出售区域的定价计划计算
- 续费会校验账号在当前租期下是否可续费
- 若混合选择多个账号，所有账号都必须支持当前续费天数
- 续费订单创建成功后进入异步处理
- 若传 idempotency_key，同一用户下相同键且请求参数一致会返回原订单；请求参数不一致会返回错误

成功响应示例：

```
{
  "code": 0,
  "message": "success",
  "data": {
    "order_no": "20260411000124",
    "merchant_order_no": "M202604110002",
    "total_amount": "6",
    "status": "provisioning",
    "created_at": "2026-04-11T10:05:00+08:00",
    "updated_at": "2026-04-11T10:05:00+08:00"
  }
}
```

10. 查询订单列表

10.1 GET /orders

查询参数：

参数	必填	说明
page	否	页码，默认 1
page_size	否	每页数量，默认 10，最大 100
status	否	订单状态过滤
start_date	否	创建时间起始
end_date	否	创建时间结束

示例：

```
GET /api/v1/orders?page=1&page_size=10&status=completed
```

响应示例：

```
{
  "code": 0,
  "message": "success",
  "data": [
    {
      "order_no": "20260411000123",
      "merchant_order_no": "M202604110001",
      "total_amount": "6",
      "order_type": "purchase",
      "status": "completed",
      "created_at": "2026-04-11T10:00:00+08:00",
      "updated_at": "2026-04-11T10:03:00+08:00",
      "regions": [
        {
          "region_id": 1,
          "region_name": "香港",
          "country_code": "HK",
          "quantity": 2,
          "unit_price": "3",
          "line_amount": "6",
          "subnets": [
            {
              "subnet_cidr": "86.53.92.0/24",
              "quantity": 2
            }
          ]
        }
      ]
    }
  ]
}
```

注意：

- 当前实现返回的是当前页订单数组
- 当前响应不返回 total、page、page_size

11. 查询订单详情

11.1 GET /orders/{order_no}

示例:

```
GET /api/v1/orders/20260411000123
```

响应示例:

```

{
  "code": 0,
  "message": "success",
  "data": {
    "order_no": "20260411000123",
    "merchant_order_no": "M202604110001",
    "total_amount": "6",
    "status": "completed",
    "reason": null,
    "error_message": null,
    "created_at": "2026-04-11T10:00:00+08:00",
    "updated_at": "2026-04-11T10:03:00+08:00",
    "proxy_accounts": [
      {
        "order_proxy_account_id": 1001,
        "username": "user_a",
        "password": "pass_a",
        "ip_address": "1.2.3.4",
        "host": "1.2.3.4",
        "port": 30001,
        "proxy_type": "http",
        "supported_proxy_types": ["http", "socks5"],
        "proxy_endpoints": [
          {
            "protocol": "http",
            "host": "1.2.3.4",
            "port": 30001,
            "username": "user_a",
            "password": "pass_a"
          },
          {
            "protocol": "socks5",
            "host": "1.2.3.4",
            "port": 30002,
            "username": "user_a",
            "password": "pass_a"
          }
        ],
        "expired_at": "2026-05-11T10:00:00+08:00",
        "expired_at_timestamp": 1778464800,
        "created_at": "2026-04-11T10:03:00+08:00",
        "created_at_timestamp": 1775872980,
        "region_id": 1,
        "region_name": "香港",
        "country_code": "HK",
        "subnet_cidr": "86.53.92.0/24",
        "supported_duration_days": [7, 30]
      }
    ]
  }
}

```

说明:

- `proxy_accounts` 中的 `order_proxy_account_id` 可直接用于后续续费
- `supported_proxy_types` 表示账号支持的代理协议，当前可能包含 `http`、`socks5`

- `proxy_endpoints` 按协议返回实际连接信息；HTTP 与 SOCKS5 端口不同时，应使用对应 `protocol` 下的 `host`、`port`、`username`、`password`
- 顶层 `host`、`port`、`username`、`password` 和 `proxy_type` 是兼容字段；新接入建议优先读取 `proxy_endpoints`
- `supported_duration_days` 表示该账号当前支持的续费天数列表
- 如账号带有对应网段信息，则会返回 `subnet_cidr`
- 当订单状态为 `failed` 或 `partial_failed` 时，`reason` 只会返回稳定的公开原因码：`insufficient_stock`、`provisioning_failed` 或 `system_error`
- `error_message` 只包含可公开的通用失败说明，不会返回内部接口、IP、端口、账号标识、堆栈信息或底层网络及数据库错误

12. 订单状态说明

Open API 对外实际需要关注的订单状态如下：

状态	说明
<code>provisioning</code>	分配中
<code>completed</code>	已完成
<code>failed</code>	失败
<code>partial_failed</code>	部分失败
<code>cancelled</code>	已取消
<code>refunded</code>	已退款
<code>partially_refunded</code>	部分退款

说明：

- 部分成功、部分失败时，对外统一返回 `partial_failed`
- 已成功交付的账号仍会出现在订单详情的 `proxy_accounts` 中
- `unpaid`、`paid` 属于内部兼容状态，不作为 Open API 对外状态说明

失败原因码：

原因码	说明
<code>insufficient_stock</code>	当前库存不足
<code>provisioning_failed</code>	代理资源开通失败
<code>system_error</code>	订单处理失败，且没有更具体的可公开原因

13. 接入建议

13.1 下单建议

- 查询地区请以 `pricing_plans` 和 `stocks_by_duration` 为准
- 购买前请先确认目标租期下 `supported=true`
- 若平台已向商户开放来源限制能力，可在新购接口中传 `providers`

13.2 续费建议

- 优先保存 `order_proxy_account_id`
- 续费时直接按 `order_proxy_account_ids` 提交
- 续费前建议先查订单详情确认账号仍属于当前用户