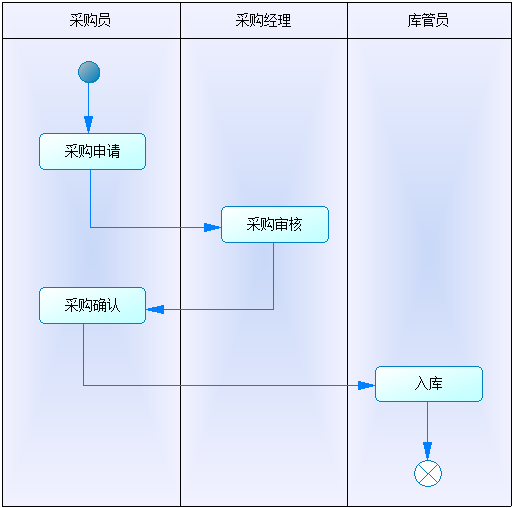
ERP项目笔记 Day-06

1. 采购主线业务

采购业务流程活动图

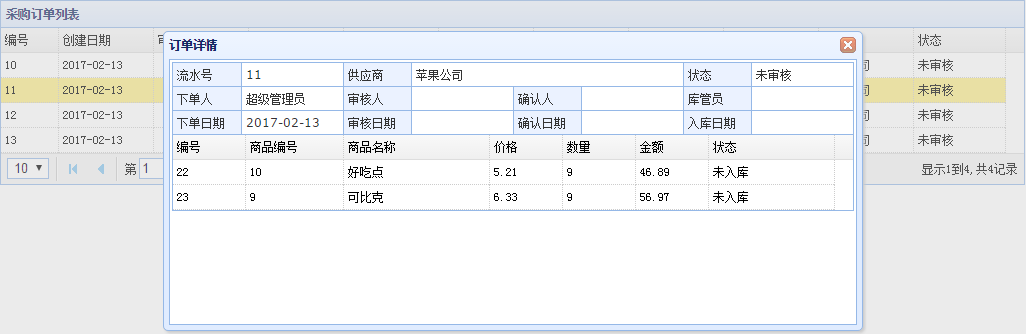


1. 采购订单查询
   1. 需求分析

显示所有的采购订单

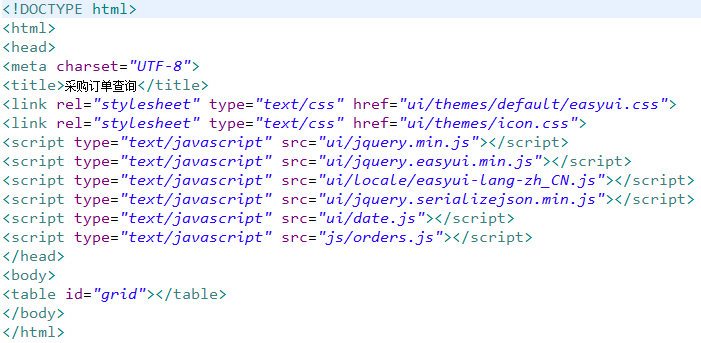


双击其中一行，弹出窗口显示订单信息和明细



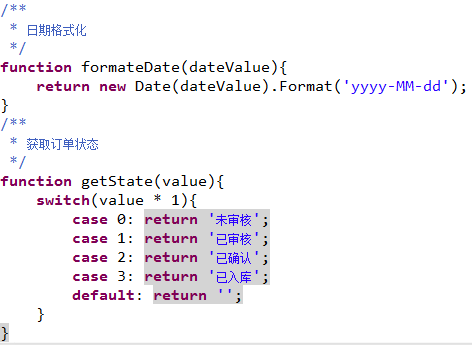
* 1. 采购订单列表

1. 修改orders.html



1. 创建orders.js文件

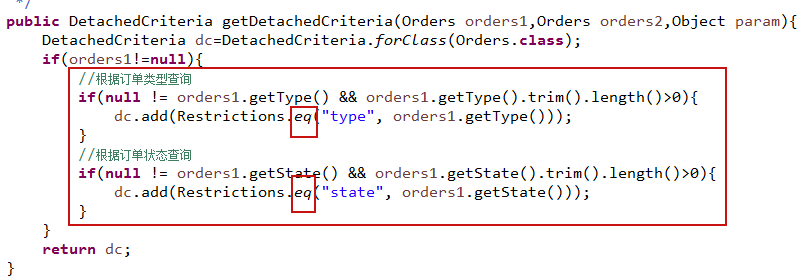
添加方法：



添加初始化方法：



1. 修改OrdersDao的getDetachedCriteria方法:



1. 显示员工名称和供应商名称，修改如下:

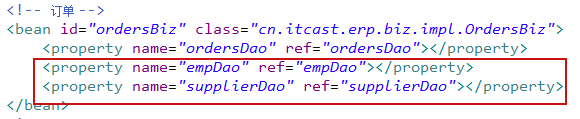
Orders实体类修改，添加红框中的属性，并生成getter与setter方法：



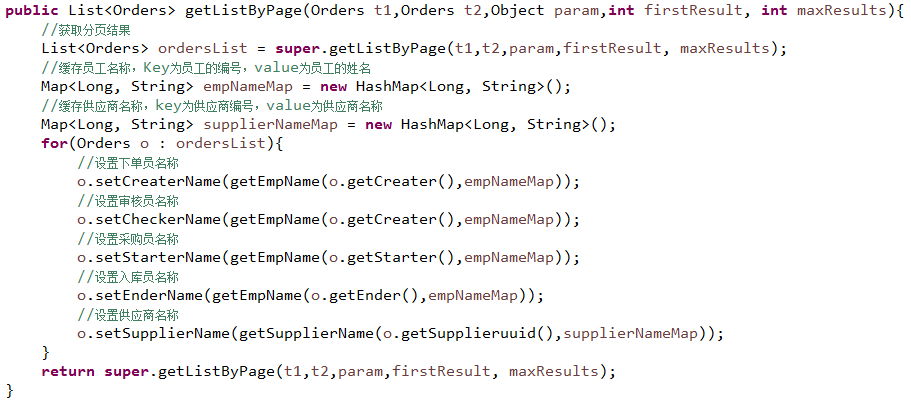
修改OrdersBiz，添加属性EmpDao和supplierDao，并生成getter与setter



配置spring文件：



重写listByPage方法：



增加两个私有方法，分别用来获取员工名称和供应商名称





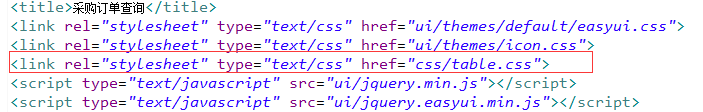
* 1. 采购订单明细
     1. 添加弹出明细窗口

1. 修改orders.html，加入弹出窗口的内容



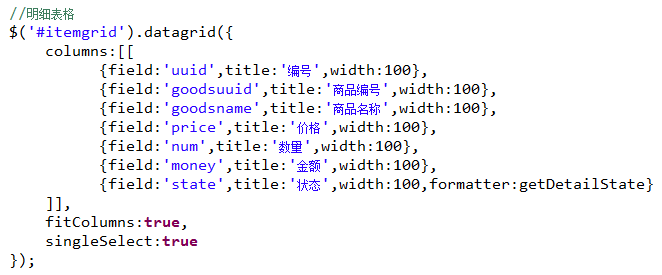
1. 添加table美化css引用

复制项目必备里的table.css到webapp/css下，并在orders.html的header标签中添加引用

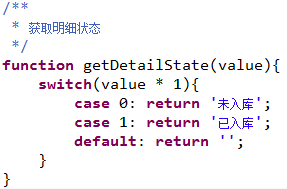


* + 1. 加载明细表格

1. 在orders.js中添加itemgrid的初始化



添加getDetailState方法:

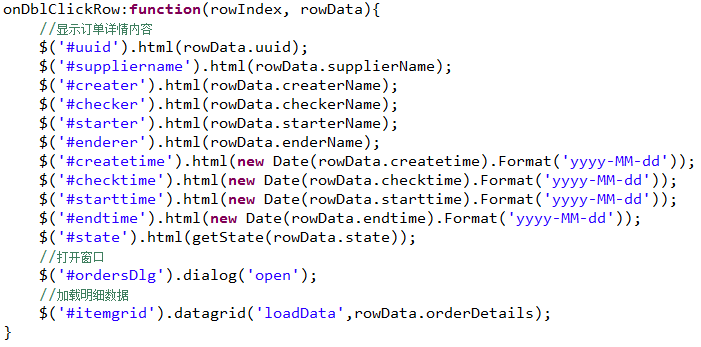


* + 1. 双击弹出明细窗口

1. 当双击采购订单列表中的某一行时弹出，我们来添加双击行的事件

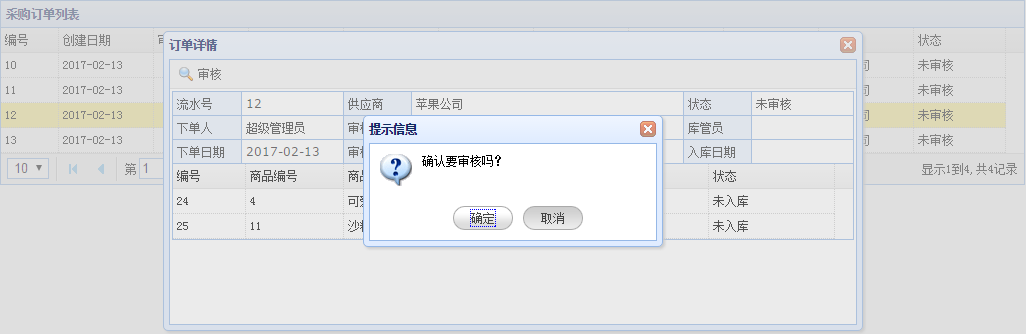


给grid表格添加双击事件，弹出窗口后加载订单信息，把订单详细信息显示到table，同时加载订单里的商品列表：



1. 采购订单审核
   1. 需求与实现思路分析

需求：列出所有“未审核”的采购订单，双击某一行时弹出详细窗口，点击“审核”按钮后弹出确认框，确认后提示“审核”结果。审核成功，则关闭详细窗口并刷新“未审核”列表。如下图

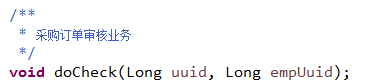


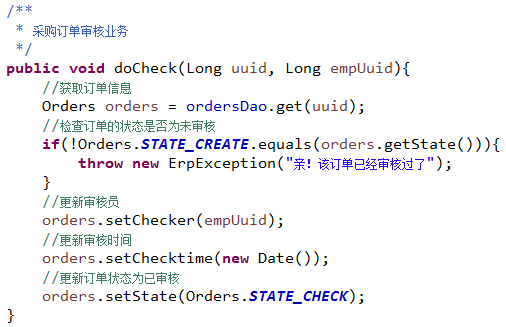
实现思路：

1. 订单的状态：

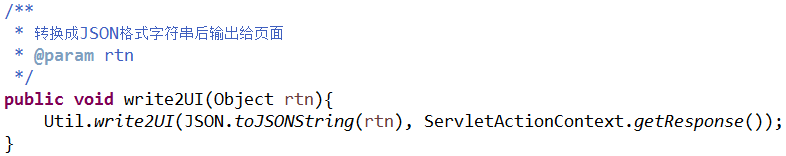
0: 未审核， 1：已审核， 2：已确认， 3：已入库

1. 业务OrdersBiz添加doCheck方法，用来修改订单的状态state，同时设置审核人，审核时间
2. OrdersAction添加请求调用方法doCheck，接收订单编号，调用审核业务，返回审核结果给前端
3. 实现未审核订单列表，双击后打开详细窗口，点击“审核”按钮完成审批操作，再刷新未审核列表
   1. 后端代码实现
4. 在IOrderBiz中添加doCheck方法后，在OrderBiz中实现该方法，代码如下：





1. 抽出输出JSON字符串代码，放到BaseAction中



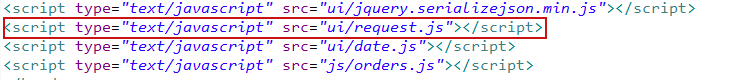
1. 在OrdersAction中添加doCheck方法，如下：



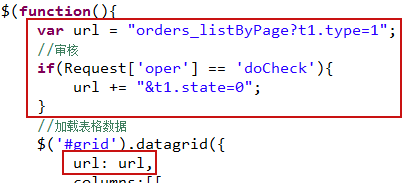
* 1. 前端实现

审核功能有一大部分的页面与查询页面相同，因此我们可以采用静态页面传参的方式来区分是查询还是审核

1. 修改orders.html，添加request.js。通过url传参oper=doCheck来进入采购订单审核的页面



1. 修改orders.js，添加静态传参修改:



1. 判断是否需要添加“审核”按钮



1. 添加doCheck方法:



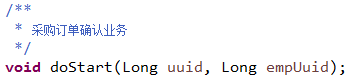
1. 采购订单确认
   1. 需求与实现思路

需求与实现跟审核类似，列表里只列出state=1即已审核的订单；提交数据后把订单的状态改成已确认state=2

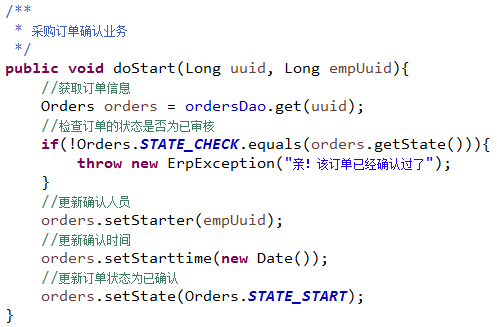
* 1. 后端实现

1. 修改OrdersBiz添加doStart方法，同时添加接口

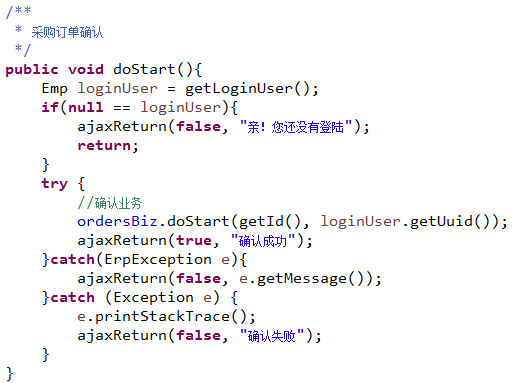
IOrdersBiz中添加方法：



OrdersBiz中实现方法：

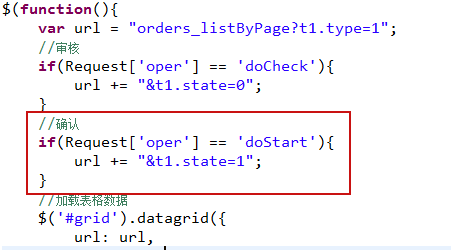


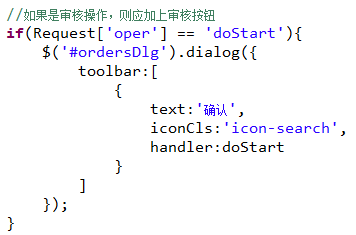
1. 修改OrdersAction，添加doStart方法，供ajax调用



* 1. 前端实现

1. 修改orders.js，添加对oper=doStart的判断,动态添加工具栏按钮，具体跟oper的操作类型有关



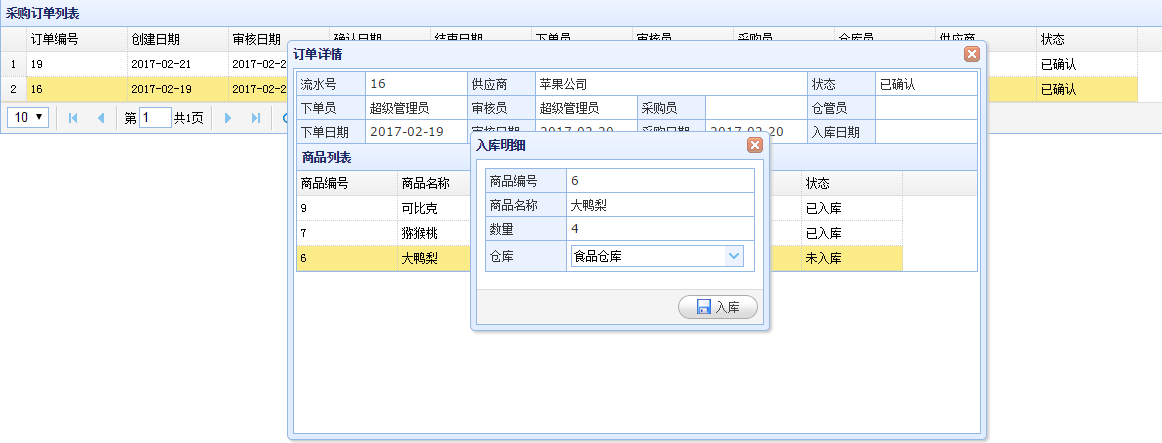


1. 订单确认跟订单审核有很多相同的地主，我们可把原有的doCheck复制过来修改成doStart方法：



1. 采购订单入库（重点）
   1. 需求分析

需求：对已确认的订单进行入库，执行入库后将会增加库存。



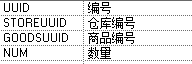
分析：订单入库的列表里只显示已确认的订单，即订单的state=2。双击某一行后可以弹出的窗口中看到订单的明细，双击明细中的某一行弹出窗口，显示订单商品明细信息，包括商品编号、名称、数量和自己管理的仓库下拉列表。选择仓库后确认，则将此商品入库，入库后更新明细记录的状态

* 1. 入库相关表结构分析

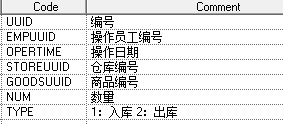
1. 仓库表（STORE)



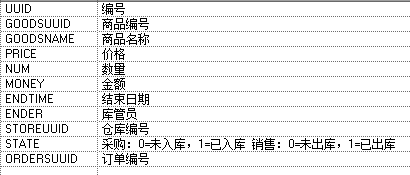
1. 商品仓库库存表（STOREDETAIL）



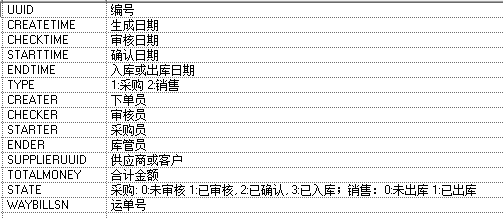
1. 商品仓库库存变动记录表（STOREOPER）



1. 订单明细表（ORDERDETAIL）



1. 订单表（ORDERS）

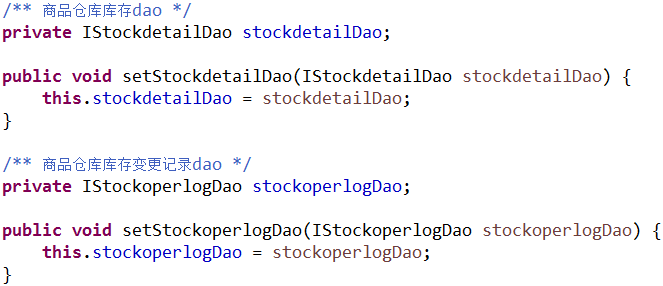


* 1. 入库业务

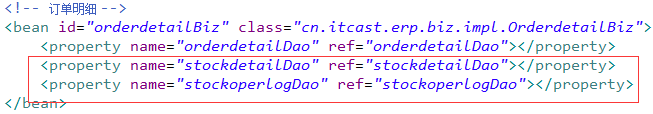
1. 更新订单明细表里的状态字段，改成1，表示已经入库了，同时要记录操作人、操作时间、入到哪个仓库
2. 变更商品库存数量。如果存在商品编号和仓库编号都一样的记录，则更新数量；否则插入一条新的记录，记录该商品在哪个仓库下的数量
3. 记录入库操作记录，添加商品仓库库存更新记录，即需要往STOCKOPERLOG表插入记录。
4. 检查是否需要更新订单的状态的判断。检查订单下所有的明细是否都已经入库了，如果都入库了则需要更新订单的状态为3（已入库），同时设置 入库时间、入库操作人
   1. 代码实现
      1. 后端
5. 添加引用：

我们要更新或增加商品仓库库存数量，要用到对应原数据访问，因此需要添加StockdetailDao的引用；要记录“商品仓库库存变更记录”，我们也要添加StockoperlogDao的引用

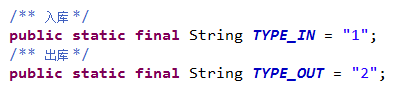
修改OrderdetailBiz，添加这两个dao的引用



修改资源文件application\_biz.xml:

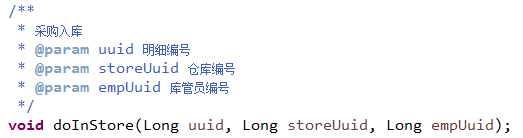


1. Storeoper实体类中添加测量定义：



1. IOrderdetailBiz添加doInStore方法，并在OrderdetailBiz中实现

接口方法：

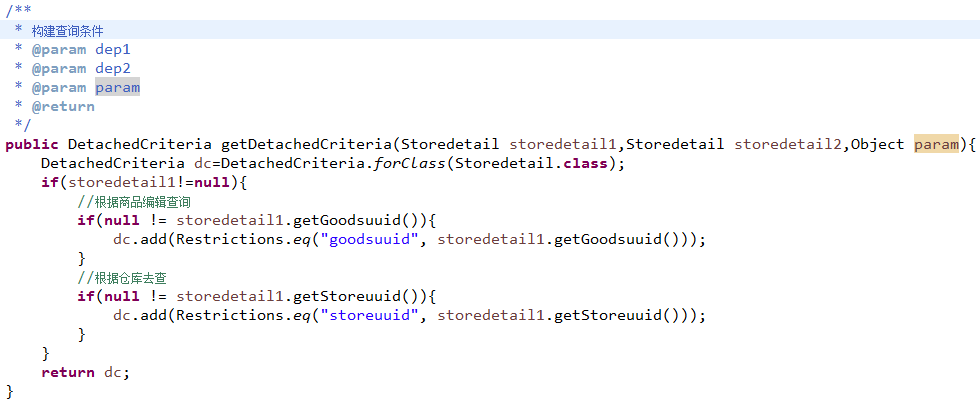


业务实现：

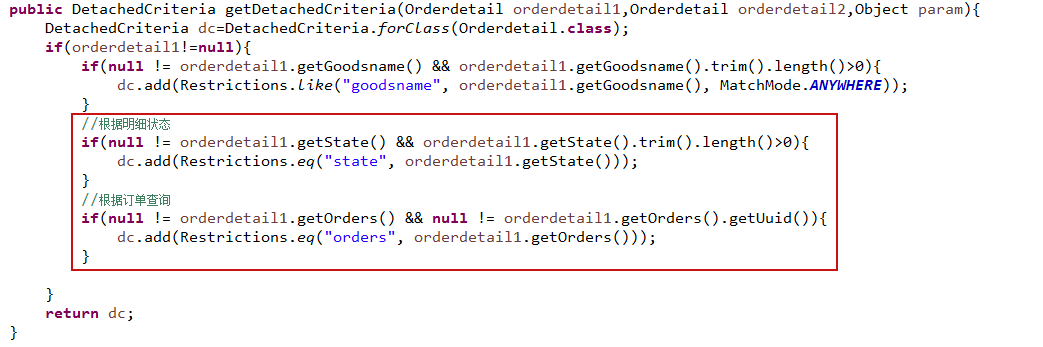




1. 修改StoredetailDao的getDetachedCriteria方法：

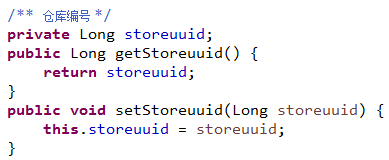


1. 在OrderdetailDao中添加：



1. 修改OrderdetailAction,

添加storeuuid：



加入doInStore方法，供页面入库ajax调用:



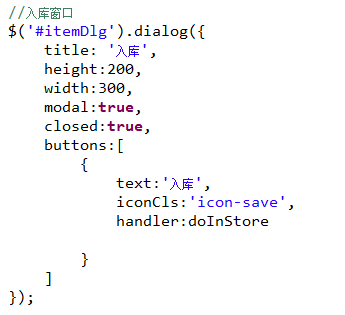
* + 1. 前端

1. 添加入库窗口，修改orders.html

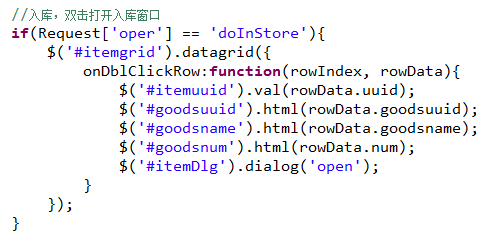
修改orders.js，添加入库窗口，只有在入库页面才显示



1. 入库窗口初始化：



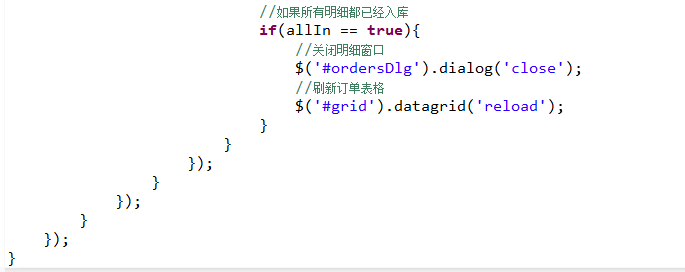
1. 动态添加双击行事件：



1. 提交入库请请求

添加doInStore方法：



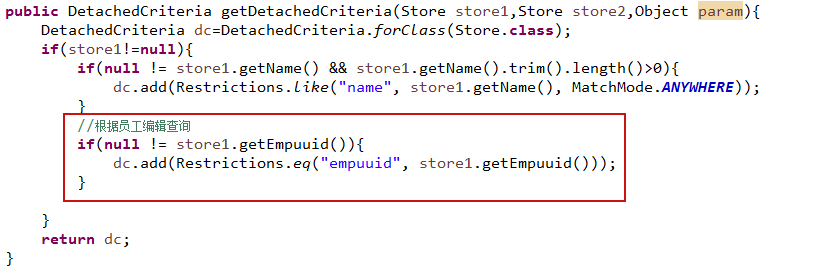


* 1. 显示当前登陆用户的仓库列表

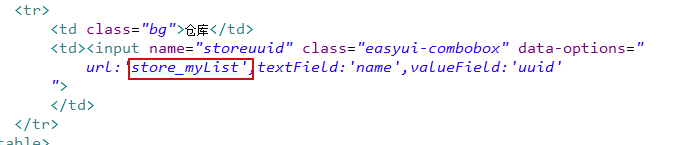
修改StoreAction，创建myList方法如下：



修改StoreDao，添加empuuid的查询条件



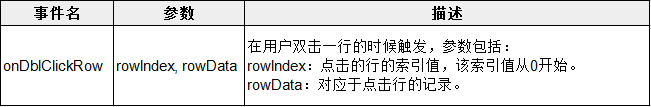
修改orders.html中入库窗口下的仓库：



1. 挂菜单

把昨天和今天所做的采购管理功能都挂上菜单

1. 总结:
   1. Easyui
      1. datagrid



* 1. 项目经验

1. 采购业务流程
2. 入库业务流程与实现