

批量创建机器人虚拟昵称

接口描述

- 1、批量创建机器人虚拟昵称
- 2、(timestamp, appId) 参与sign签名, 并和sign一起通过url传递, 请求体参数不参与签名, 通过post请求体传递【请设置请求头contentType:application/json】
- 3、接口支持https协议

接口URL

http://api.polyv.net/live/v4/global/robot/save-batch

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](#)
- 2、一次性创建的机器人信息数量不能超过200个
- 3、昵称支持重复设置

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
name	true	String	虚拟昵称，必填，不支持emoji表情，不超过20个字符
avatar	false	String	机器人头像，不传或者传空，则设置默认头像

示例

```
http://api.polyv.net/live/v4/global/robot/save-batch?
appId=frlr1zazn3&sign=FAA7C22E73FA521B4CA48BB543252059&timestamp=1679023736161
```

请求体参数：

```
[
  {
    "name": "虚拟昵称1",
    "avatar": "https://liveimages.videocc.net/uploaded/images/2023/03/gj5fk2a06r.png"
  },
  {
    "name": "虚拟昵称2"
  }
]
```

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());
/**
 * 批量创建机器人虚拟昵称
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void robotSaveBatchTest() 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/global/robot/save-batch";

    //http 调用逻辑
    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp", timestamp);

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

    url = HttpUtil.appendUrl(url, requestMap);

    List<Object> jsonList = new ArrayList<>();
    Map<String, String> jsonMap = new HashMap<>();
    jsonMap.put("name", "虚拟昵称1");
    jsonMap.put("avatar", "https://liveimages.videocc.net/uploaded/images/2023/03/gj5fk2a06r.png");
    jsonList.add(jsonMap);
    jsonMap = new HashMap<>();
    jsonMap.put("name", "虚拟昵称2");
    jsonList.add(jsonMap);

    String response = HttpUtil.postJsonBody(url, JSON.toJSONString(jsonList), null);

    log.info("测试批量创建机器人虚拟昵称成功: {}", 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": "99d7e672d04d42698d9c95fb70e7c9ab.66.16790237605309137",
  "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
}
```