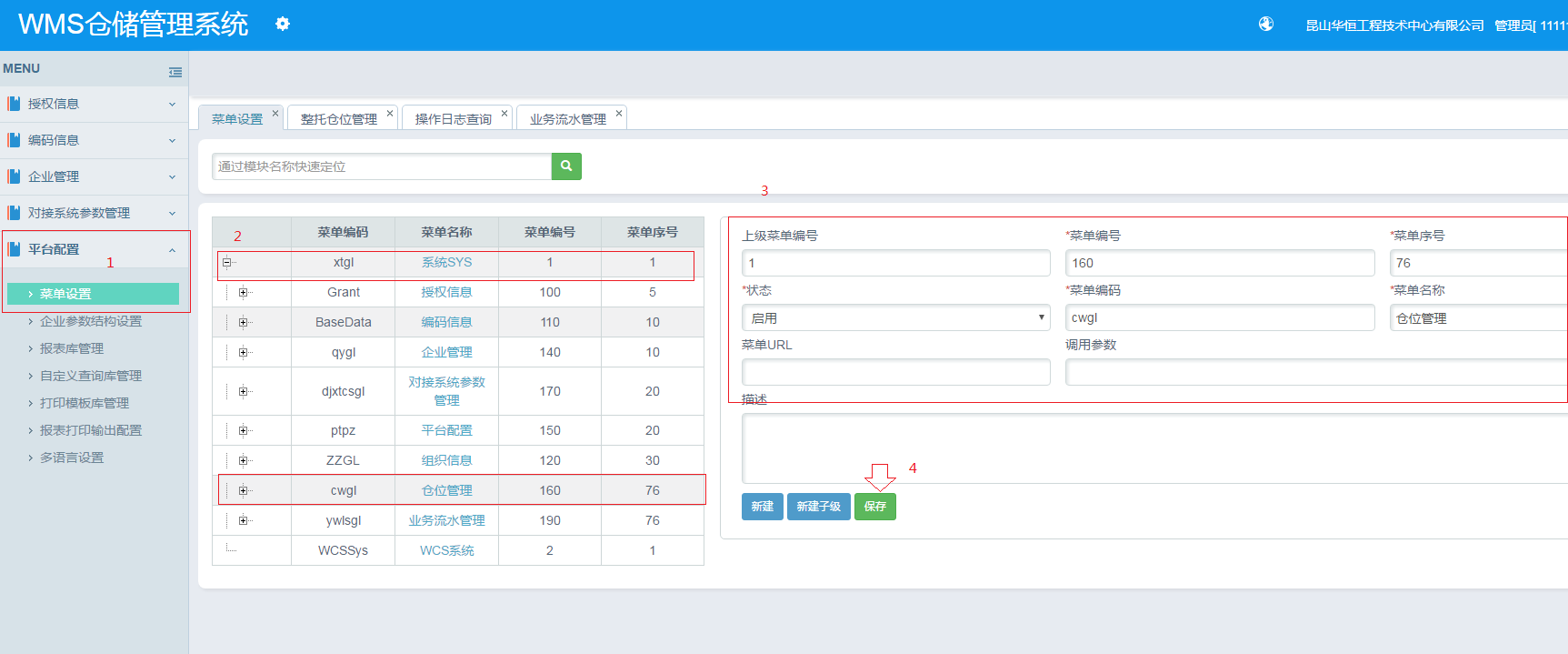
用户管理界面开发手册

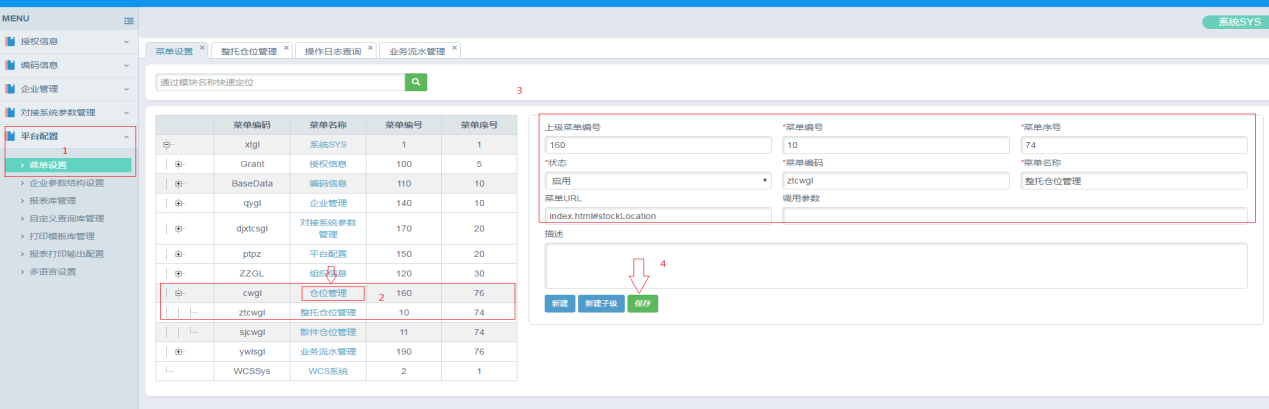
采用HTML+AngularJS实现在界面上的数据展示

以添加整托仓位管理功能为例

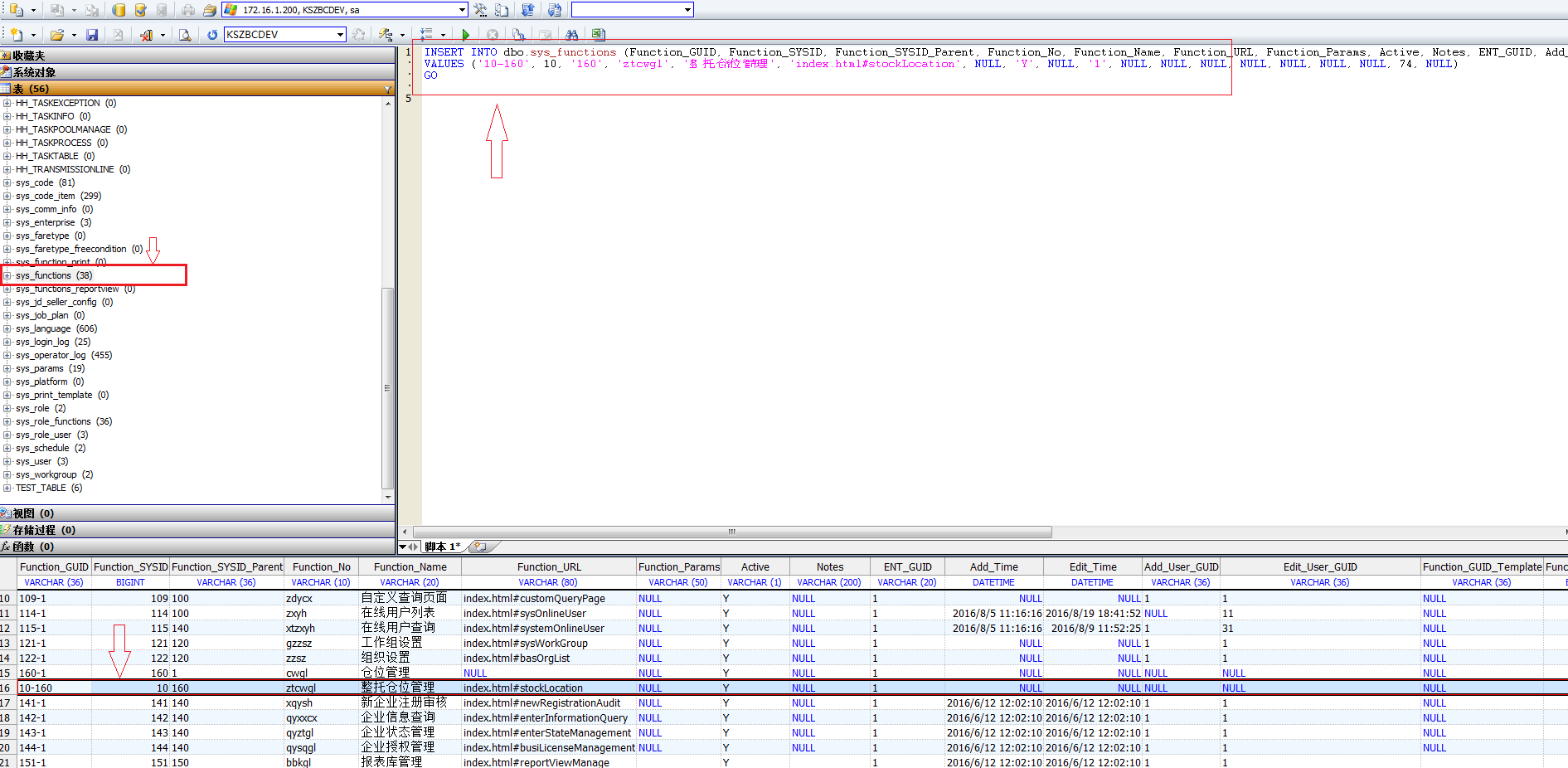
1.在平台配置->菜单设置->新建子级中添加菜单“仓位管理”

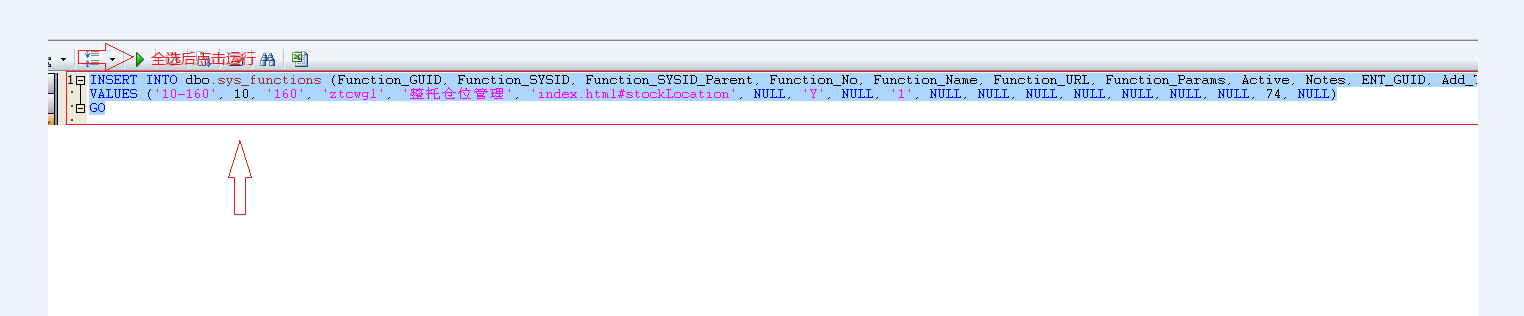


2在平台配置->菜单设置->新建子级中添加菜单“整托仓位管理”

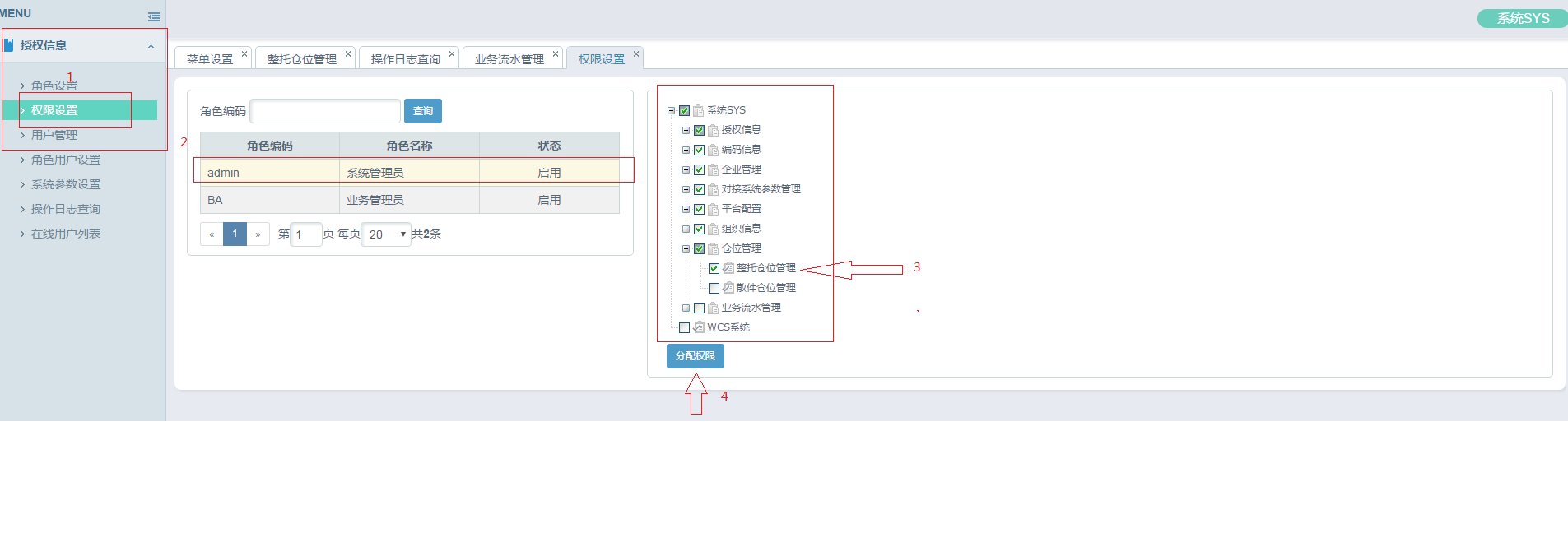


（或在数据库中直接添加“整托仓位管理”子菜单）

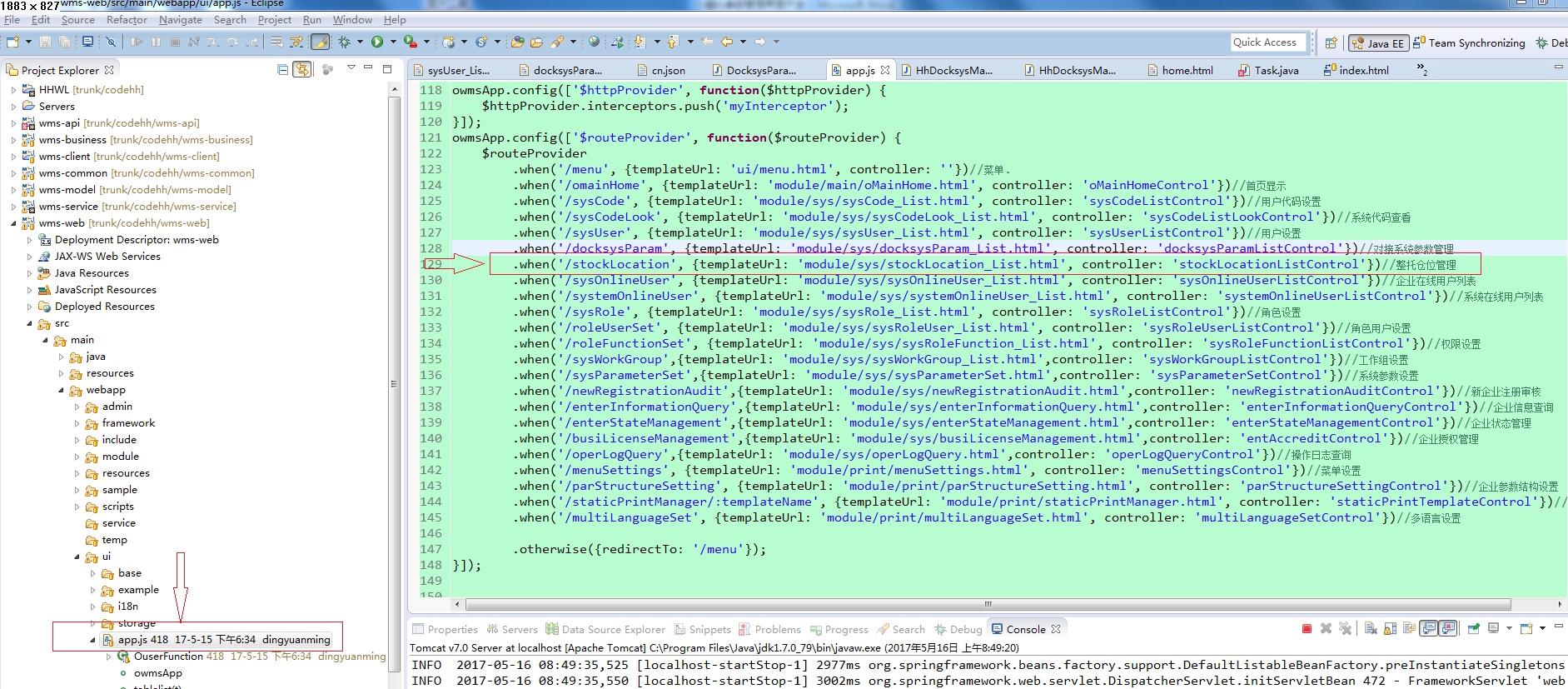




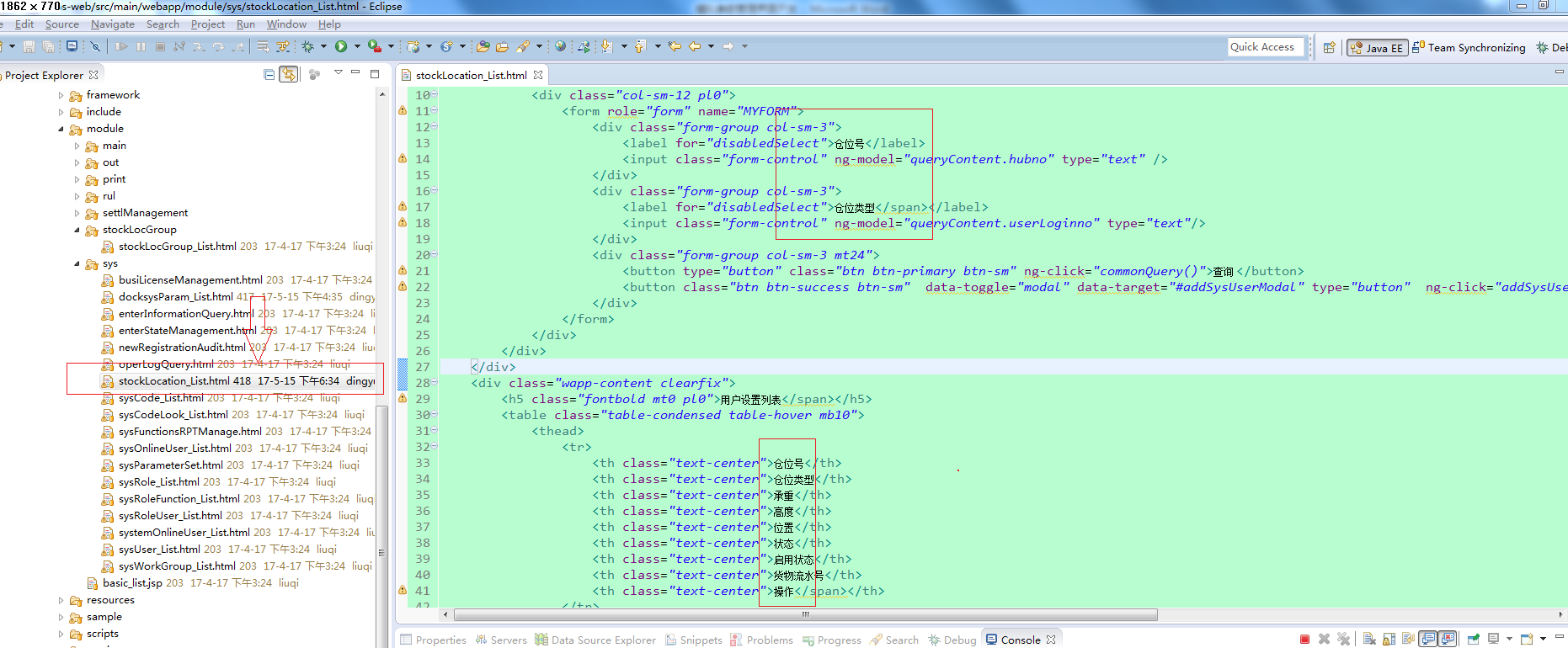
3在权限设置中分配权限



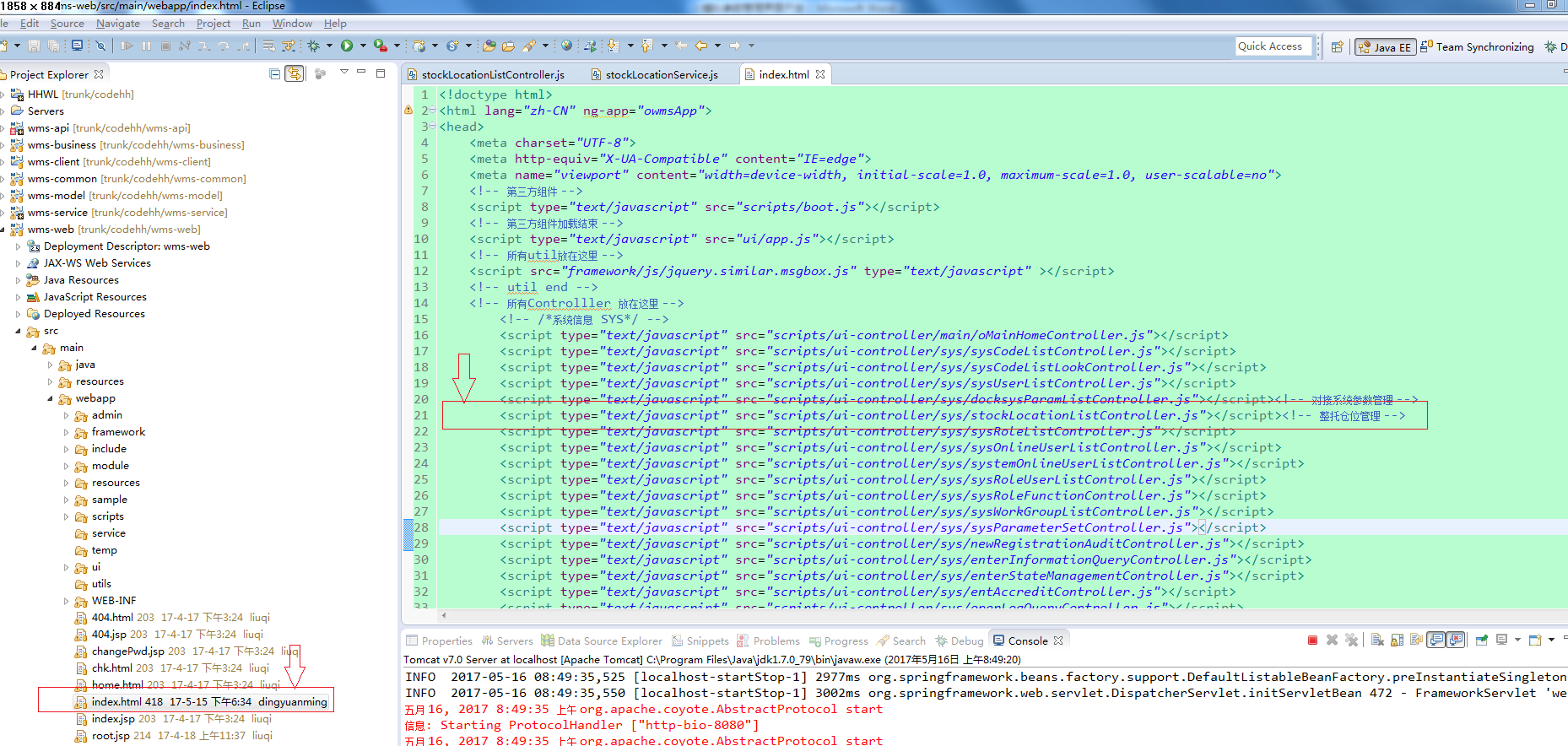
开发新界面时需在app.js(统一管理路径为owms-web\src\main\webapp\ui\app.js)内新增对应关联关系，如：.when('/stockLocation', {templateUrl: 'module/sys/stockLocation\_List.html', controller: 'stockLocationListControl'})//整托仓位管理

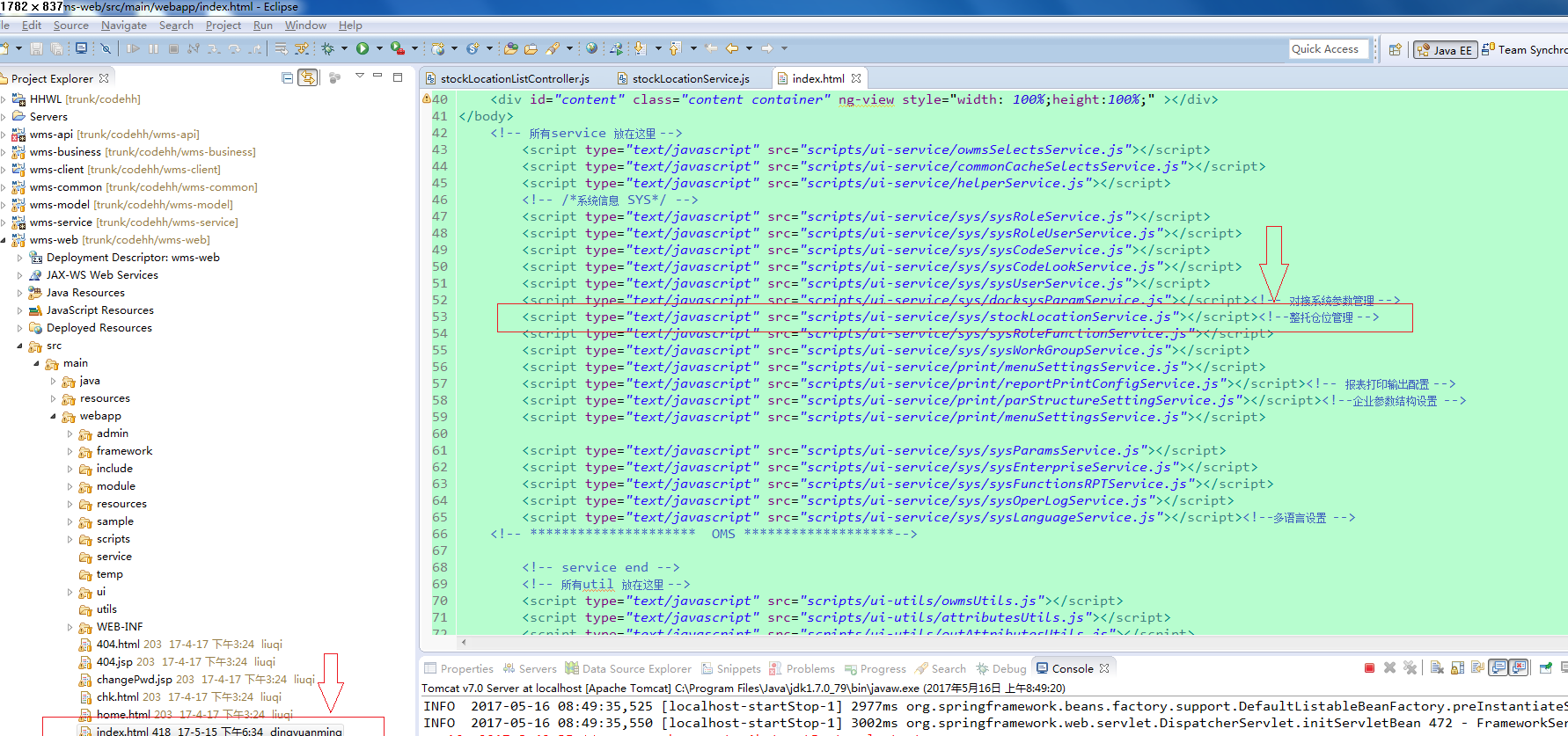


其中stockLocation\_List.html 为页面展示的文件

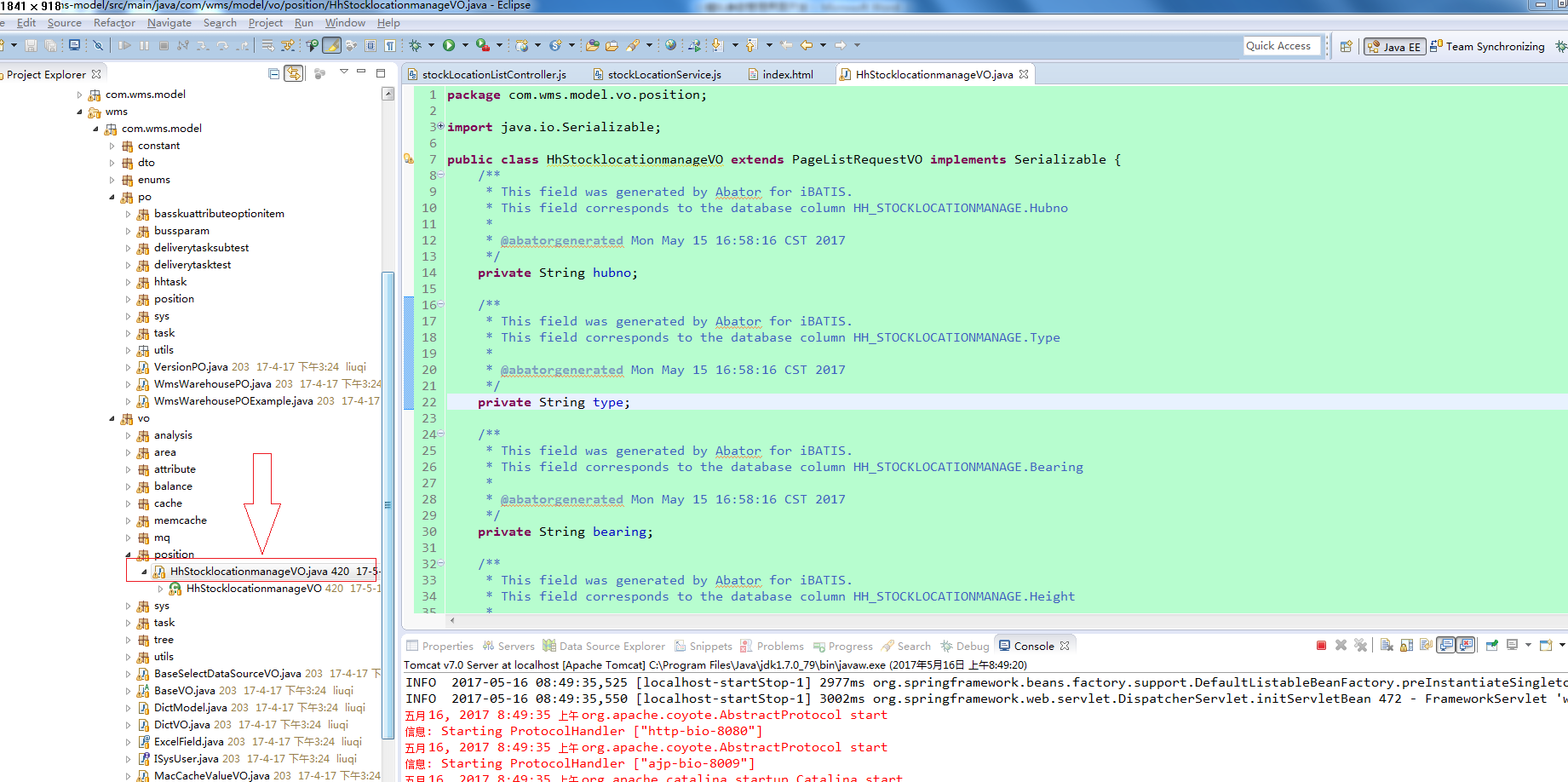


统一加载路径为owms-web\src\main\webapp\index.html，开发新界面新增的\*.js文件（stockLocationService.js）需在index.html内添加引用。

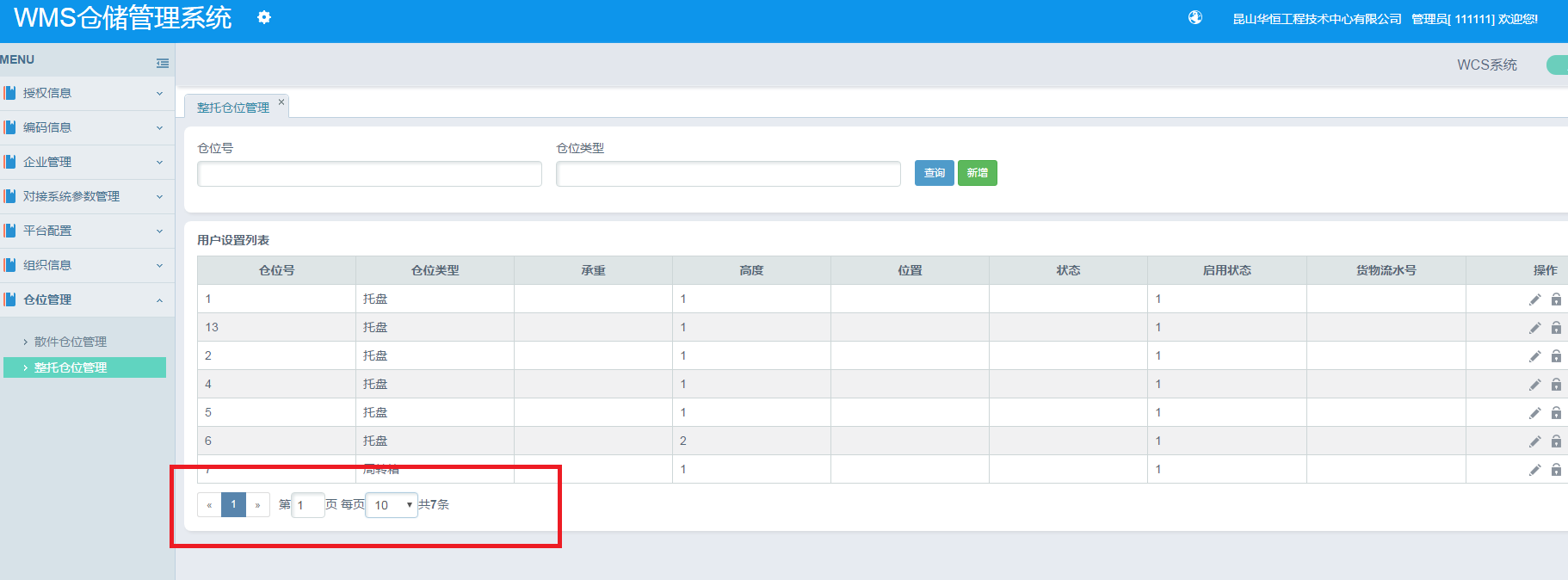




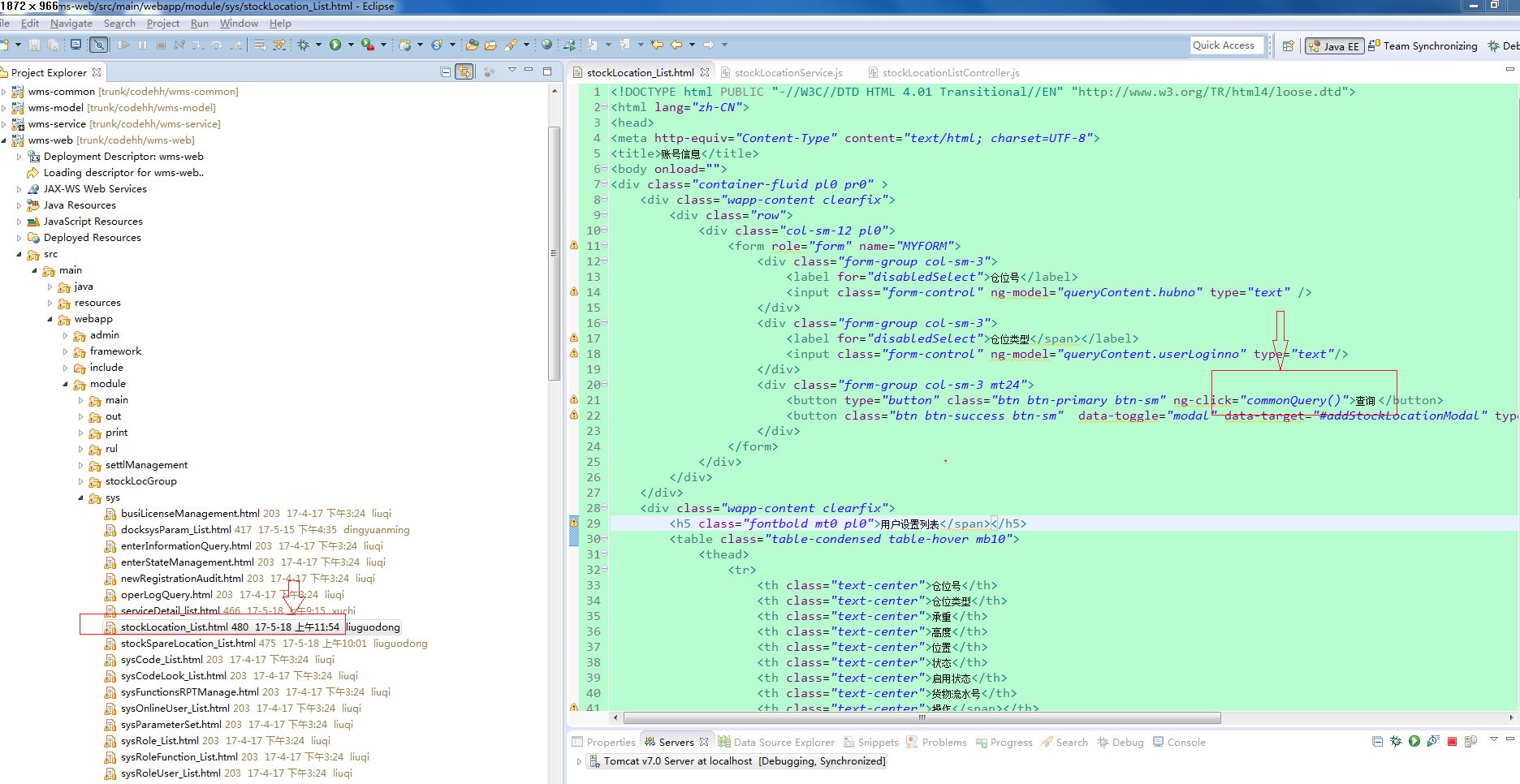
界面展示的数据对象为VO、根据页面数据展示要求予以定义，具体文件路径为： /wms-model/src/main/java/com/wms/model/vo/position/HhStocklocationmanageVO.java



分页查询功能实现：



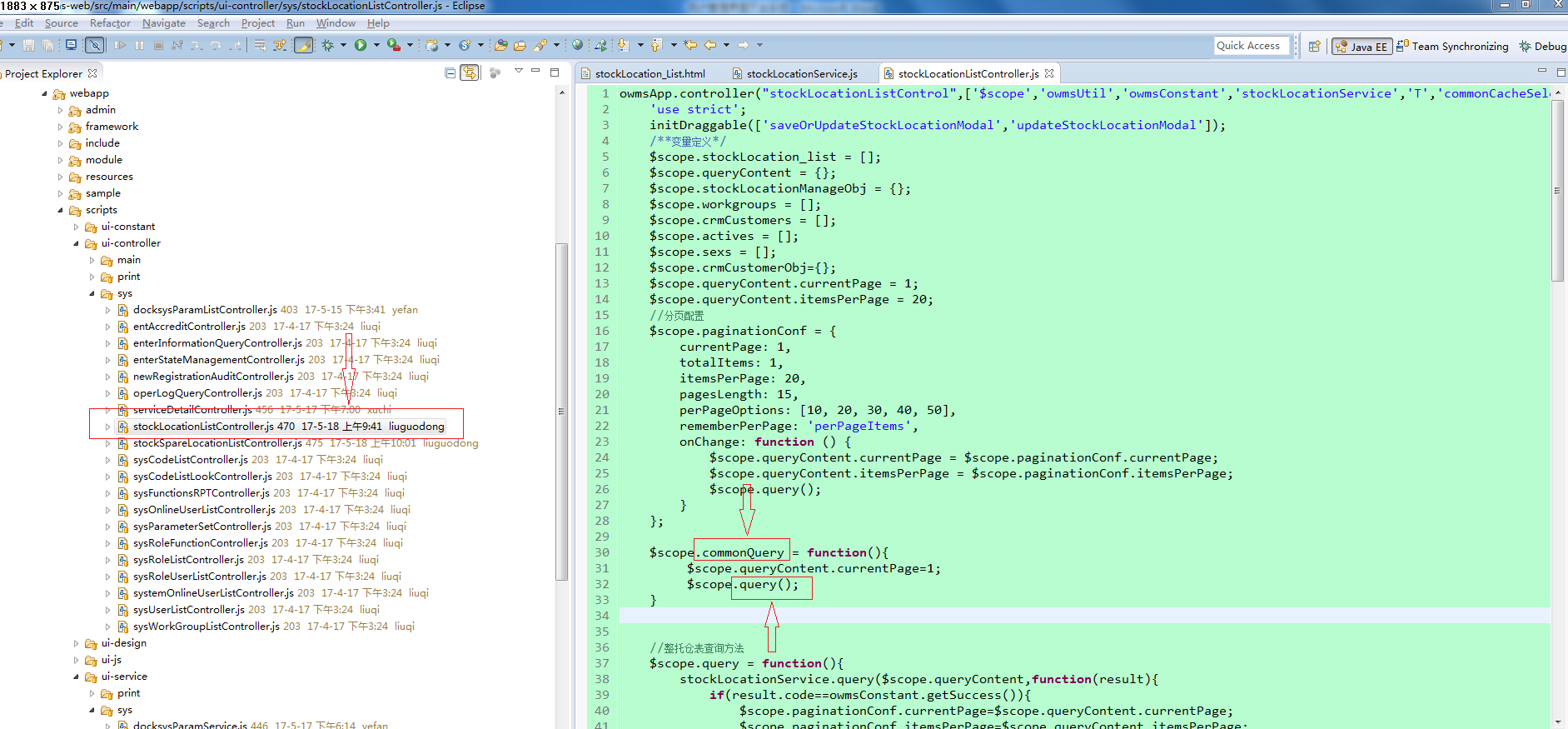
点击界面“查询”按钮（界面对应的html文档为stockLocation.html），调用commonQuery()方法



controller: '\*'为对应页面数据处理的js文件，controljs文件内处理界面数据并调用servicejs文件内action方法，此处即调用query（），controljs按照Angular JS语法予以编写、servicejs为普通js语法编写；

stockLocationListController.js文件的具体路径为：

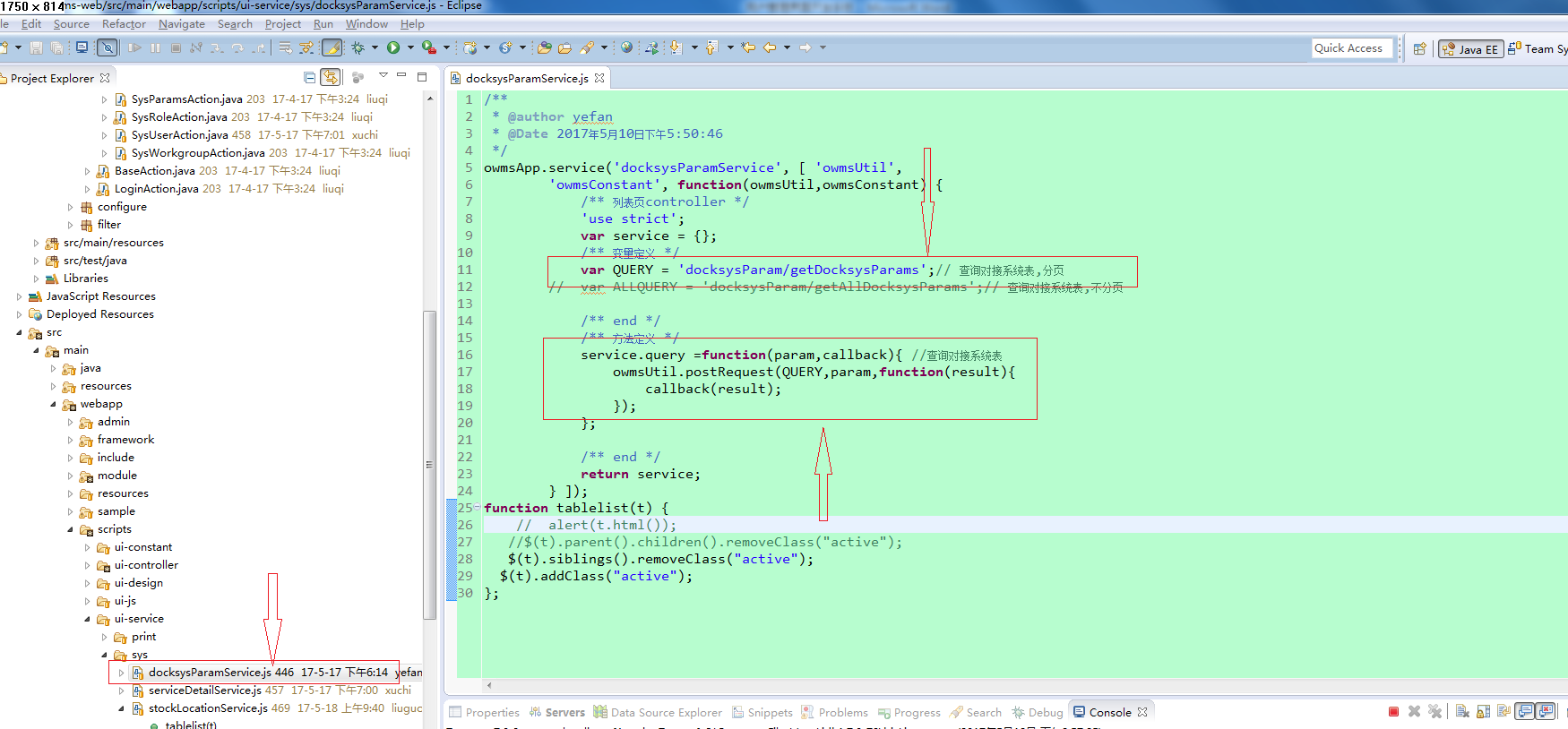
/wms-web/src/main/webapp/scripts/ui-controller/sys/stockLocationListController.js



stockLocationService.js文件的具体路径为：

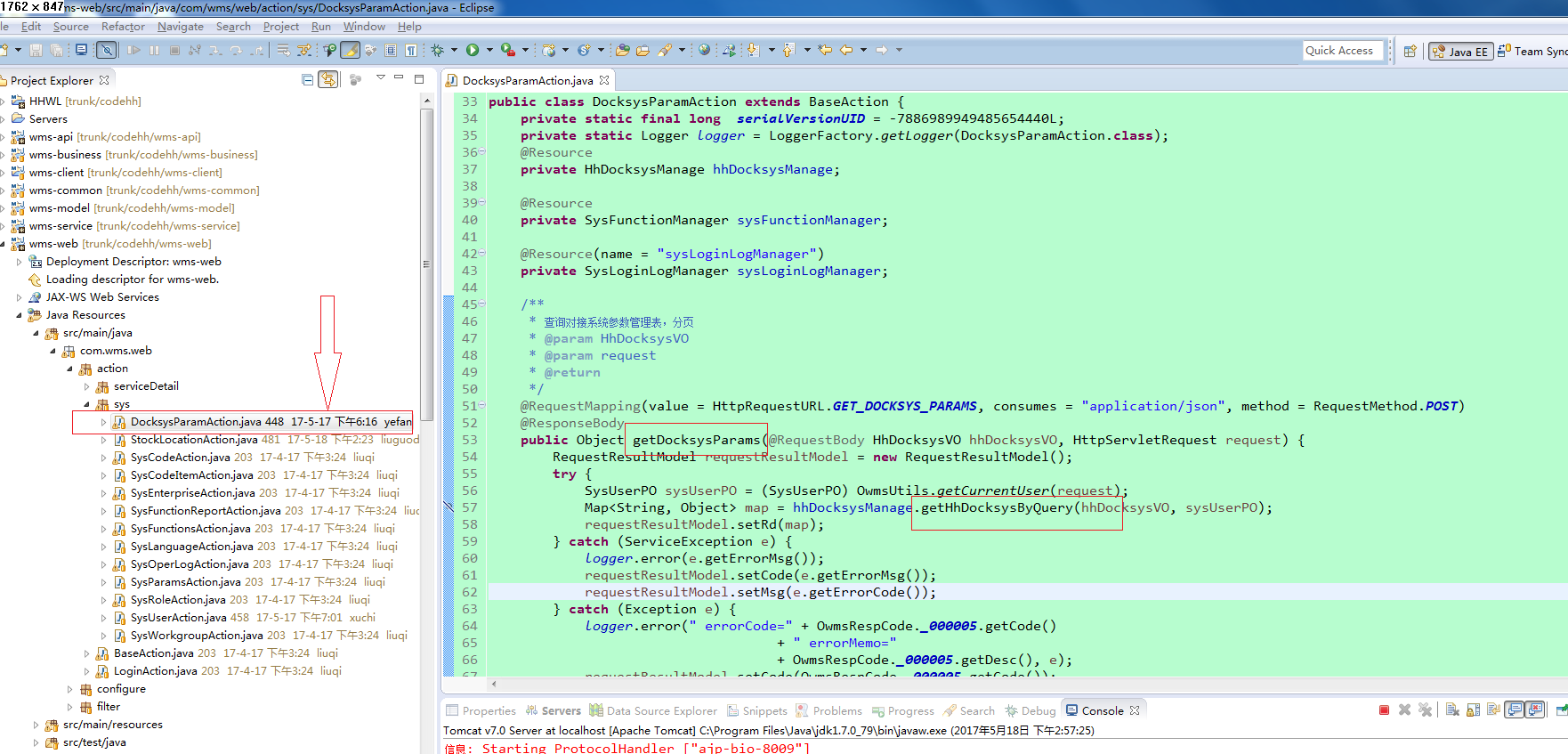
/wms-web/src/main/webapp/scripts/ui-service/sys/ stockLocationService.js

servicejs实现展示层与控制层方法的交互，servicejs方法通过POST请求调用action（\*.do）方法，即getDocksysParams方法；



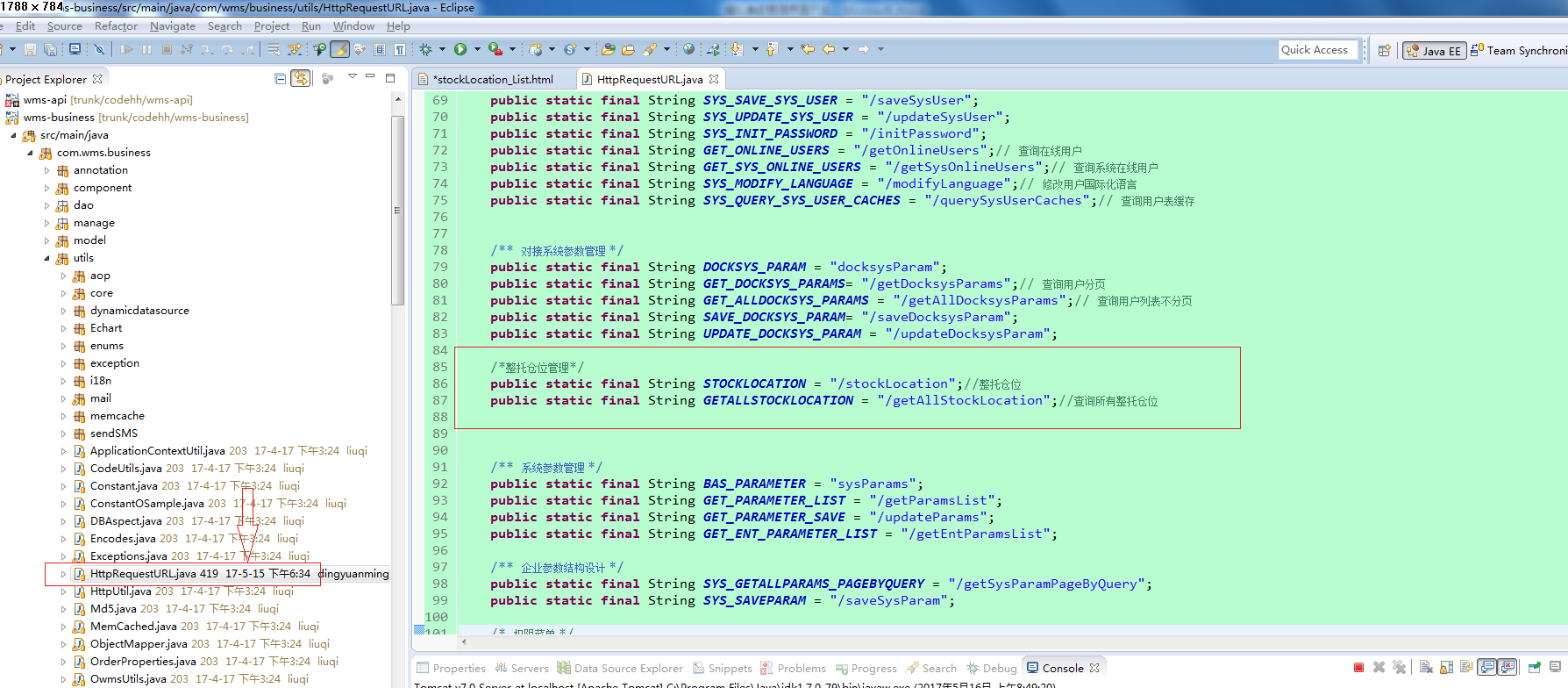
action（\*.do）方法调用

servicejs实现展示层与控制层方法的交互，servicejs方法通过POST请求调用action（\*.do）方法，路径为/wms-web/src/main/java/com/wms/web/action/sys/StockLocationAction.java，



调用方法的统一配置文件为

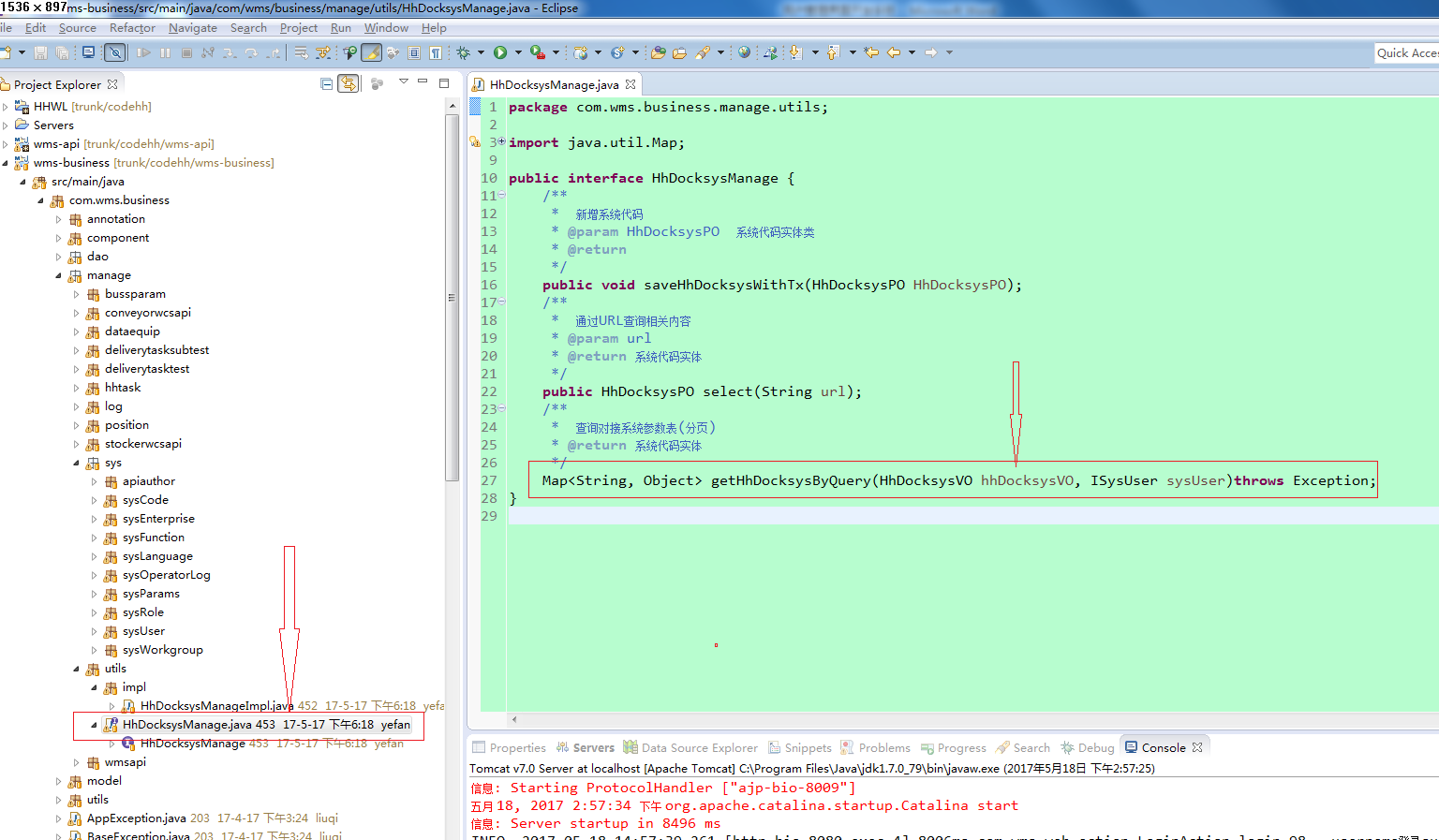
owms-business\src\main\java\com\odianyun\owms\business\utils\HttpRequestURL.java；

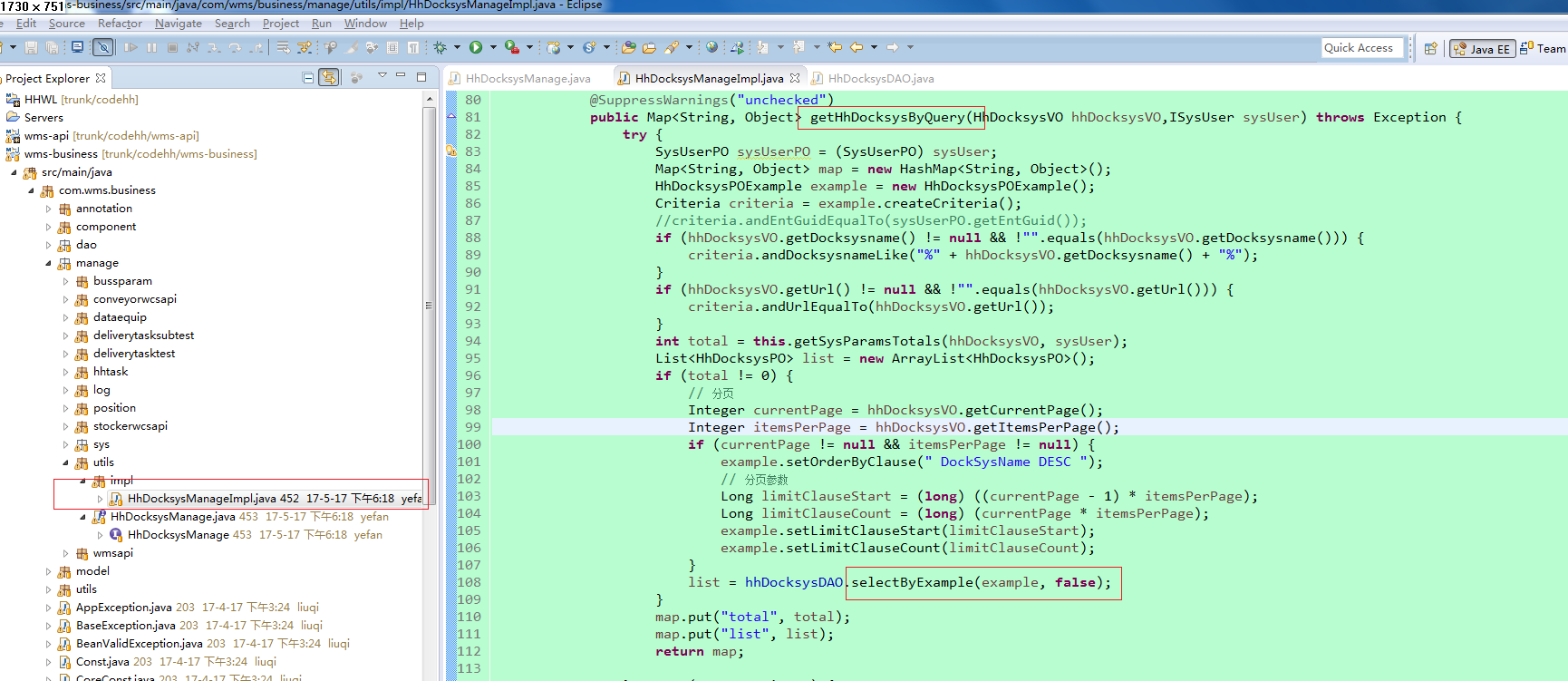


action（\*.do）方法内调用业务逻辑方法，此处getDocksysParam()方法调用业务逻辑层的getDocksysByQuery();，业务逻辑处理方法的具体路径为/wms-business/src/main/java/com/wms/business/manage/position/HhStocklocationManage.java

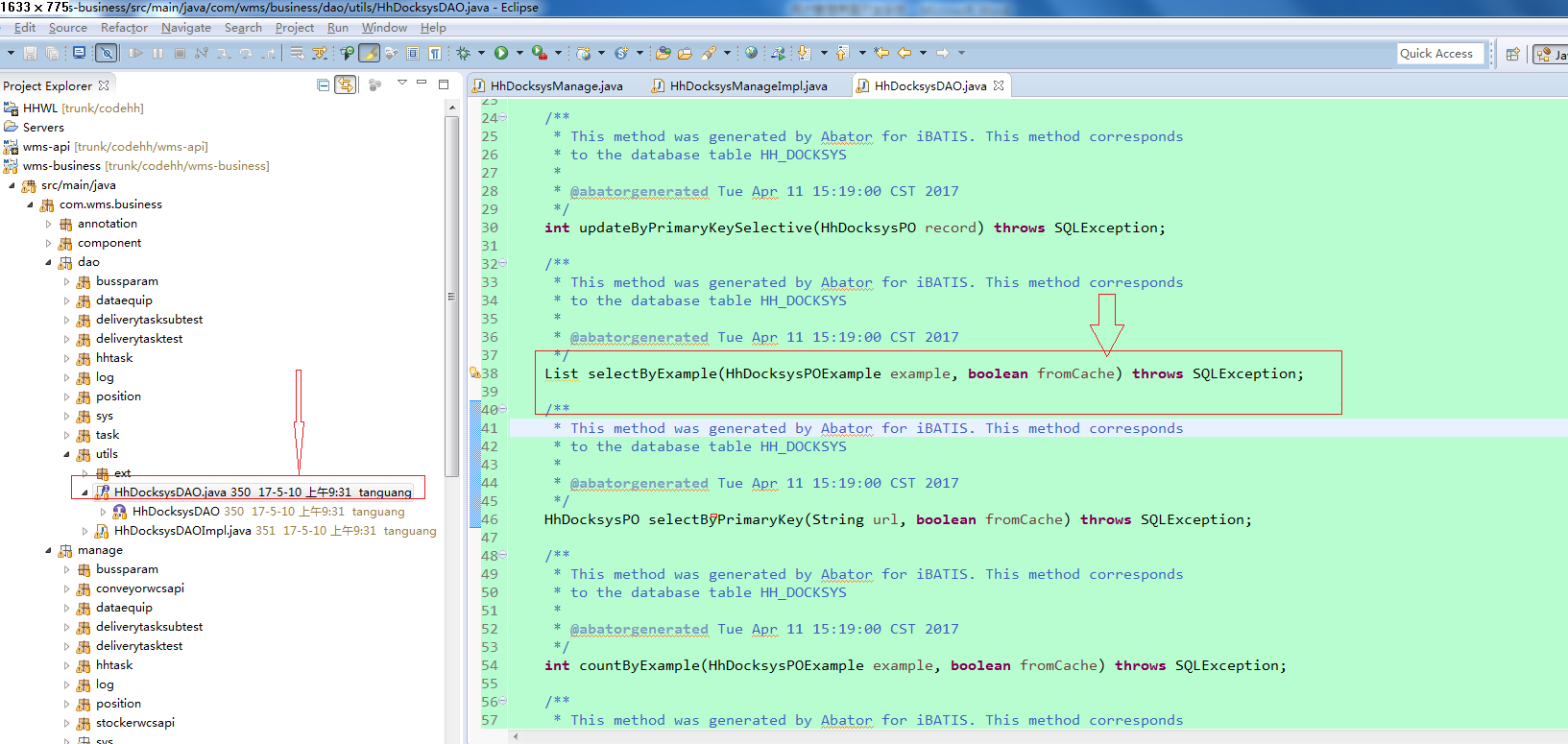
/wms-business/src/main/java/com/wms/business/manage/position/impl/HhStocklocationManageImpl.java

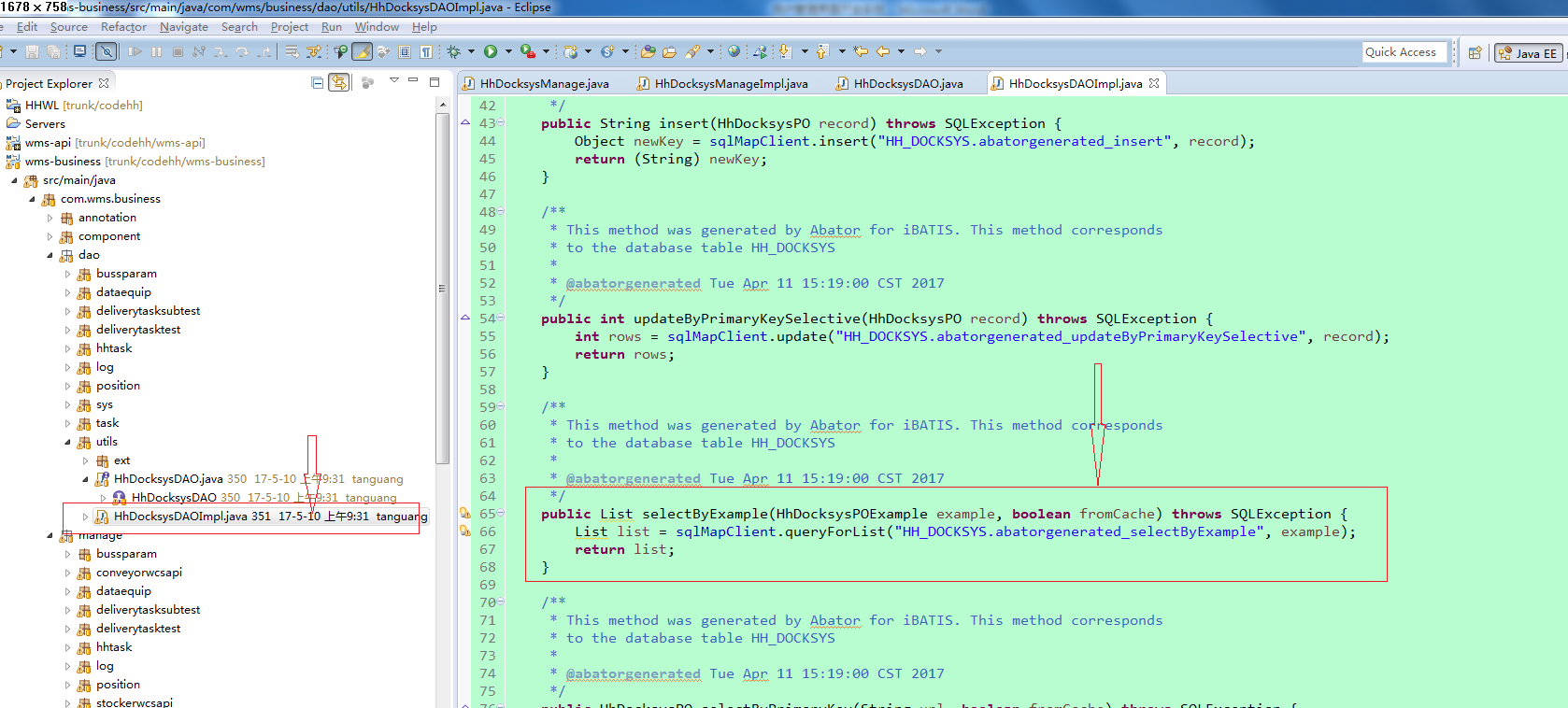
getDocksysByQuery()为业务层的分页查询方法



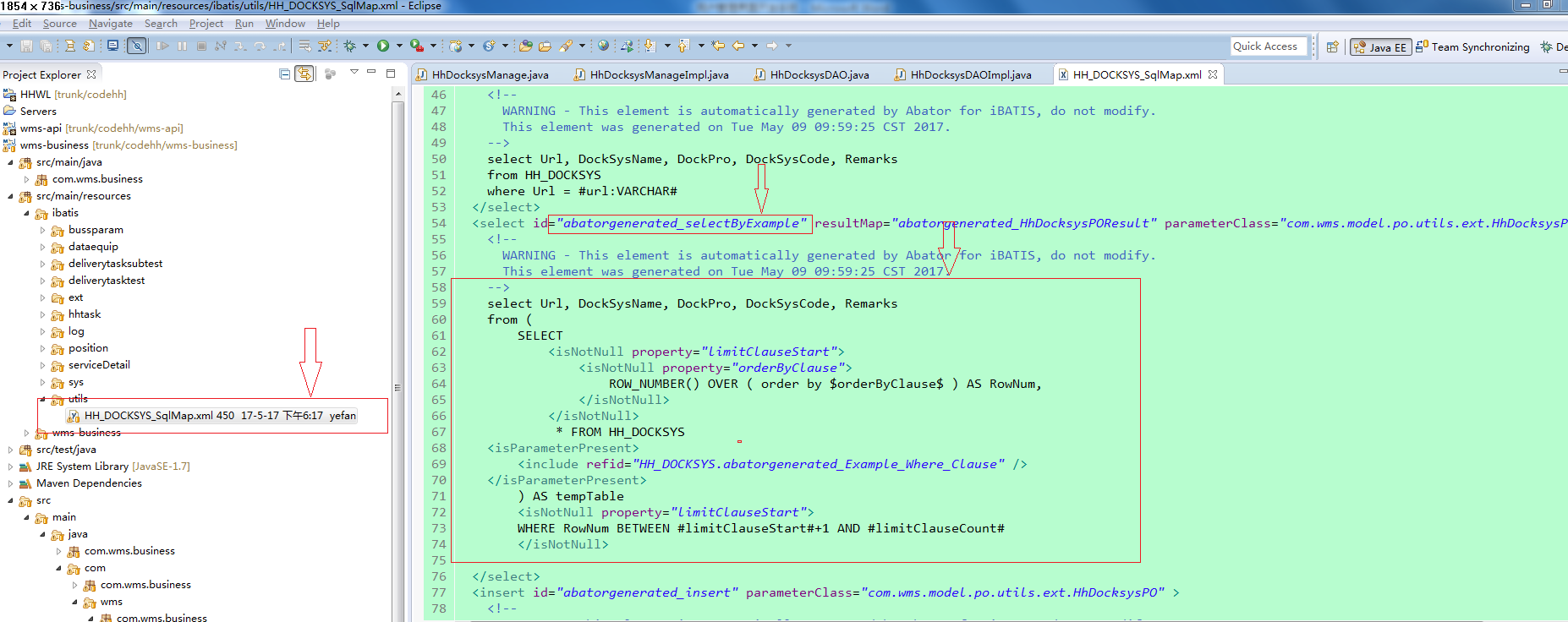


selectByQuery()调用DAO层的selectByExample();





对应的SQL语句为id=“abatorgenerated\_selectByExample”（此处注意修改abator自动生成的SQL语句为以下的形式）



数据库操作方法调用Spring的mapping文件，即

/wms-business/src/main/resources/ibatis/position/HH\_STOCKLOCATIONMANAGE\_SqlMap.xml内具体的SQL语句。

数据操作SQL语句编写规范

SQL语句编写需严格按照mybatis规范执行，自动生成文件内的SQL语句不允许修改。自定义Mapper方法的反馈对象可设置为VO、以方便直接反馈界面使用。

关于操作数据库的PO层,DAO层文件，以及HH\_MSGINFOLOG\_SqlMap.xml文件均由obator自动生成。（具体数据操作方法可参考接口开发文档）。

**实现各个页面中的增加功能：**

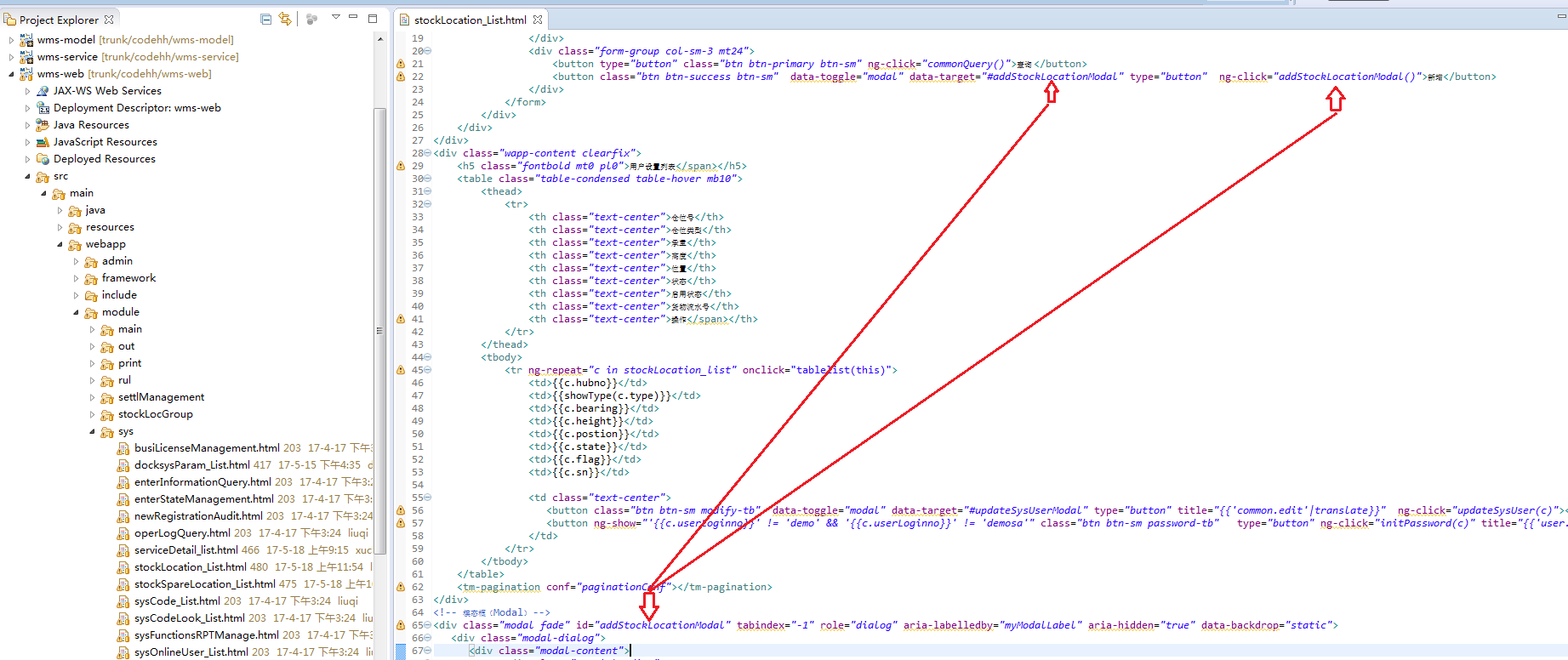
**以整托仓位管理页面中的新增功能为例；**

**1.功能介绍：**

**点击整托仓位管理页面中的新增按钮，跳转到新增页面，输入响应的参数（根据需求文档），点击保存，成功则跳出新增成功提示框；失败则跳出新增失败提示框。**

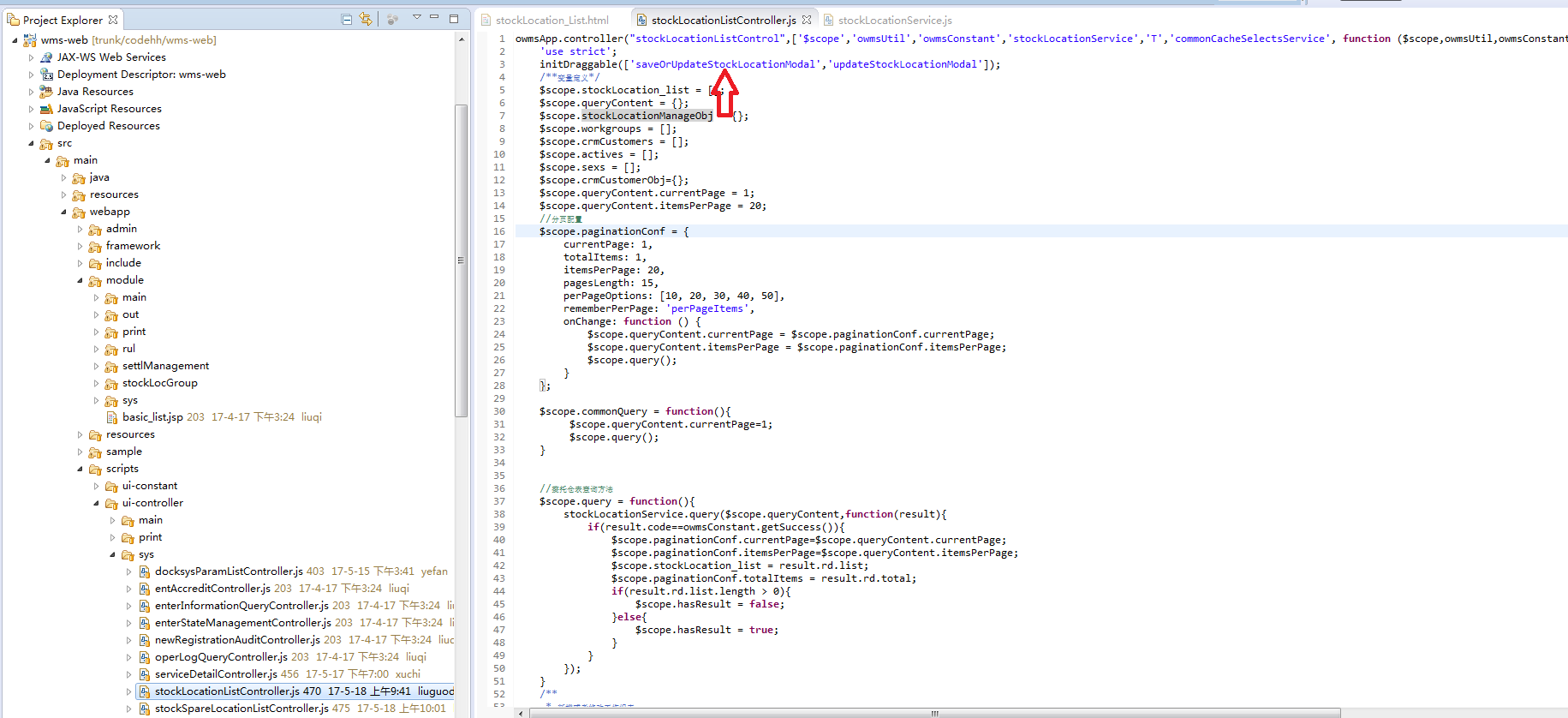
**2.实现步骤：**

**1).在stockLocation\_List.html中找到新增按钮对应的button，将ng-click单击事件的方法改成与模态框的id属性相同。data-target属性也改成和模态框id属性相同。**

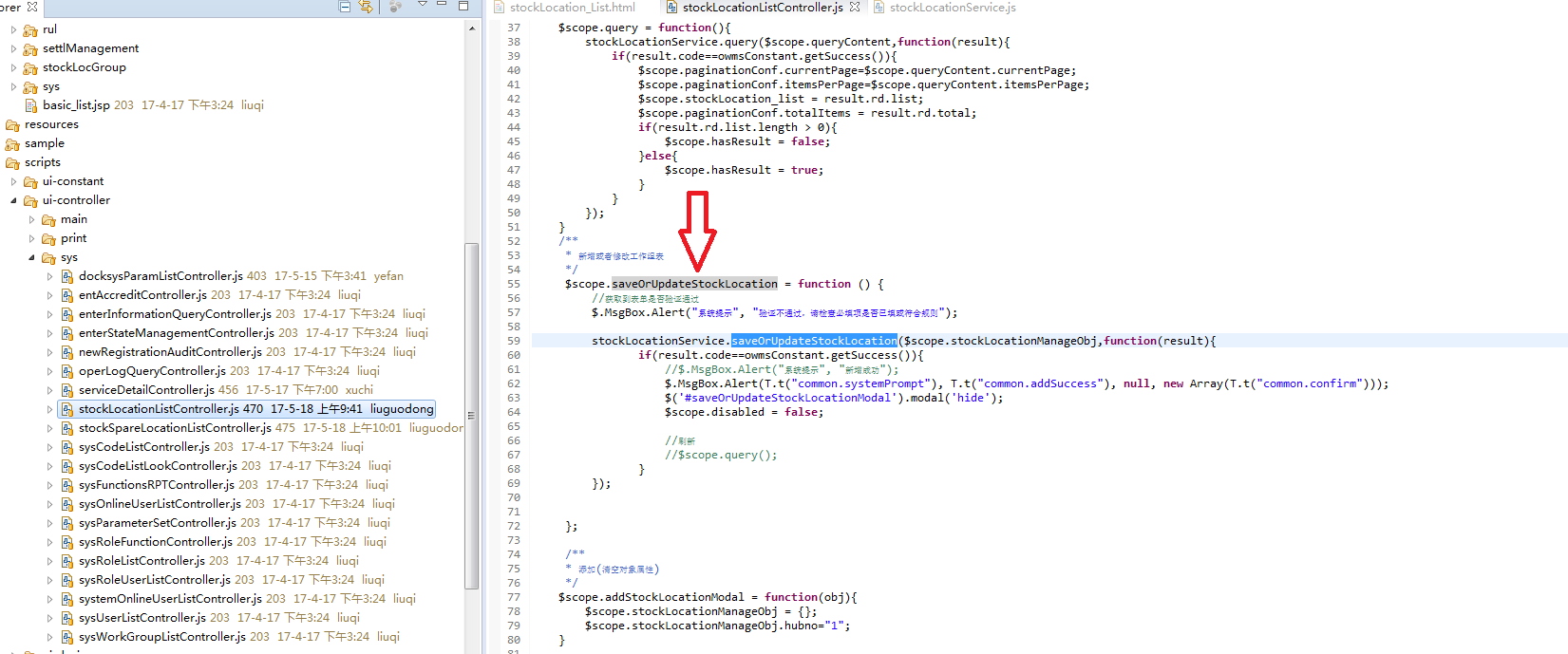
****

**2).模态框中的表单对应的提交方法ng-submit="saveOrUpdateStockLocation()" 与StockLocationController.js文件中的initDraggable([' saveOrUpdateStockLocation ',''])一致。**

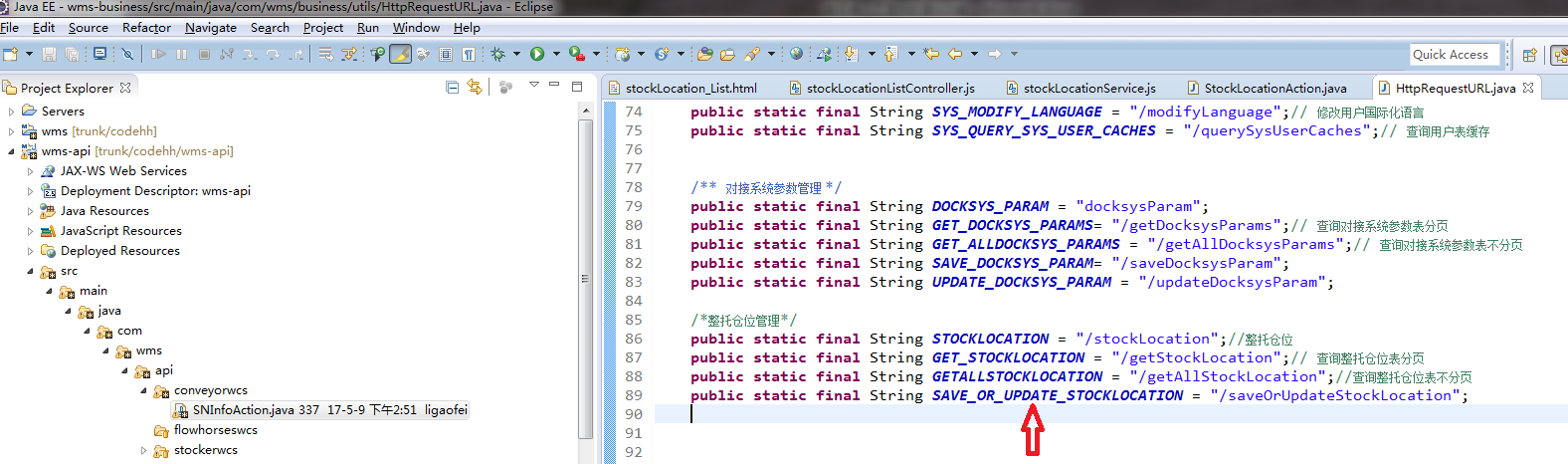
****

****

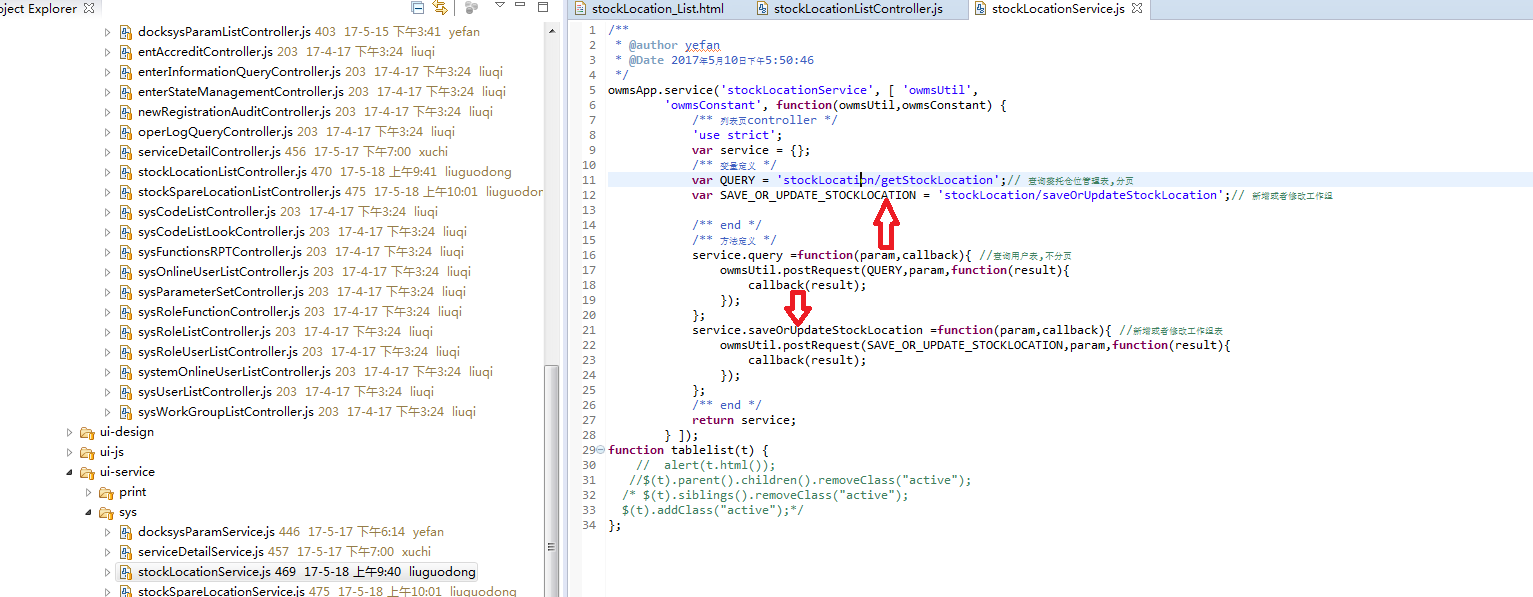
**3).并在StockLocationController.js添加响应的saveOrUpdateStockLocation()方法。需要更改方法中对应的数据。**

****

**4).该方法会调用stockLocationService.js文件，需要在service文件中添加新增参数对象SAVE\_OR\_UPDATE\_STOCKLOCATION，并将该对象保存在HttpRequestURL文件中。**

****

**该对象在下方的saveOrUpdateStockLocation()调用，代表向后台发送请求的请求路径。**

****

**5).执行该方法后会调用后台StockLocationAction文件中的saveOrUpdateStockLocation()方法，接收表单中的数据，接受表单中的数据后，通过hhStocklocationManage对象调用saveStockLocationTithTx方法，将参数传入service层。**

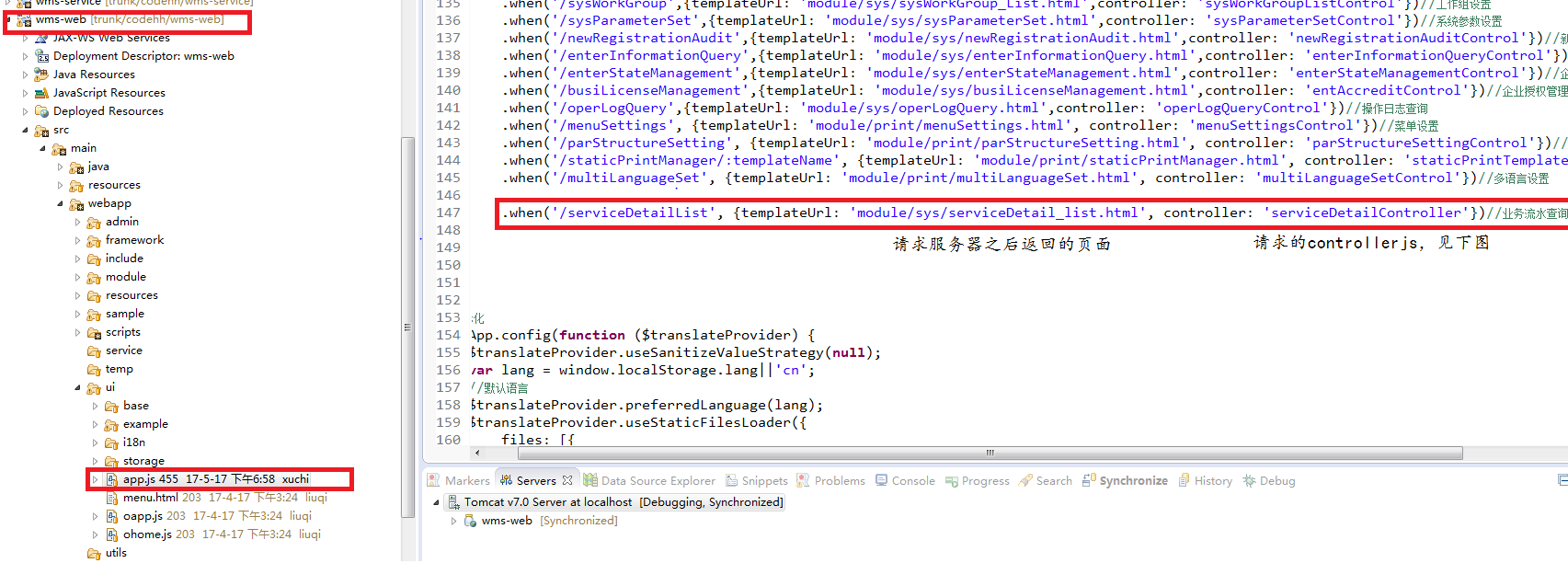
****

**6).service层接收到参数后调用DAO层方法，有DAO层将数据写入数据库中。**

****

**实现各个页面中的条件查询功能：**

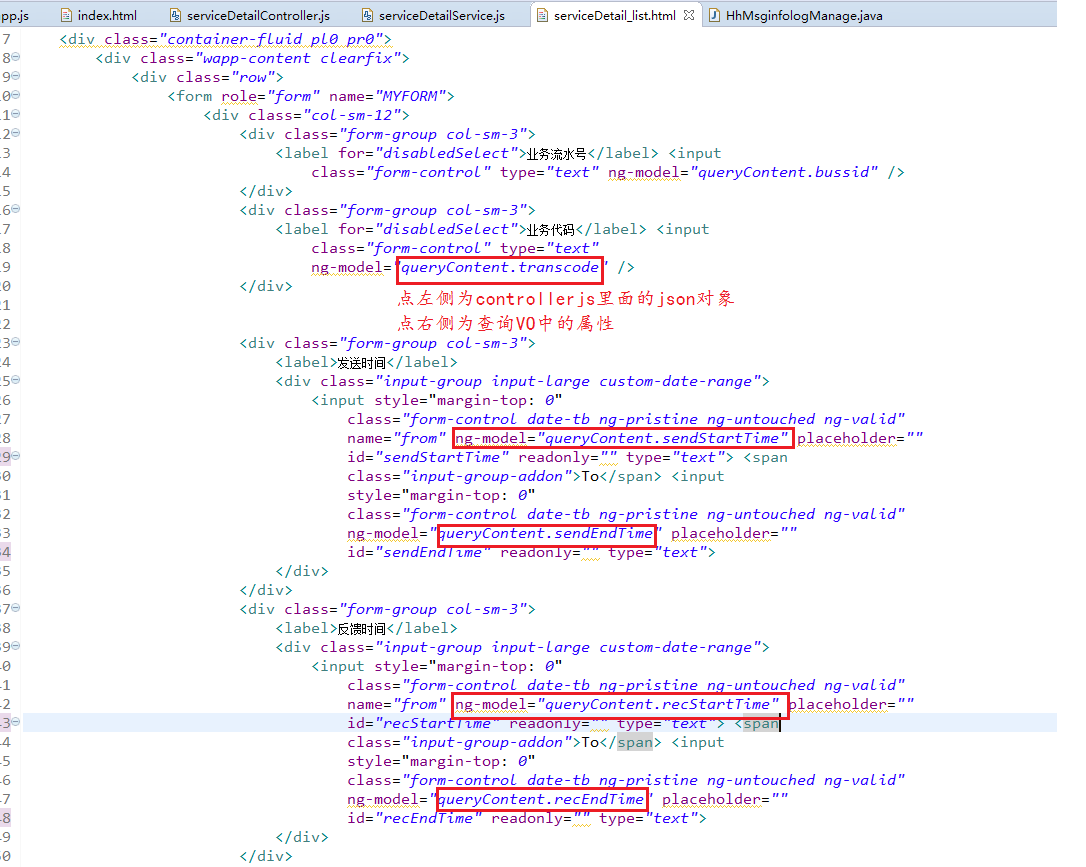
**以业务流水查询为例**



修改app.js配置



修改页面html中的rel配置.



修改页面html中的查询参数的配置.



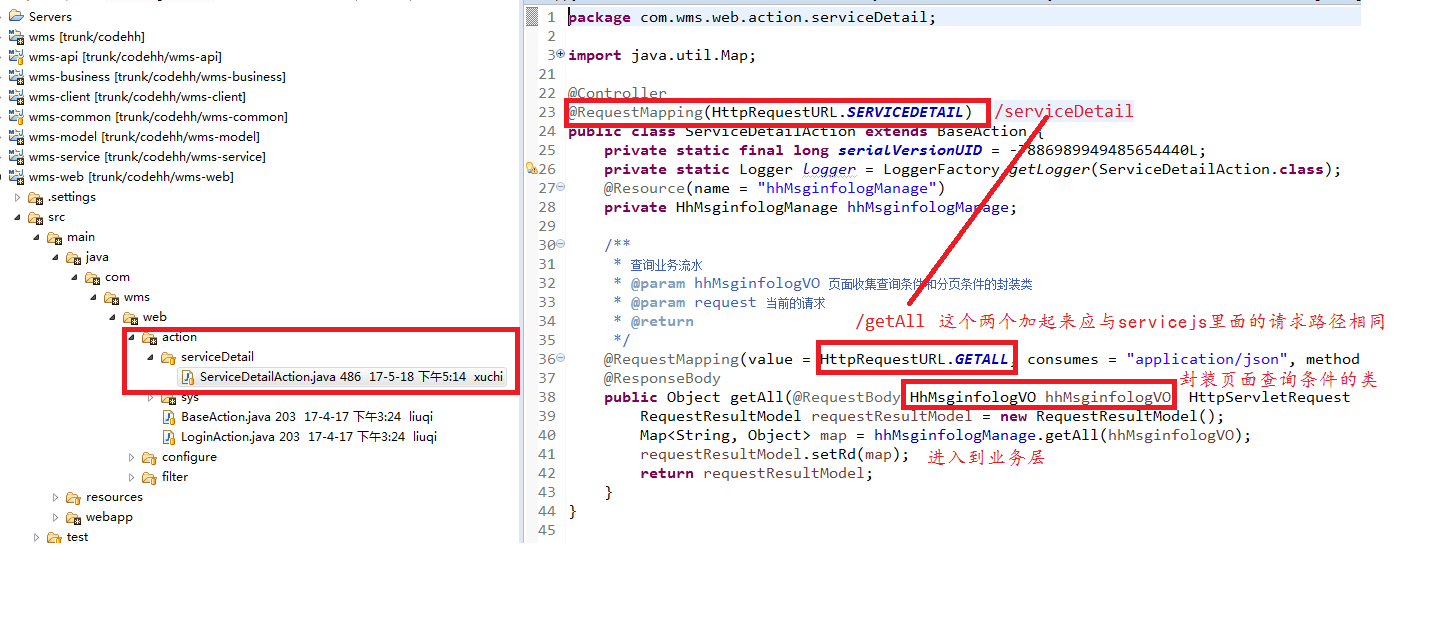
修改页面controllerjs配置.



修改页面servicejs配置，通过此处后即可进入到服务器端.

**以下是服务器端**

1、控制层



2、业务层

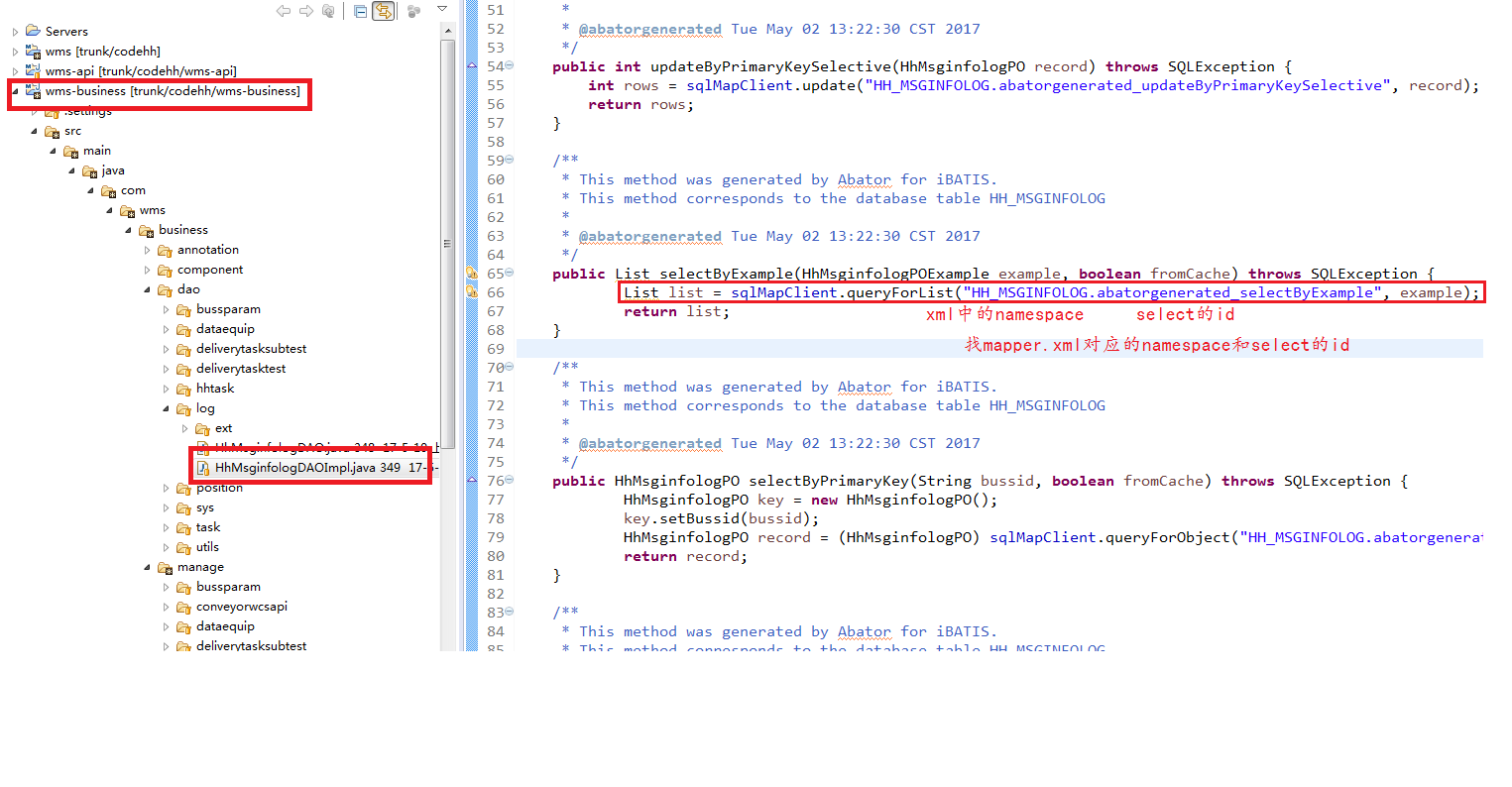
2.1总的业务层



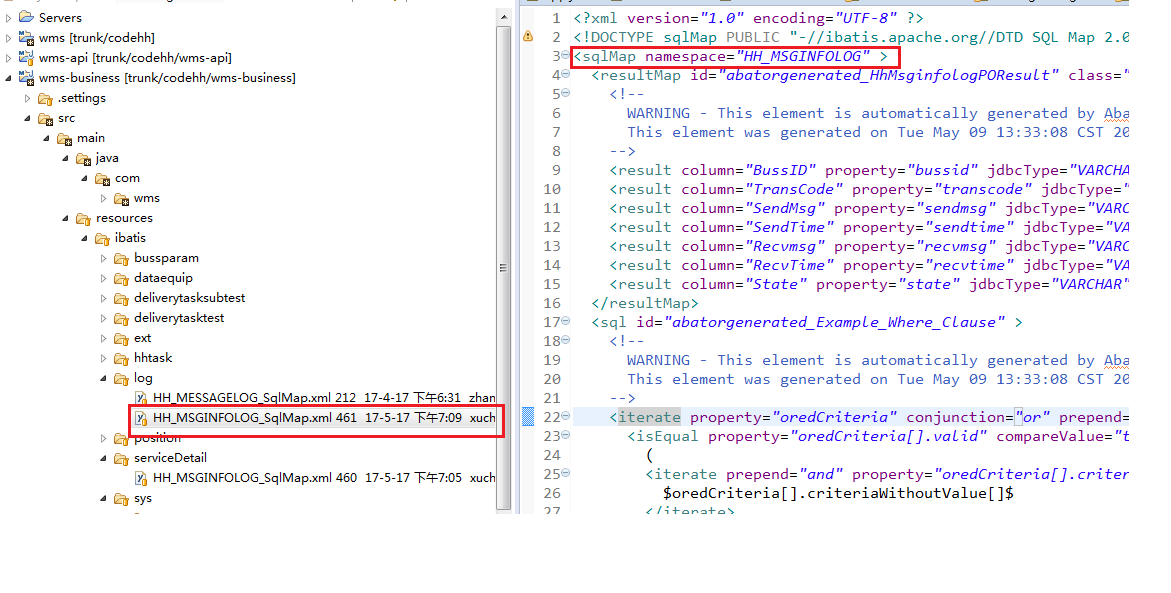
2.2添加查询条件



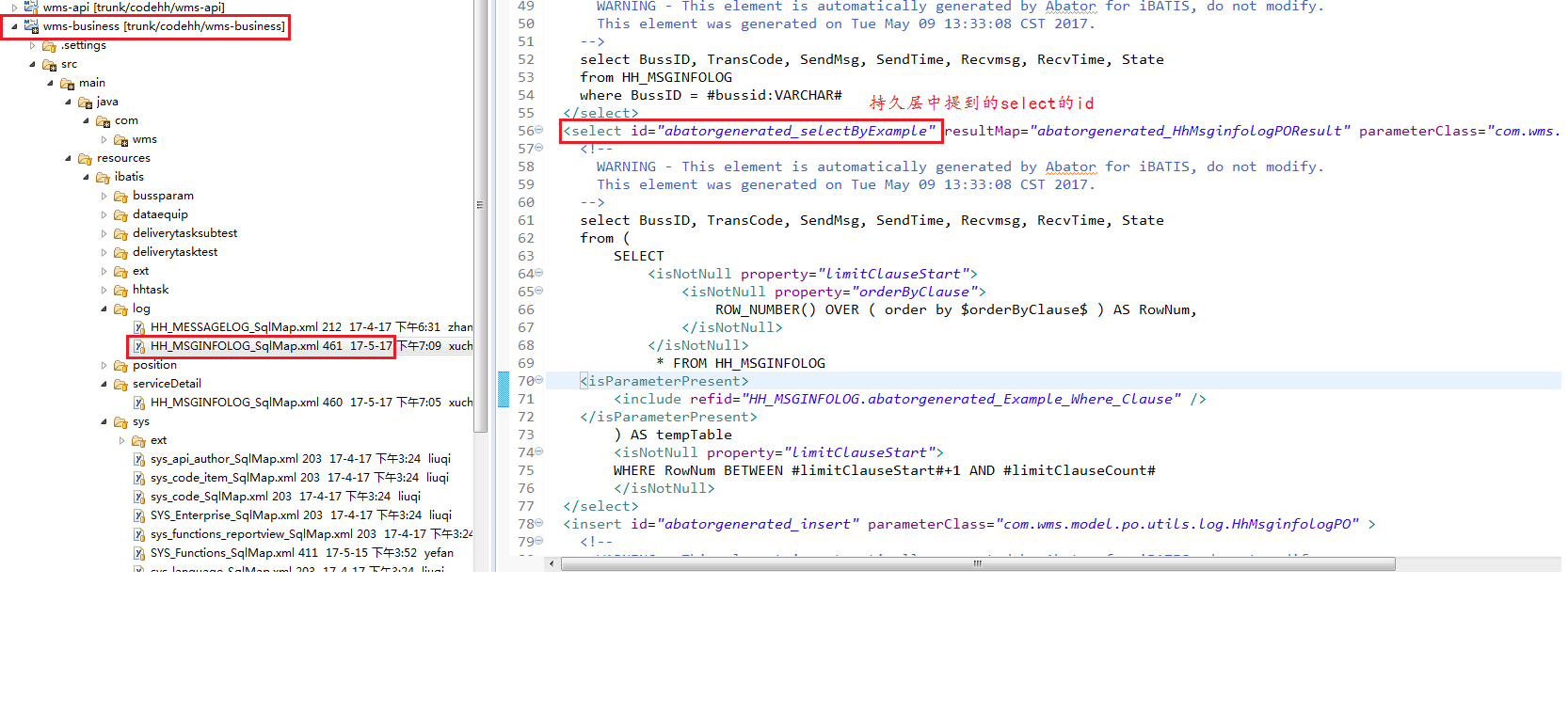
3.持久层操作数据库查询



mapper.xml



mapper.xml中提到的namespace



mapper.xml中提到的select的id