1、创建频道并设置白名单

描述

由于创建频道同时设置观看条件为白名单，由于白名单列表为空，则会抛出异常；则需要先创建频道号，再添加频道白名单，再设置观看权限

单元测试

请确认已经全局调用了初始化系统

```
    /**
     * 1、创建并初始化频道
     * 2、设置频道主要观看条件白名单（主要观看条件白名单和次要观看条件白名单不互用）
     * 3、设置观看条件为白名单
     * @throws IOException
     * @throws NoSuchAlgorithmException
     */
    @Test
    public void testCreateChannelInitWriteList() throws IOException, NoSuchAlgorithmException {
        LiveChannelInitRequest liveChannelInitRequest = new LiveChannelInitRequest();
        LiveChannelInitResponse liveChannelInitResponse = null;
        try {
            //1、创建并初始化频道
            LiveChannelInitRequest.BasicSetting basicSetting = new
LiveChannelInitRequest.BasicSetting().setName(
                    "创建并初始化频道-白名单观看")
                    .setChannelPasswd(getRandomString(6))
                    .setAutoPlay(1)
                    .setPlayerColor("#666666")
                    .setScene(LiveConstant.SceneType.ALONE.getDesc())
                    .setCategoryId(340019)
                    .setMaxViewer(0)
                    .setStartTime(null)
                    .setDesc("这是一个描述")
                    .setPublisher("sadboy主讲")
                    .setLinkMicLimit(-1)
                    .setPureRtcEnabled("N")
                    .setReceiveChannelIds(null)
                    .setOnlyOneLiveEnabled("N");
            liveChannelInitRequest.setBasicSetting(basicSetting);
            liveChannelInitResponse = new
LiveChannelOperateServiceImpl().createChannelInit(liveChannelInitRequest);
            Assert.assertNotNull(liveChannelInitResponse);
            if (liveChannelInitResponse != null) {
                //to do something ......
                log.debug("测试创建并初始化频道 验证码观看创建成功{}", JSON.toJSONString(liveChannelInitResponse));
                //2、设置频道主要观看条件白名单（主要观看条件白名单和次要观看条件白名单不互用）
                LiveCreateChannelWhiteListRequest liveCreateChannelWhiteListRequest =
                        new LiveCreateChannelWhiteListRequest();
                Boolean liveCreateChannelWhiteListResponse;

liveCreateChannelWhiteListRequest.setChannelId(liveChannelInitResponse.getChannelId()).setRank(1)
                        //此处一般设置为观众手机号或者学生学号等
                        .setCode(String.valueOf(System.currentTimeMillis()))
                        //此处设置为观众名称
                        .setName("学员1");
                liveCreateChannelWhiteListResponse = new LiveWebAuthServiceImpl().createChannelWhiteList(
                        liveCreateChannelWhiteListRequest);
                Assert.assertNotNull(liveCreateChannelWhiteListResponse);
                if (liveCreateChannelWhiteListResponse) {
                    //to do something ......
                    log.debug("测试添加单个白名单-全局白名单成功");
                    //3、设置观看条件为白名单
                    LiveUpdateChannelAuthRequest liveUpdateChannelAuthRequest = new
LiveUpdateChannelAuthRequest();
```

```
                    Boolean liveUpdateChannelAuthResponse;
                    LiveChannelSettingRequest.AuthSetting authSetting = new
LiveChannelSettingRequest.AuthSetting().setAuthType(
                            LiveConstant.AuthType.PHONE.getDesc())
                            .setEnabled("Y").setRank(1)
                            .setAuthTips("请输入手机号");
                    List<LiveChannelSettingRequest.AuthSetting> authSettings =
                            new ArrayList<LiveChannelSettingRequest.AuthSetting>();
                    authSettings.add(authSetting);
                    liveUpdateChannelAuthRequest.setChannelId(liveChannelInitResponse.getChannelId())
                            .setAuthSettings(authSettings);
                    liveUpdateChannelAuthResponse = new LiveWebAuthServiceImpl().updateChannelAuth(
                            liveUpdateChannelAuthRequest);
                    Assert.assertNotNull(liveUpdateChannelAuthResponse);
                    if (liveUpdateChannelAuthResponse) {
                        log.debug("测试设置观看条件成功");
                    }
                }
            }
        } catch (PloyvSdkException e) {
            //参数校验不合格 或者 请求服务器端500错误，错误信息见PloyvSdkException.getMessage(),B
            log.error(e.getMessage(), e);
            // 异常返回做B端异常的业务逻辑，记录log 或者 上报到ETL 或者回滚事务
            throw e;
        } catch (Exception e) {
            log.error("SDK调用异常", e);
            throw e;
        }
    }
```