排序接口

`vvmusiceditPriorityDrawPicType`

从1-3
index是3 action是1
从3-1
index是1 action是2

```
    /**
     * 直播素材分类顺序
     */
    @RequestMapping(value = "vvmusiceditPriorityDrawPicType")
    @ResponseBody
    public Map<String, Object> vvmusiceditPriorityDrawPicType() throws Exception {
        Map<String, Object> result = new HashMap<String, Object>();

        Long Id = Long.parseLong(request.getParameter("typeid"));
        Integer action = Integer.parseInt(request.getParameter("action")); // 1、从前往后移  // 2、从后往前（包括位置不变）
//      Short Type = Short.parseShort(request.getParameter("type"));
        Long index = Long.parseLong(request.getParameter("index"));
//      Long channelId = Long.parseLong(request.getParameter("channelid"));
        // String opr = index.longValue() == 1 ? "置顶" : "移动";

        if (action == 1) {
            index = index - 1;
        } else {
            index = index - 2;
        }

        Integer beforeId = null;
        Integer afterId = null;
        BigDecimal before = null;
        BigDecimal after = null;

        LiveDrawPicRoomType recEx = new LiveDrawPicRoomType();

        if (index < 0) {
            index = (long) 0;
//          recEx.setRows(1);
             recEx.setViewNumber(1);
        } else {
//          recEx.setRows(2);
             recEx.setViewNumber(2);
        }

        recEx.setStartIndex(index);
//      recEx.setType(Type);
//      recEx.setChannelid(channelId);
        List<LiveDrawPicRoomType> recList = liveDrawPicTypeService.getBefAndAftRec(recEx);
        if (recList == null || recList.size() <= 0 || recList.size() > 2) {
```

```
            result.put("state", 2000);
            result.put("msg", "不存在");
            return result;
        }

        if (recList.size() == 1) {
            LiveDrawPicRoomType rec = recList.get(0);
            if (index == 0) {
                afterId = rec.getTypeid();
                after = rec.getPriority();
                before = new BigDecimal(0);
            } else {
                afterId = rec.getTypeid();
                after = rec.getPriority().add(new BigDecimal(1));
                before = rec.getPriority();
            }
        } else {
            for (int i = 0; i < recList.size(); i++) {
                LiveDrawPicRoomType rec = recList.get(i);
                if (i == 0) {
                    beforeId = rec.getTypeid();
                    before = rec.getPriority();
                } else {
                    afterId = rec.getTypeid();
                    after = rec.getPriority();
                }
            }
        }

        BigDecimal priority = getPriority(beforeId, afterId, before, after,minIncrea);

        // MpObjChannelExample recEx2 = new MpObjChannelExample();
        //
recEx2.createCriteria().andChannelidEqualTo(channelId).andStatusEqualTo((short)1);//andMp
idEqualTo
        // ShufflingFigure recEx2=new ShufflingFigure();
        LiveDrawPicRoomType targetList = (LiveDrawPicRoomType)
liveDrawPicTypeService.findById(Id);

        if (targetList == null) {
            result.put("state", 2000);
            result.put("msg", "待修改素材不存在");
            return result;
        }

        LiveDrawPicRoomType rec = targetList;
        rec.setPriority(priority);

        liveDrawPicTypeService.update(rec);

        // 记录操作日志
        // mpcmsOprLogService.doOprLog(getUserInfo(), MODULE_RECED_ID, opr,
        // mpId);
//      mpcmsOprLogService.doOprLog(getUserInfo(), "抠图素材管理", "排序", Id.toString(),
" ", "vv直播");
        result.put("state", 1000);
```

```
        return result;
    }
```

```
    private BigDecimal getPriority(Integer beforeId, Integer afterId,
            BigDecimal before, BigDecimal after, BigDecimal increa) {
        BigDecimal priority = null;

        // 前后值相等，重新排序
        if (before.equals(after)) {
            priority = doResize(beforeId, afterId);

        } else {
            // 前值+间隔>=后值，间隔/10，继续尝试
            if (before.add(increa).compareTo(after) >= 0) {
                increa = increa.divide(new BigDecimal(10));
                // 间隔小于等于极限间隔，重新排序
                if (increa.compareTo(resize) <= 0) {
                    priority = doResize(beforeId, afterId);
                } else {
                    priority = getPriority(beforeId, afterId, before, after,
                            increa);
                }

            } else {
                priority = before.add(increa);

            }
        }

        return priority;
    }
```

```
    private BigDecimal doResize(Integer beforeId, Integer afterId) {
        LiveDrawPicRoomType m = new LiveDrawPicRoomType();
//      m.setType(type);
//      m.setChannelId(channelId);
        List<LiveDrawPicRoomType> recList = liveDrawPicTypeService.getAllRec(m);
        BigDecimal order = new BigDecimal(1);
        BigDecimal priority = null;

        if (recList != null && recList.size() > 0) {
            for (LiveDrawPicRoomType rec : recList) {
                rec.setPriority(order);

                if (beforeId != null) {
                    if (rec.getTypeid().equals(beforeId)) {
                        priority = rec.getPriority().add(new BigDecimal(0.1));
                    }
                } else {
                    if (rec.getTypeid().equals(afterId)) {
                        priority = rec.getPriority().subtract(
                                new BigDecimal(0.1));
```

```
                    }
                }
                order = order.add(new BigDecimal(1));

            }

            liveDrawPicTypeService.resizePriority(recList);

        } else {
            priority = new BigDecimal(1000.000);
        }

        return priority;

    }
```