4、枚举 3种数据字典的使用（枚举-下拉菜单和绑定）

类别 DDKJ ; 经验

# 1、Js中简单的下拉

## 1、js中配置数据字典

$scope.qtypeList = [  
 {"type":'',"desc":"全部"}  
 ,{"type":1,"desc":"试客"}  
 ,{"type":5,"desc":"试客企业版登录"}  
 ,{"type":6,"desc":"试客企业版无登录"}  
];

## 2、前台开始绑定

类型：  
<div class="input-group">  
 <select style="width: 100px;" class="form-control"

ng-options="item.type as item.desc for item in qtypeList"

ng-model="query.type" >  
 </select>  
</div>

# 2、后端传来的map参数绑定（最好不是map而是对象，因为map的话，不知道）

#### 1、controller

@RequestMapping("map")  
@ResponseBody  
public Map map(){  
 Map<String,String> map = new HashMap<>();  
 map.put( "1","实地拜访");  
 map.put( "2","电话联系");  
 map.put( "3","邮件");  
 map.put( "4","微信");  
 return map;  
}

#### 2、请求赋值

$http.get(***ctx*** + '/contact/map')  
 .success(function(data, status) {  
 $scope.typeList = data;  
 })  
 .error(function(data, status) {  
 ngDialog.alert('查询失败');  
 });

### 3、前台开始绑定

<option ng-repeat="(key,value) in typeList"><a ng-click="chooseType(key,value)">{{***value***}}</a></ option >

## 4、其实之前是这样的

$scope.typeList=[  
 {label:'全部',value:''},  
 {label:'实地拜访',value:1},  
 {label:'电话联系',value:2},  
 {label:'邮件',value:3},  
 {label:'微信',value:4},  
 {label:'QQ',value:5},  
 {label:'钉钉',value:6},  
];  
  
  
$scope.typeName=$scope.typeList[0].label;

<div class="dropdown adDropdown">  
 <button class="btn btn-default dropdown-toggle multi-select flex-around-center" style="width: 130px;" type="button" id="dropdownMenu1" data-toggle="dropdown" aria-haspopup="true" aria-expanded="true">  
 {{typeName}}  
 <span class="caret"></span>  
 </button>  
 <ul class="dropdown-menu" aria-labelledby="dropdownMenu1">  
 <li ng-repeat="item in typeList"><a ng-click="chooseType(item)">{{***item***.label}}</a></li>  
 </ul>  
</div>

# 3、后端传来的枚举类开始绑定（可以输入选择）

## 1、controller获取枚举对象集合（使用对象对这个集合进行 包装了）

@RequestMapping("getWorkflowTaskType4Select")  
@ResponseBody  
public ResponseBean getWorkflowTaskType4Select() {  
 try {  
 List<EnumWorkflowTaskType> typeList = EnumWorkflowTaskType.getTypeList();  
 List<LabelValueBean> labelValueBeanList = new ArrayList<>();  
  
 for (EnumWorkflowTaskType type : typeList){  
 if (type.getType().compareTo(EnumWorkflowTaskType.***未知***.getType()) == 0){ //排除未知的枚举  
 continue;  
 }  
 LabelValueBean bean = new LabelValueBean(type.getDes(),String.valueOf(type.getType()));  
 labelValueBeanList.add(bean); //标签bean中添加枚举  
 }  
 return ResponseBean.buildSuccess(labelValueBeanList); //Object 对象存入  
 } 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***);  
 }  
}

### 1、枚举 EnumWorkflowTaskType 类

public enum EnumWorkflowTaskType {  
  
 ***预开发票申请***(1,"预开发票申请"),  
 ***特批额度申请***(2,"特批额度申请"),  
 ***特批单价申请***(3,"特批单价申请"),  
 ***应用推广代金券申请***(4,"应用推广代金券申请"),  
 ***多点宝代金券申请***(5,"多点宝代金券申请"),  
 ***汇款底单审核***(6,"汇款底单审核"),  
 ***客户合同审核***(7,"客户合同审核"),  
 ***拜访客户审核***(8,"拜访客户审核"),  
 ***客户实名认证审核***(9,"客户实名认证审核"),  
 ***客户发票信息审核***(10,"客户发票信息审核"),  
 ***应用审核***(11,"应用审核"),  
 ***资源包退款审核***(12,"资源包退款审核"),  
 ***客户提现审核***(13,"客户提现审核"),  
 ***对接接口审核***(14,"对接接口审核"),  
 ***商务或运营变更审核***(15,"商务或运营变更审核"),  
 ***多点账户变更审核***(16,"多点账户变更审核"),  
 ***快启代金券申请***(17,"快启代金券申请"),  
 ***客户发票作废申请***(18,"客户发票作废申请"),  
 ***客户发票作废重开申请***(19,"客户发票作废重开申请"),  
 ***客户发票部分作废申请***(20,"客户发票部分作废申请"),  
 ***创建资源包价格范围申请***(21,"创建资源包价格范围申请"),  
 ***财务流水手动录入申请***(22,"财务流水手动录入申请"),  
 ***合同章申请***(23,"合同章申请"),  
 ***结算单盖章申请***(24,"结算单盖章申请"),  
 ***客户变更账号信息申请***(25,"客户变更账号信息申请"),  
 ***助手管理信息审核申请***(26,"助手管理信息审核申请"), //张宇晋添加  
  
 ***多点宝媒体审核***(51,"多点宝媒体审核"),  
 ***多点宝广告审核***(52,"多点宝广告审核"),  
  
 ***未知***(99,"未知")  
 ;  
  
 public Integer type;  
 public String des;  
  
 EnumWorkflowTaskType(Integer type, String des) {  
 this.type = type;  
 this.des = des;  
 }  
  
  
 public Integer getType() {  
 return type;  
 }  
  
 public String getDes() {  
 return des;  
 }  
  
 public static boolean checkExist(Integer type){  
 if(type == null) return false;  
 for (EnumWorkflowTaskType value : EnumWorkflowTaskType.values()){  
 if (value.type.compareTo(type) == 0){  
 return true;  
 }  
 }  
 return false;  
 }  
  
  
  
 public static EnumWorkflowTaskType getEnumCashFlowType(Integer type){  
 if(type == null) return EnumWorkflowTaskType.***未知***;  
 for (EnumWorkflowTaskType value : EnumWorkflowTaskType.values()){  
 if (value.type.compareTo(type) == 0){  
 return value;  
 }  
 }  
 return EnumWorkflowTaskType.***未知***;  
 }  
  
 public static String getDes(Integer type){  
 if(type == null) return EnumWorkflowTaskType.***未知***.des;  
 for (EnumWorkflowTaskType value : EnumWorkflowTaskType.values()){  
 if (value.type.compareTo(type) == 0){  
 return value.des;  
 }  
 }  
 return EnumWorkflowTaskType.***未知***.des;  
 }  
  
 public static List<EnumWorkflowTaskType> getTypeList(){  
 return Arrays.asList(values());  
 }  
}

### 2、下拉列表专用 -标签类Bean LabelValueBean

/\*\*  
 \* 下拉列表用  
 \*/  
public class LabelValueBean implements Serializable{  
 private static final long ***serialVersionUID*** = -1211726511402154326L;  
  
 private String label;  
 private String value;  
  
 private Boolean checked = false;  
  
 public LabelValueBean() {  
 }  
  
 public LabelValueBean(String label, String value) {  
 this.label = label;  
 this.value = value;  
 }  
  
 public String getValue() {  
 return value;  
 }  
  
 public void setValue(String value) {  
 this.value = value;  
 }  
  
 public String getLabel() {  
 return label;  
 }  
  
 public void setLabel(String label) {  
 this.label = label;  
 }  
  
 public Boolean getChecked() {  
 return checked;  
 }  
  
 public void setChecked(Boolean checked) {  
 this.checked = checked;  
 }  
}

## 2、前端调用获取Date

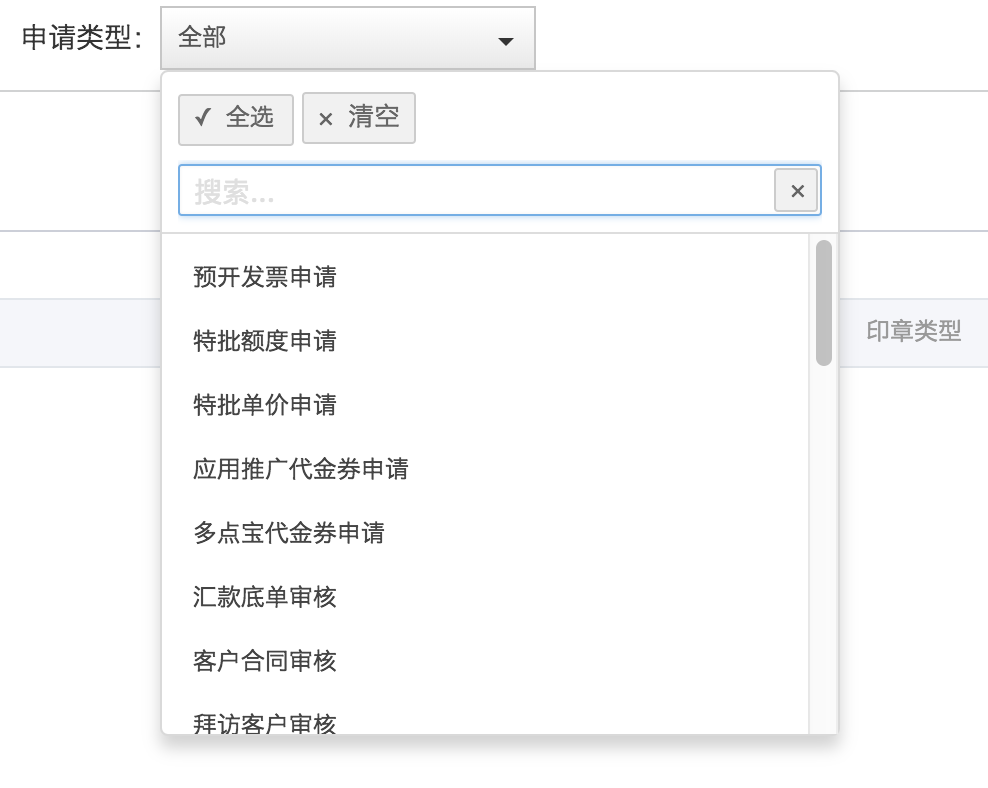
### 1、加载页面verifySignetData data数据一定要放到这个获取的方法后面

$scope.localLang = {  
 selectAll: "全选",  
 selectNone: "清空",  
 reset: "重置",  
 search: "搜索...",  
 nothingSelected: "全部" //default-label is deprecated and replaced with this.  
};  
  
$scope.typeList = [];  
$scope.outputModelType = {outputData: []};  
  
$scope.getWorkflowTaskType4Select = function(){  
 $http.post(***ctx*** + '/sys/default/appr/getWorkflowTaskType4Select',{})  
 .success(function(data, status) {  
 if(data.success){  
 $scope.typeList = data.result;  
 } else {  
 ngDialog.alert(data.message);  
 }  
 })  
 .error(function(data, status) {  
 ngDialog.alert('获取审批类型失败');  
 });  
};  
  
$scope.verifySignetData(0);  
$scope.getWorkflowTaskType4Select(); //注意上面加载数据的顺序

## 3、前端开始绑定

申请类型：  
<div class="dropdown input-group select-wrap adManager">  
 <div  
 isteven-multi-select  
 input-model="typeList"  
 output-model="outputModelType.outputData"  
 button-label="label"  
 item-label="label"  
 tick-property="checked"  
 search-property="label"  
 translation="localLang"  
 helper-elements="all none filter"  
 output-properties="value label"  
 max-height="250px"  
 >  
 </div>  
</div>

## 4、页面展示



## 5、前端点击查询，调用后台数据 。绑定数据字典选中的数据，（注意：这里是一个集合数组了哦）

$scope.verifySignetData = function(page){  
 $scope.query.page = page;  
 $scope.query.taskTypeList = [];  
  
 angular.forEach( $scope.outputModelType.outputData, function( value, key ) {  
 $scope.query.taskTypeList.push(value.value);  
 });  
  
 $http.post(***ctx*** + '/verify/signet/data',$scope.query)  
 .success(function(data, status) {  
 $scope.pageResult = data;  
 $scope.verify = undefined;  
 $scope.query.pageSize=$scope.pageResult.size;  
 })  
 .error(function(data, status) {  
 $scope.processing = false;  
 ngDialog.alert('加载数据失败');  
 });  
};

## 6、后端接收数据

#### 1、ctroller

@RequestMapping("data")  
@ResponseBody  
public Page<SignetApplyRecord> data(@RequestParam(defaultValue = "0") Integer page, SignetQuery query, @RequestParam(defaultValue = "15")Integer pageSize){  
 SysAdminUser adminUser = AppSessionHelper.getSessionUser();  
 Pageable pageable = new PageRequest(page, pageSize);  
 return verifySignetService.findVerifySignetData(adminUser.getId(),pageable,query);  
}

### 2、这里重点查看下这个query 对象

public class SignetQuery implements Serializable {  
  
 private static final long ***serialVersionUID*** = -4564423981924197001L;  
  
 private Long userId;  
 private Integer offset;  
 private Integer limit;  
 private Date startDate;  
 private Date endDate ;  
  
 private Long currAdmId4Auth; //当前登录用户 用于查询权限  
  
 private String userParam;  
 private Integer signetType; //EnumWorkflowTaskType  
 private Integer taskType; //EnumWorkflowTaskType 中 关于印章的类型  
 private Integer status;  
  
 private List<Integer> taskTypeList; //申请状态 EnumWorkflowTaskStatus

### 3、观察这个List在mybatis中的查询情况

<select id="findVerifySignetList" resultType="com.duodian.admore.entity.db.signet.SignetApplyRecord">  
 SELECT t.*\**,d.nickName userNickName,ifnull(e.realName,f.realName) userAuthName,b.name applyAdmName,ura.remittanceAccount  
 FROM signet\_apply\_record t  
 LEFT JOIN user d ON d.id = t.userId  
 LEFT JOIN user\_auth\_personal e on e.userId = t.userId AND e.status = 1  
 LEFT JOIN user\_auth\_enterprise f on f.userId = t.userId AND f.status = 1  
 LEFT JOIN user\_email\_account g ON g.userId = t.userId  
 left join sys\_admin\_user b on b.id = t.applyAdmId  
 LEFT JOIN user\_remittance\_account ura ON ura.userId = t.userId  
 <if test="currAdmId4Auth != null">  
 LEFT JOIN sys\_admin\_user\_customer h ON h.admId = #{currAdmId4Auth} AND h.userId = t.userId  
 LEFT JOIN (SELECT wt.pid,count(*\**) cc FROM workflow\_task\_examine\_details wte LEFT JOIN workflow\_task wt ON wt.id = wte.taskId WHERE wt.taskType = 23 AND wte.admId = #{currAdmId4Auth} GROUP BY wt.pid) w ON w.pid = t.id  
 </if>  
 <where>  
 <include refid="findVerifySignetWhere"></include>  
 </where>  
 order by t.id desc  
 <if test="offset != null and limit != null">  
 limit #{offset}, #{limit}  
 </if>  
</select>  
  
<select id="countVerifySignet" resultType="java.lang.Long">  
 SELECT count(*\**) FROM signet\_apply\_record t  
 LEFT JOIN user d ON d.id = t.userId  
 LEFT JOIN user\_auth\_personal e on e.userId = t.userId AND e.status = 1  
 LEFT JOIN user\_auth\_enterprise f on f.userId = t.userId AND f.status = 1  
 LEFT JOIN user\_email\_account g ON g.userId = t.userId  
 left join sys\_admin\_user b on b.id = t.applyAdmId  
 <if test="currAdmId4Auth != null">  
 LEFT JOIN sys\_admin\_user\_customer h ON h.admId = #{currAdmId4Auth} AND h.userId = t.userId  
 LEFT JOIN (SELECT wt.pid,count(*\**) cc FROM workflow\_task\_examine\_details wte LEFT JOIN workflow\_task wt ON wt.id = wte.taskId WHERE wt.taskType = 23 AND wte.admId = #{currAdmId4Auth} GROUP BY wt.pid) w ON w.pid = t.id  
 </if>  
 <where>  
 <include refid="findVerifySignetWhere"></include>  
 </where>  
</select>  
  
<sql id="findVerifySignetWhere">  
 <if test="startDate != null">  
 <![CDATA[ and t.cdate >= #{startDate} ]]>  
 </if>  
 <if test="endDate != null">  
 <![CDATA[ and t.cdate <= #{endDate} ]]>  
 </if>  
 <if test="userParam != null and userParam != '' and userParam != 'undefined'">  
 AND (t.userId = #{userParam} OR d.nickName LIKE CONCAT('%',#{userParam},'%' ) OR e.realName LIKE  
 CONCAT('%',#{userParam},'%' ) OR f.realName LIKE CONCAT('%',#{userParam},'%' ) OR g.email LIKE  
 CONCAT('%',#{userParam},'%' ))  
 </if>  
 <if test="taskType != null">  
 and t.taskType = #{taskType}  
 </if>  
 <if test="taskTypeList != null and taskTypeList.size() > 0">  
 and t.taskType IN  
 <foreach collection="taskTypeList" index="index" item="at" open="(" separator="," close=")">  
 #{at}  
 </foreach>  
 </if>  
 <if test="signetType != null">  
 and t.type = #{signetType}  
 </if>  
 <if test="status != null">  
 and t.status = #{status}  
 </if>  
 <if test="currAdmId4Auth != null ">  
 <![CDATA[ and (h.admId = #{currAdmId4Auth} OR w.cc > 0) ]]>  
 </if>  
</sql>

# 4、枚举数据字典的使用

## 1、在从后台向前台取得数据的时候（类型都是Integer）

### 1、int比较

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;

public String getTypeStr() {  
 if(this.type == null) return "未知";  
 if(this.type.intValue() == ***TYPE\_SHIKE***){  
 return "试客";  
 }else if(this.type.intValue() == ***TYPE\_OUTSIDE***){  
 return "OPEN平台";  
 }else if(this.type.intValue() == ***TYPE\_MOREAPPS***){  
 return "MOREAPPS";  
 }else if(this.type.intValue() == ***TYPE\_SHIKE\_OTHER***){  
 return "试客其它";  
 }else if(this.type.intValue() == ***TYPE\_SHIKE\_ENTERPRISE\_LOGIN***){  
 return "试客企业版登录";  
 }else if(this.type.intValue() == ***TYPE\_SHIKE\_ENTERPRISE\_LOGIN\_NO***){  
 return "试客企业版无登录";  
 }else {  
 return "未知平台";  
 }  
}

### 2、integer 的比较

public String getHaveBackstageDesc(){  
 if (this.haveBackstage == null){  
 return "未知";  
 }  
 if (this.haveBackstage.compareTo(0) == 0){  
 return "无后台";  
 } else if (this.haveBackstage.compareTo(1) == 0){  
 return "有后台";  
 }else {  
 return "未知";  
 }  
  
}

### 3、直接通过枚举获取数据

public String getTaskTypeDesc() {  
 return EnumWorkflowTaskType.getDes(this.taskType);  
}

# 5、对于默认数据的选中

## 1.写死

<div class="form-item flex-col-center">  
 <div class="item-title"><span class="adError">\*</span> 销售阶段</div>  
 <div class="input-group">  
 <select class="item-select" ng-model="chance.status" required>  
 <option value="">请选择</option>  
 <option value="1" ng-selected="chance.status==1" >初期沟通</option>  
 <option value="2" ng-selected="chance.status==2" >意向需求</option>  
 <option value="3" ng-selected="chance.status==3" >产品试用</option>  
 <option value="4" ng-selected="chance.status==4" >业务洽谈</option>  
 <option value="5" ng-selected="chance.status==5" >即将下单</option>  
 <option value="6" ng-selected="chance.status==6" >成功合作</option>  
 </select>  
 </div>

## 2、其他则是根据status即可自动选中

# 6、接3，下拉选中一个

## 1、contrler获取数据

@RequestMapping("findContactById")  
@ResponseBody  
public ResponseBean findContactById(Long id){  
 try {  
 Contact contact = contactService.findContactById(id);  
 CustomerChance customerChance = customerChanceService.findCustomerChanceById(contact.getCustomerId());  
 contact.setAppName(customerChance.getAppName());  
 return ResponseBean.buildSuccess(contact);  
 } catch (AppException e) {  
 return ResponseBean.buildFailure(e.getMessage());  
 }  
}

## 2、js控制

$scope.localLang = {  
 selectAll: "全选",  
 selectNone: "清空",  
 reset: "重置",  
 search: "搜索...",  
 nothingSelected: "请选择..." //default-label is deprecated and replaced with this.  
};

$scope.getCustomerList = function(){  
 $http.post(***ctx*** + '/contact/getCustomerList')  
 .success(function(data, status) {  
 if(data.success){  
 $scope.appNameList = data.result;  
 ngDialog.closeMouseDialog();  
 } else {  
 ngDialog.alert(data.message);  
 }  
 })  
 .error(function(data, status) {  
 ngDialog.alert('操作失败');  
 });  
};

## 3、前端页面

<div class="select-wrap adManager">  
 <div  
 isteven-multi-select  
 input-model="appNameList"  
 output-model="outputModelTypeModel.outputData"  
 button-label="appName"  
 item-label="appName"  
 tick-property="checked"  
 search-property="appName"  
 translation="localLang"  
 helper-elements="all none filter"  
 output-properties="id appName"  
 max-height="200px"  
 selection-mode="single"  
 >  
 </div>  
</div>

## 4、页面展示



## 5、选中保存时候的一部分方法

if($scope.outputModelTypeModel.outputData[0]==undefined||$scope.outputModelTypeModel.outputData[0]==null||$scope.outputModelTypeModel.outputData[0]==''){  
 ngDialog.alert("请输入产品名称");  
 return ;  
}  
$scope.contact.customerId = $scope.outputModelTypeModel.outputData[0].id;

## 6、编辑编辑默认选中

$scope.appNameList[0].appName=$scope.contact.appName;  
$scope.appNameList[0].id=$scope.contact.customerId;  
$scope.appNameList[0].checked =true;



# 7、点击下拉框进行监听，同时设置另外的值（想法：watch进行监听 输入框选中的值）

<div class="form-divider"></div>  
 <div class="form-item flex-col-center">  
 <div class="item-title">老后台客户</div>  
 <div class="select-wrap adManager">  
 <div  
 isteven-multi-select  
 input-model="olduserList"  
 output-model="olduserListOutputModel.outputData"  
 button-label="customerName"  
 item-label="customerName"  
 tick-property="checked"  
 search-property="customerName"  
 translation="localLang"  
 helper-elements="filter"  
 output-properties="customerId customerName"  
 selection-mode="single"  
 max-height="250px"  
∂∂ >  
 </div>  
 </div>  
 </div>

$scope.olduserList = [];  
$scope.olduserListOutputModel = {outputData: []};  
$scope.getOldUserList = function () {  
 $http.post(***ctx*** + '/workflow/oldBackGroundBalance/transfer/getOldUserList', {})  
 .success(function (data, status) {  
 if (data.success) {  
 $scope.olduserList = data.result;  
 } else {  
 ngDialog.alert(data.message);  
 }  
 })  
 .error(function (data, status) {  
 ngDialog.alert('获取老后台客户数据失败');  
 });  
};  
$scope.query.balance ="请选择一个老后台客户";  
$scope.$watch('olduserListOutputModel.outputData[0]', function() {  
 for(var i= 0;i< $scope.olduserList.length ;i++){  
 if($scope.olduserList[i].customerId==$scope.olduserListOutputModel.outputData[0].customerId){  
 $scope.query.balance = $scope.olduserList[i].balance ;  
 }  
 }  
  
});

# 8、点击保存判断下拉框是否已经输入了值

if ($scope.olduserListOutputModel.outputData == null || $scope.olduserListOutputModel.outputData.length <= 0){  
 toastr.warning('请选择一个旧后台客户', '提示');  
 return;  
}

# 10、接1、2

## 1、controller

@RequestMapping("getDesignerList")  
@ResponseBody  
public ResponseBean getDesignerList(){  
 List<EnumDesigner> designerList = EnumDesigner.getTypeList();  
 List<EnumData> list = new ArrayList<>();  
 for (EnumDesigner designer : designerList){  
 EnumData data = new EnumData();  
 if(designer.getType() == EnumDesigner.***未知***.getType()) continue;  
 data.setType(designer.getType());  
 data.setDes(designer.getDes());  
 list.add(data);  
 }  
 return ResponseBean.buildSuccess(list);  
}

## 2、枚举

package com.duodian.admore.zhaobutong.enums;  
  
import java.util.Arrays;  
import java.util.List;  
  
/\*\*  
 \* Created by james on 2018/3/14.  
 \* 设计师枚举  
 \*/  
public enum EnumDesigner {  
 ***陆天博***(1,"陆天博"),  
 ***史雪东***(2,"史雪冬"),  
 ***程博***(3,"程博"),  
 ***李雪娇***(4,"李雪娇"),  
  
  
 ***未知***(99,"未知")  
 ;  
  
  
 public Integer type;  
 public String des;  
  
 EnumDesigner(Integer type, String des) {  
 this.type = type;  
 this.des = des;  
 }  
  
 public static List<EnumDesigner> getTypeList() {  
 return Arrays.asList(values());  
 }  
  
  
 public Integer getType() {  
 return type;  
 }  
  
 public String getDes() {  
 return des;  
 }  
  
 public static boolean checkExist(Integer type){  
 if(type == null) return false;  
 for (EnumDesigner value : EnumDesigner.values()){  
 if (value.type.compareTo(type) == 0){  
 return true;  
 }  
 }  
 return false;  
 }  
  
  
  
 public static EnumDesigner getEnumDesigner(Integer type){  
 if(type == null) return EnumDesigner.***未知***;  
 for (EnumDesigner value : EnumDesigner.values()){  
 if (value.type.compareTo(type) == 0){  
 return value;  
 }  
 }  
 return EnumDesigner.***未知***;  
 }  
  
 public static String getDes(Integer type){  
 if(type == null) return EnumDesigner.***未知***.des;  
 for (EnumDesigner value : EnumDesigner.values()){  
 if (value.type.compareTo(type) == 0){  
 return value.des;  
 }  
 }  
 return EnumDesigner.***未知***.des;  
 }  
}

### 2.2、枚举返回

package com.duodian.admore.zhaobutong.bean;  
  
import java.io.Serializable;  
  
/\*\*  
 \* Created by james on 2018/3/15.  
 \*/  
public class EnumData implements Serializable {  
  
 private Integer type;  
 private String des;  
  
 public Integer getType() {  
 return type;  
 }  
  
 public void setType(Integer type) {  
 this.type = type;  
 }  
  
 public String getDes() {  
 return des;  
 }  
  
 public void setDes(String des) {  
 this.des = des;  
 }  
}

## 3、js接收返回值

$scope.getDesignerList = function(){  
 $http.post(***ctx*** + '/material/getDesignerList')  
 .success(function(data, status) {  
 if(data.success){  
 $scope.designerList = data.result;  
 } else {  
 ngDialog.alert(data.message);  
 }  
 })  
 .error(function(data, status) {  
 ngDialog.alert('获取设计师列表失败');  
 });  
};

## 4、前台展示

<div class="form-divider"></div>  
<div class="form-item form-remark">  
 <div class="item-title">设计师</div>  
 <div class="input-group">  
 <select class="item-select" ng-options="item.type as item.des for item in designerList" ng-model="material.designer"></select>  
 *<#--<select class="item-select" ng-model="material.designer" required>  
 <option value="">请选择</option>  
 <option ng-repeat="item in designerList">{{item}}</option>  
 </select>-->* </div>  
</div>

# 11、数据库中的字段内容

public ResponseBean getSysAdminUserSelect() {  
 try {  
 List<LabelValueBean> labelValueBeanList = new ArrayList<>();  
 List<SysAdminUser> sysAdminUsers = sysAdminUserRepository.findByStatus(EnumDelete.***可用***.status);  
 for (SysAdminUser sysAdminUser : sysAdminUsers) {  
 //由于目前可能被多个数据库使用，通过id判断过来是不准确的，在一键迁移的时候，主键的id不一样是 小当优惠券的id，所以我通过这种方式来进行处理了  
 LabelValueBean bean = new LabelValueBean(sysAdminUser.getName(), String.valueOf(sysAdminUser.getAdminId()));  
 labelValueBeanList.add(bean); //标签bean中添加枚举  
 }  
 return ResponseBean.buildSuccess(labelValueBeanList); //Object 对象存入  
 } catch (AppException e) {  
 ExceptionLogUtils.log(e, this.getClass());  
 return ResponseBean.buildFailure(e.getCode(), e.getMessage());  
 } catch (Exception e) {  
 ExceptionLogUtils.log(e, this.getClass());  
 return ResponseBean.buildFailure(e.getMessage());  
 }  
}