

# 1. 代码获取

---

1. 后端代码获取，命令行输入：

```
git clone -b lab6 https://seecoder-gitlab-gitlab.seec.seecoder.cn/zhaoxr429/erp-backend.git
```

在现有项目的基础上新建lab6分支，将clone下来的项目覆盖你们的lab6分支，注意不要把clone下来的项目中的 `.git` 文件复制进去

2. 前端代码获取，命令行输入：

```
git clone https://seecoder-gitlab-gitlab.seec.seecoder.cn/zhaoxr429/erpfrontend.git
```

要求与后端项目一致，自行创建lab6分支

# 2. 后端代码说明

---

## 已实现的功能

### 进货

进货单制定 `PurchaseService.makePurchaseSheet`

根据状态获取进货单 `PurchaseService.getPurchaseSheetByState`

进货单审批 `PurchaseService.approval`

### 进货退货

制定进货退货单 `PurchaseReturnsService.makePurchaseReturnsSheet`

根据状态获取进货退货单 `PurchaseReturnsService.getPurchaseReturnsSheetByState`

进货退货单审批 `PurchaseReturnsService.approval`

### 商品出入库

制定商品入库单 `WarehouseService.productWarehousing`

制定商品出库单 `WarehouseService.productOutOfWarehouse`

审批入库单 `WarehouseService.approvalInputSheet`

审批出库单 `WarehouseService.approvalOutputSheet`

## 库存盘点

库存盘点 WarehouseService.warehouseCounting

## 其他

登录认证 UserService.login/register/auth

## 待实现的功能

### 销售

销售单制定 SaleService.makeSaleSheet

(可参考进货单制定的相关方法逻辑)

根据状态获取销售单 SaleService.getSaleSheetByState

(可参考根据状态获取进货单相关方法逻辑)

销售单审批 SaleService.approval

(可参考进货单审批相关方法逻辑)

获取某个销售人员某段时间内消费总金额最大的客户

SaleService.getMaxAmountCustomerOfSalesmanByTime

(考察对多个包同时操作)

### 库存查看

- 库存查看： 一个时间段内的入库数量合计
  - warehouseService.getWarehouseInputProductQuantityByTime
- 库存查看： 一个时间段内的出库数量合计
  - warehouseService.getWarehouseOutProductQuantityByTime
- 库存查看： 查询指定时间段内出/入库数量/金额/商品信息/分类信息
  - warehouseService.getWarehouseIODetailByTime

(考察对多个包同时操作)

### 库存盘点excel导出（自行建立相关方法）

库存盘点 excel导出

(考察对网络请求的理解)

### 销售退货（自行建立相关方法）

制定销售退货单

(可参考进货退货单制定的相关方法逻辑)

根据状态获取销售退货单

(可参考根据状态获取进货退货单相关方法逻辑)

销售退货单审批

(可参考进货退货单审批的相关方法逻辑)

(另:

1. 销售退货需要具体到每个商品退多少个，优惠券不退，价格按照单价和折扣来退；退的时候有一个问题，即退回仓库的哪一个批次 [ 因为销售的时候会卖出同一产品的不同批次 ] —— 你可以按照某种你喜欢的方式自动退回仓库，合理即可。举个例子：销售出货时出了50台1批次苹果（售价5000元）和50台2批次苹果（售价5200元），现在退了80台，那么可以按照售价由低到高的顺序逐一退回仓库，这里时退回了50台1批次苹果和30台2批次苹果。
2. 销售退货与销售要关联起来 )

## 客户（自行建立相关方法）

增加客户

删除客户

## 3. 前端说明

---

### 已实现的功能

#### 进货

PurchaseView.vue

使用到了组件PurchaseList.vue

#### 进货退货

PurchaseReturnView.vue

使用到了组件PurchaseReturnList.vue

#### 商品出入库

InventoryManagement.vue

使用到了组件PurchaseList.vue和SaleList.vue

#### 库存查看

InventoryView.vue

需要大家实现后端接口

## 库存盘点

InventoryCheck.vue

需要大家实现导出excel功能(后端也需要编写相应的逻辑)

## 销售

SaleView.vue

使用到了组件SaleList.vue

前端页面大致实现，需要大家实现后端接口以及添加 获取某个销售人员某段时间内消费总金额最大的客户

## 客户管理

CustomerView.vue

查看所有客户

## 未实现的功能

### 销售退货

可以参考进货退货，请大家自行实现

## 库存盘点excel导出

对接后端接口

## 客户管理

需要大家实现增加客户和删除客户

## 4. 其他注意点

---

1. 前后端流水线构建时，注意分支名的填 lab6 ，如果不行请填写 origin/lab6 。
2. 前端新建流水线时，请选择 node10 模板， node16 模板暂时有点问题，部署时将 package.json 改为 "node-sass": "^4.12.0" ，本地运行时，请根据本机 node 版本选择 node-sass 或者 sass 。
3. 后端 application.yml 中数据库配置（线上/本机），本机调试时请将 .sql 文件导入本地数据库，完成代码后，部署到devcloud上时也别忘了初始数据库和数据表。

## 5. lab6要求

---

1. 完成前后端代码未实现部分
2. 完成代码后请将本地test结果截图放入后端lab6根目录下
3. 作业ddl：2022/6/26 23:59:59

