

查询福袋中奖者分页列表

接口描述

- 1、查询指定福袋活动的中奖者分页列表

2、接口支持https协议

接口URL

http://api.polyv.net/live/v4/channel/lucky-bag/winner-page

[在线API调用](#)

请求方式

GET

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、若活动ID不存在或未发起则会响应400异常

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 】
activityId	true	Long	福袋活动ID
sessionId	false	String	场次ID，不传则查询所有场次
currentPage	false	Integer	页码，默认1
pageSize	false	Integer	每页条数，默认10

示例

```
http://api.polyv.net/live/v4/channel/lucky-bag/winner-page?
appId=frlr1zazn3&timestamp=1665629374000&sign=6CFA3141718E1AF0155889EEA2988206&activityId=100001&sessionId=abc123&currentPage=1&pageSize=10
```

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

参数名	类型	说明
error	Object	状态码非200时的错误信息【详见 Error字段描述 】
data	Object	中奖者分页数据【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
contents	Array	中奖者记录列表【详见 contents字段说明 】
totalItems	Long	总记录数
totalPages	Integer	总页数
pageNumber	Integer	当前页码
pageSize	Integer	每页条数

contents参数描述

参数名	类型	说明
id	Long	记录ID
viewerId	String	观众ID
viewerNickName	String	观众昵称

参数名	类型	说明
prizeType	String	奖品类型：cash-现金红包 custom-自定义奖品
prizeInfo	String	奖品信息，JSON字符串，结构见 prizeInfo字段说明
issueStatus	Integer	发放状态：0-待发放
outTradeErrCode	String	发放错误码
outTradeErrDesc	String	发放错误描述
createTime	String	创建时间

prizeInfo参数描述

prizeInfo 为 JSON 字符串，根据奖品类型（prizeType）不同，包含以下字段：

参数名	类型	说明
type	String	奖品类型：cash-现金红包 custom-自定义奖品
amount	BigDecimal	现金金额，单位：元（type=cash时返回）
name	String	自定义奖品名称（type=custom时返回）
image	String	自定义奖品图片URL（type=custom时返回）

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(ChannelLuckyBagTest.class);
/**
 * 查询福袋中奖者分页列表
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void testGetWinnerPage() 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/channel/lucky-bag/winner-page";
    String activityId = "100001";
    String sessionId = "abc123";
    String currentPage = "1";
    String pageSize = "10";

    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp", timestamp);
    requestMap.put("activityId", activityId);
    requestMap.put("sessionId", sessionId);
    requestMap.put("currentPage", currentPage);
    requestMap.put("pageSize", pageSize);
    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));

    String response = HttpUtil.get(url, requestMap);

    log.info("测试查询福袋中奖者分页列表, 返回值: {}", 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": "fc0894ce-08df-4376-92e2-fd0531cbe8ba",
  "data": {
    "contents": [
      {
        "id": 1001,
        "viewerId": "viewer001",
        "viewerNickName": "张三",
        "prizeType": "cash",
        "prizeInfo": "{\"type\": \"cash\", \"amount\": 1.00}",
        "issueStatus": 0,
        "outTradeErrCode": null,
        "outTradeErrDesc": null,
        "createTime": "2026-06-01T10:30:00"
      },
      {
        "id": 1002,
        "viewerId": "viewer002",
        "viewerNickName": "李四",
        "prizeType": "custom",
        "prizeInfo": "{\"type\": \"custom\", \"name\": \"精美礼品\", \"image\": \"https://example.com/gift.png\"}",
        "issueStatus": 0,
        "outTradeErrCode": null,
        "outTradeErrDesc": null,
        "createTime": "2026-06-01T10:30:00"
      }
    ],
    "totalItems": 2,
    "totalPages": 1,
    "pageNumber": 1,
    "pageSize": 10
  },
  "success": true
}

```

异常示例

```

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

```