# 修改默认模板内容保护（防录屏）设置

## 接口描述

```
1、更新内容保护默认模板设置
2、接口支持https协议
```

## 接口URL

```
http://api.polyv.net/live/v4/user/template/marquee/update
```

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】 |

| 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】 |
| timestamp | true | String | 当前13位毫秒级时间戳，3分钟内有效 |

## 请求体参数描述

| 参数名 | 必选 | 类型 | 说明 |
| --- | --- | --- | --- |
| enable | true | String | 内容保护开关<br>Y：开启<br>N：关闭 |
| antiRecordType | false | String | 内容保护方式，内容保护开关开启时必填<br>marquee：跑马灯<br>watermark：水印 |
| modelType | false | String | 内容保护类型，内容保护开关开启时必填，水印方式时设置自定义URL无效<br>fixed：固定<br>nickname：用户名<br>diyurl：url自定义设置，仅跑马灯方式有效 |
| content | false | String | 内容保护类型是固定值时为设置内容<br>URL自定义设置时为网址，需要携带 http:// 或 https://<br>内容保护类型为登录用户名时可不传，固定值和url自定义设置时必传 |

| 参数名 | 必选 | 类型 | 说明 |
| --- | --- | --- | --- |
| opacity | false | Integer | 跑马灯透明度，范围0-99<br>水印透明度，范围0-100 |
| fontSize | false | String | 字体大小<br>内容保护方式为跑马灯时：设置数值，范围1-256<br>内容保护方式为水印时：<br>small：小<br>middle：中<br>large：大 |
| fontColor | false | String | 跑马灯字体颜色，色值，例如：#FFFFFF |
| showMode | false | String | 跑马灯显示方式<br>roll：滚动<br>flicker：闪烁 |
| doubleEnabled | false | String | 双跑马灯开关<br>Y：开启<br>N：关闭 |
| autoZoomEnabled | false | String | 自定义缩放开关<br>Y：开启<br>N：关闭 |

## 示例

```
http://api.polyv.net/live/v4/user/template/marquee/update?
appId=frlr1zazn3&sign=EC5A0EF79FAE452F1393A33B10AD6173&timestamp=1677201932605
```

请求体json参数：

```
{
    "doubleEnabled": "N",
    "autoZoomEnabled": "Y",
    "enable": "Y",
    "fontSize": "20",
    "showMode": "flicker",
    "modelType": "fixed",
    "opacity": "80",
    "content": "测试跑马灯内容test",
    "fontColor": "#ff4d4f",
    "antiRecordType": "marquee"
}
```


## 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 |
| data | Object | 成功响应的数据 |
| error | Object | 状态码非200时的错误信息【详见Error参数描述】 |
| 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 |
| 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 |

## 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 |

## 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(getClass());
/**
 * 更新内容保护默认模板设置
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void updateMarqueeTemplateTest() 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/template/marquee/update";
    String antiRecordType = "marquee";
    String autoZoomEnabled = "Y";
    String content = "测试跑马灯内容test";
    String doubleEnabled = "N";
    String enable = "Y";
    String fontColor = "#ff4d4f";
    String fontSize = "20";
    String modelType = "fixed";
    String opacity = "80";
    String showMode = "flicker";

    //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("antiRecordType", antiRecordType);
    jsonMap.put("autoZoomEnabled", autoZoomEnabled);
    jsonMap.put("content", content);
    jsonMap.put("doubleEnabled", doubleEnabled);
    jsonMap.put("enable", enable);
    jsonMap.put("fontColor", fontColor);
    jsonMap.put("fontSize", fontSize);
    jsonMap.put("modelType", modelType);
    jsonMap.put("opacity", opacity);
    jsonMap.put("showMode", showMode);

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

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

    log.info("测试更新内容保护默认模板设置成功：{}", 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": "a5e9b1d15f544d7a9db6f038156a851d.67.16772019328880111",
    "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
}
```