

1.1.1 Json 框架

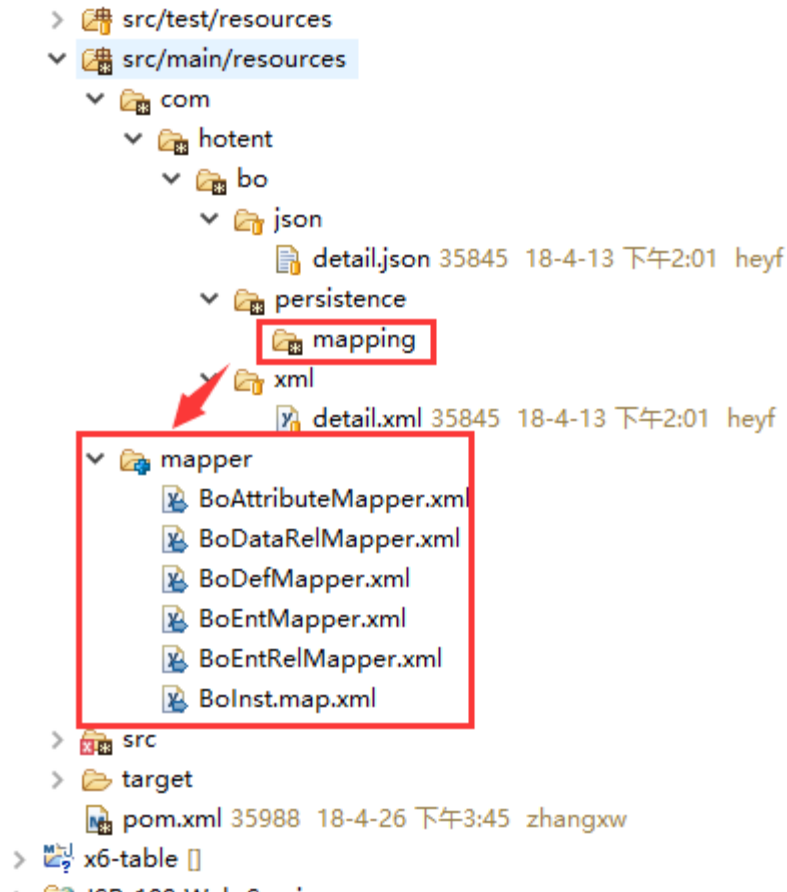
json 框架仅使用 jackson，JsonUtil 为工具类，常用注解有 @JsonIgnore
@JsonProperty("name")

1.1.2 Dao 改造

1、所有的 dao 继承 MyBatisDao

```
1 package com.hotent.bo.persistence.dao;
2
3 import com.hotent.base.dao.MyBatisDao;
4
5
6 /**
7  * bo实例
8  *
9  * @company 广州宏天软件股份有限公司
10 * @author heyifan
11 * @email heyf@jee-soft.cn
12 * @date 2018年4月13日
13 */
14 public interface BoInstDao extends MyBatisDao<String, BoInst> {
15 }
```

2、map 文件迁移到 mapper 目录,命名方式由 xxx.map.xml 修改为 xxxMapper.xml



- 3、删除所有 dao 的实现类，只需要保留 dao 接口类（MyBatis 框架会自动代理生成实现类），特别注意：在删除实现类之前确认实现类中调用的 sql 语句 key 要与方法名一致：

```
24 public class BOEntRelDaoImpl extends MyBatisDaoImpl<String, BOEntRel> implements BOEntRelDao{
25
26     @Override
27     public String getNamespace() {
28         return BOEntRel.class.getName();
29     }
30
31
32     @Override
33     public List<BOEntRel> getByDefId(String defId) {
34         List<BOEntRel> list=this.getByKey("getByDefId", defId);
35         return list;
36     }
37
38
39     @Override
40     public void removeByDefId(String defId) {
41         this.deleteByKey("removeByDefId", defId);
42     }
43
44
45     @Override
46     public List<BOEntRel> getByEntId(String entId) {
47         List<BOEntRel> list=this.getByKey("getByEntId", entId);
48         return list;
49     }
50 }
51
52
```

- 4、另外在 dao 定义的方法中，如果入参超过一个，需要在每个参数前面加上注解。

1.1.3 控制层 Controller

- 1、入参、出参尽量具体一些，比如：指定泛型、不要用 Map、String、JsonNode 作为参数。

```
78 /**
79  * 流程任务表单列表(分页条件查询)数据
80  * @param request
81  * @param reponse
82  * @return
83  * @throws Exception
84  * @PageJson
85  * @exception
86  * @since 1.0.0
87  */
88 @RequestMapping(value="list", method=RequestMethod.POST, produces={"application/json; charset=utf-8"})
89 @ApiOperation(value = "流程任务表单列表(分页条件查询)数据", httpMethod = "POST", notes = "流程任务表单列表(分页条件查询)数据")
90 public PageList<BpmFormDef> list(@ApiParam(name="queryFilter", value="通用查询对象")@RequestBody QueryFilter queryFilter) throws Exception{
91     PageList<BpmFormDef> list = (PageList<BpmFormDef>) bpmFormDefManager.query(queryFilter);
92     List<JsonNode> bpmFormDefList = new ArrayList<JsonNode>();
93     for(BpmFormDef bpmFormDef : list.getRows()){
94         Map formJson = new HashMap(); //JSONObject.fromObject(bpmForm);
95         formJson.put("desc", bpmFormDef.getName());
96         formJson.put("key", bpmFormDef.getKey());
97         formJson.put("id", bpmFormDef.getId());
98         formJson.put("name", bpmFormDef.getName());
99         formJson.put("type", bpmFormDef.getType());
100         //计算bo对象
101         String expand = bpmFormDef.getExpand();
102         if(StringUtil.isNotEmpty(expand)){
103             JsonNode expandJson = JsonUtil.toJsonNode(expand);
104             if(expandJson.get("boDefList")!=null){
105                 formJson.put("boDefList", expandJson.get("boDefList").toString());
106             }
107         }
108         bpmFormDefList.add(JsonUtil.toJsonNode(formJson));
109     }
110     PageList pageJson = new PageList(bpmFormDefList);
111     pageJson.setPage(list.getPage());
112     pageJson.setPageSize(list.getPageSize());
113     pageJson.setRows(list.getRows());
114     pageJson.setTotal(list.getTotal());
115     return pageJson;
116 }
```

- 2、入参过多，而且多为简单类型时将其封装为一个查询对象，以**Params 命名。

```

@ApiOperation(value = "加载编辑器设计模式的模板列表", httpMethod = "POST", notes = "加载编辑器设计模式的模板列表")
public CommonResult chooseDesignTemplate(@ApiParam(name="subject", value="标题")@RequestBody String subject,
    @ApiParam(name="categoryId", value="")@RequestBody String categoryId,
    @ApiParam(name="formDesc", value="表单描述")@RequestBody String formDesc,
    @ApiParam(name="isSimple", value="true将只允许选择一行")@RequestBody Boolean isSimple) throws Exception{
    Map mv = new HashMap();

    String templatePath = FormUtil.getDesignTemplatePath();
    String xml = FileUtil.readByClassPath(templatePath+"designtemps.xml");
    Document document = Dom4jUtil.loadXml(xml);
    Element root = document.getRootElement();
    @SuppressWarnings("unchecked")
    List<Element> list = root.elements();
    String reStr = "[";
    for (Element element : list){
        String alias = element.attributeValue("alias");
        String name = element.attributeValue("name");
        String templateDesc = element.attributeValue("templateDesc");
        if (!reStr.equals("")){
            reStr += ",";
        }
        reStr += "{name:'"+name+"', alias:'"+alias+"', templateDesc:'"+templateDesc+"'}";
    }
    reStr += "]";

    mv.put("subject", subject);
    mv.put("categoryId", categoryId);
    mv.put("formDesc", formDesc);
    mv.put("temps", reStr);
    mv.put("isSimple", isSimple);
    return new CommonResult(true, null, mv);
}
```

- 3、一个 Controller 方法中不要写太多的逻辑代码，尤其是不能多次操作数据库。

```
29 @RestController
30 @RequestMapping("/form/rights/v1")
31 @Api(tags="formRightsController")
32 @SuppressWarnings({ "unchecked", "rawtypes" })
33 public class FormRightsController extends BaseController {
34     @Resource
35     private BpmFormRightManager bpmFormRightManager;
36
37     @Resource
38     private BpmFormDefManager bpmFormDefManager;
39
40
41     @RequestMapping(value="getPermission", method=RequestMethod.POST, produces={"application/json; charset=utf-8"})
42     @ApiOperation(value="获取表单权限", httpMethod="POST", notes="获取表单权限")
43     public CommonResult getPermission(@ApiParam(name="flowKey", value="流程key")@RequestBody String flowKey,
44         @ApiParam(name="formKey", value="表单KEY")@RequestBody String formKey,
45         @ApiParam(name="nodeId", value="节点ID")@RequestBody String nodeId,
46         @ApiParam(name="type", value="权限类型")@RequestBody int type,
47         @ApiParam(name="parentflowKey", value="父流程定义")@RequestBody String parentflowKey) {
48         if (type == 1) {
49             bpmFormRightManager.remove(formKey, flowKey, nodeId, parentflowKey);
50         }
51
52         JSONObject json = bpmFormRightManager.getPermissionSetting(formKey, flowKey, parentflowKey, nodeId, type);
53         List<Map<String, String>> tableSn = bpmFormRightManager.getTableOrderBySn(formKey);
54         Map map = new HashMap();
55         map.put("json", json);
56         /*map.put("permissionList", PermissionUtil.getPermissionList("formPermissionCalcList"));*/
57         map.put("tableSn", tableSn);
58         return new CommonResult(true, "", map);
59     }
60
61 }
```