

SHA1签名生成规则

1.获取 **userid**、**secretkey** 签名关键信息，具体见[获取开发密钥](#)，**secretkey**密钥用于生成签名，作为通信数据安全的关键信息，**严禁保存在客户端直接使用，所有API都必须通过客户自己服务器中转调用POLYV服务器获取响应数据**

2.采用**SHA1**签名算法生成签名，具体加密计算方法如下：

2.1将请求所需参数（**参数值非空的参数**）按照**参数名字典顺序**排列，采用HTTP GET请求参数连接方式（key=value&key=value&key=value）拼接参数名与参数值，例如：
date=2020-10-13&pstime=1621997347695&userid=1b448be323

2.2并在字符串**尾部**加上**secretkey**，以secretKey为abc为例，得到：
date=2020-10-13&pstime=1621997347695&userid=1b448be323abc

2.3然后计算**SHA1**，将**SHA1**结果转为**大写字母**，作为sign;

常见问题：

- 1.字符串拼接时没有将参数值为null的参数剔除；
- 2.签名自用字符集必须为UTF-8，若不指定，可能采用平台默认字符集，导致错误；

快速接入基础代码请[下载相关依赖源码](#)，[点击下载源代码](#)，下载后加入到自己的源码工程中即可。测试用例中的**HttpUtil.java** 和 **VodSignUtil.java** 都包含在下载文件中。

强烈建议您使用[点播Java SDK](#)完成API的功能对接，点播Java SDK 对API调用逻辑、异常处理、数据签名、HTTP请求线程池进行了统一封装和优化。

```
package net.polyv.common;

import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.UUID;

import org.junit.Test;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import net.polyv.util.VodSignUtil;

/**
 * @author: thomas
 */
public class VodSignTest {

    private static final Logger log = LoggerFactory.getLogger(VodSignTest.class);

    @Test
    public void buildSign() throws UnsupportedEncodingException, NoSuchAlgorithmException {
        String userid = "XXXXXXXX";

        String secretkey = "XXXXXXXXXXXXXXXXXXXXXXXXX";

        long timestamp = System.currentTimeMillis();
        Map<String, String> paramMap = new HashMap<String, String>();
        //公共参数
        paramMap.put("userid", userid);
        paramMap.put("ptime", Long.toString(timestamp));
        //业务参数
        paramMap.put("requestId", UUID.randomUUID().toString().replaceAll("-", ""));
        paramMap.put("vid", "1b448be3233659acf35d430ba9210bd4_1");

        String sign = VodSignUtil.getSign(paramMap, secretkey);
        log.debug("生成签名: {}", sign);
    }
}
```