

创建商品

接口描述

1、创建标签
2、接口支持https协议

接口URL

https://api.polyv.net/live/v4/user/product/save

[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
name	true	String	商品名称
cover	false	String	封面图, normal类型商品值不为空
linkType	true	Integer	商品链接类型, 10:通用链接,11:多平台链接 ,12:调用原生方法
link	true	String	商品链接不能为空
productType	false	String	商品类型，可选值有： normal: 普通商品 finance: 金融商品 position:职位商品 不传时默认为 normal
pcLink	false	String	pc端链接
mobileLink	false	String	移动端链接
wxMiniprogramOriginalId	false	String	微信小程序原始ID
wxMiniprogramLink	false	String	微信小程序链接
mobileAppLink	false	String	手机APP链接
iosLink	false	String	ios链接

参数名	必选	类型	说明
androidLink	false	String	android链接
otherLink	false	String	其他端链接
features	false	String	商品特色,卖点标签, 可以多个,json数组
tagIds	false	Array	商品标签ID列表
btnShow	false	String	按钮显示文案
productDesc	false	String	商品描述
productDetail	false	String	商品详情
ext	false	String	扩展信息: JSON对象字符串: {"coverList":["封面图地址, 最多15个"], "videoList": [{"videoSource":"视频来源", "videoId":"视频ID"}]} 【详情见 EXT 】 cover和ext选择其中一个字段传, 如果同时传以ext为准
-	-	-	以下为普通商品专用参数 (productType=normal)
priceType	false	String	商品实际价格类型, 可选值有: AMOUNT: 金额 CUSTOM: 自定义 不传时默认为 AMOUNT
realPrice	false	Float	商品实际价格-金额, priceType=AMOUNT 时必传, 最高精确度为两位小数, 如 12.34
customPrice	false	String	商品实际价格-自定义, priceType=CUSTOM 时必传, 最多20个字符

参数名	必选	类型	说明
originalPriceType	false	String	商品原始价格（划线价格）类型，可选值有： AMOUNT: 金额 CUSTOM: 自定义 默认为 AMOUNT
price	false	Float	商品原始价格（划线价格）-金额， originalPriceType=AMOUNT 时生效，最高精确度为两位小数，如 12.34
customOriginalPrice	false	String	商品原始价格（划线价格）-自定义， originalPriceType=CUSTOM 时生效，最多20个字符 备注：如果 price 和 customOriginalPrice 都为空，则表示不启用原始价格（划线价格）
-	-	-	以上为普通商品专用参数 (productType=normal)

EXT

参数名	必选	类型	说明
coverList	true	Array	封面图地址数组，最多15个
videoList	false	Array	视频数组，最多1个【详情见 ChannelProductExtVideo 】需要开通素材库才能用

参数名	必选	类型	说明
jumpWay	false	String	购买方式为外链购买-通用链接的跳转方式: POP_UP-弹出框打开、 NEW_WINDOW-新窗口打开、 CURRENT_WINDOW-当前窗口打开

ChannelProductExtVideo

参数名	必选	类型	说明
videoid	true	String	视频ID
videoSource	true	String	视频来源: MATERIAL_LIBRARY-素材库

示例

```
https://api.polyv.net/live/v4/user/product/save?  
appId=fr1r1zazn3&sign=012332FBD8DBCFC068AFCB484A2ADECB&timestamp=1670550787318
```

请求体json参数：

```
{
  "productType": "normal",
  "linkType": 10,
  "link": "http://123.456.com",
  "pcLink": "",
  "mobileLink": "",
  "wxMiniprogramOriginalId": "",
  "wxMiniprogramLink": "",
  "mobileAppLink": "",
  "iosLink": "",
  "androidLink": "",
  "otherLink": "",
  "cover": "//liveimages.videocc.net/uploaded/images/2023/11/gqzaoq7pk3.png",
  "name": "平台商品",
  "price": 0,
  "realPrice": 0,
  "features": "[\"\\\", \"\\\"]",
  "tagIds": [1001, 1002],
  "productDesc": "",
  "productDetail": ""
}
```

响应参数描述

参数名	类型	说明
code	Integer	响应状态码，200为成功返回，非200为失败
status	String	响应结果，由业务决定，成功返回success，失败返回error
data	String	成功时，内容为商品id
success	Boolean	响应结果，由业务决定，成功返回true，失败返回false
error	Object	状态码非200时的错误信息【详见 Error字段说明 】
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 final Logger log = LoggerFactory.getLogger(getClass());
/**
 * 新增商品
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void createProductTest() 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/product/save";
    //业务参数
    String name = "平台商品";
    String productType = "normal";
    Integer linkType = 10;
    String link = "http://123.456.com";
    String cover = "//liveimages.videooc.net/uploaded/images/2023/11/gqzaoq7pk3.png";

    //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("name", name);
    jsonMap.put("linkType", linkType);
    jsonMap.put("link", link);
    jsonMap.put("productType", productType);
    jsonMap.put("cover", cover);

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

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

    log.info("测试新建商品结果: {}", 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": "78de516b69364aeabe7d080e7228cc5c.77.16705507903976231",
  "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
}
```