

发送阿里云短信通知

接口描述

- 1、根据用户配置的模板发送短信通知
- 2、(timestamp, appId) 参与sign签名, 并和sign一起通过url传递, 请求体参数不参与签名, 通过post请求体传递【请设置请求头contentType:application/json】
- 3、接口支持https协议

接口URL

http://api.polyv.net/live/v4/user/sms/send

请求方式

POST

接口约束

- 1、接口同时支持HTTP 、HTTPS , 建议使用HTTPS 确保接口安全, 接口调用有频率限制, [详细请查看](#)
- 2、接口调用频率限制, 单个用户、IP每日短信发送上限2000, 单个手机号每日发送上限10条
- 3、使用前账号需要配置阿里云短信模板

请求参数描述

参数名	必选	类型	说明
appId	true	String	账号appId【详见 获取密钥 】
timestamp	true	Long	当前13位毫秒级时间戳, 3分钟内有效

参数名	必选	类型	说明
sign	true	String	签名，为32位大写的MD5值，生成签名的appSecret密钥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【详见 签名生成规则 】

请求体参数描述

参数名	必选	类型	说明
phoneNumbers	true	Array	手机号，长度11位，最大200个
templateParamNames	true	Array	短信模板的参数名
templateParamValues	true	Array	短信模板参数名对应的参数值，与参数名一一对应

示例

```
http://api.polyv.net/live/v4/user/sms/send?
appId=frlr1zazn3&timestamp=1656903040000&sign=9E0DCCEAB6A9D255717C9BEAB66CFDCA
```

请求体json参数：

```
{
  "phoneNumbers": [1768909,17680476163],
  "templateParamNames": ["code"],
  "templateParamValues": [9802]
}
```

响应参数描述

参数名	类型	说明
code	Integer	状态码，与 http 状态码相同，用于确定基本的响应状态
status	String	响应结果，由业务决定，成功返回 success，失败返回error
success	Boolean	是否成功响应
requestId	String	请求ID，每次请求生成的唯一的UUID，仅可用于排查、调试，不应该和业务挂上钩
error	Object	状态码非200时的错误信息【详见 Error字段描述 】
data	Array	【详见 data字段说明 】

Error参数描述

参数名	类型	说明
code	Integer	错误代码，用于确定具体的错误原因
desc	String	错误描述，与 error.code 对应

data参数描述

参数名	类型	说明
phoneNumber	String	手机号

参数名	类型	说明
status	Integer	短信状态 -1：发送失败，手机号超过每日发送频次限制 -2：发送失败，手机号不合法（非数字、长度错误） 11：发送失败（调用阿里云短信接口失败） 12：发送成功（调用阿里云短信接口成功，进入短信发送队列等待发送，并非成功发送到手机）
serialNo	String	发送成功时返回阿里云回执ID流水号

Java请求示例

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。

强烈建议您使用[直播Java SDK](#)完成API的功能对接，直播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。

```

private static final Logger log = LoggerFactory.getLogger(AccountTest.class);
/**
 * 发送阿里云短信通知
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void sendSmsTest() throws IOException, NoSuchAlgorithmException {
    //公共参数,填写自己的实际参数
    String appId = super.appId;
    String appSecret = super.appSecret;
    String timestamp = String.valueOf(System.currentTimeMillis());

    //业务参数
    String url = "http://api.polyv.net/live/v4/user/sms/send";
    List<String> phoneNumbers = Arrays.asList("17677777777");
    List<String> templateParamNames = Arrays.asList("code");
    List<String> templateParamValues = Arrays.asList("9876");

    //http 调用逻辑
    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp", timestamp);

    Map<String, Object> jsonMap = new HashMap<>();
    jsonMap.put("phoneNumbers", phoneNumbers);
    jsonMap.put("templateParamNames", templateParamNames);
    jsonMap.put("templateParamValues", templateParamValues);

    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));

    url = HttpUtil.appendUrl(url, requestMap);
    String response = HttpUtil.postJsonBody(url, JSON.toJSONString(jsonMap), null);

    log.info("测试发送短信通知成功: {}", response);
    //do somethings
}

```

响应示例

系统全局错误说明详见[全局错误说明](#)

成功示例

```
{
  "code": 200,
  "status": "success",
  "requestId": "f2a4a564-4dd6-4a9f-b58e-05580cd69939",
  "data": [
    {
      "serialNo": "605320356903663783^0",
      "phoneNumber": "17680476163",
      "status": 12
    },
    {
      "serialNo": null,
      "phoneNumber": "1768909",
      "status": -2
    }
  ],
  "success": true
}
```

异常示例

```
{
  "code": 400,
  "status": "error",
  "requestId": "d310b70bc329403f87f77f9203d50f89.128.16360831552223589",
  "error": {
    "code": 20001,
    "desc": "application not found."
  },
  "success": false
}
```