

UrbanOps 开放 API 对接接口文档

版本: v1.0

更新日期: 2026-06-11

适用对象: 第三方系统对接开发人员

鉴权方式: HMAC-SHA256 API 签名

目录

- 概述
- 签名鉴权机制
- 通用规范
- 接口 1: 人员查询
- 接口 2: 班组查询
- 接口 3: 考勤查询
- 接口 4: 工单查询
- 接口 5: 物料查询
- 接口 6: 全量工单统计
- 错误码说明
- 附录: 客户端签名代码示例

1. 概述

1.1 基本信息

项目	说明
接口协议	HTTPS
接口地址	<code>https://{host}/open-api/</code>
鉴权方式	HMAC-SHA256 API 签名（详见第2节）
响应格式	JSON
字符编码	UTF-8

1.2 接口列表

#	接口名称	路径	方法	说明
1	人员查询	/open-api/staff/query	GET	获取人员档案，支持按部门筛选
2	班组查询	/open-api/teams/query	GET	获取班组信息，含班组长和人员数量
3	考勤查询	/open-api/attendance/query	GET	获取考勤打卡数据
4	工单查询	/open-api/tasks/query	GET	获取工单全生命周期数据
5	物料查询	/open-api/materials/query	GET	获取物料消耗与照片数据
6	全量工单统计	/open-api/full-order-statistics/query	GET	获取公司/班组工单/巡查/养护完成数量

1.3 接入前提

接入前需向 UrbanOps 平台申请以下凭证（由运维人员配置到 Redis）：

参数	说明
appId	应用唯一标识（由平台分配）
appSecret	应用密钥（妥善保管，不可泄露）

2. 签名鉴权机制

2.1 签名参数

每个请求必须携带以下 4 个请求头：

请求头	类型	必填	说明
appId	String	✓	平台分配的应用 ID
timestamp	String	✓	当前 Unix 时间戳（毫秒），与服务器时间误差不能超过 60 秒
nonce	String	✓	随机字符串，长度 ≥ 10，每次请求必须不同（防重放）
sign	String	✓	签名字符串（SHA256 计算结果，小写十六进制）

2.2 签名算法

签名字符串构造规则

签名字符串 = queryParams (按key字母序排列的value拼接)
+ headerParams (按key字母序排列的value拼接)
+ appSecret

具体步骤：

1. **排序 Query 参数值**：将 URL 中所有 query 参数按 key 字母升序排列，拼接所有 value
2. **拼接 Header 参数值**：按 key 字母升序（即 `appId` → `nonce` → `timestamp`），依次拼接它们的值
3. **追加 appSecret**：在最后拼接 `appSecret`
4. **计算 SHA256**：对上述完整字符串计算 SHA256 哈希，转为小写十六进制字符串

签名公式

```
sign = SHA256(  
    sortedQueryParamValues  
    + appId值 + nonce值 + timestamp值  
    + appSecret  
)
```

注意：GET 请求无 body，body 部分为空字符串（不参与签名）。

2.3 签名示例

假设：

- `appId` = `my_app_001`
- `appSecret` = `abc123xyz`
- 请求 URL: `/open-api/staff/query?deptId=100`

步骤拆解：

```
queryParams(sorted by key):  
    deptId=100  
    → values = "100"  
  
headerParams(sorted by key):  
    appId=my_app_001, nonce=a1b2c3d4e5f6, timestamp=1749696000000  
    → values = "my_app_001" + "a1b2c3d4e5f6" + "1749696000000"  
  
signStr = "100" + "my_app_001" + "a1b2c3d4e5f6" + "1749696000000" + "abc123xyz"  
  
sign = SHA256(signStr)
```

2.4 签名校验规则

校验项	规则	失败返回
appld 非空	必须传递	400 签名不正确
timestamp 时效	<code> serverTime - timestamp ≤ 60s</code>	400 签名不正确
nonce 长度	<code>len(nonce) ≥ 10</code>	400 签名不正确
sign 非空	必须传递	400 签名不正确
签名比对	<code>serverSign == clientSign</code>	400 签名不正确
nonce 防重放	同一 appld + nonce 组合只能使用一次	900 存在重复请求

3. 通用规范

3.1 通用响应格式

```
{
  "code": 0,
  "msg": "成功",
  "data": [ ... ]
}
```

字段	类型	说明
code	Integer	0 表示成功, 非 0 表示失败
msg	String	提示信息
data	Array	业务数据列表（根据查询条件返回所有匹配记录）

3.2 日期时间格式

所有日期时间字段统一使用格式：`yyyy-MM-dd HH:mm:ss`

示例：`2026-06-11 09:00:00`

4. 接口 1：人员查询

4.1 基本信息

项目	说明
路径	/open-api/staff/query
方法	GET
说明	获取人员基础档案信息，支持按部门、姓名、手机号等条件筛选

4.2 请求参数

参数	类型	必填	说明
name	String	否	姓名，模糊匹配
mobile	String	否	手机号，模糊匹配
deptId	Long	否	部门ID，精确匹配
status	Integer	否	状态：0=启用，1=禁用
busiLine	String	否	业务线：yl=园林，wy=物业，sz=市政

4.3 响应字段

字段	类型	说明
id	Long	人员ID
username	String	用户名
nickname	String	姓名
mobile	String	手机号
deptId	Long	部门ID
deptName	String	部门名称（如"养护一班"）
postIds	Array<Long>	岗位ID列表
postNames	String	岗位名称（逗号分隔，如"班组长,技术员"）
busiLine	String	业务线
status	Integer	状态（0=启用，1=禁用）
cardNumber	String	工卡号（设备编号，用于考勤打卡）
isInner	Integer	是否内部人员（1=内部，2=外部）
createTime	String	入职日期（格式：yyyy-MM-dd HH:mm:ss）

4.4 请求示例

```
curl -X GET "https://test.jichengshanshui.com.cn:28302/open-api/staff/query?deptId=100" \
-H "appId: my_app_001" \
-H "timestamp: 1749696000000" \
-H "nonce: a1b2c3d4e5f6g7h8" \
-H "sign: 3a5f8c9e2b1d4f6a8c0e2b4d6f8a0c2e4b6d8f0a"
```

4.5 响应示例

```
{
  "code": 0,
  "msg": "成功",
  "data": [
    {
      "id": 1001,
      "username": "zhangsan",
      "nickname": "张三",
      "mobile": "13800138001",
      "deptId": 100,
      "deptName": "养护一班",
      "postIds": [10, 20],
      "postNames": "班组长,技术员",
      "busiLine": "yl",
      "status": 0,
      "cardNumber": "DEV001",
      "isInner": 1,
      "createTime": "2024-01-15 00:00:00"
    }
  ],
  "total": 150
}
```

5. 接口 2：班组查询

5.1 基本信息

项目	说明
路径	/open-api/teams/query
方法	GET
说明	获取组织架构班组信息，含班组长、所属养护所、人员数量

5.2 请求参数

参数	类型	必填	说明
name	String	否	班组名称，模糊匹配
parentId	Long	否	父部门ID（上级组织），精确匹配
status	Integer	否	状态：0=启用，1=禁用

5.3 响应字段

字段	类型	说明
id	Long	班组ID
name	String	班组名称
parentId	Long	父部门ID
parentName	String	父部门名称（所属养护所）
leaderUserId	Long	班组长人员ID（Staff.id）
leaderUserName	String	班组长姓名
memberCount	Long	人员数量
sort	Integer	排序
status	Integer	状态（0=启用，1=禁用）

5.4 请求示例

```
curl -X GET "https://test.jichengshanshui.com.cn:28302/open-api/teams/query" \
-H "appId: my_app_001" \
-H "timestamp: 1749696000000" \
-H "nonce: b2c3d4e5f6g7h8i9" \
-H "sign: ..."
```

5.5 响应示例

```
{
  "code": 0,
  "msg": "成功",
  "data": [
    {
      "id": 100,
      "name": "养护一班",
      "parentId": 1,
      "parentName": "蓟城山水集团",
      "leaderUserId": 1001,
      "leaderUserName": "张三",
      "memberCount": 15,
      "sort": 1,
      "status": 0
    }
  ],
  "total": 12
}
```

6. 接口 3：考勤查询

6.1 基本信息

项目	说明
路径	/open-api/attendance/query
方法	GET
说明	获取电子围栏打卡与位置数据，支持按班组、人员、时间范围筛选

6.2 请求参数

参数	类型	必填	说明
userId	Long	否	人员ID，精确匹配
deptId	Long	否	部门ID（班组），精确匹配
busiLine	String	否	业务线
punchType	String	否	打卡类型（上班/下班）
startTime	String	否	开始时间（格式：yyyy-MM-dd HH:mm:ss）
endTime	String	否	结束时间（格式：yyyy-MM-dd HH:mm:ss）

6.3 响应字段

字段	类型	说明
id	Integer	记录ID
deviceCode	String	设备编码
deviceName	String	设备名称
punchType	String	打卡类型
userId	Long	人员ID
nickname	String	人员姓名
deptId	Long	部门ID
deptName	String	部门名称
address	String	打卡地址
latGcj02	BigDecimal	高德纬度（GCJ-02坐标系）
lngGcj02	BigDecimal	高德经度（GCJ-02坐标系）
lat	BigDecimal	WGS84纬度
lng	BigDecimal	WGS84经度
createTime	String	打卡时间

注意：当前版本暂不支持"轨迹点集""是否佩戴工卡""记录仪状态"字段（数据库无对应存储）。待业务系统确认后将补充。

6.4 响应示例

```
{
  "code": 0,
  "msg": "成功",
  "data": [
    {
      "id": 1,
      "deviceCode": "DEV001",
      "deviceName": "打卡机01",
      "punchType": "上班",
      "userId": 1001,
      "nickname": "张三",
      "deptId": 100,
      "deptName": "养护一班",
      "address": "北京市西城区xx路xx号",
      "latGcj02": 39.9042,
      "lngGcj02": 116.4074,
      "lat": 39.9010,
      "lng": 116.4030,
      "createTime": "2026-06-11 08:30:00"
    }
  ],
  "total": 320
}
```

7. 接口 4：工单查询

7.1 基本信息

项目	说明
路径	/open-api/tasks/query
方法	GET
说明	获取工单全生命周期数据，含发布时间、状态、接单渠道、共同处理人

7.2 请求参数

参数	类型	必填	说明
orderNo	String	否	工单号，模糊匹配
orderName	String	否	工单名称，模糊匹配
busiLine	String	否	业务线
status	Integer	否	审批状态
buzStatus	String	否	业务状态
workerId	Long	否	接单人ID
startDate	String	否	发布时间起始（yyyy-MM-dd HH:mm:ss）
endDate	String	否	发布时间截止（yyyy-MM-dd HH:mm:ss）

7.3 响应字段

字段	类型	说明
id	Long	工单ID
orderNo	String	工单号
orderName	String	工单名称
orderType	String	工单类型：Q=快速，C=普通，O=其他
busiLine	String	业务线
commitDate	String	发布时间（工单派发时间）
finishDate	String	完成时间
userId	Long	发布人ID
userName	String	发布人姓名
workerId	Long	接单人ID
workerName	String	接单人姓名
coHandlers	Array<String>	共同处理人姓名列表
woSourceName	String	接单渠道（小程序/人工派单等）
status	Integer	审批状态
buzStatus	String	业务状态
remark	String	备注
photos	Array<PhotoVO>	工单附件照片

PhotoVO 结构：

字段	类型	说明
busiType	String	业务类型：01 =问题照片， 02 =处理中， 03 =完成
fileName	String	文件名
url	String	文件URL

7.4 响应示例

```
{
  "code": 0,
  "msg": "成功",
  "data": [
    {
      "id": 5001,
      "orderNo": "W0202606110001",
      "orderName": "树木修剪",
      "orderType": "C",
      "busiLine": "yl",
      "commitDate": "2026-06-11 09:00:00",
      "finishDate": null,
      "userId": 1,
      "userName": "管理员",
      "workerId": 1001,
      "workerName": "张三",
      "coHandlers": ["李四", "王五"],
      "woSourceName": "小程序",
      "status": 1,
      "buzStatus": "processing",
      "remark": "紧急处理",
      "photos": [
        {
          "busiType": "01",
          "fileName": "problem.jpg",
          "url": "https://xxx.com/file/abc.jpg"
        }
      ]
    }
  ],
  "total": 85
}
```

8. 接口 5：物料查询

8.1 基本信息

项目	说明
路径	/open-api/materials/query
方法	GET
说明	获取物料消耗记录与人机材照片数据，支持按工单筛选

8.2 请求参数

参数	类型	必填	说明
orderNo	String	否	工单号，精确匹配
classifyName	String	否	物料分类名，模糊匹配
typeName	String	否	物料类型名，模糊匹配
userId	Long	否	使用人ID
startDate	String	否	开始时间（yyyy-MM-dd HH:mm:ss）
endDate	String	否	结束时间（yyyy-MM-dd HH:mm:ss）

8.3 响应字段

字段	类型	说明
id	Long	记录ID
orderNo	String	关联工单号
materialName	String	物料名称
specifications	String	规格
userCount	Long	使用数量
unitName	String	单位
classifyName	String	物料分类
typeName	String	物料类型
typeDetailName	String	物料明细类型
userId	Long	使用人ID
userName	String	使用人姓名
source	Integer	来源：1=工单，2=巡检，3=养护
createTime	String	上传时间
photos	Array<PhotoVO>	人机材照片（结构同工单接口）

8.4 响应示例

```
{
  "code": 0,
  "msg": "成功",
  "data": [
    {
      "id": 3001,
      "orderNo": "W0202606110001",
      "materialName": "高效氯氰菊酯",
      "specifications": "500ml/瓶",
      "userCount": 5,
      "unitName": "瓶",
      "classifyName": "药品",
      "typeName": "杀虫剂",
      "typeDetailName": "有机磷",
      "userId": 1001,
      "userName": "张三",
      "source": 1,
      "createTime": "2026-06-11 14:30:00",
      "photos": [
        {
          "busiType": "02",
          "fileName": "material.jpg",
          "url": "https://xxx.com/file/def.jpg"
        }
      ]
    }
  ],
  "total": 45
}
```

9. 接口 6：全量工单统计

9.1 基本信息

项目	说明
路径	/open-api/full-order-statistics/query
方法	GET
说明	获取各公司/班组在一段时间内实际完成的工单数量、巡查记录数量、养护记录数量

9.2 请求参数

参数	类型	必填	说明
companyId	Long	否	公司/班组ID，精确匹配
companyName	String	否	公司/班组名称，模糊匹配
type	Integer	否	类型：1=按公司，2=按班组
startDate	String	否	统计开始日期（格式：yyyy-MM-dd，返回开始日期>=该值的记录）
endDate	String	否	统计结束日期（格式：yyyy-MM-dd，返回结束日期<=该值的记录）

9.3 响应字段

字段	类型	说明
id	Long	记录ID
companyId	Long	公司/班组ID
companyName	String	公司/班组名称
type	Integer	类型：1=按公司，2=按班组
startDate	String	统计开始日期
endDate	String	统计结束日期
workOrderCount	Integer	工单完成数量
inspectionCount	Integer	巡查记录数量
maintenanceCount	Integer	养护记录数量
remark	String	备注
createTime	String	创建时间

9.4 请求示例

```
curl -X GET "https://test.jichengshanshui.com.cn:28302/open-api/full-order-statistics/query?type=
-H "appId: my_app_001" \
-H "timestamp: 1749696000000" \
-H "nonce: f6g7h8i9j0k1l2m3" \
-H "sign: ..."
```

9.5 响应示例

```
{
  "code": 0,
  "msg": "成功",
  "data": [
    {
      "id": 1,
      "companyId": 100,
      "companyName": "养护一班",
      "type": 1,
      "startDate": "2026-06-01",
      "endDate": "2026-06-30",
      "workOrderCount": 85,
      "inspectionCount": 120,
      "maintenanceCount": 60,
      "remark": "二季度统计",
      "createTime": "2026-06-11 10:00:00"
    }
  ]
}
```

10. 错误码说明

10.1 签名相关错误

10.2 业务相关错误

code	msg	说明
400	签名不正确	签名验证失败（appId不存在/timestamp超时/nonce长度不足/签名不匹配）
900	存在重复请求	nonce 已被使用（重放攻击检测）

10.3 通用错误

code	msg	说明
1-100-020-001	人员查询失败	人员查询内部错误
1-100-020-002	班组查询失败	班组查询内部错误
1-100-020-003	考勤查询失败	考勤查询内部错误
1-100-020-004	工单查询失败	工单查询内部错误
1-100-020-005	物料查询失败	物料查询内部错误

| 1-900-005-001 | 全量工单统计不存在 | 记录不存在 |
| 1-900-005-002 | 相同公司/班组同时间段统计已存在 | 重复数据校验 |

10.3 通用错误

code	msg	说明
400	参数校验失败	请求参数格式或值不正确
500	系统内部错误	服务器内部异常

11. 附录：客户端签名代码示例

10.1 Java 示例

```
import cn.hutool.crypto.digest.DigestUtil;
import cn.hutool.http.HttpRequest;
import cn.hutool.http.HttpResponse;
import cn.hutool.core.util.RandomUtil;

import java.util.Map;
import java.util.TreeMap;

public class OpenApiClient {

    private static final String BASE_URL = "https://test.jichengshanshui.com.cn:28302";
    private static final String APP_ID = "your_app_id";
    private static final String APP_SECRET = "your_app_secret";

    public static String call(String path, Map<String, Object> params) {
        String timestamp = String.valueOf(System.currentTimeMillis());
        String nonce = RandomUtil.randomString(16);
        String sign = buildSign(params, timestamp, nonce);

        String url = BASE_URL + path;
        HttpResponse response = HttpRequest.get(url)
            .form(params)
            .header("appId", APP_ID)
            .header("timestamp", timestamp)
            .header("nonce", nonce)
            .header("sign", sign)
            .execute();
        return response.body();
    }

    private static String buildSign(Map<String, Object> params,
                                    String timestamp, String nonce) {
        StringBuilder sb = new StringBuilder();

        // 1. 排序 query 参数值
        if (params != null && !params.isEmpty()) {
            new TreeMap<>(params).values().stream()
                .map(String::valueOf).forEach(sb::append);
        }

        // 2. header 参数值 (appId, nonce, timestamp 字母序)
        sb.append(APP_ID);
        sb.append(nonce);
        sb.append(timestamp);

        // 3. 追加 appSecret + SHA256
        sb.append(APP_SECRET);
        return DigestUtil.sha256Hex(sb.toString());
    }
}
```

10.2 Python 示例

```
import hashlib
import time
import random
import string
import requests

BASE_URL = "https://test.jichengshanshui.com.cn:28302"
APP_ID = "your_app_id"
APP_SECRET = "your_app_secret"

def build_sign(params, timestamp, nonce):
    parts = []
    if params:
        for key in sorted(params.keys()):
            parts.append(str(params[key]))
    parts.append(APP_ID)
    parts.append(nonce)
    parts.append(timestamp)
    parts.append(APP_SECRET)
    sign_str = "".join(parts)
    return hashlib.sha256(sign_str.encode()).hexdigest()

def call(path, params=None):
    if params is None:
        params = {}
    timestamp = str(int(time.time() * 1000))
    nonce = "".join(random.choices(string.ascii_lowercase + string.digits, k=16))
    sign = build_sign(params, timestamp, nonce)
    headers = {
        "appId": APP_ID,
        "timestamp": timestamp,
        "nonce": nonce,
        "sign": sign
    }
    response = requests.get(BASE_URL + path, params=params, headers=headers)
    return response.json()
```

10.3 Shell (cURL) 示例

```
#!/bin/bash
APP_ID="your_app_id"
APP_SECRET="your_app_secret"
TIMESTAMP=$(date +%s%3N)
NONCE=$(openssl rand -hex 16)

# 示例: /open-api/staff/query?deptId=100
# query params 排序后 values: deptId=100 → "100"
QUERY_VALUES="100"
HEADER_VALUES="${APP_ID}${NONCE}${TIMESTAMP}"
SIGN_STR="${QUERY_VALUES}${HEADER_VALUES}${APP_SECRET}"
SIGN=$(echo -n "$SIGN_STR" | openssl dgst -sha256 -hex | awk '{print $2}')

curl -X GET "https://test.jichengshanshui.com.cn:28302/open-api/staff/query?deptId=100" \
-H "appId: ${APP_ID}" \
-H "timestamp: ${TIMESTAMP}" \
-H "nonce: ${NONCE}" \
-H "sign: ${SIGN}"
```

本文档基于 2026-06-11 代码生成，如有接口变更请同步更新。