签名生成规则

1.获取userId 、 appId 、 appSecret关键信息，具体见获取开发密钥， `appSecret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**MD5**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，连接参数名与参数值，例如：
ts1552447784505userIde6b23c6f51videoIde6b23c6f51c4b1cb9f0302a92ed42440_eviewerIdabcd1234viewerIp127.0.0.1;

2.2并在拼接字符串**首尾**加上appSecret，加上appSecret，以appSecret为abc为例，得到：
abcts1552447784505userIde6b23c6f51videoIde6b23c6f51c4b1cb9f0302a92ed42440_eviewerIdabcd1234viewerIp127.0.0.1abc

2.3然后采用UTF-8编码计算**MD5**，将**MD5**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 LiveSignUtil.java** 都包含在下载文件中。

```
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.LiveSignUtil;


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

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

    @Test
    public void buildSign() throws UnsupportedEncodingException, NoSuchAlgorithmException {
        String appId = "XXXXXXXX";
        String userId = "XXXXXXXX";
        String appSecret = "XXXXXXXXXXXXXXXXXXXXXXXX";

        long timestamp = System.currentTimeMillis();
        Map<String, String> paramMap = new HashMap<String, String>();
        //公共参数
        paramMap.put("appId", appId);
        paramMap.put("timestamp", Long.toString(timestamp));
        //业务参数
        paramMap.put("lessonId","2149813");

        String sign = LiveSignUtil.getSign(paramMap, appSecret);
        log.debug("生成签名：{}",sign);

    }


}
```