9、对象中嵌套list对象 前台传递json数组

# 1、前台页面

## 解释

## 1、checked 表示是否选中，有时候json数组中没有，需要到后台中进行通过验证是否存在加上

## 2、weigth：表示权重，有时候，有时候josn数组中没有，需要到后台中进行通过验证是否存在加上

## 3、name 和value是不会消失的

[{

"name": "渠道",

"checked": true,

"weight": 1,

"value": 1

}, {

"name": "官网",

"checked": true,

"weight": 14,

"value": 2

}, {

"name": "广告",

"checked": false,

"weight": 0,

"value": 3

}, {

"name": "官网企业版有登录",

"checked": false,

"weight": 0,

"value": 4

}, {

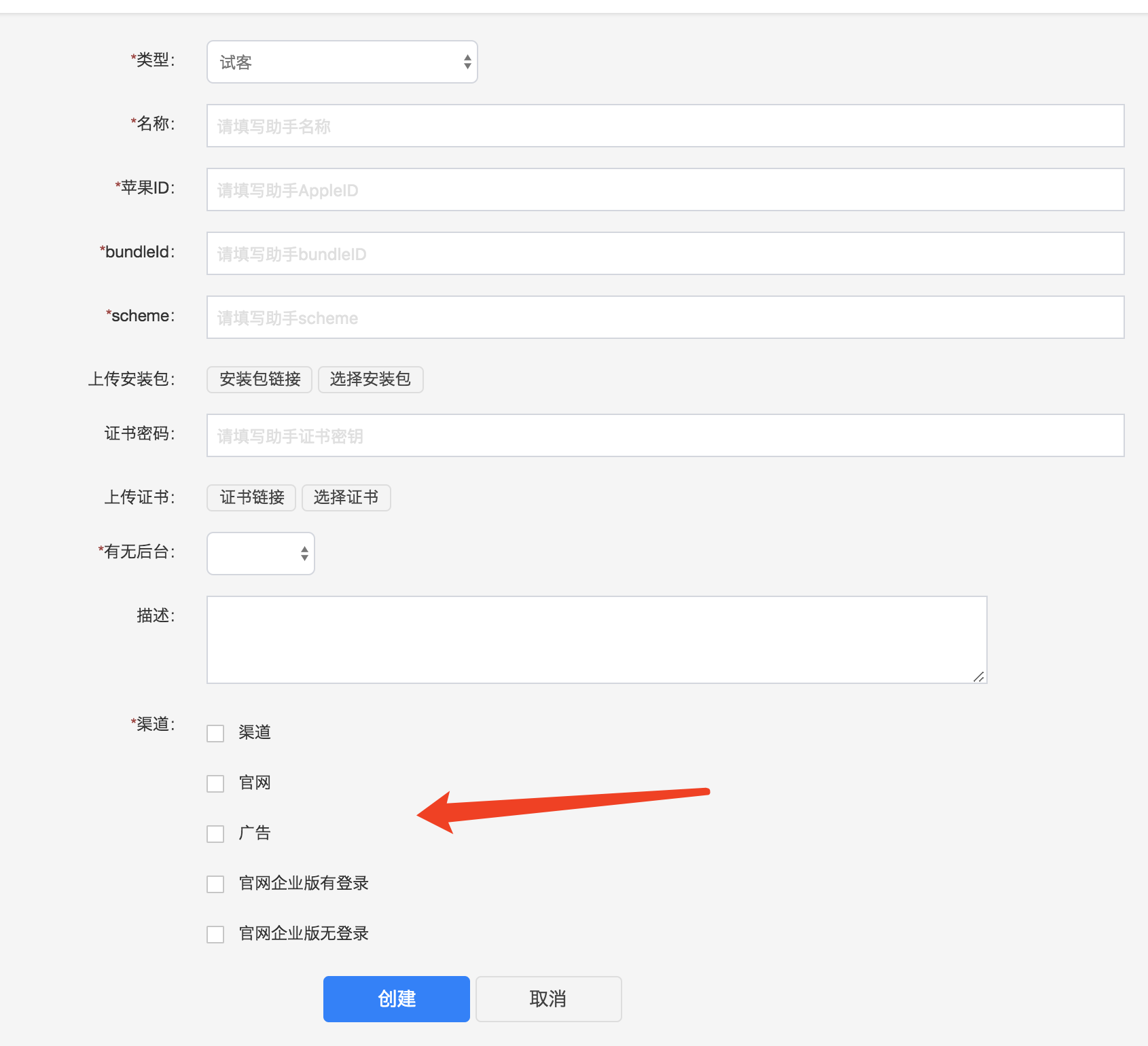
"name": "官网企业版无登录",

"checked": false,

"weight": 0,

"value": 5

}]



# 1、实体类

## 解释

### 1、channelJson 用来存放上面的json数组

### 2、channelJsonDesc ，private List<ChannelJson> channelJsonList; 用来将json数组解析后展示给前台 进行显示，这里需要注意的是，我这类的这两个字段并没有在该数据库实体类类的功能中使用到，而是在新版审核的时候，查看详情需要用到，所以才加上了。这个功能其实是在DTO SkinAppInfoData中

### 3、具体看DTO SkinAppInfoData

## 1、实体类 SkinAppInfoCheck

@Entity  
@Table(name = "skin\_app\_info\_check")  
public class SkinAppInfoCheck implements Serializable {  
  
  
 @Column(length = 50)  
 private String trackId; //应用苹果id  
 private String appid; //应用id  
 private String appSecret; //应用加密key  
 private String name; //名称  
 private String icon; //图标  
 private Integer type; //类型  
 private String bundleId; //bundleId  
 private String scheme; //scheme  
 private Integer status; //状态  
 private String downloadUrl; //下载url  
 private String filePath; //文件路径  
 private String remark; //备注  
 private String apnscer; //证书路径  
 @Column(length = 30)  
 private String password; //证书路径  
 private Long userId; //用户id  
 @Temporal(TemporalType.***TIMESTAMP***)  
 @Column(columnDefinition="TIMESTAMP DEFAULT CURRENT\_TIMESTAMP",insertable = true,updatable = false)  
 private Date cdate;  
  
 private Date udate;  
  
 private Integer haveBackstage; //有无后台  
  
 private String makerMemo; //标记备注  
  
  
 private String channelJson; //渠道 关联对象 ChannelJson.java  
  
 @Transient  
 private String channelJsonDesc;  
  
 @Transient  
 private List<ChannelJson> channelJsonList;

public StringBuilder getChannelJsonDesc() {  
 StringBuilder stringBuilder = new StringBuilder();  
 if(channelJsonList.size()!=0){  
 for(ChannelJson channelJson:channelJsonList){  
 if(channelJson.getChecked()==true){  
 stringBuilder.append(channelJson.getName()+":"+channelJson.getWeight()+"\n");  
  
 }  
 }  
 }  
 return stringBuilder;  
}

## 2、前后台交互DTO SkinAppInfoData List< ChannelJson > 其实只用来用提供给channelJsonDesc 服务的 ，并不是前后台传递的参数，传递的参数只有String 类型的 channelJson

public class SkinAppInfoData implements Serializable {  
  
 public static final int ***TYPE\_SHIKE*** = 1;  
 public static final int ***TYPE\_OUTSIDE*** = 2;  
 public static final int ***TYPE\_MOREAPPS*** = 3;  
 public static final int ***TYPE\_SHIKE\_OTHER*** = 4;  
 public static final int ***TYPE\_SHIKE\_ENTERPRISE\_LOGIN*** = 5;  
 public static final int ***TYPE\_SHIKE\_ENTERPRISE\_LOGIN\_NO*** = 6;  
  
 private static final long ***serialVersionUID*** = -1L;  
  
 private String appid; //应用id  
 private String appSecret; //应用加密key  
 private String name; //名称  
 private String icon; //图标  
 private Integer type; //类型  
 private String trackId; //应用苹果id  
 private String bundleId; //bundleId  
  
 private String makerMemo; //标记备注  
  
 private Integer haveBackstage; //有无后台  
  
 private String haveBackstageDesc; //有无后台  
  
 private String channelJson;  
  
 private String channelJsonDesc;  
  
 private List<ChannelJson> channelJsonList;  
  
 public StringBuilder getChannelJsonDesc() {  
 StringBuilder stringBuilder = new StringBuilder();  
 if(channelJsonList.size()!=0){  
 for(ChannelJson channelJson:channelJsonList){  
 if(channelJson.getChecked()==true){  
 stringBuilder.append(channelJson.getName()+":"+channelJson.getWeight()+"\n");  
  
 }  
 }  
 }  
 return stringBuilder;  
 }  
  
 public List<ChannelJson> getChannelJsonList() {  
 return channelJsonList;  
 }  
  
 public void setChannelJsonList(List<ChannelJson> channelJsonList) {  
 this.channelJsonList = channelJsonList;  
 }  
}

# 2、前台绑定checkbox复选框

## 1、js添加默认数据

$scope.channelLists=[  
 {name:"渠道",value:'1',weight:0,checked:false},  
 {name:"官网",value:'2',weight:0,checked:false},  
 {name:"广告",value:'3',weight:0,checked:false},  
 {name:"官网企业版有登录",value:'4',weight:0,checked:false},  
 {name:"官网企业版无登录",value:'5',weight:0,checked:false}  
];

## 2、html 绑定json数组

<div class="form-group">  
 <label class="col-sm-2 control-label"><span class="danger">\*</span>渠道：</label>  
 <div class="col-sm-6 ">  
 <div class="channel-list" ng-repeat="item in channelLists">

<input type="checkbox" ng-checked="item.checked==true" name="channel" value="{{***item***.value}}" ng-click="selectChannel(item)">  
 <span class="channel-name" ng-bind="item.name"></span>  
 <input class="form-control" ng-model="item.weight" name="channelQuality" style="width: 160px;" type="number"

ng-if="item.checked" min="0" step="1" max="100" placeholder="权重值最大为100" required>  
 </div>  
 </div>  
</div>

### 3、js设置checkbox绑定出现权值输入框

$scope.selectChannel=function (item) {  
 if(item.checked==true){  
 item.checked=undefined;  
 }else{  
 item.checked=true;  
 }  
}

# 3（重点）、点击保存开始传递json数组（）

## 1、 需要将json数组变成json字符串

### 1、angular.copy($scope.channelLists); //去掉$$hashKey属性

### 2、$scope.skin.channelJson = new *FormData*(); new出一个json 数组（个人理解）

### 3、将json数组转化为 字符串 angular.*toJson*($scope.json);

$scope.json =angular.copy($scope.channelLists); //去掉$$hashKey属性  
var channelJson = angular.*toJson*($scope.json);

$scope.skin.channelJson = new ***FormData***();  
$scope.skin.channelJson = channelJson;

## 2、因为这里的数据时通过DTO得到的，DTO 中包含了List<ChannelJson> channelJsonList; ,但是这种方式传参失败了，所以设置为null。可能是前后台只能传递List简单数组，不能传递list对象吧。

$scope.saveSkin = function(){  
 if($scope.skin.status==1){ //待审核不能重复提交  
 ngDialog.alert("您已经提交审核，请等待审核完成重新提交")  
 }else {  
 if(!$scope.processing){  
 $scope.processing = true;  
 //传递的参数是 时间戳  
 $scope.skin.cdate = undefined;  
  
 $scope.skin.status = 1; //1 表示待审核，默认是待审核  
  
   
  
 $scope.json =angular.copy($scope.channelLists); //去掉$$hashKey属性  
 var channelJson = angular.*toJson*($scope.json);

$scope.skin.channelJson = new ***FormData***();  
 $scope.skin.channelJson = channelJson;

$scope.skin.channelJsonList = null //传递数据失败，因为不能传递List对象

$http.post(***ctx*** + '/skin/save',$scope.skin)  
 .success(function(data, status) {  
 $scope.processing = false;  
 if(data.success){  
 $scope.add = false;  
 $scope.editFlag = false;  
 toastr.success('操作成功','提示');  
 $scope.skinData();  
 } else {  
 ngDialog.alert(data.message);  
 }  
 })  
 .error(function(data, status) {  
 ngDialog.alert('保存助手数据失败');  
 });  
 }  
 }  
};

# 4、Controller接收

@RequestMapping("save")  
@ResponseBody  
public ResponseBean saveSkins(String apprIds, String ccIds, SkinAppInfoData skinAppInfoData){  
  
 try {  
 SysAdminUser adminUser = AppSessionHelper.getSessionUser();  
  
 skinsService.saveSkin( apprIds, ccIds, adminUser.getId(),skinAppInfoData);  
  
 return ResponseBean.buildSuccess();  
 } catch (AppException e) {  
 return ResponseBean.buildFailure(e.getCode(),e.getMessage());  
 } catch (Exception e) {  
 logger.error(e.getMessage(), e);  
 return ResponseBean.buildFailure(ErrorCodeEnum.***系统错误***);  
 }  
}

# 6、（重点）service ，对传来的Json进行价值的获取，排除掉错误的数据 （比如weight,用户没有输入值，给了空值，但是weight是Integer类型）

JSONArray jsonArray = JSONArray.fromObject(skinAppInfoData.getChannelJson());  
  
 List<ChannelJson> channelJsons = new ArrayList<>();  
 for (int i =0; i < jsonArray.size(); i++) {  
 ObjectMapper objectMapper = new ObjectMapper();  
 try {  
 ChannelJson channelJson = objectMapper.readValue(jsonArray.get(i).toString(),ChannelJson.class);  
 //防止传来的参数，防止没有weight和checked 防止checked为false，但是weigth还有值  
 if("".equals(channelJson.getWeight())||channelJson.getWeight()==null||

"".equals(channelJson.getChecked())||null==channelJson.getChecked()||channelJson.getChecked()==false){  
 channelJson.setWeight(0);  
 channelJson.setChecked(false);  
 }  
 channelJsons.add(channelJson);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
}  
  
  
 appInfo.setChannelJson(JSONArray.fromObject(channelJsons).toString());

# 7、前台分页展示

## 1、观察页面



# 8、Controller Service前台获取数据

## 1、解释 ：上面7中的前台分页显示的内容其实，就是后台组织好的一个字符串，但是这个字符串需要提前组织好List<ChannelJson>

@Override  
public Page<SkinAppInfoData> findSkinAppInfoList(Pageable pageable, Object... param) throws AppException {  
 Map data = MyBatisHelper.mergeParameterMap(pageable,param);  
 List<SkinAppInfoData> dataList = skinsMapper.findSkinList(data);  
  
 for(SkinAppInfoData skinAppInfoData :dataList){  
 List<ChannelJson> channelJsonList = new ArrayList<>();  
 if(skinAppInfoData.getChannelJson()!=null&&!"".equals(skinAppInfoData.getChannelJson())) {  
 JSONArray jsonArray = JSONArray.fromObject(skinAppInfoData.getChannelJson());  
 for (int i = 0; i < jsonArray.size(); i++) {  
 ObjectMapper objectMapper = new ObjectMapper();  
 try {  
 ChannelJson channelJson = objectMapper.readValue(jsonArray.get(i).toString(), ChannelJson.class);  
 channelJsonList.add(channelJson);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 skinAppInfoData.setChannelJsonList(channelJsonList);  
  
 }  
  
 Long count = skinsMapper.countSkinList(data);  
 return new PageImpl<SkinAppInfoData>(dataList,pageable,count);  
}

# 9.（重点 ）选中编辑预览，让checkbox也能够复制预览，js编辑按钮函数 如下，

## 1、将Json字符串 转话为Json 数组 angular.*fromJson*(item.channelJson) ;

## 2、因为每次我们选中一条行数据编辑的时候， $scope.channelLists 的值就会变化，，所以当如果没有复选框数据的行的时候，要对她进行归零，也就是添加一个$scope.channelListsEmpt 重新赋值给它;

//临时存放，防止发生channelLists改变  
 $scope.channelListsEmpt =[  
 {name:"渠道",value:'1',weight:0,checked:false},  
 {name:"官网",value:'2',weight:0,checked:false},  
 {name:"广告",value:'3',weight:0,checked:false},  
 {name:"官网企业版有登录",value:'4',weight:0,checked:false},  
 {name:"官网企业版无登录",value:'5',weight:0,checked:false}];

$scope.edit = function(item) {  
 $scope.skin = angular.copy(item);  
 if($scope.skin.channelJsonList.length>0){  
 $scope.channelLists = angular.*fromJson*(item.channelJson) ;  
 }else {  
 $scope.channelLists = $scope.channelListsEmpt; //临时数组  
 }  
 $scope.add = false;  
 $scope.editFlag = true;  
};

# 10、在我们希望，通过其他方式修改，只留一个checkbox 有效数据的时候，（从前台来的只有一个主键和一个value值，也就是将除了拥有value的json字符串，其他的字符串制作成初始数据）

@Override  
public void updateChannelStatus(String appId,String channelListsEnableValue) {  
  
 SkinAppInfoCheck skinAppInfoCheck = skinAppInfoCheckDAO.findByTrackId(appId);  
 String channelJsonString = skinAppInfoCheck.getChannelJson();  
  
 JSONArray jsonArray = JSONArray.fromObject(channelJsonString);  
 List<ChannelJson> channelJsons = new ArrayList<>();  
 for (int i =0; i < jsonArray.size(); i++) {  
 ObjectMapper objectMapper = new ObjectMapper();  
 try {  
 ChannelJson channelJson = objectMapper.readValue(jsonArray.get(i).toString(),ChannelJson.class);  
 if(channelJson.getValue()!=Integer.parseInt(channelListsEnableValue)){  
 channelJson.setChecked(false);  
 channelJson.setWeight(0);  
 }  
 channelJsons.add(channelJson);  
  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 }  
  
 skinAppInfoCheck.setChannelJson(JSONArray.fromObject(channelJsons).toString());  
 skinAppInfoCheckDAO.save(skinAppInfoCheck);  
  
  
}