# 文件上传

File文件上传:标识ssbm

文件信息实体类FileInfo-id编号

文件上传记录ImportRecord 上传记录id 文件fileid=FileInfo.id

文件File上传列表:根据标识ssbm查询

ImportRecord 导入记录id 导入文件fileid

上传文件中的数据:参数为id fileid

数据正常：fileid-FileInfo + handler + ImportParams = ImportRecord更新次数

数据异常：Import\_\*\*\_cwsj 设置导入记录id

查看错误数据:参数为导入记录id

Import\_\*\*\_cwsj 由id查询列表grid

导出错误数据：参数为导入记录id

List< Import\_\*\*\_cwsj > = \*\*Service.getDao().queryByIrid(id);

# 时间查询sqsj

Huji\_gaclDto ：@CondtionExpression 模糊查询like 等值查询equals

1.中间值

Date sqsj\_s (value = "sqsj", type = CondtionType.greaterthanOrequal)

Date sqsj\_e (value = "sqsj", type = CondtionType.lessthanOrequal)

2.数据字典

String tjzt (value = "tjzt.code", type = CondtionType.equal)

3.关联查询

String xm (joinName="record\_gacl",value = "xm", type = CondtionType.equal)

Huji\_gacl:

@OneToMany(targetEntity = Record\_gacl.class)

@JoinColumn(name="huji\_gacl\_id")

private List<Record\_gacl> record\_gacl;

Record\_gacl:

@Getter

@Setter

@Des("公安车辆个人车辆信息核对户表id")

@Column(name = "huji\_gacl\_id")

private Long huji\_gacl\_id;

# 核对与导入/出

首页面: 查询+核对+数据列表（类型转换+单个成员点击响应）

单个成员核对页面+成员编号id：个人信息+绑定数据grid（增加+修改）+删除（绑定数据id）+完成核对（成员编号id）

绑定数据页面+成员编号id：增加+修改

# 工具

## combo

<td><@c.combo code="SSZT" name="sszt"/></td>

var khx = $("#"+modelName+"\_khx1").combobox('getValue');

## Dto(List)

@Des("角色权限进行判断收文状态")

@CondtionExpression(prameType=String.class,value="swzt.code",type=CondtionType.in)

private List<String> swzts;

List<String> list = new ArrayList<String>();

list.add("1");

list.add("2");

list.add("3");

list.add("4");

list.add("5");

list.add("6");

map.put("swzts", list);

## Panel加载报表

|  |
| --- |
| function getBaoBiao(){  var formData = qf1.serializeObject({transcript:"overlay"});  $("#"+modelName+"\_table").panel({  href : URL.loadExport,  queryParams : formData,  onLoad : function(){  closeLoadingDiv();  }  });  } |

## 右下角提示

|  |
| --- |
| /\*\*发布\*\*/  function release(id){  $.ajax({  type : "post",  dataType : "json",  url :URL.release,  data : {id:id},  success : function(data) {  showMsg(data.returnMsg);  if (data.success) {  $.messager.progress('close');  grid.datagrid('reload');  }else{  $.messager.alert('系统提示','请进行指标内容的配置，否则无法进行发布!');  }  }  });  } |

## Tree(选择事件)

<td><@c.getOrgTree name="ssdq" code="" required="false"/></td>

//tree不可多选

<@c.getBmdwTree name="fgbm" code="bmdw"

multiple="true" onBeforeCheck="fgbmCheck(node,checked)"

required="true" id="yhgl\_fgbm" />

function fgbmCheck(node,checked){

var ok = true;

var newValue = node.id;

$.ajax({

url : ctx + "/oabg/yhgl/getFgbms",

type : "post",

async:false,

success : function(data){

if(data.indexOf(newValue) != -1 ){

showMsg("该部门已被分管");

ok = false;

}

}

});

return ok;

}

## 必填红星

<@c.redstar/>

## 时间控件

<td><input name="sssj" class="easyui-datebox"

data-options="editable:false,icons:

[{iconCls:'icon-combo-clear',handler :

function(e) { $(e.data.target).combo('clear');}}]"></td>

|  |
| --- |
| //指标年度--  // Date ndkhid ;  // Date ndkhid1 ;  // if(!"".equals(StringSimple.nullToEmpty(map.get("zbnd")))){  // String str =(String) map.get("zbnd");  // int nd = Integer.parseInt(str);  // ndkhid1 = sdf.parse(""+nd+"-12-31 23:59:59");  // nd-=1;  // ndkhid = sdf.parse(""+nd+"-12-31 23:59:59");  // }else{//默认为当前年份的数据  // Calendar cal = Calendar.getInstance();  // int year = cal.get(Calendar.YEAR);  // int pyear =year -1;  // ndkhid = sdf.parse(""+pyear+"-12-31 23:59:59");  // ndkhid1 = sdf.parse(""+year+"-12-31 23:59:59");  // } |

|  |
| --- |
| datetimebox  function getNowFormatDate() {  var date = new Date();  var seperator1 = "-";  var month = date.getMonth() + 1;  var strDate = date.getDate();  if (month >= 1 && month <= 9) {  month = "0" + month;  }  if (strDate >= 0 && strDate <= 9) {  strDate = "0" + strDate;  }  var currentdate = date.getFullYear() + seperator1 + month + seperator1 + strDate;  return currentdate;  } |

## 年度控件

<input name="zbnd" class="textbox Wdate"

onclick="WdatePicker

({dateFmt:'yyyy',minDate:'2000',maxDate:'2050'})"

style="width:150px"/>

## 年度控件Change事件

|  |
| --- |
| <td><input id="ssgl\_fqss\_addForm\_zbnd" name="zbnd" class="textbox Wdate" onFocus="WdatePicker({dateFmt:'yyyy',minDate:'2000',maxDate:'2050',onpicked:function(){changeZbnd();}})" style="width:150px;"/></td>  <script type="text/javascript">  function changeZbnd(){  var zbnd = $("#ssgl\_fqss\_addForm\_zbnd").val();  $("#ssgl\_fqss\_addForm\_khx").combobox({  url : ctx + "/ds/getKhxx?zbnd="+zbnd  });  }  </script>  <td class="label">考核项:<@c.redstar/></td>  <td><@c.combo id="ssgl\_fqss\_addForm\_khx" name="getKhxx" code="getKhx" required="true"/></td> |

|  |
| --- |
| /\*\*  \* 根据指标年度自动获取该年度存在的考核项  \* @author wsz  \* @created 2017年8月12日  \* @return  \*/  @ResponseBody  @RequestMapping(value ="/ds/getKhxx")  public String getKhxx(String zbnd){  Map<String,Object> map = new HashMap<String,Object>();  if(!"".equals(StringSimple.nullToEmpty(zbnd))){  map.put("zbnd", zbnd);  }  List<Zbpz\_ndkh> klist=zbpz\_ndkhService.query(Zbpz\_ndkhDto.class,map);  Set<String> kset=new HashSet<String>();  for(Zbpz\_ndkh z:klist){  kset.add(z.getKhx());  }  List<Map<String,Object>> khxs=new ArrayList<Map<String,Object>>();  for(String khx:kset){  Map<String,Object> tmap=new HashMap<String,Object>();  tmap.put("value", khx);  tmap.put("text", khx);  khxs.add(tmap);  }  return JSON.toJSONString(khxs);  } |

## 字典按钮

<@command menu="SSGL\_FQSS"

name="FQSS\_ADDBTN,FQSS\_DELETEBTN"/>

绑定事件

$("a[data-action=FQSS\_ADDBTN").bind("click",function(){

openAddWin();

});

隐藏

$("a[data-action='swng\_swclform']").hide();

## 查询/清空按钮

<td><a id="ssgl\_fqss\_query" class="easyui-linkbutton"

data-options="plane:false,iconCls:'iconfont icon-standard-zoom'">查询</a></td>

<td><a id="ssgl\_fqss\_clear" class="easyui-linkbutton"

data-options="plain:false,iconCls:'l-btn-icon icon-standard-cross'">清空</a></td>

$("#"+modelName+"\_query").bind("click",function(){

var formData=queryForm.serializeObject({

transcript:"overlay"

});

grid.datagrid("load",formData);

});

## Tab分割

data-options="href:'${s.ctx}/jxdftj/khbm/bbForm'"

## 弹窗新页面

function openAddWin(){

var win=$("<div id="+modelName+"\_addForm></div>").window({

title : "新增申诉",

height : 350,

width : 1000,

maximized:true, //页面自适应最大化

href :URL.addSsUrl,

onLoad : function(){

closeLoadingDiv();

},

onClose : function(){

win.window("destroy");

}

});

}

接收数据:

<script type="text/javascript">

var sid = '${id!}';

</script>

## 获取所选列表grid

var row=$("#zbpz\_ejzbmk\_grid").datagrid("getSelected");

function gridChecked(grid){

var rows=grid.datagrid("getChecked");

if(rows.length > 0){

return $.map(rows,function(n){

return n.id;

}).join(",");

}

## Grid联动

第一个gird：双击修改自己；单击刷新第二个grid

第二个grid：点击新增，必须先选择了第一个grid

var row=$("#zbpz\_ejzbmk\_grid").datagrid("getSelected");

|  |
| --- |
| /\*\* 渲染一级指标模块Grid\*\*/  function yjzbmkRenderGrid() {  $("#zbpz\_yjzbmk\_grid").datagrid({  url : URL.yjzbmkGridData,  queryParams:{  ndkhid : ndkhid  },  columns : [[  {field : "id",checkbox : true},  {field : "zbmk",title : "指标模块",align : "center",width : "200",halign : "center"},  ]],    onDblClickRow:function(index, row){  var zbmkid=row.id;  zbmkAddWin(ndkhid,zbmkid);  },  onClickRow:function(index, row){  var zbmkid=row.id;  var zbnr\_grid=$("#zbpz\_yjzbnr\_grid");  var zbmk\_grid=$("#zbpz\_yjzbmk\_grid");  zbnrRenderGrid(zbnr\_grid,zbmkid,zbmk\_grid);  }  });  } |

|  |
| --- |
| /\*\* 渲染指标内容Grid\*\*/  function zbnrRenderGrid(zbnr\_grid,zbmkid,zbmk\_grid) {  zbnr\_grid.datagrid({  url : URL.zbnrGridData+"?zbmkid="+zbmkid,  columns : [[  {field : "id",checkbox : true},  {field : "zbnr",title : "指标内容",align : "center",width : "200",halign : "center"},  {field : "xtpt",title : "系统平台",align : "center",width : "200",halign : "center"},  {field : "zbzdz",title : "指标最低值",align : "center",width : "120",halign : "center"},  {field : "jcfz",title : "基础分值",align : "center",width : "120",halign : "center"},  {field : "zjfz",title : "增加分值",align : "center",width : "120",halign : "center"},  {field : "zjz",title : "增加值",align : "center",width : "120",halign : "center"},  ]],  onDblClickRow:function(index, row){    var rows=zbmk\_grid.datagrid("getSelected");  if(rows==null){  alert("请点击选择一个指标模块！");  }else{  zbnrAddWin(rows.id,row.id);  }  },  });  } |

## grid按钮

var row=$("#zbpz\_yjzbmk\_grid").datagrid("getSelected");

<table id="ssgl\_fqss\_grid" toolbar="div[data-id=ssgl\_fqss\_list\_grid\_bar]">

</table>

<div data-id="ssgl\_fqss\_list\_grid\_bar" style="padding:2px">

<@command menu="SSGL\_FQSS" name="FQSS\_ADDBTN"/>

</div>

## 输入框字段长度

页面限制输出长度一般为数据库设置的长度的一半

<input name="ssnr" class="easyui-textbox"

data-options="multiline:true,required:true,validType:['maxLength[50]']" style="width:300px;height:100px;">

## 金额类型留2位

## 百分百

public String getGcb(Double df,Double total){

if(total == 0D){

return "0.0%";

}else{

return String.format("%.2f", (df\*100/total)) +"%";

}

}

## Form序列化

var formData = queryForm.serializeObject();

grid.datagrid('reload',formData);

var formdata = form.serializeObject({transcript:"overlay"});

## 导出传参时序列化

|  |
| --- |
| function exportBb(){  var formData = qf1.serialize();  var zbnd = $("#" + modelName + "\_zbnd1").val();  var khx = $("#" + modelName + "\_khx1").combobox('getValue');  if(zbnd == "" || khx == ""){  $.messager.alert('系统提示','请输入相关导出信息!');  }else{  window.open(ctx + "/jxdftj/khbm/export?" + formData);  }  } |

|  |
| --- |
| @RequestMapping("export")  public void export(HttpServletResponse response){  Map<String,Object> data = new HashMap<String, Object>();  String title ="收文登记表";  data.put("list", dagl\_gddj\_service.getSwList());  data.put("timeCondition", new SimpleDateFormat("yyyy-MM-dd").format(new Date()));  data.put("userCondition", userService.getCurrentUser().getRealname());  ModelExport.exportWithModel("swcx/swdjb.xls", title, data, response);  } |

## Form清空与加载

queryForm.form('clear');

$("#" + modelName + "\_form").form("load", data);

## Input数据获取

var value = $(this).textbox("getValue");

## 隐藏Input

$("input").attr("disabled",true);

## 获取.class多个数据

|  |
| --- |
| $(".rgpf").each(function(){  //考核对象 + 系统总评 + 人工总评  var khdx = $(this).data("khdx");  var xtzp = $(this).data("xtzp");  var value = $(this).textbox("getValue");  rgpf += ndkhid+"="+zbnd+"="+khx+"="+khdx+"="+xtzp+"="+value+"&";  }); |

# 申诉与回复

## 申诉对象Ssxx

申诉人员id + 申诉相关信息 + 申诉状态1/2/3 + 接受对象（考核部门）id

## 2. SsxxDto

申诉人id + 申诉相关信息 + 申诉状态 + 申诉时间 + 考核部门id + 考核部门可查询列表

### 2.1 被审核对象的列表Map

时间查询：

|  |
| --- |
| String sj=StringSimple.nullToEmpty(map.get("sssj"));  try{  if(!"".equals(sj)){  map.put("sssj\_s", sdf.parse(sj+" 00:00:00"));  map.put("sssj\_e", sdf.parse(sj+" 24:59:59"));  }  }catch(Exception e){  logger.error("context",e);  } |

申诉人id:只有是当前用户自己发起的申诉，才能查看到对应的申诉列表

|  |
| --- |
| //添加申诉人id -- 只能查看自己发送的申诉信息  map.put("ssrid", userService.getCurrentUser().getId()); |

申诉状态：直接查询出3种申诉状态，1已发送2可回复3可修改

页面所需信息： 申诉对象id + 申诉时间 + 申诉相关信息 +申诉内容 + 申诉状态 + 接受对象id + 接受对象姓名

### 2.2 考核部门的查看申诉信息列表

时间查询：

|  |
| --- |
| String sj=StringSimple.nullToEmpty(map.get("sssj"));  try{  if(!"".equals(sj)){//设置全天时间段  map.put("sssj\_s", sdf.parse(sj+" 00:00:00"));  map.put("sssj\_e", sdf.parse(sj+" 24:59:59"));  }  }catch(Exception e){  logger.error("context",e);  } |

只能查看发送给自己的申诉信息：

|  |
| --- |
| //添加考核对象(发送)id -- 只能查看自己接受到的申诉信息  map.put("fsid", userService.getCurrentUser().getId()); |

只能查看申诉人员已发送（提交）的申诉信息：1已发送2可回复

|  |
| --- |
| //考核部门只能查询  List<String> list=new ArrayList<String>();  list.add("1");  list.add("2");  map.put("sszts", list);  @Des("考核部门-只能查询已发送/可回复状态申诉信息")  @CondtionExpression(prameType=String.class,value="sszt.code",type=CondtionType.in)  private List<String> sszts; |

页面设计为全部可回复状态：

页面所需信息：申诉信息id + 申诉时间 + 申诉相关信息 + 申诉内容 + 申诉状态 + 申诉人员id + 申诉人员姓名

## 3. 回复

### 3.1 回复对象实体SsxxComment

所属申诉对象 + 申诉id + 回复人员对象（+人员id） + 所属回复（指向同类） + 回复内容 + 回复时间

|  |
| --- |
| @Getter  @Setter  @Des("所属回复")  @ManyToOne(targetEntity = SsxxComment.class)  @JoinColumn(name = "parentCotentId", nullable=true)  private SsxxComment parentContent; |

### 3.2 SsxxCommentDto

查询回复信息列表，只根据申诉id判断。获取列表时，时间升序asc,即最新的在后面。

|  |
| --- |
| //所属申诉信息id  @CondtionExpression(value="sid",type=CondtionType.equal)  private Long sid; |

查询到的回复信息格式为：List<Map<String,Object>>

所需信息有：申诉信息id + 回复时间 + 回复内容 + 父回复信息id + 父回复信息的回复人员姓名

### 3.3 进行回复

所需参数有:申诉信息id +回复相关信息（时间，内容）+ 所属父回复 + 回复人员

回复人员即为当前操作用户，可直接调用。回复成功时修改申诉状态为2可回复。

传入参数有：申诉信息id + 回复内容 + 父回复id（为空，则为第一次回复）

|  |
| --- |
| public Result addNewComment(String sid,String content,String pid){  //评论用户+所属申诉对象  User u=userService.getCurrentUser();  Ssxx s = ssxxService.getDao().findOne(Long.parseLong(sid));    SsxxComment sc = new SsxxComment();  sc.setUser(u);  sc.setSsxx(s);  sc.setSid(s.getId());  sc.setUid(u.getId());  sc.setHfsj(new Date());  //判断是否有父回复信息,没有便是第一次回复  if( !"".equals(StringSimple.nullToEmpty(pid))){  SsxxComment psc = this.getDao().findOne(Long.valueOf(pid));  sc.setParentContent(psc);  }  //回复内容  sc.setContent(content);  //存在回复-更改该条申诉对象的状体1-已发送 2-可回复 3可发送/删除  Sszt sszt = new Sszt();  sszt.setCode("2");  s.setSszt(sszt);  ssxxService.save(s);  this.save(sc);  return new Result(true,"回复成功!");  } |

## 4. 申诉人员

### 4.1 新增申诉信息

申诉人id（即当前申诉人员的用户）+ 申诉相关信息（时间、内容、年度、考核项） + 申诉状态默认为3可修改 + 考核部门id

|  |
| --- |
| @ResponseBody  @RequestMapping(value="saveSsxx")  public Result saveSsxx(Ssxx s){  //申诉人id-即当前进行申诉的人员id  User u = userService.getCurrentUser();  if(u == null){  return new Result(false,"发送失败");  }  Ssxx save=new Ssxx();  save.setSsrid(u.getId());  save.setSsr(u.getRealname());  save.setZbnd(s.getZbnd());  save.setSssj(new Date());  save.setKhx(s.getKhx());  //发送对象id-发送对象，即考核部门人员id  Long fsid=Long.parseLong(s.getFsdx());  User u1=userService.getDao().findOne(fsid);  save.setFsid(u1.getId());  save.setFsdx(u1.getRealname());  //新添加的申诉信息对象，默认为可发送/可删除状态  Sszt sszt=new Sszt();//1-已发送 2可回复 3可发送/可删除  sszt.setCode("3");  save.setSszt(sszt);  //设置发送内容  save.setSsnr(s.getSsnr());  ssxxService.save(save);  return new Result(true,"新增成功");  } |

### 4.2 删除申诉信息

只能删除申诉状态为3可修改的申诉信息。一般进行逻辑删除。

传入该条申诉信息id + 获取对应数据库实体信息 + 逻辑删除。

|  |
| --- |
| @ResponseBody  @RequestMapping(value="deleteSs")  public Result deleteSs(String id){  try{  Ssxx s=ssxxService.getDao().findOne(Long.parseLong(id));  //逻辑删除  ssxxService.logicDeleteModel(s);  return new Result(true,"删除成功");  }  catch(Exception e){  logger.error("context",e);  return new Result(false,"删除失败");  }  } |

### 4.3 提交申诉信息

只能提交申诉状态为3可修改的申诉信息。只需修改状态为1可发送即可。

传入该条申诉信息id + 修改申诉状态为1 + 保存更新

|  |
| --- |
| @ResponseBody  @RequestMapping(value="tjSs")  public Result tjSs(String id){  Long ssid=Long.parseLong(id);  Sszt sszt=new Sszt();//1已发送 2可回复 3可提交/删除  sszt.setCode("1");  Ssxx s=ssxxService.getDao().findOne(ssid);  s.setSszt(sszt);  ssxxService.save(s);  return new Result(true,"提交成功");  } |

### 4.4 查看与回复

申诉人员只能查看2可回复状态的申诉回复信息页面。需要先获取该条申诉信息的回复列表。Ajax获取后,已JQuery append()方式进行展示。

|  |
| --- |
| public List<Map<String, Object>> getListGrid(String sid){  Map<String,Object> map = new HashMap<String,Object>();  //只有能查看该申诉信息的考核部门人员才能获取对应的回复信息  map.put("sid", Long.parseLong(sid));  Page<SsxxComment> page = this.queryForPage(SsxxCommentDto.class, map);  List<SsxxComment> scs = page.getContent();  List<Map<String, Object>> list = new ArrayList<Map<String,Object>>();  for(SsxxComment sc: scs){  Map<String, Object> mapscs = new HashMap<String, Object>();  //本次回复  mapscs.put("id", sc.getId());  mapscs.put("name", sc.getUser().getRealname());  mapscs.put("hfsj", sc.getHfsj());  mapscs.put("content", sc.getContent());  //被回复  mapscs.put("pid", sc.getParentContent() == null ? "" : sc.getParentContent().getId());  mapscs.put("pname",sc.getParentContent()==null?"" : sc.getParentContent().getUser().getRealname());  list.add(mapscs);  }  return list;  } |

当考核部门进行回复时，将该条申诉信息状态修改为2可回复，此时申诉人员可以查看并进行回复。

实例代码如3.3。

## 5. 考核部门人员

进行申诉查看功能，包括：查看提交的申诉信息、回复申诉信息。

# 6.Controller

|  |
| --- |
| org.springframework.web.servlet.mvc.AbstractController |

extends

|  |
| --- |
| AbstractController  @Override  protected ModelAndView  handleRequestInternal(HttpServletRequest request, HttpServletResponse response)  throws Exception {  return super.handleRequest(request, response);  } |

extends

|  |
| --- |
| SimplewebBaseController  @InitBinder  protected void initBinder(WebDataBinder binder) {}  private class DateEditor extends PropertyEditorSupport {} |

# 7.动态table

行：每一考核对象+各二级指标部门的评分

列：固定首列 + 各二级指标部门的评分 +系统评分 +人工评分(人工输入)

|  |
| --- |
| 输出: Map<String,Object> result = new HashMap<String, Object>(); |

|  |
| --- |
| 动态列:  List<Zbpz\_ejzbfbb> ejzbfbb = ndkh.getEjzbfbb();  List<Map<String,Object>> ejzb = new ArrayList<Map<String,Object>>();    for(Zbpz\_ejzbfbb ze : ejzbfbb){  if(!(ze == null)){  Map<String,Object> head = new HashMap<String,Object>();  head.put("ejzb", ze.getEjzb());  head.put("id", ze.getId());  ejzb.add(head);  }  }  Collections.sort(ejzb, new Comparator<Map<String,Object>>() {  @Override  public int compare(Map<String, Object> o1, Map<String, Object> o2) {  Long id1 = Long.parseLong(StringSimple.nullToEmpty(o1.get("id")));  Long id2 = Long.parseLong(StringSimple.nullToEmpty(o2.get("id")));  return id1 > id2 ? 1 : (id1 == id2 ? 0 : -1);  }  });    result.put("ejzb", ejzb); |

|  |
| --- |
| 动态行:  List<Zbpz\_zbdx> zbdxList = ndkh.getZbdx();  //存储数据列表  List<Map<String,Object>> lm = new ArrayList<Map<String,Object>>();  if(!zbdxList.isEmpty()){  for(int i=0;i<zbdxList.size();i++){  Map<String,Object> data = new HashMap<String,Object>();  Zbpz\_zbdx zz = zbdxList.get(i);  //指标对象  data.put("zbdx", zz.getZbdx());  data.put("ndkhid", zz.getNdkhid());  //二级指标部门列表 ->评分列表  Double sum =0D;  List<Map<String,Object>> ejpf = new ArrayList<Map<String,Object>>();  for(int j = 0; j<ejzbfbb.size() ;j++){  Map<String,Object> pf = new HashMap<String,Object>();  Zbpz\_ejzbfbb ze = ejzbfbb.get(j);  //获取绩效分评  Jxfp jxfp = jxfpDao.queryJxfp(zz.getZbdx(), String.valueOf(ndkh.getId()), ze.getEjzb());  Double d = jxfp.getRgpf();  sum += d;  //存储评分  pf.put("pf", d);  pf.put("id", ze.getId());  ejpf.add(pf);  }  Collections.sort(ejzb, new Comparator<Map<String,Object>>() {  @Override  public int compare(Map<String, Object> o1, Map<String, Object> o2) {  Long id1 = Long.parseLong(StringSimple.nullToEmpty(o1.get("id")));  Long id2 = Long.parseLong(StringSimple.nullToEmpty(o2.get("id")));  return id1 > id2 ? 1 : (id1 == id2 ? 0 : -1);  }  });  //系统总评(平均值)  data.put("xtzp", sum/ejzbfbb.size());  //二级分评  data.put("ejpf", ejpf);  lm.add(data);  }  }  result.put("dataList", lm); |

|  |
| --- |
| 页面：  <div data-options="region:'center',border:false" style="padding:3px;">  <form id="jxzp\_form">  <input id="jxzp\_ndkhid" name="jxzp\_ndkhid" type="hidden" >  <table width="100%" class="formtable">  <tbody>  <tr style="height: 10px;">  <td align="center">指标对象</td>  <#list result.ejzb as h>  <td align="center" >${h.ejzb!}</td>  </#list>  <td align="center">系统评分</td>  <td align="center">人工评分</td>  </tr>  <#list result.dataList as data>  <tr style="height: 10px;" >  <td style="text-align: center;">${data.zbdx!}</td>  <!-- 中间评分列 -->  <#list data.ejpf as pf>  <td style="text-align: center;">${pf.pf!}</td>  </#list>  <td style="text-align: center;">${data.xtzp!}</td>  <!-- 人工评分 -->  <td style="text-align: center;">  <input name="rgpf" data-zbnd="" data-khx=""  data-khdx="${data.zbdx!}" data-xtzp="${data.xtzp!}"  class="rgpf easyui-numberbox" required="required"/></td>  </tr>  </#list>  </tbody>  </table>  </form>  </div> |

# 8.流程查看

1.当前用户

1.1 考核对象

1.2 年度考核

2.年度考核-指标级

2.1. 指标级为1

2.1.1. 指标发布状态2

2.1.1.1 数据全部上报

2.1.2.

2.2. 指标级为2

# 9.报表统计

## 1.动态列

## 2.复杂行

# 10.Echarts

|  |
| --- |
| JSONObject jo = new JSONObject();  JSONArray name = new JSONArray();  JSONArray value = new JSONArray();    JSONObject j1 = new JSONObject();  j1.put("name", "武进区");  j1.put("value", 20);    JSONObject j2 = new JSONObject();  j2.put("name", "溧阳市");  j2.put("value", 30);    JSONObject j3 = new JSONObject();  j3.put("name", "天宁区");  j3.put("value", 40);    JSONObject j4 = new JSONObject();  j4.put("name", "新北区");  j4.put("value", 50);    JSONObject j5 = new JSONObject();  j5.put("name", "金坛区");  j5.put("value", 60);    name.add(j1.get("name"));  name.add(j2.get("name"));  name.add(j3.get("name"));  name.add(j4.get("name"));  name.add(j5.get("name"));    value.add(j1);  value.add(j2);  value.add(j3);  value.add(j4);  value.add(j5);    jo.put("name", name);  jo.put("value", value);  jo.put("title", "安置处");    Map<String,Object> m = new HashMap<String,Object>();  m.put("jo1", jo); |

## 饼状图

|  |
| --- |
| /\*\* 饼状图 \*\*/  function seriBingTu(data,target){  var chart = echarts.init(document.getElementById(target));  chart.setTheme("macarons");  option = {  title : {  text : data.title,  x : 'center'  },  tooltip : {  trigger: 'item',  formatter: "{a} <br/>{b} : {c} ({d}%)"  },  legend : {  orient: 'vertical',  x: 'left',  data: data.name  },  series :[  {  name: '各项得分',  type: 'pie',  radius : '55%',  center: ['50%', '60%'],  data:data.value,  itemStyle: {  emphasis: {  shadowBlur: 10,  shadowOffsetX: 0,  shadowColor: 'rgba(0, 0, 0, 0.5)'  }  }  }  ]  };  chart.setOption(option);  } |

## 柱状图

|  |
| --- |
| /\*\*柱状图 \*\*/  function seriZhuTu(data,target){  var chart = echarts.init(document.getElementById(target));  chart.setTheme("macarons");  option = {  color: ['#561712'],  title : {  text: data.title,  x:'center'  },  tooltip : {  trigger: 'axis',  axisPointer : { // 坐标轴指示器，坐标轴触发有效  type : 'shadow' // 默认为直线，可选为：'line' | 'shadow'  }  },  grid :{  left :'3%',  right : '4%',  bottom : '3%',  containLabel: true  },  xAxis : [  {  type : 'category',  data : data.name,  axisTick: {  alignWithLabel: true  }  }  ],  yAxis : [  {  type : 'value'  }  ],  series :[  {  name: '评分',  type : 'bar',  barWidth : '60%',  data : data.value  }  ]  };  chart.setOption(option);  } |

# 11.用户管理StringKit

## 1.var row = gridSelectedValid(grid);

## 2. var ids=gridCheckedValid(grid);

# 12.接口获取数据

|  |
| --- |
| package simple.project.communal.service;  import java.util.ResourceBundle;  import org.apache.axis.client.Call;  import org.apache.axis.client.Service;  import com.alibaba.fastjson.JSONArray;  /\*\*  \* @author sxm  \* @created 2017年8月16日  \*/  @org.springframework.stereotype.Service  public class ClintService {  /\*\*  \* 接口  \*/  public JSONArray getReplyInfo(String t\_pt,String t\_method){  ResourceBundle xtpt = ResourceBundle.getBundle("xtpt");  String pt=xtpt.getString(t\_pt);  String method=xtpt.getString(t\_method);    String endpoint = "http://"+pt;  Service service = new Service();  String result="";  Call call;  try {  call = (Call) service.createCall();  call.setTargetEndpointAddress(endpoint);  call.setOperationName(method);    call.addParameter("tjny", org.apache.axis.encoding.XMLType.XSD\_STRING, javax.xml.rpc.ParameterMode.IN);    call.setReturnType(org.apache.axis.encoding.XMLType.XSD\_STRING);  call.setUseSOAPAction(true);  result = (String) call.invoke(new Object[]{"2017-07"});  } catch (Exception e) {  e.printStackTrace();  }  return JSONArray.parseArray(result);  }    } |

|  |
| --- |
| package simple.project.communal.service;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import simple.base.utils.StringSimple;  import simple.project.communal.entities.Zbpz\_zbnr;  import com.alibaba.fastjson.JSONArray;  import com.alibaba.fastjson.JSONObject;  /\*\*  \* @author sxm  \* @created 2017年8月16日  \*/  @Service  public class XtsjService {    @Autowired  private ClintService clintService;      /\*\*  \* 获取系统数据  \* @author sxm  \* @created 2017年8月16日  \* @param fbnr  \* @return  \*/  public String getXtsj(Zbpz\_zbnr zbnr,String ssdq){  String xtsj="";  //获取系统平台code  String xtpt=StringSimple.nullToEmpty(zbnr.getXtpt().getCode());  String nr=StringSimple.nullToEmpty(zbnr.getZbnr().getCode());    if("999".equals(nr)){  xtsj="0";  }else{  //获取指标内容code  String zbnr\_code=nr.substring(1, nr.length());  //接入平台url  String t\_pt="";  //调用方法  String t\_method="";  if("1".equals(xtpt)){  t\_pt="shjzpt";  t\_method="shjzpt\_method";  }else if("2".equals(xtpt)){  t\_pt="zljpt";  t\_method="zljpt\_method";  }else if("3".equals(xtpt)){  t\_pt="ylfwpt";  t\_method="ylfwpt\_method";  }    JSONArray t = clintService.getReplyInfo(t\_pt,t\_method);  for (Object object : t) {  JSONObject obj = JSONObject.parseObject(object.toString());  if(obj.get("ssdq").equals(ssdq)){  xtsj=obj.getString(zbnr\_code);  }  else if(obj.get("ssdq").equals("3204")){  xtsj="0";  }  }  }    return xtsj;  }  } |

|  |
| --- |
| /\*\*  \* 获取系统  \* @author sxm  \* @created 2017年8月16日  \* @param fbnr  \* @return  \*/  public String getXtsj(Sjsb\_fbnr fbnr){  String xtsj="";  //获取系统平台code  String xtpt=StringSimple.nullToEmpty(fbnr.getXtpt().getCode());  String zbnr=StringSimple.nullToEmpty(fbnr.getZbnr().getCode());  if("999".equals(zbnr)){  xtsj="0";  }else{  //获取指标内容code  String zbnr\_code=zbnr.substring(1, zbnr.length());  //获取考核用户ssdq  String ssdq=userService.getCurrentUser().getOrg().getOrgCode();  //接入平台url  String t\_pt="";  //调用方法  String t\_method="";  if("1".equals(xtpt)){  t\_pt="shjzpt";  t\_method="shjzpt\_method";  }else if("2".equals(xtpt)){  t\_pt="zljpt";  t\_method="zljpt\_method";  }else if("3".equals(xtpt)){  t\_pt="ylfwpt";  t\_method="ylfwpt\_method";  }    JSONArray t = clintService.getReplyInfo(t\_pt,t\_method);  for (Object object : t) {  JSONObject obj = JSONObject.parseObject(object.toString());  if(obj.get("ssdq").equals(ssdq)){  xtsj=obj.getString(zbnr\_code);  }  else if(obj.get("ssdq").equals("3204")){  xtsj="0";  }  }  }    return xtsj;  } |