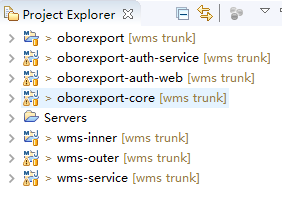
用Eclipse以Maven导入工程后

|  |  |  |
| --- | --- | --- |
| **工程名** | **WMS中功能** | **规划功能** |
| oborexport | 最外层 |  |
| oborexport-core | 核心SDK |  |
| oborexport-auth-service |  | 预留SSO |
| oborexport-auth-web |  | 预留SSO |
| wms-service | WMS业务逻辑 |  |
| wms-outer | WMSWEB服务 | 规划对外WEB服务，WMS中实际对外接口和站点 |
| wms-inner |  | 规划运营端WEB服务，WMS中没用到 |

1. **数据库读写分离:** 在service方法上加@ReadOnly注解，该service方法获取的数据源是从库数据库，查从库，实现类com.quincy.core.aop.ReadOnlyAop；
2. **缓存:** 在service方法上加@Cache注解，会按方法名、参数值为key值对返回值进行缓存，expire参数代表缓存时间(单位: 秒)，如果不加永不失效，实现类com.quincy.core.aop.CacheAop；
3. **分布式锁:** 在service方法上加@Synchronized注解，实现类com.quincy.core.aop. SynchronizedAop；
4. **获取数据库连接对象，绕过Mybatis直接使用jdbc:** 在service方法上加@ConnectionInjector注解，在service方法定义时加java.sql.Connection类型参数，在调用时传空即可，框架会自动从连接池中取出一个连接对象给它赋值，使用完毕不需要关闭，框架会自动关闭，实现类com.quincy.core.aop.ConnectionInjectorAop；
5. **获取Jedis对象，操作redis:** 在service方法上加@JedisInjector注解，在service方法定义时加redis.clients.jedis.Jedis类型参数，在调用时传空即可，框架会自动从连接池中取出一个连接对象给它赋值，使用完毕不需要关闭，框架会自动关闭，实现类com.quincy.core.aop.JedisInjectorAop；
6. **如果请求需要验证验证码:** 在Controller方法上加@VCodeRequired注解，如果验证码输入有误，如果是json返回{"status":-4, "msg":"验证码输入有误。"}，如果是普通页面重定向至提示页面，实现类com.quincy.core.interceptor.VCodeInterceptor；
7. **OPEN API签名校验:** 在接收数据的Controller方法上加@OpenApiSignatureRequired注解，如果签名有误，如果是json返回{"status":”0x001”, "msg":"签名验证失败。"}，如果是XML返回<Result><Code>0x001</Code><Message>签名验证失败</Message></Result>，实现类com.oborexport.wms.interceptor.OpenApiAuthValidationInterceptor。为确保HTTP接口接收数据的完整性和不可抵赖性，需要验证数字签名，先为每一对接客户端生成一对公、私钥存在数据库该客户对应的记录中，将私钥以邮件形式发送至客户接口人邮箱，在调用我方接口时需要先对发送的数据做MD5摘要，再用私钥用RSA算法对摘要签名，在调用接口时将该签名一并回传。如果传输的数据和钱相关，必须要让客户自己生成密钥对，将公钥发送给我方，客户自己保存私钥，以排除我方伪造数据和签名的嫌疑；
8. **本地Session和缓存Session切换: 修改两处配置，**(1) com.oborexport.wms.WmsConstant中变量IMPL\_AUTHORIZATION的值; (2)wms-outer工程中com-wmsweb.properties文件中impl.authorization的值，authorizationSessionServiceImpl为本地Session，authorizationCacheServiceImpl为缓存Session。如果部署单个WEB就可以满足需求，全部设置为authorizationSessionServiceImpl，如果需要部署多个WEB达到负载均衡全部设置为authorizationCacheServiceImpl；
9. **动静分离:** js、css、图片等静态文件集中存放，在打线上包时会将静态文件排除在外，以达到部署时将其与动态请求分离的目的，使其不占用有限的WEB服务资源，如果是超大型项目还可方便地将其部署在CDN上，在WMS中存放在wms-outer/src/main/webapp/resource目录中。在代码中引用静态文件要求必须加前缀，ftl模板中<@property key="prefix.resource"/>，js代码中resourcePrefix。在开发模式下前缀为/resource，以传统方式访问静态文件，在线上模式下前缀为静态文件域名<http://resource.wms.oborexpress.com>，直接指向静态文件目录，以静态方式访问；
10. **ftl自定义标签:** 
    1. <@property key="properties配置文件中的key值"/>: 引用properties文件中配置的值;
    2. <@i18n key="国际化资源文件中的key值" />：显示国际化文本。如果文本中有占位符，将文本按顺序以半角逗号分隔，文本有3种来源，如果以property：开头，从properties文件中取对应的值，如果以i18n:开头，从国际化资源文件中取对应的文本，什么都不加直接原封不动输出，例如:

<@i18n key="国际化资源文件中的key值">property:配置文件中的key值,i18n:国际化文件中的key值,原封不动的文本</@i18n>

* 1. <@attr key="给定的key值" />: 取给定的key值对应的值，locale取国际化locale，uri\_without\_first取除了url中locale段以外的uri，url取请求的url

1. **Ajax上传文件:** 引入obor.js，例子如下:

$("#file1").ajaxUploadFiles({

url: "/"+locale+"/management/areapricerule/import/"+type+"/"+$("input[name='areaPriceRuleExpress']:checked").val()+"/"+$("input[name='warehouse']:checked").val(),

maxSize: 1024\*1024\*2,

acceptableTypes: ["xlsx"],

validationFailed: **function**(msg) {

layer.close(popLayer);

layer.msg(msg);

},

complete: **function**(status, responseText) {

layer.close(popLayer);

},

success: **function**(data) {

alert(data.msg);

**if**(data.status==0) {

$(top.location).attr("href", siginUrl);

}

},

failure: **function**(status, responseText) {

alert("failure---"+status+": "+responseText);

}

});