## 库存管理子系统



listStockUpDetail.action

listStockUpDetail.action

listStockUp.action

<action name="listStockUp " class="stockUpAction " method="list">

<result name="success"> / </result>

</action>

public List<StockUp> findAllStockUps();

<action name="listStockUpDetail " class="stockUpAction " method="listDetail">

<result name="success"> / </result>

</action>

public List<StockUpDetail> findStockUpById(int stockUpId);



两按钮通过判断备货单状态显示

finishDelivery.action

printDelivery.action

<action name=" printDelivery " class="stockUpAction " method=" print ">

<result name="success"> / </result>

</action>

public List<StockUpDetail> findStockUpById(int stockUpId);

<action name=" finishDelivery " class="stockUpAction " method=" finish ">

<result name="success"> / </result>

</action>

public void finishDelivery (int stockUpId)





|  |
| --- |
| Class:StockUpService |
| /\*返回所有备货单列表\*/  public List<StockUp> findAllStockUps();  字段包括：序号（id、不是数据库字段，从1开始递增）、备货单编号（stockUp .stockUpID）、订单编号（stockUp.orderID）、添加用户（user.userName）、添加时间（stockUp .addTime）、备货单状态（stockUp.state） |
| /\*根据“备货单编号”查找备货单信息\*/  public List<StockUpDetail> findStockUpById(int stockUpId);  字段包括：备货单编号（stockUp.stockUpID）、备货单状态（stockUp.state）、备货单添加用户（user.userName）、备货单添加时间（stockUp.addTime）、订单顾客姓名（user.userName）、订单下单时间（order. orderTime）、订单收货人（order. receiver）、订单收货地址（order. receiveAddress）、备货单明细序号（id、不是数据库字段，从1开始递增）、备货单明细编号（stockUpDetail. stockUpDetailID）、产品编号（stockUpDetail. productID）、产品名称（product. productName）、原厂编号（product. supplierID）、产品规格（product. productSize）、备货数量（product. stockUpAmount） |
| /\*根据“备货单编号”更新备货单状态为“已处理”，同时向应收账表（receivable）和应收账明细表（receivableDetail）添加一条记录\*/  public void finishDelivery (int stockUpId) |



listPurchase.action

listPurchaseDetail.action

listPurchaseDetail.action

<action name="listPurchase " class="purchaseAction " method="list">

<result name="success"> / </result>

</action>

public List<Purchase> findAllPurchases();

<action name="listPurchaseDetail " class=" purchaseAction " method="listDetail">

<result name="success"> / </result>

</action>

public List<PurchaseDetail> findPurchaseById(int purchaseId);



finishPurchase.action

<action name="finishPurchase" class="purchaseAction" method="finish">

<result name="success"> / </result>

</action>

public void finishPurchase (int purchaseId)



|  |
| --- |
| Class:PurchaseService |
| /\*返回所有进货通知单列表\*/  public List<Purchase> findAllPurchases();  字段包括：序号（id、不是数据库字段，从1开始递增）、进货通知单编号（purchase.purchase ID）、添加用户（user.userName）、添加时间（purchase.addTime）、进货通知单状态（purchase.state） |
| /\*根据“进货通知单编号”查找进货通知单信息\*/  public List<PurchaseDetail> findPurchaseById(int purchaseId);  字段包括：进货通知单编号（purchase. purchaseID）、进货通知单状态（purchase.state）、添加用户（user.userName）、添加时间（purchase. addTime）、序号（id、不是数据库字段，从1开始递增）、产品名称（product. productName）、原厂编号（product. supplierID）、产品规格（product. productSize）、进货数量（product. stockUpAmount） |
| /\*根据“进货通知单编号”更新进货通知单状态为“已处理”，同时根据findPurchaseById 中的产品编号更新库存（原库存＋进货库存）\*/  public void finishPurchase (int purchaseId) |



productDetail.action

listProduct.action

deleteProduct.action

productDetail.action

listUnsafeProduct.action

<action name="listProduct" class=" productAction " method="list">

<result name="success">/</result>

</action>

public List<Product> findAllProducts();

<action name=" listUnsafeProduct" class=" productAction " method="unsafe">

<result name="success">/</result>

</action>

public List<Product> findUnsafeProducts();

<action name="productDetaill" class=" productAction " method="detail">

<result name="success">/</result>

</action>

public Product findProductById(int productId);

<action name="deleteProduct" class=" productAction " method="delete">

<result name="success">/</result>

</action>

public void delete(int productId);





updateProduct.action

<action name="updateProduct" class=" productAction " method="update">

<result name="success">/</result>

</action>

public void update(Product product);



addAgainPurchase.action

productDetail.action

deleteProduct.action

<action name="addAgainPurchase" class=" againPurchaseAction " method="add">

<result name="success">/</result>

</action>

public void add(List<Product> unsafeProducts);



addProduct.action

addProduct.jsp

<action name="addProduct" class=" ProductAction " method="delete">

<result name="success">/</result>

</action>

public void add(Product product);

|  |
| --- |
| Class:ProductService |
| /\*返回所有产品列表\*/  public List<Product> findAllProducts();  字段包括：序号（id、不是数据库字段，从1开始递增）、产品编号（product. productID）、产品名称（product. productName）、销售价（product. salePrice）、采购价（product. purchasePrice）、库存量（product. stock）、安全库存（product. safeStock） |
| /\*返回所有库存量<安全库存量的产品列表\*/  public List<Product> findUnsafeProducts();  字段包括：序号（id、不是数据库字段，从1开始递增）、产品编号（product. productID）、产品名称（product. productName）、销售价（product. salePrice）、采购价（product. purchasePrice）、库存量（product. stock）、安全库存（product. safeStock） |
| /\*根据“产品编号”查找产品信息 \*/  public Product findProductById(int productId);  字段包括：产品编号（product.productId）、产品名称（product. productName）、产品规格（product.productSize）、原厂编号（product. supplierID）、销售价（product. salePrice）、采购价（product. purchasePrice）、库存量（product. stock）、安全库存（product. safeStock） |
| /\*根据“产品编号”删除该产品 \*/  public void delete(int productId); |
| /\*根据product的产品编号更新产品信息 \*/  public void update(Product product); |
| /\*根据findUnsafeProducts()方法的返回值向再订货通知单表（againPurchase）和其明细表（againPurchaseDetail）添加一条记录，再购数量（purchaseAmount）为安全库存量－库存量，再订货通知单状态（state）为“未处理”\*/  public void add(List<Product> unsafeProducts); |
| /\*向产品表（product）添加一条记录 \*/  public void add(Product product); |

|  |
| --- |
| Class:againPurchaseService |
| /\*根据findUnsafeProducts()方法的返回值向再订货通知单表（againPurchase）和其明细表（againPurchaseDetail）添加一条记录，再购数量（purchaseAmount）为安全库存量－库存量，再订货通知单状态（state）为“未处理”\*/  public void add(List<Product> unsafeProducts); |