

流程：

1：主持人创建主持房间

2: 主持人根据pk报名用户，邀请主播加入PK，通过Broadcast通知主播

3：如果主播长时间没有应答，取消邀请，通过Broadcast通知主播

4：主播加入房间，通过Broadcast通知主持人和其他加入pk的房间，接口返回，主持人和其他加入pk的房间的信息

5:提供踢出pk，主动断开pk，通知主持人房间和所有pk的房间。

文档

1，直接创建主持人房间，把Room的type 设为4

2: 把普通房间变为主持人房间：GET /api/v1/room/pk/hall

SDK: LiveRoomClient.createHall();

示例

请求url: GET http://121.42.181.209/api/v1/room/pk/hall

响应: Room

3: 主持人邀请主播，主播需要创建房间:

GET /api/v1/room/pk/invite/{actorId}

SDK: LiveRoomClient.pkInvite(int actorId);

示例

请求url: GET http://121.42.181.209/api/v1/room/pk/invite/9

响应: Baseresponse

4: 主持人取消邀请: GET /api/v1/room/pk/invite/cancel/{actorId}

SDK: LiveRoomClient.pkInviteCancel(int actorId);

示例

请求url: GET http://121.42.181.209/api/v1/room/pk/invite/cancel/9

响应: Baseresponse

5 主播加入主持人房间: GET /api/v1/room/pk/join/{actorId}

GET: LiveRoomClient.pkJoin(int actorId);

示例

请求url: GET http://121.42.181.209/api/v1/room/pk/join/8

响应: RoomPkInfo

6: 主持人踢出pk: /api/v1/room/pk/kick/{actorId}

SDK: LiveRoomClient.pkKick(9);

示例

请求url: GET http://121.42.181.209/api/v1/room/pk/kick/9

响应RoomPkInfo

7:pk主播主动退出: /api/v1/room/pk/leave/{主持人id}

SDK: LiveRoomClient.pkLeave(8);

示例

请求url: GET http://121.42.181.209/api/v1/room/pk/leave/8

响应: RoomPkInfo

8:获取pk房间信息: GET /api/v1/room/pk/info/{actorId}

SDK: LiveRoomClient.pkRoomInfo(int actorId);

actorId: 当前用户所在房间的主播id

使用场景: 当用户进入房间后检测到state为pk状态时, 使用此接口, 返回的RoomPkInfo的actionRoom 为空。

示例

请求url: GET http://121.42.181.209/api/v1/room/pk/info/27

响应: RoomPkInfo

9:pk房间重联: GET /api/v1/room/pk/rejoin

SDK: LiveRoomClient.rejoin();

配合8接口使用更合理一点, 下一板加入状态机, 可能更合理一点

请求url: GET http://121.42.181.209/api/v1/room/pk/rejoin

响应: RoomPkInfo

```
public class RoomPkInfo {  
    private Room actionRoom; //加入pk的房间或者离开pk的房间  
    private Room hallRoom; //主持人的房间  
    private Collection<Room> pkRooms; //当前所有pk的房间  
}
```

当RoomPkInfo 为加入房间返回的数据时: pkRooms 包含actionRoom

```
public class BroadcastEvent {  
    public static final int TYPE_GIFT = 1;  
    public static final int TYPE_ROOM_MEMBER = 10;  
    public static final int TYPE_ROOM_MEMBER_KICK = 11;  
    public static final int TYPE_ROOM_MEMBER_SILENCE = 12;  
    public static final int TYPE_ROOM_MEMBER_AGENT = 13;  
    public static final int TYPE_ROOM_MEMBER_AGENT_CANCEL = 14;  
    public static final int TYPE_USER_LEVEL = 20;  
    public static final int TYPE_MESSAGE_LETTER = 30;  
    public static final int TYPE_MESSAGE_FLY = 31;  
    public static final int TYPE_JOIN_ASK = 40;  
    public static final int TYPE_JOIN_CHECK = 41;  
    public static final int TYPE_JOIN = 42;  
    public static final int TYPE_JOIN_LEAVE = 43;  
    public static final int TYPE_JOIN_OPEN = 44;  
    public static final int TYPE_FANS_ADD = 50;
```

```

    public static final int TYPE_SHARE_ROOM= 60;
    public static final int TYPE_ROOM_DESTORY= 80;
    public static final int TYPE_ROOM_RECOVERY= 81;
    public static final int TYPE_ROOM_PK_INVITE= 70;
    public static final int TYPE_ROOM_PK_JOIN= 72;
    public static final int TYPE_ROOM_PK_LEAVE= 73;
    public static final int TYPE_ROOM_PK_INVITE_CANCEL= 74;
    public static final int TYPE_ROOM_PK_REJOIN= 75;
    private int type;
    private Map<String,String> data = new HashMap<String, String>();
}

```

```

public class Room implements Serializable{
    private static final long serialVersionUID = 1L;
    public static final int STATE_CREATE =0;
    public static final int STATE_CLOSE =1;
    public static final int STATE_LEAVE =2;
    public static final int STATE_NET_TERRIBLE=3;
    public static final int STATE_NORMAL =4;
    public static final int STATE_PK =5;//房间处于pk
    public static final int STATE_JOIN =6;

    public static final int TYPE_NORMAL=1;
    public static final int TYPE_SOUND =2;
    public static final int TYPE_GAME = 3;
    public static final int TYPE_HALL =4;//主持人房

    public static final int ORIENTATION_LANDSCAPE = 0;
    public static final int ORIENTATION_VERTICAL = 1;

    private String id;
    Indexed private int uid;
    Indexed private int state;
    Indexed private Integer type;
    Indexed private Integer gameId;
    Indexed private Long heartbeatTime;
    private Map<Integer,ActorStreamAddress> playAddresses;
    private long createTime;
    private int audienceCount;
    private int medium;
    private String subject;
    private String channel;
    private Collection<String> tags;
    private Double lon;
    private Double lat;
    private String location;
    private Integer hot;

```

```
private GiftGuard giftGuard;  
private Integer screenOrientation;  
private String poster;  
private Integer joinActorCount;  
}
```