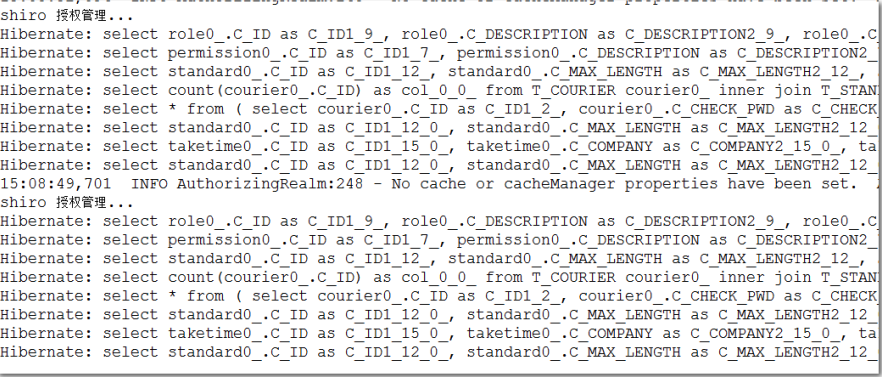
# 主要讲解点：

1. Spring整合缓存框架ehcache的运用（重点）
2. 配送页面及服务端代码的完成（重点）
3. 运输配送管理列表查询功能（重点）

# 以授权数据查询为例讲解缓存的作用

缓存可以提高查询性能，对同一批数据进行多次查询时，第一次查询走数据库，查询数据后，将数据保存在内存中，第二次以后查询可以直接从内存获取数据，而不需要和数据库进行交互

控制台抓包分析



每次访问一个需要被权限控制资源时，调用Realm的授权方法，根据当前用户查询角色（role）和权限（permission）信息，每次调用都会查询一次数据库

问题：为什么使用ehcache而不使用redis缓存

1. Shiro默认对ehcache的支持
2. 在后台管理系统中ehcache使用非常普遍

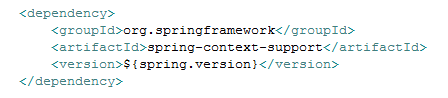
# 二、spring整合ehcache管理CacheManager

第一步：在common\_parent导入ehcache\_maven坐标

1）Ehcache开发包



2）Spring整合ehcache包 spring-context-support包



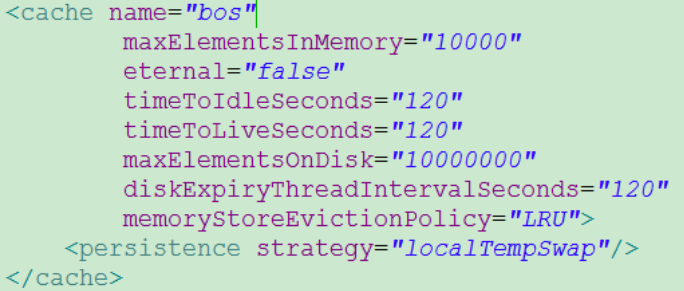
第二步：使用ehcache，导入ehcache.xml配置文件

1）解压ehcache-core.jar包，将ehcache-failsafe.xml 复制src/main/resources 改名为ehcache.xml



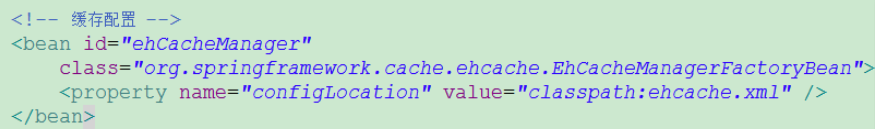


2）自定义缓存区



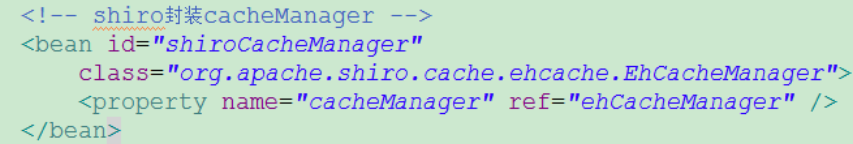
第三步：配置spring整合ehcache

将ehcacheManager交给spring管理



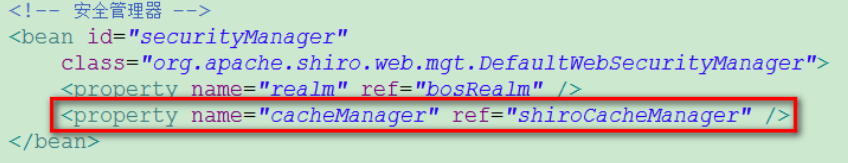
# 三、shiro整合ehcache实现授权数据缓存

第四步：配置shiro封装缓存管理器



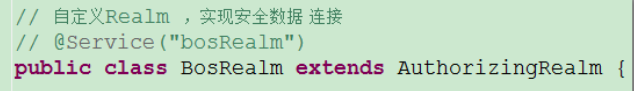
第五步：将shiro的缓存管理器，注入到安全管理器中

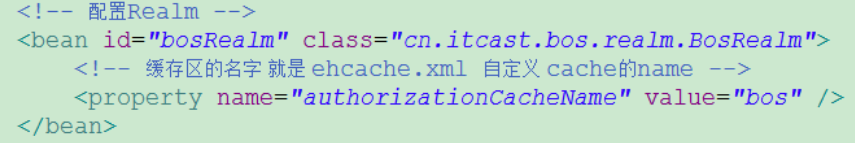
应用程序--->Subject--->SecurityManager--->Realm--->安全数据



第六步：对认证数据、授权数据哪些进行缓存?

配置Realm





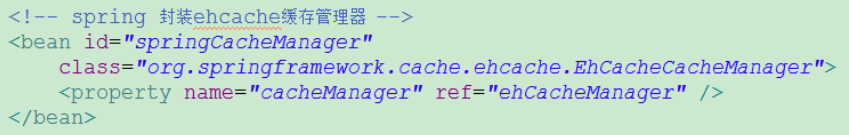
注意：使需要缓存对象，实现Serializable接口



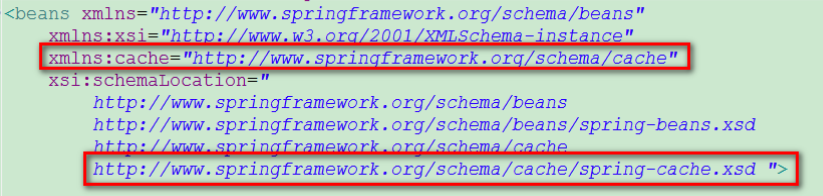
# 四、@Cacheable、@CacheEvict缓存注解的使用

参考spring整合ehcache管理CacheManager，前三步是一样的，接下来继续配置

第四步：配置spring缓存管理器，封装ehcache自带CacheManager

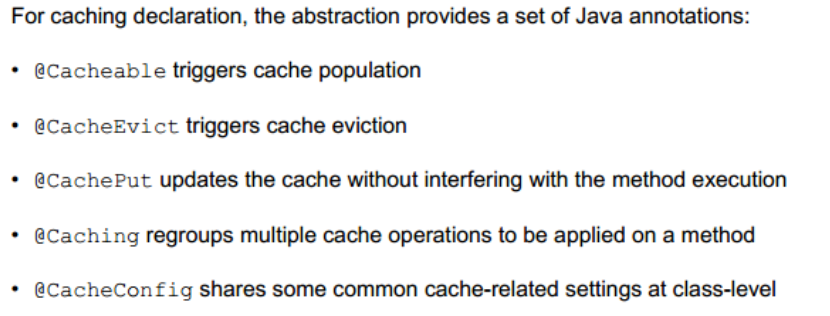


第五步：在applicationContext.xml中引入cache名称空间



第六步：激活spring缓存注解

当spring整合缓存框架后，提供了一套注解



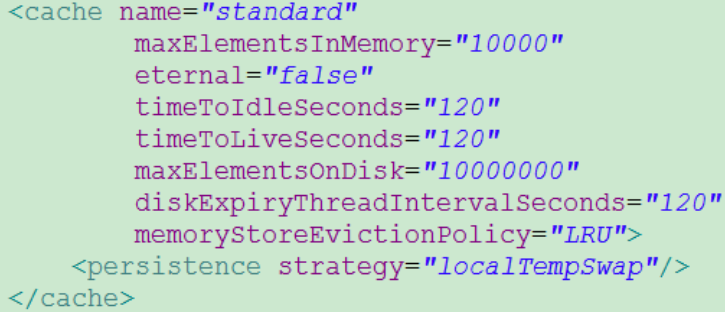


第七步：在被spring管理bean对象方法上使用@Cacheable、@CacheEvict

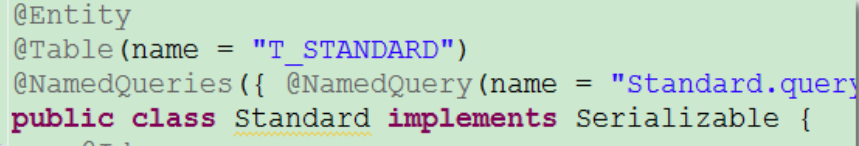
@Cacheable应用缓存区，对方法返回结果进行缓存，用于查询方法

@CacheEvict清除缓存区数据，用于增加、修改、删除方法

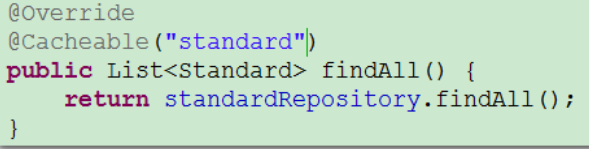
1）在ehcache.xml自定义缓存区

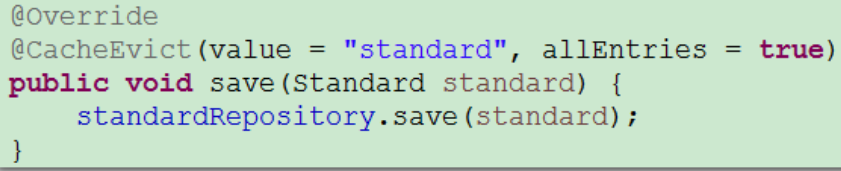


2）对实体类进行序列化



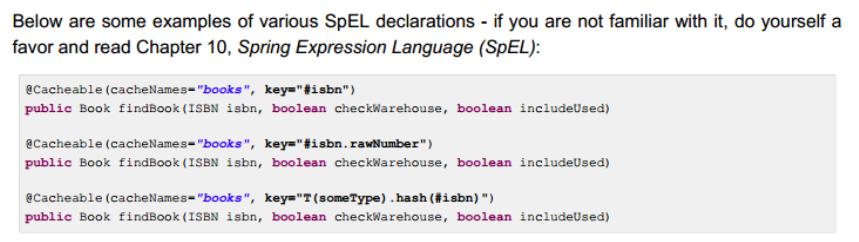
3）在StandardServiceImpl增加注解



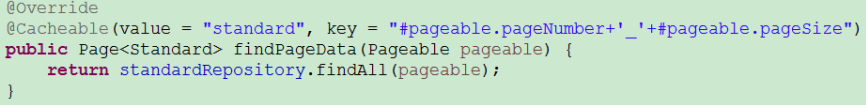


# 五、设置@Cacheable的key对有参数方法进行缓存

有参数的方法如何对结果数据进行缓存



针对数据在不同条件进行不同缓存，设置@Cacheable注解key属性



# 六、第十章主要内容概述

1、物流项目中转业务，在运单产生后，如何对物流运输进行管理（出入库操作、到达网点操作、开始配送、客户签收录入）

2、物流项目报表模块业务简介，主流服务器端报表技术（JasperResport）和客户端报表技术（Highcharts）

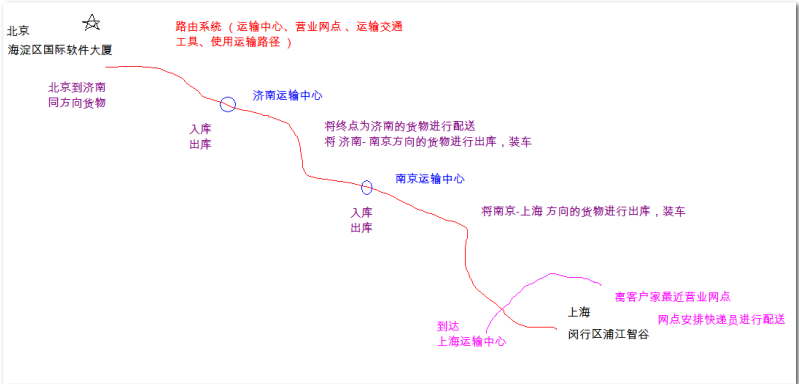
1. 项目Linux服务器部署操作（虚拟机安装、linux操作系统安装、主要软件安装配置使用）

# 七、中转业务模块需求分析

业务需求分析







分拣管理模块：管理中转配送过程中，到达每个运输中心 出入库操作

入库：进行运转中心仓库

出库：离开运转中心仓库

盘库：盘点仓库中货物

合包：将同方向小件货物，进行整合打包

拆包：将整合打包的货物拆解

# 八、导入运输配送管理页面和实体类

1、在T\_MENU表插入菜单信息

导入t\_menu\_transit.sql



2、导入页面

将transitinfo.html复制到pages/transit目录

将transit.json复制到data目录

将icon1复制到images（存放自定义地图标记）



3、导入中转配送相关实体类

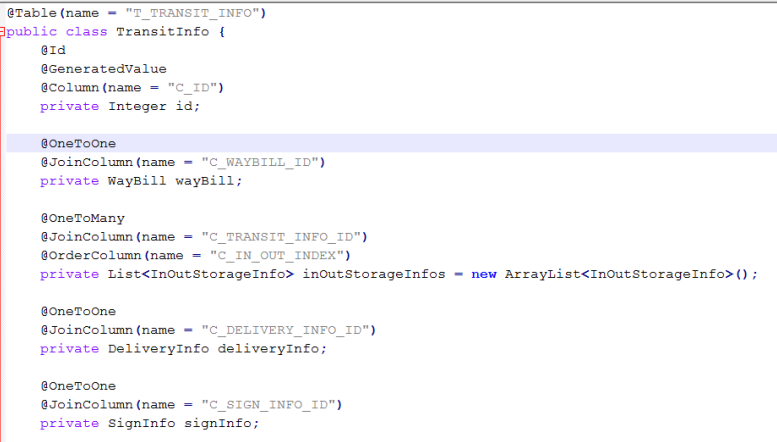
运单（快递单）对货物进行物流配送---WayBill

整个物流配送信息---TransitInfo

出入库信息（出入运输中心、出入网点）---InOutStorageInfo

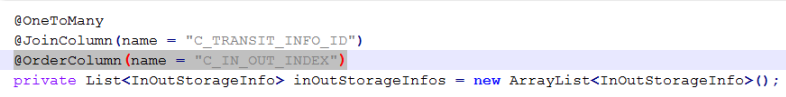
配送信息（当货物到达网点后，需要指定快递员进行配送）---DeliveryInfo

签收信息（客户签收货物信息）---SignInfo



注意点：一对多如果不关注集合对象顺序，使用Set，关注集合对象顺序使用List

保证顺序在数据表建立索引数据列，存放list集合下标@OrderColumn



# 九、开启中转配送(运单管理页面发起请求)

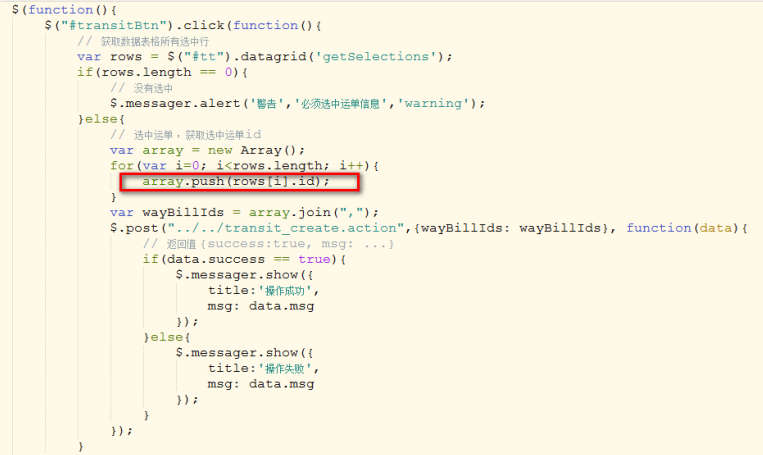
在waybill\_manage.html中添加按钮“中转配送”---TransitInfo对象产生

1. 页面加入按钮



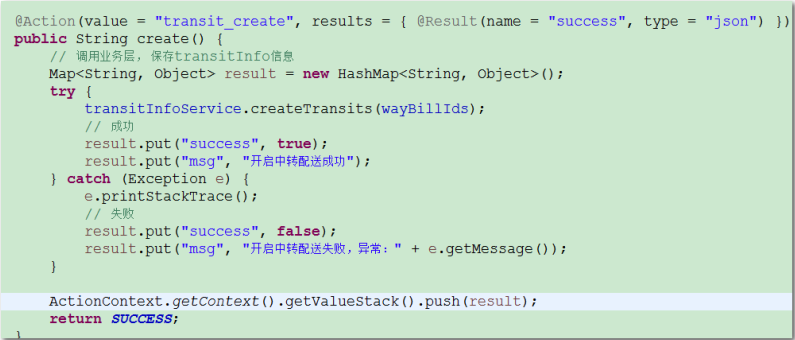
1. 通过js对按钮，添加点击事件

将选中运单，提交给服务器，开始进行中转配送（产生对应Transit对象）

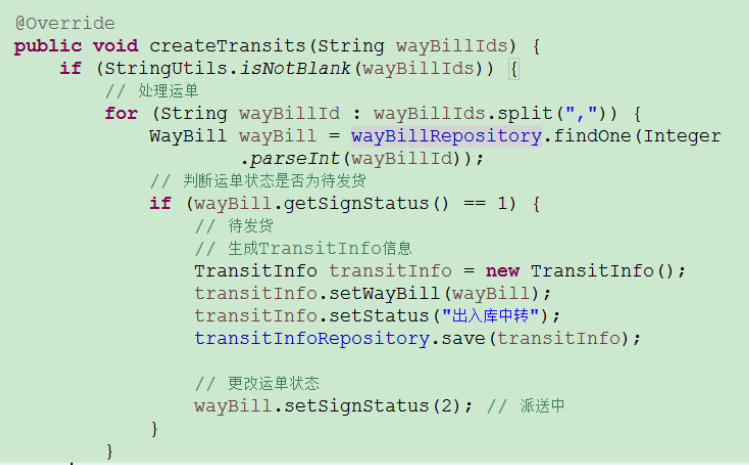


# 十、开启中转配送(服务器端代码实现)

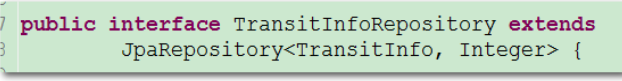
1. 编写TransitInfoAction，提供create方法



1. 编写业务层方法



5、编写DAO



# 十一、修复之前代码运单保存无状态Bug

运单录入时，可以根据运单号查询运单信息，进行保存

问题：运单不能任何时候都进行修改，只能在发货前进行修改

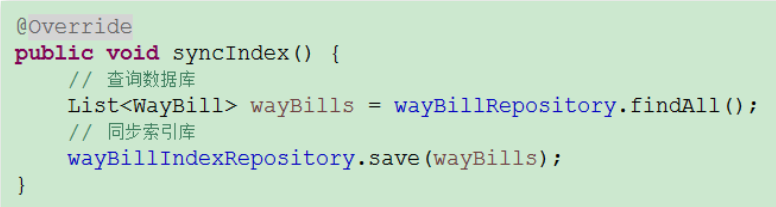


# 十二、修复运单数据库和索引库不一致Bug

为了数据库数据和索引库数据一致，可以编写一个程序实现一致，将所有运单查询出来，调用WayBillIndexRepository的save方法，更新索引库

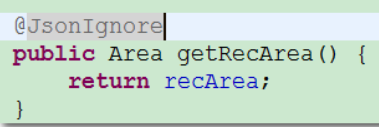
PS：可以启动定时任务Quartz每天定时同步数据库信息和索引库信息

在WayBillService添加方法



问题：在程序中保存数据到索引库，发生异常jackson数据转换异常，造成死循环栈溢出

解决@JsonIgnore注解



# 十三、运输配送管理显示百度地图运输路径

1、准备工作



运输路径，调用“路由系统”获取两点运输路径，运输路径经过沿途运输中心

1. 由路由系统规划路径，标记点（运输中心），调用百度地图API将点连线



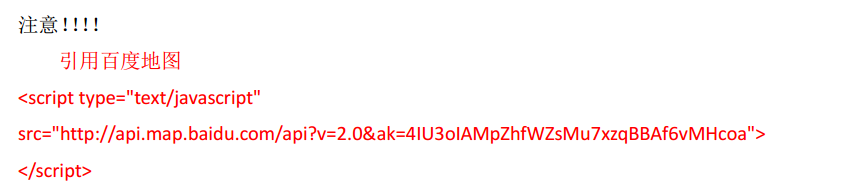
2）绘制两点驾车路线





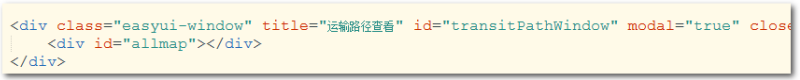


2、编辑transitinfo.html显示两点运输路径

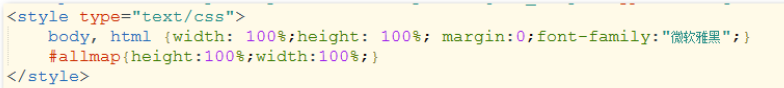


1. 首先根据起点和终点显示路径

在显示地图位置，添加div



简单设置css样式



显示地图



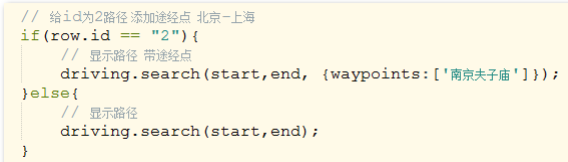




# 十四、运输配送管理显示百度地图运输路径（途经点）

2）带途经点的路径

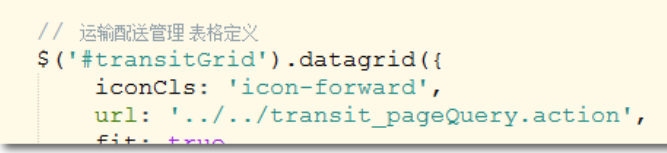
Driving.search（起点，终点，{waypoints【途经点】：[途经点信息]}）



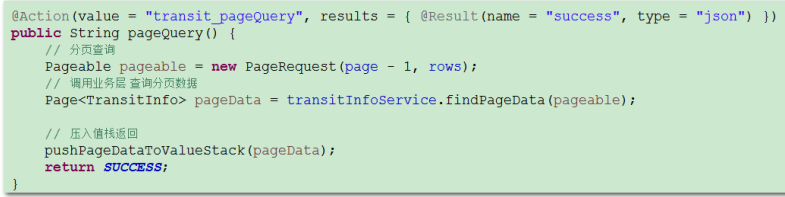


# 十五、运输配送管理列表查询功能

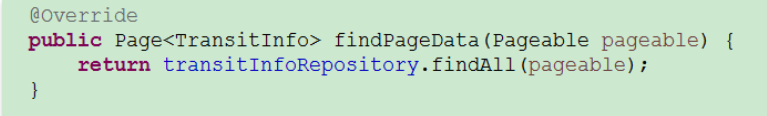
1. 修改transitinfo.html的datagrid的url地址



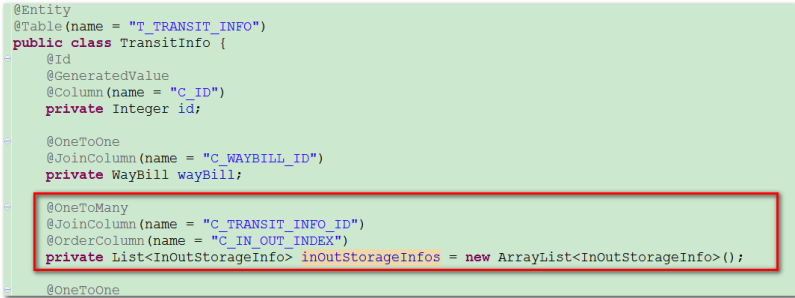
2、在TransitInfoAction添加pageQuery方法



1. 业务层



4、转换json关联集合问题



注意点：如果需要将集合数据转换为json返回，可以有以下几种策略

第一种：在业务层手动查询

第二种：配置立即抓取

第三种：OpenEntityManagerInView

