

修改配置信息

接口描述

- 1、修改配置信息

2、接口支持https协议

接口URL

https://api.polyv.net/live/v4/user/viewer-user-system/update-config

[在线API调用](#)

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](#)

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
mobileLoginEnabled	true	String	手机登录开关，值为Y或N
wxWorkLoginEnabled	true	String	企微登录开关，值为Y或N
viewerWeixinAuthExpired	false	Integer	微信授权有效期，单位天，范围0-180
collectMobileEnabled	false	String	引导绑定手机号开关，值为Y或N
guestModeEnabled	false	String	游客模式登录开关，值为Y或N
touristExternalHrefEnabled	false	String	游客外部登录开关，值为Y或N
touristExternalHrefConfig	false	Object	游客外部登录链接配置【详见 touristExternalHrefConfig字段说明 】

touristExternalHrefConfig参数描述

参数名	必选	类型	说明
pcLink	false	String	pc端链接
mobileLink	false	String	移动端链接

参数名	必选	类型	说明
androidLink	false	String	安卓端链接
iosLink	false	String	ios端链接
otherLink	false	String	其他端链接
wxMiniprogramLink	false	String	微信小程序链接
wxMiniprogramOriginalId	false	String	微信小程序原始ID
wxMiniprogramAppId	false	String	微信小程序APP ID
mobileAppLink	false	String	手机APP端链接
harmonyLink	false	String	鸿蒙端链接
jumpWay	false	String	跳转方式，默认 NEW_WINDOW

示例

```
https://api.polyv.net/live/v4/user/viewer-user-system/update-config?
appId=frlr1zazn3&sign=012332FBD8DBCFC068AFCB484A2ADECB&timestamp=1670550787318
```

请求体json参数：

```
{
  "mobileLoginEnabled": "Y",
  "wxWorkLoginEnabled": "N",
  "viewerWeixinAuthExpired": 30,
  "collectMobileEnabled": "Y",
  "guestModeEnabled": "Y",
  "touristExternalHrefEnabled": "Y",
  "touristExternalHrefConfig": {
    "mobileLink": "https://example.com/mobile",
    "pcLink": "https://example.com/pc",
    "jumpWay": "NEW_WINDOW"
  }
}
```

响应体json:

```
{
  "code": 200,
  "status": "success",
  "requestId": "97d9bc9c-c0d2-4639-acad-54b28335004c",
  "data": null,
  "success": true
}
```

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	Object	响应数据，成功时返回null

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
-	-	成功时返回null

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 final Logger log = LoggerFactory.getLogger(getClass());
/**
 * 修改配置信息
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void updateConfigTest() 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());

    String url = "https://api.polyv.net/live/v4/user/viewer-user-system/update-config";
    //业务参数
    String mobileLoginEnabled = "Y";
    String wxWorkLoginEnabled = "N";
    Integer viewerWeixinAuthExpired = 30;
    String collectMobileEnabled = "Y";
    String guestModeEnabled = "Y";
    String touristExternalHrefEnabled = "Y";

    //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("mobileLoginEnabled", mobileLoginEnabled);
    jsonMap.put("wxWorkLoginEnabled", wxWorkLoginEnabled);
    jsonMap.put("viewerWeixinAuthExpired", viewerWeixinAuthExpired);
    jsonMap.put("collectMobileEnabled", collectMobileEnabled);
    jsonMap.put("guestModeEnabled", guestModeEnabled);
    jsonMap.put("touristExternalHrefEnabled", touristExternalHrefEnabled);

    Map<String, Object> touristExternalHrefConfig = new HashMap<>();
    touristExternalHrefConfig.put("mobileLink", "https://example.com/mobile");
    touristExternalHrefConfig.put("pcLink", "https://example.com/pc");
    touristExternalHrefConfig.put("jumpWay", "NEW_WINDOW");
    jsonMap.put("touristExternalHrefConfig", touristExternalHrefConfig);

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

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

    log.info("测试修改配置信息成功: {}", 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": "97d9bc9c-c0d2-4639-acad-54b28335004c",
  "data": null,
  "success": true
}
```

异常示例

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