

设计文档 副本

1、需求

📁 娱乐模式PK

2、需求分析

📄 00_需求分析.xmind

直播间pk玩法

匹配机制

- 优先匹配跟自己段位《段位信息是由后台通过积分系统来实现》相同的房间
- 30s内匹配不到则范围扩大到相邻段位，最多只扩大一个段位
- 不会匹配到3周内人匹配过的对手
- 匹配15分钟自动超时，自动关闭PK功能并toast：暂时匹配不到对手哦，请稍后再试
- 每一周默认15分钟，结束后统计魅力值高低决定输赢
- 公屏文案：匹配成功后系统文案：已匹配到xxx《段位》对手yyy《对手名称》，开始PK！

开发功能

- 白名单查询接口《区分房间类型》
 - 判断是否在冷却期内
 - 参加比赛接口
 - 把嘉宾位请下桌
 - 进入匹配池
 - 维护各段位匹配池参与关系
 - 匹配核心逻辑
 - 匹配池逻辑
 - 不会匹配到3周内人匹配过的对手
 - 超过30秒匹配时间，段位提升一级进行匹配
 - 记录当前处于那个匹配池
 - 匹配成功
 - 记录已匹配关系
 - 双方房间发送公屏：已匹配到xxx《段位》对手yyy《对手名称》，PK开始！
 - 声音互通
 - toast：暂时匹配不到对手哦，请稍后再试
 - 移出匹配列表
 - 比赛中逻辑
 - 监听送礼流水，在PK期间增加对应的pk积分
 - 比赛计时15分钟(可配置)
 - 比赛结束逻辑
 - 比较PK积分
 - 魅力值高于对手，则结算的时候判定为PK胜利，我方获得1个段位星星
 - 赢
 - 发送公屏：恭喜！PK胜利获得1星，当前段位xxx《当前段位》
 - 判定后直接进入惩罚阶段
 - 输
 - 平局
 - 我方投降
 - 对方投降
 - 惩罚阶段
 - PK结束后继续保持连接2分钟《倒计时2分钟》
 - 胜利方可直接退出结束该场比赛
 - 敌方不可提前退出，点击退出PK后toast：请先完成惩罚哦~
 - 段位机制
 - 批量加减星功能
 - 针对不同段位批量加减星
- 直播间轮询接口
 - 修改重上列表获取uid方式，增加channel映射关系能进入声音互通的channel
- 查询比赛信息接口《修改娱乐轮询》
 - 比赛双方信息
 - 选手(多人)
 - 贡献榜前3列表
 - 房间信息
 - 段位
 - pk值
 - 比赛结果
 - 选手(单人)
 - 用户信息
 - 段位
 - pk值
 - 比赛结果
- 比赛信息
 - 比赛开始时间
 - 比赛结束时间
 - 匹配模式
 - 电台模式
 - 娱乐模式
 - PKtips
 - PKtips由固定文案库，随机读取组成，5s切换一条，区分模式读取文案库
 - 比赛状态
 - 匹配中、成功匹配/比赛开始、比赛中、比赛结束
- 退出pk
 - 作为比赛失败进行结算
 - 冷却PK功能30分钟
 - 发送退出PK通知事件
- 声音开/关接口
 - 恢复声音连接
 - 只有主动断开声音的人才能恢复
 - 公屏
 - 发起方：房主已恢复声音连接
 - 被动方：对手已恢复声音连接
 - 被动断开的人发送公屏给对方
 - 公屏
 - 发起方：房主已断开声音连接
 - 被动方：对手邀请房主打开麦克风交流~
 - 断开声音连接
 - 公屏
- 互娱后台条件
 - 发送topic
 - 互娱后台配置条件
- 分发页标记
- 埋点数据统计

互娱后台条件

分发页标记

开启PK

- 房间增加PK中的标记信息
- 房间类型
 - 娱乐模式《厅》
 - 电台模式
- 可见范围
 - 白名单内的房间的房主/主持人/管理员可见
 - 将嘉宾拖下桌同时开启PK

PK匹配阶段

- 服务端匹配逻辑
- 前端轮询

PK中阶段

- 声音关联
 - 把两边房间的声音连接起来，默认连线，可关闭
- 投降
 - 点击投降后，弹窗确认：当前仍在PK中，投降后将认输并受到PK惩罚，确认投降？取消/确定，点击确定后退出PK并toast：本场PK已投降。
 - 在PK中主动投降的一方，默认判定为输方，同时冷却PK功能30分钟
- 双方比赛信息
 - 匹配成功后右侧展示对方的封面和房间名
- 静音开关
- 送礼贡献者
 - 双方均展示送礼最高的三个人，按照送礼魅力值计算
- 送礼比拼进度条
 - 按照当场PK的送礼魅力值来定，最终多的一方获胜
- 记分方式
 - PK时段内整个房间的魅力值总收入

PK结束

- 胜利
 - 魅力值高于对手，则结算的时候判定为PK胜利
 - 同时公屏系统提示：恭喜！PK胜利获得1星，当前段位xxx《当前段位》
- 失败
 - 魅力值低于对手，则结算的时候判定为PK失败，我方损失1个段位星星
 - 同时结果宣布15秒后，恢复房间初始状态，公屏系统提示：很遗憾！PK失败损失1星，当前段位xxx《当前段位》
- 平局
 - 魅力值与对手一样，则结算的时候判定为PK失败，我方损失1个段位星星
 - 同时结果宣布15秒后，恢复房间初始状态，公屏系统提示：很遗憾！PK平局损失1星，当前段位xxx《当前段位》
- 致方投降
 - 不判断魅力值，直接当作我方PK失败，我方损失1个段位星星
 - 不宣布结果，直接恢复房间初始状态，公屏系统提示：PK遗憾损失1星，当前段位xxx《当前段位》
- 对方投降
 - 不判断魅力值，直接当作我方PK胜利，我方获得1个段位星星
 - 宣布对方投降的结果15秒后，直接恢复房间初始状态，公屏系统提示：对方投降！PK胜利获得1星，当前段位xxx《当前段位》

基础设施

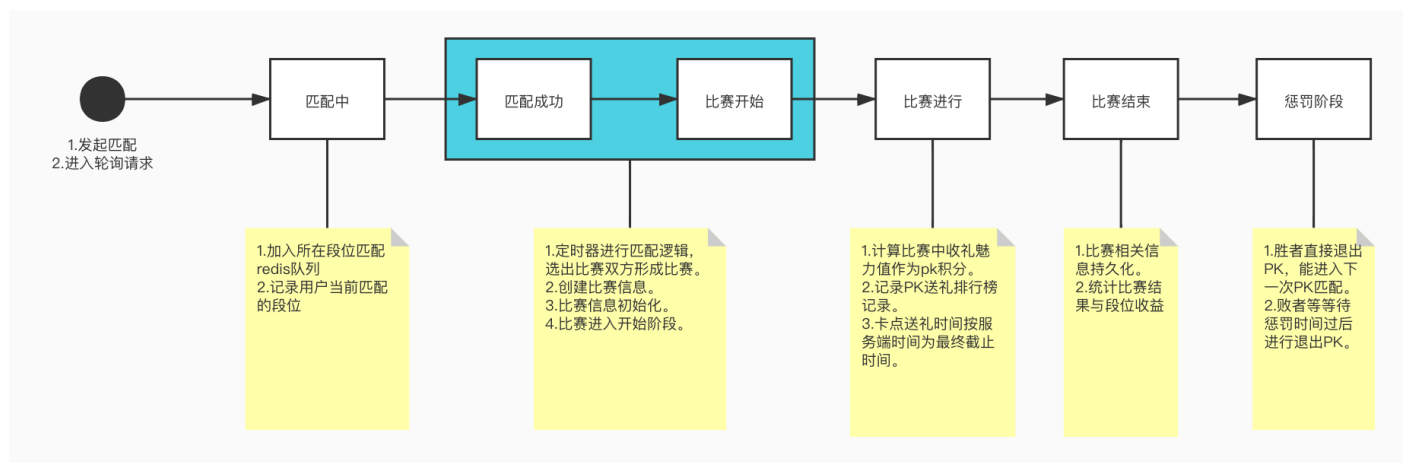
- Redis数据结构
 - 用户比赛信息
 - 比赛贡献榜
 - 排名积分
 - 各段位匹配池
 - 用户匹配成功对象记录，保留最近3个(可配)
- 数据表
 - 比赛记录表
 - 用户积分排名表
 - lz_pp_activity(比赛)
 - web_pp_h5
- 涉及服务

段位机制

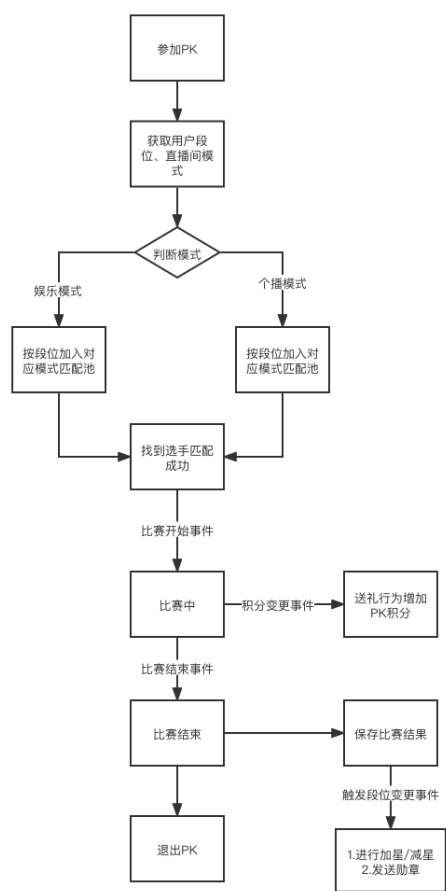
- 积分
 - PK时间内厅的总收益魅力值
 - 按用户维度记录段位
- 段位
 - 青铜、白银、黄金、铂金、钻石、星耀、王者、荣耀王者
 - 青铜1-5星，白银1-5星，黄金1-5星，.....，钻石1-6星，星耀1-6星，王者1-50星，荣耀王者51-100星
 - 青铜1再输无法再往下掉，银死青铜1
 - 勋章逻辑
 - 用户更换新段位的时候，给用户发放对应段位的勋章《显示在资料卡和公屏发言昵称后，已有逻辑》
 - 批量加减星，针对不同段位批量加减星

3、功能设计

3.1 比赛机制



流程图



3.2 声音互通

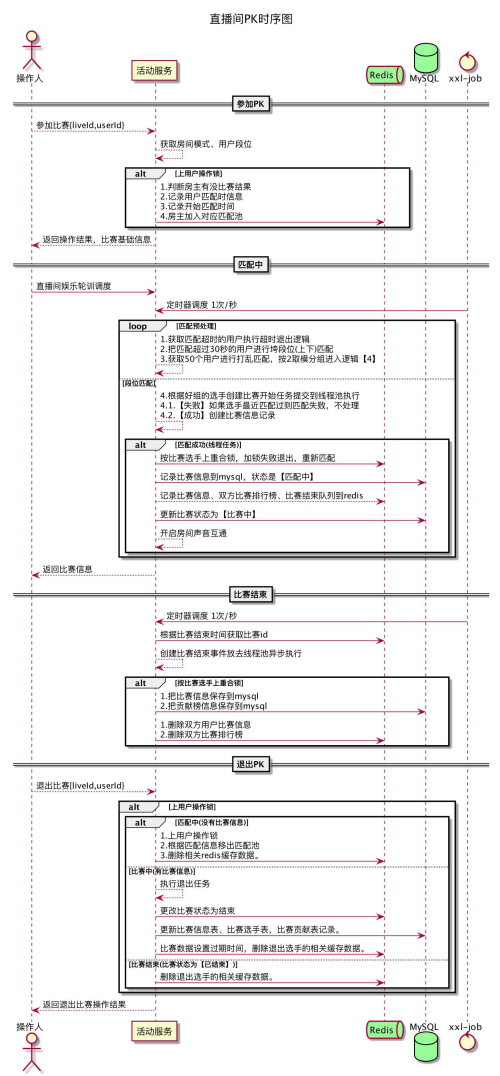
声音相关 lz_pp_call 接口

ppChannelLiveService: - getUserId32, 查询用户的绑定声网id (一个32位的id)

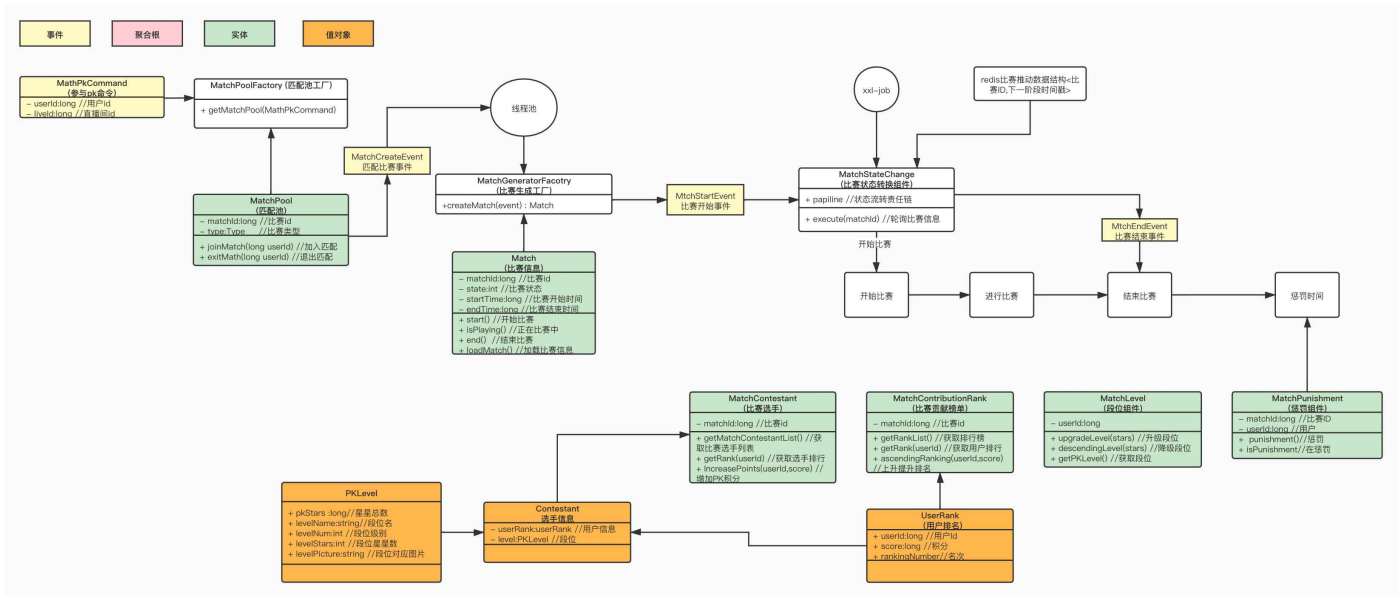
CallServiceImpl: - getLineCode, 根据liveId查channelId

房间互动服务

3.3 功能时序图



3.4 实体设计



3.3 包路径

调用链 api -> application.service -> domain.service-> domain.model

Java

```

1  api    //dc对外接口
2  --impl //dc实现类
3
4  application //应用
5  --event //应用事件/命令
6  --service //应用服务（跨领域调用的业务逻辑在这里组装）
7
8  domain //领域
9  --match //比赛信息
10 ----model //实体 放带有行为的实体
11 -----valueobject //值对象
12 -----factory //领域工厂 创建实体
13 -----repository //资源库 实体所拥有的资源(相当于是组合事务的地方，相当于原来manager)
14 ----event //领域 事件/命令 推动实体事件的类
15 ----service //领域服务 组合的业务逻辑(存放组合逻辑，例如不是实体自身行为的方法)
16
17 --rank //贡献榜
18 --player //参赛选手
19 --level //段位
20 --punishment //惩罚
21 --publicscreen //公屏
22
23 scheduler //调度
24 infrastructure //基础设施
25 --mapper //mybatis组件
26 --redis //redis组件
27 --kafka //kafka组件

```

4、Redis存储结构设计

用途	数据结 构	key名(都是大写)	值内容	时效性
用户比赛关联	set	live_pk_match_player_\${njId}	matchId	比赛结束数据持久化 后删除
用户比赛中信 息	hash	live_pk_match_info_\${matchId}	match:\${比赛信息} state:\${比赛状态} 关闭声音用户:\${操作 人Id}	比赛结束数据持久化 后删除
比赛选手比拼	zset	live_pk_match_player_competi	key:贡献榜用户id	比赛结束数据持久后

榜		<i>tion_list_\${matchId}</i>	sorce: 本场送礼价值	删除
比赛贡献榜	zset	live_pk_match_rank_\${njId}_\${matchId}	key:贡献榜用户id sorce: 本场送礼价值	比赛结束数据持久后删除
各段位匹配池 参赛用户列表	zset	live_pk_match_\${段位}	key:比赛用户id sorce: 参加pk开始时间戳	
用户匹配成功 对象记录	zset	live_pk_match_user_sccuess_\${userId}_\${房间类型}	key:已匹配选手userId socre: 匹配成功时间戳	根据配置只保留最近几条匹配数据
用户惩罚	set	<i>live_pk_user_penalty_\${userId}</i>		30分钟过期（可配）
用户当前匹配的段位	hset	<i>live_pk_match_level_\${userId}</i>	value:段位	匹配成功删除
比赛进程推动	zset	live_pk_match_next_stage	value:比赛id socre:下一阶段时间戳	
记录用户匹配时间	zset	live_pk_match_user_time	value:用户ID socre: 参加匹配时间	

5、Mysql存储层设计

- gift_record //复用送礼流水表
- match_gift_record //复用比赛送礼流水关联
- live_match_info //直播间比赛信息表
- live_match_result //直播间比赛结果表
- live_match_contribution_rank //直播间贡献榜表
- user_level_info //用户段位记录表

live_match_info (直播间比赛信息表)

SQL

```
1 CREATE TABLE `live_match_info` (  
2   `id` bigint(20) NOT NULL COMMENT '比赛id',  
3   `mode` int(1) NOT NULL COMMENT '模式 1娱乐模式、2电台模式',  
4   `state` int(1) NOT NULL COMMENT '状态 1比赛开始 2比赛中 3比赛结束',  
5   `start_time` datetime NOT NULL COMMENT '开始时间',  
6   `end_time` datetime NOT NULL COMMENT '结束时间',  
7   `modify_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '记录修改时间',  
8   `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '记录创建时间',  
9   PRIMARY KEY (`id`)  
10 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='直播间比赛信息表';
```

live_match_result(直播间比赛结果表)

SQL

```
1 CREATE TABLE `live_match_result` (  
2   `id` bigint(20) NOT NULL COMMENT '主键',  
3   `live_match_id` bigint(20) NOT NULL COMMENT '直播间比赛ID',  
4   `user_id` bigint(20) NOT NULL COMMENT '选手ID',  
5   `result_score` int(20) NOT NULL DEFAULT '0' COMMENT '结果分值 加减星星',  
6   `pk_score` int(20) NOT NULL DEFAULT '0' COMMENT 'PK分值',  
7   `result_status` tinyint(2) NOT NULL DEFAULT '0' COMMENT '结果状态 0-无状态 1-赢 2-输 3-平局',  
8   `modify_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',  
9   `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
10  PRIMARY KEY (`id`),  
11  KEY `idx_match_user_id` (`live_match_id`, `user_id`) USING BTREE  
12 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='直播间比赛结果';
```

live_match_contribution_rank (直播间贡献榜表)

SQL

```
1 CREATE TABLE `live_match_contribution_rank` (  
2   `id` bigint(20) NOT NULL COMMENT '主键',  
3   `live_id` bigint(20) NOT NULL COMMENT '直播间ID',  
4   `match_id` bigint(20) NOT NULL COMMENT '比赛ID',  
5   `user_id` bigint(20) NOT NULL COMMENT '贡献用户ID',  
6   `charm_score` int(20) NOT NULL DEFAULT '0' COMMENT '贡献魅力值',  
7   `lizhi_score` int(20) NOT NULL DEFAULT '0' COMMENT '贡献PP钻',  
8   `modify_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',  
9   `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
10  PRIMARY KEY (`id`),  
11  KEY `idx_match_user_id` (`match_id`,`user_id`) USING BTREE  
12 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='直播间贡献榜';
```

user_level_info(用户段位记录表)

SQL

```
1 CREATE TABLE `user_level_info` (  
2   `id` bigint(20) NOT NULL COMMENT '主键',  
3   `user_id` bigint(20) NOT NULL COMMENT '用户ID',  
4   `star` int(10) NOT NULL DEFAULT '0' COMMENT '拥有星星数',  
5   `modify_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',  
6   `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
7   PRIMARY KEY (`id`),  
8   KEY `idx_match_user_id` (`user_id`) USING BTREE  
9 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='用户段位记录表';
```

6、任务后台kafka结构体

lz_topic_pp_live_match_event (直播间比赛事件)

JSON

```
1 {
2   long appId , //appID
3   long matchId, //比赛ID
4   long liveId, //直播间ID
5   long njId, //房主ID
6   long userId, //用户ID
7   int result, //结果 1赢 2输 3平局
8   int surrender, //投降 0否 1是
9   long startTime, //比赛开始时间
10  long endTime, //比赛结束时间
11  long score, //分值
12  long goldCoinCount//单场收礼金币数
13 }
```

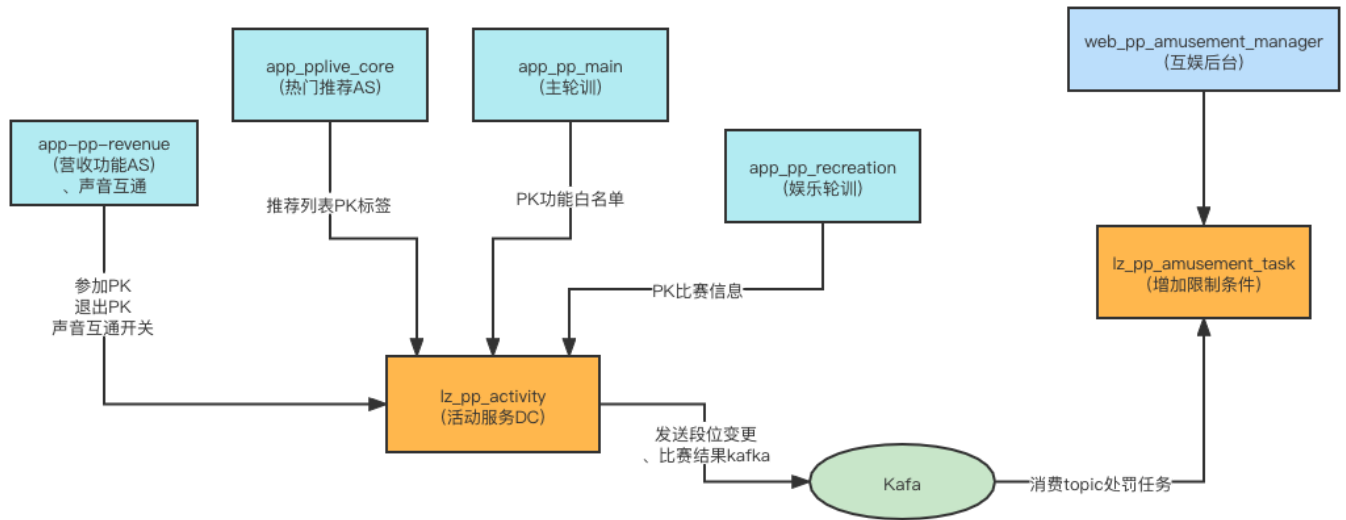
lz_topic_pp_user_level_event（段位事件）

JSON

```
1 {
2   long appId, //appID
3   long liveId, //直播间ID
4   long matchId, //比赛ID
5   long njId, //房主ID
6   long userId, //用户ID
7   int event, //行为标记 1升 2降
8   int oldStar, //原来星星数
9   int curStar, //现在星星数
10  int oldLevel, //原来段位
11  int curLevel, //现在段位
12  int createTime //创建时间
13 }
```

7、涉及服务

核心服务关系图



分支名：feat_v3.8.0_pkPlay

服务名	协议	功能	处理情况
app-pp-revenue	新as协议 1、PK操作接口 RequestPPPKOperation 2、PK中开关麦接口 RequestPPPKMicOperation	比赛对外提供接口功能	
lz_pp_activity		比赛相关逻辑	
app_pplive_core	RecommendLiveCardsService	推荐列表数据修改，增加房间pk状态。	
app_pp_main	RequestLiveMainData	主轮询接口增加是否白名单	
app_pp_recreation	ResponseLiveFunModePolling	娱乐轮询接口修改，增加比赛相关信息。	
web_pp_amusement_manager		互娱后台功能 1.配置kafka 2.增加用户限制条件	
lz_pp_amusement_task		互娱后台功能 1.增加用户限制条件	
app_pp_common	ResponsePPACData	增加 serverTimeStamp字段	
长连接PUSH		增加structPPCallChannel	

lz_pp_call			
------------	--	--	--

8、开发任务

类型	工作项目	描述	计划耗时(h)
设计	需求分析		8
	详细设计		16
协议	协议设计		4
接口开发	白名单查询接口	根据房间id区分房间类型返回PK开关判断 app_pp_main项目中的RequestLiveMainData	1
	参加比赛接口	1.判断是否在冷却期内; 2.把嘉宾位请下麦; 3.调用 比赛匹配机制	2
	直播间辅助轮询接口	1.修改麦上列表获取uid方式，增加channel映射关系 能进入声音互通的channel; 3.返回CallChannel事件	2
	娱乐轮询接口	1.比赛信息 2.比赛选手双方信息 3.返回CallChannel信息 app-pp-recreation项目中的 ResponseLiveFunModePolling	4
	退出pk	匹配中： 1.退出redis匹配池 2.删除当前匹配段位redis数据 比赛中： 1.作为比赛失败进行结算 2.加入redis惩罚，冷却PK功能30分钟 3.发送退出PK通知事件 比赛完： 1.胜利方退出比赛 2.失败方进入惩罚时间，惩罚完才能退出(校验惩罚时间)	4

	声音开/关接口	<p>恢复声音连接:</p> <ol style="list-style-type: none"> 1.只有主动断开声音的人才能恢复，建立新通道逻辑 2.发送公屏-发起方：房主已恢复声音连接 -被动方：对手已恢复声音连接 3.被动方操作-发送公屏给主动方：对手邀请房主打开麦克风交流~ 4.推送CallChannel事件 <p>断开声音连接:</p> <ol style="list-style-type: none"> 1.断开声音逻辑 2.发送公屏-发起方：房主已断开声音连接 -被动方：对手已断开声音连接 3.推送CallChannel事件 	6
服务端逻辑	比赛匹配机制	<ol style="list-style-type: none"> 1.根据用户段位加入不同的匹配池 2.放去30s延迟调度任务-出列提升一级段位进行匹配 3.记录比赛数据到redis 4.记录比赛数据到mysql 	4
	匹配成功	<ol style="list-style-type: none"> 1.双方移出匹配池 2.形成比赛，记录到mysql，状态为比赛中 3.删除双方当前匹配段位redis数据 4.保存信息到redis 	4
	匹配定时器	<ol style="list-style-type: none"> 1.xxl-job定时任务 2.15分钟匹配超时移出redis匹配队列 3.删除用户匹配 4.30s延迟调度出队机制 <ol style="list-style-type: none"> 4.1判断是否匹配成功 4.2更新用户匹配段位,升1级 4.3把用户加入对应段位匹配池 	4
	比赛结束机制	<ol style="list-style-type: none"> 1.把比赛信息保存到mysql 2.把贡献榜信息保存到mysql 3.pipeline命令 <ol style="list-style-type: none"> 3.1 删除双方用户比赛信息 3.2 删除双方比赛排行榜 	3
	段位机制	<ol style="list-style-type: none"> 1.用户段位星数记录表 2.星数转换相应段位 3.发送段位勋章 	4

	收礼计算收益	1.消费kafka，用redis做幂等校验 2.组合pipeline命令 2.1 redis增加积分 2.2 redis记录贡献榜 3.redis记录幂等	2
接口自测			16
运营接口	批量加减星接口（波段号、段位）	1.根据波段号进行批量加减星； 2.根据段位进行批量加减星； 3.段位变化发送对应的勋章；	3
分发页标记	首页推荐打pkTab标签		2
互娱后台条件	配置对应kafka规则		6
	用户限制需增加【PK等级】	用于筛选不同的等级用户，根据段位信息直接折算数字，设定最小值以及最高值；	3
埋点需求			4
合计			98

9、配置信息

1) Apollo配置

lz_pp_activity

Java

```
1  /**
2   * 用户惩罚时间（秒）
3   */
4  private Integer userPenaltyTime = 120;
5  /**
6   * 主动退出-惩罚时间（秒）
7   */
8  private Integer activeExitPenaltyTime = 1800;
9  /**
10   * 匹配升段时间（秒）
11   */
```

```
12 private Integer upgradeLevelTime = 30;
13 /**
14  * 分布式锁时间
15  */
16 private Integer lockTime = 30;
17
18 /**
19  * 匹配超时时间 (秒)
20  */
21 private Long matchTimeout = 54000L;
22 /**
23  * 最近匹配对象次数
24  */
25 private int matchLastCount = 3;
26 /**
27  * 比赛进行时间: (秒)
28  */
29 private Long gameTime= 54000L;
30 /**
31  * 选手榜单数量
32  */
33 private Integer playerRankListCount = 10;
34
35 /**
36  * pkTips
37  */
38 private String pkTips;
39
40 /**
41  * 比赛Key过期时间 (秒)
42  */
43 private int expirationTime = 72000;
44
45 /**
46  * 段位勋章配置
47  */
48 private String rankLevelMedalInfoJson = "[{\"rankOrder\": 1, \"medalId\": 1207},
49 {\"rankOrder\": 2, \"medalId\": 1208},{\"rankOrder\": 3, \"medalId\": 1209},{\"r
50 ankOrder\": 4, \"medalId\": 1210},{\"rankOrder\": 5, \"medalId\": 1211},{\"rankO
51 rder\": 6, \"medalId\": 1212},{\"rankOrder\": 7, \"medalId\": 1213},{\"rankOrder
52 \": 8, \"medalId\": 1214}]";
53 /**
54  * 用户段位升降kafka通知
55  */
56 private String userChangeRankLevelKafkaTopic = "lz_topic_pp_user_level_event";
57 /**
58  * 用户直播间pk比赛结果kafka通知
59  */
60 private String liveMatchResultKafkaTopic = "lz_topic_pp_live_match_result_event";
```

```

56 private String liveMatchResidualRatePro = "0.2";
57
58 /**
59  * 直播间PK比赛白名单
60  */
61 private String LiveMatchUserIdWhiteList = "";
62
63 /**
64  * 匹配过滤条件-开启
65  */
66 private boolean matchFilter = true;
67
68 /**
69  * 无视段位进行匹配-关闭
70  */
71 private boolean matchIgnoreLevelOpStatus = false;
72
73 /**
74  * 麦功能状态
75  */
76 private boolean wheatStatus = true;

```

PHP

```

1 app-pp-recreation
2 /**
3  * CDN
4  */
5 private String cdnHost = "https://cdn.lizhi.fm/";

```

2. 直播白名单数据