

创建系列直播场次

接口描述

- 1、创建系列直播场次

2、接口支持https协议

接口URL

http://api.polyv.net/live/v4/channel/session/new/create

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 】
channelId	true	Integer	频道号

请求体参数描述

参数名	必选	类型	说明
name	true	String	场次名称
planStartTime	true	String	场次开始时间 (13位时间戳)
planEndTime	true	Long	场次结束时间 (13位时间戳)
interactionScriptId	false	Integer	互动脚本ID
playType	false	String	播放方式（normal普通播放、loop循环播放）
splashImg	false	String	场次封面图
splashLargeImg	false	String	竖屏引导大图
splashImgBackgroundId	false	String	场次封面背景图ID
streamType	false	String	流类型（普通直播：client/伪直播：realDiskVideo）
videoPoolId	false	String	伪直播对应素材库视频ID（streamType为realDiskVideo时必传）

示例

```
http://api.polyv.net/live/v4/channel/session/new/create?
appId=frlr1zazn3&timestamp=1632809855078&channelId=2588188&sign=D02FA683640135D843D485EE84F6632A
```

请求体json参数：

```
{
  "name": "专题直播-新品发布会",
  "planStartTime": 1761201943000,
  "planEndTime": 1761201953000,
  "interactionScriptId": 123,
  "playType": "normal",
  "splashImg": "https://example.com/splash.jpg",
  "splashLargeImg": "https://example.com/splash-large.jpg",
  "splashImgBackgroundId": "1",
  "streamType": "client",
  "videoPoolId": "vid_abc123"
}
```

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
data	Object	场次响应对象【详见 data字段说明 】
error	Object	错误信息

data字段说明

参数名	类型	说明
channelId	Integer	频道ID
userId	String	POLYV用户ID
sessionId	String	场次ID
name	String	场次名称
endTime	Long	场次结束时间
createdTime	Long	场次开始时间
planStartTime	Long	计划开始时间
planEndTime	Long	计划结束时间
splashImg	String	封面图
splashLargeImg	String	竖屏、大图模式的封面图
splashImgBackgroundId	Integer	封面图的背景模板id
pushClient	String	推流客户端类型
type	String	场次类型(manual:手动创建场次, auto:自动创建场次)

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(ChannelOperateTest.class);
/**
 * 创建新版场次
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void testSessionNewCreateV4() throws IOException, NoSuchAlgorithmException {
    //公共参数,填写自己的实际
    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/channel/session/new/create";

    Map<String, String> query = new HashMap<>();
    query.put("appId", appId);
    query.put("timestamp", timestamp);
    query.put("channelId", "2588188");
    query.put("sign", LiveSignUtil.getSign(query, appSecret));

    Map<String, Object> body = new HashMap<>();
    body.put("name", "专题直播-新品发布会");
    body.put("planStartTime", "1761201943000");
    body.put("planEndTime", "1761201953000");
    body.put("interactionScriptId", 123);
    body.put("playType", "normal");
    body.put("videoPoolId", "vid_abc123");

    String fullUrl = HttpUtil.appendUrl(url, query);
    String response = HttpUtil.postJsonBody(fullUrl, JSON.toJSONString(body), null);
    log.info("返回值: {}", response);
}

```

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": "847f0716fa76461baf8979aaa4415dc3.67.16342672396023837",
  "data": {
    "channelId": 2588188,
    "userId": "1b448be323",
    "sessionId": "e9s2h3jd8f",
    "name": "专题直播-新品发布会",
    "startTime": 1761201943000,
    "endTime": null,
    "createdTime": 1761201943000,
    "planStartTime": 1761201943000,
    "planEndTime": 1761201953000,
    "splashImg": "https://example.com/splash.jpg",
    "splashLargeImg": "https://example.com/splash-large.jpg",
    "splashImgBackgroundId": 1,
    "pushClient": "client"
  },
  "success": true
}
```

异常示例

```
{
  "code": 400,
  "status": "error",
  "requestId": "4081dbac03e6441e8bdd301d8feee5a2.124.16360831818611581",
  "error": {
    "code": 20001,
    "desc": "application not found."
  },
  "success": false
}
```