# 课堂笔记

# SSH综合练习

主要内容：客户订单管理

功能：

1. 客户信息查询
2. 客户信息添加(文件上传)
3. 客户订单信息查询 （ajax完成分布操作）(级联查询客户)
4. 客户信息删除 (hibernate的cascade)
5. 订单信息删除

使用技术 struts2 spring hibernate jquery bootstrap

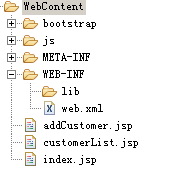
## 环境搭建

### 前端

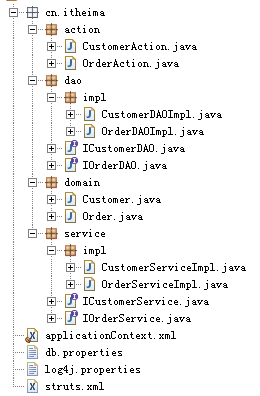
Jsp页面

jquery

bootstrap

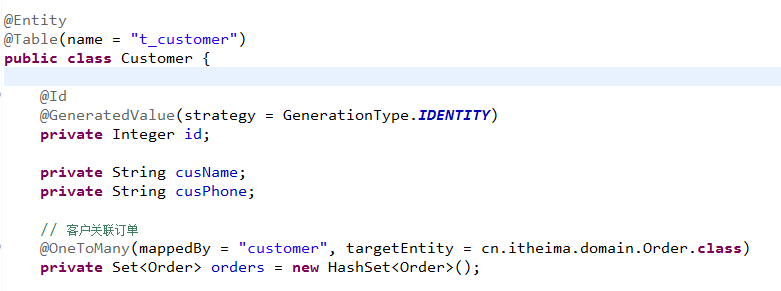


### 服务器端

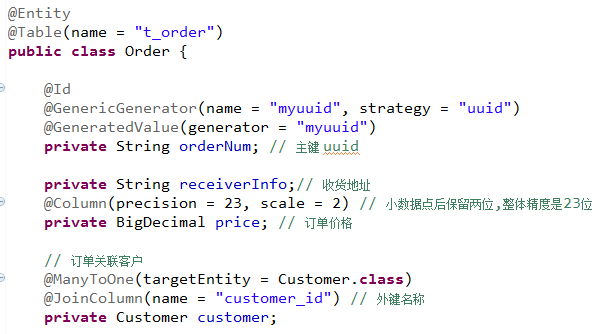


## JPA注解完成实体类配置映射

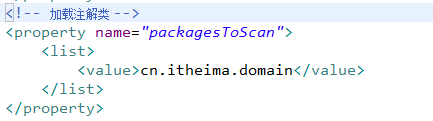
Customer



Order



在applicationContext.xml文件中配置扫描指定包下的jpa注解



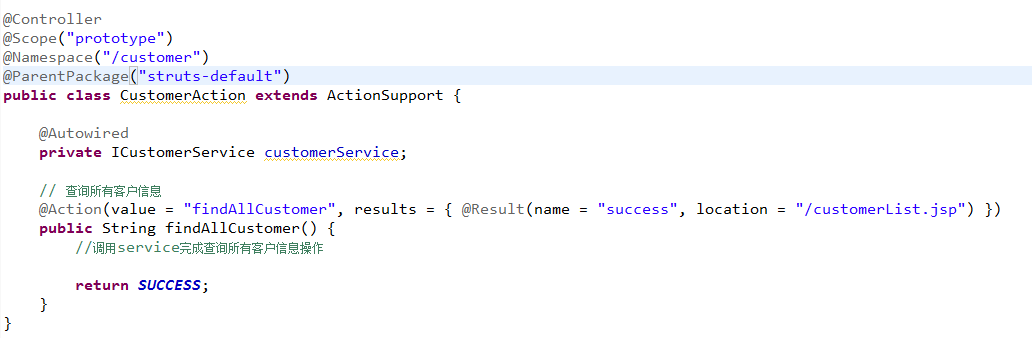
## 客户信息查询

当访问 index.jsp页面上，会有一个连接，点击连接会访问客户信息。



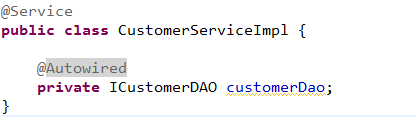
通过访问action--🡪service-🡪dao查询到客户信息，在customerList.jsp页面展示。

1. index.jsp页面连接创建
2. 在action中处理请求



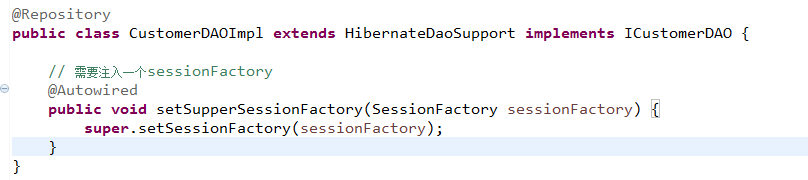
1. 定义CustomerService

使用@Service来声明service



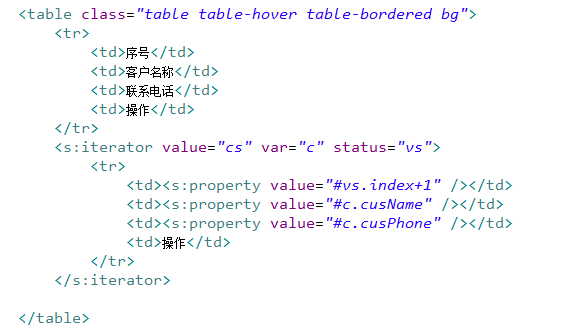
1. 定义CustomerDAO

使用@Repository来声明dao,并且要注入SessionFactory

注意：对于action,service,dao中使用注解要想生效，必须在applicationContext.xml文件中配置



1. 在customerList.jsp页面展示客户信息



## 客户信息添加

在customerList.jsp页面添加一个按钮，点击它跳转到addCustomer.jsp页面。



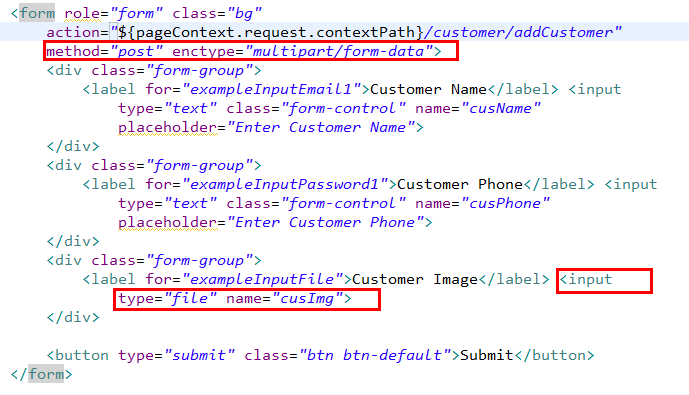
注意：客户信息添加时，要上传一个客户的照片.我们需要<input type=”file”>

问题：关于客户信息添加进需要一个照片，怎样处理?(文件上传)

### Struts2框架文件上传

文件上传时，浏览器端应该处理的事情:

1. 表单提交方式必须是POST
2. 必须使用<input type=”file”>来选择文件
3. 要设置表单encType=”multipart/form-data”

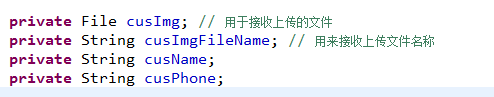


Struts2框架处理文件上传使用的是interceptor完成



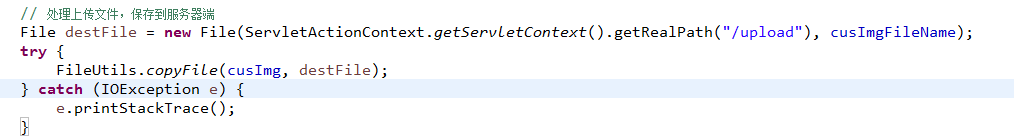
fileUpload这个interceptor就可以帮助我们完成文件上传操作

对于struts2完成文件上传非常简单，只需要在action中创建一个File来接收上传的文件就可以。

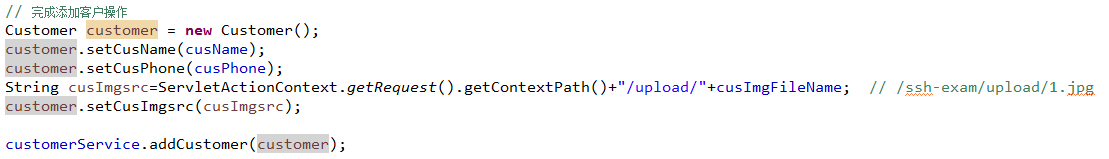


注意：要提供get/set方法就可以得到上传的信息及其它信息

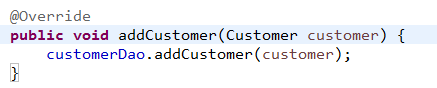
以下完成文件 上传



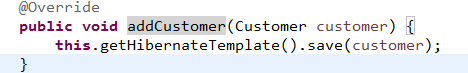
Action中完成Customer数据封装，调用service完成添加操作



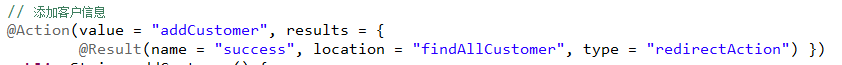
Service中调用dao完成操作



Dao中通过hibernateTemplate完成操作

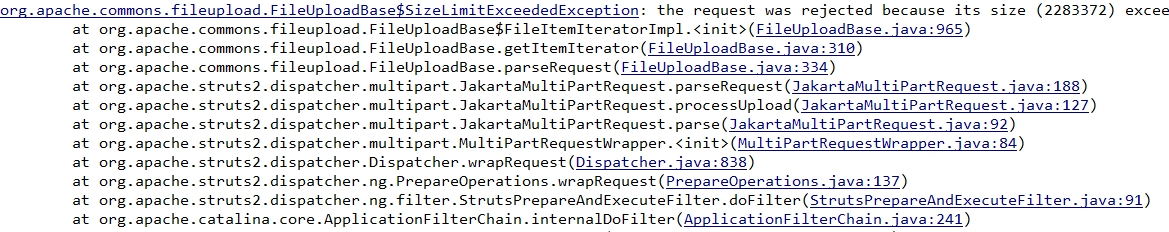


添加完成后，在跳转到customerList.jsp页面来查看

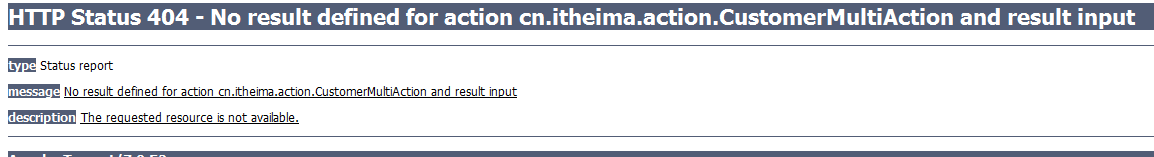


Type=”redirectAction” 相当于是重定向到一个action

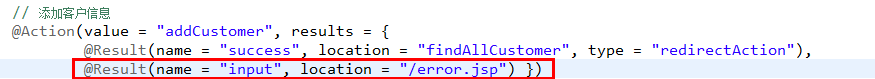
Location=”findAllCustomer”值就是action的名称



错误原因是上传文件的大小超过了允许最大上传大小。



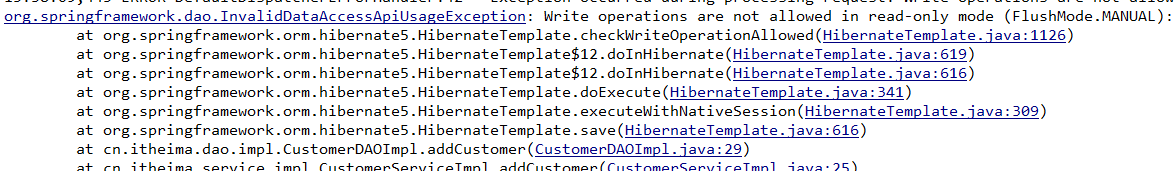
在浏览器端提示是没有input视图。



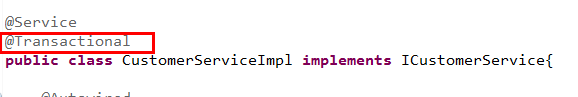
对于struts2文件上传，默认大小是在default.properties文件中定义的。

默认大小

我们可以在struts.xml文件中修改大小



现在错误的原因是事务控制



## 客户信息删除

在页面上添加一个删除连接

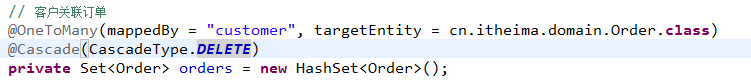


向服务器发送请求



服务器操作

注意：在删除客户时，要将客户关联的订单信息也删除。

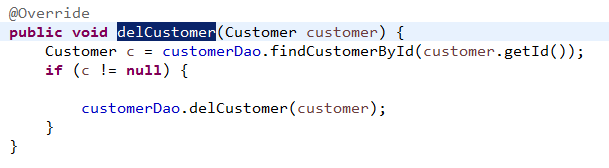


Action操作

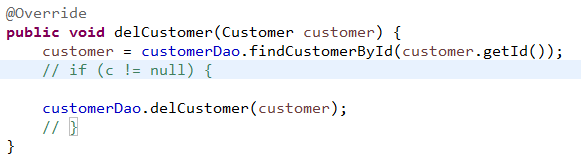


使用模型驱动来完成id值封装到customer中

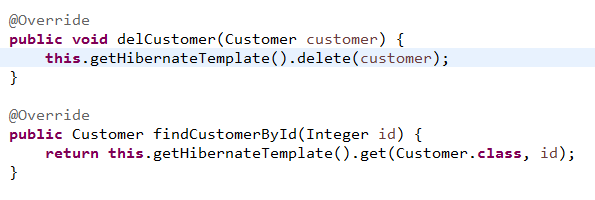
Service操作



上面的代码有问题，在一个session中具有两个相同的oid的customer对象



Dao操作

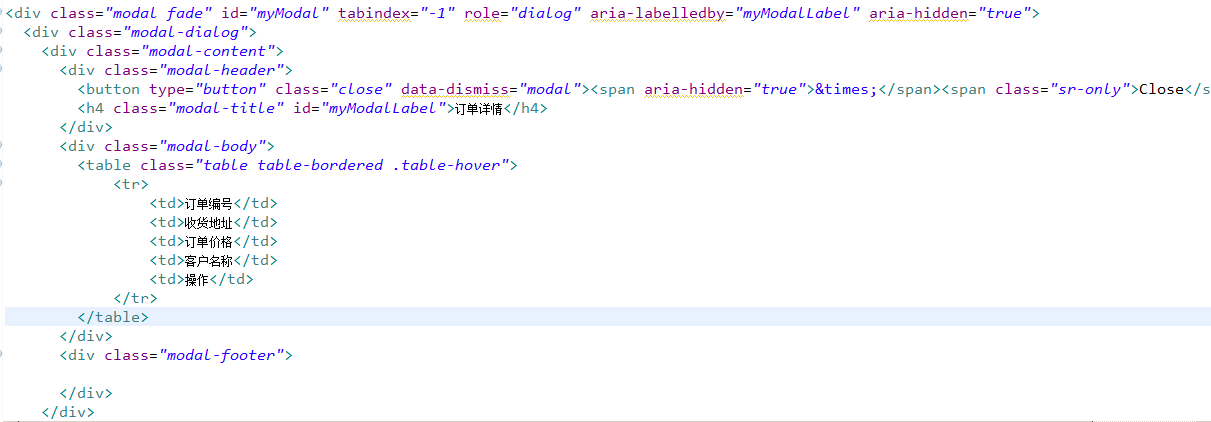


## 订单信息查询

在页面点击订单详情需要弹出一个模态框来展示订单信息。

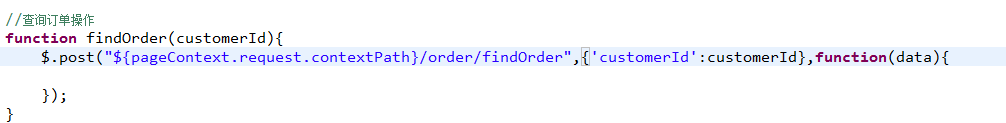
1. 要弹出模态框



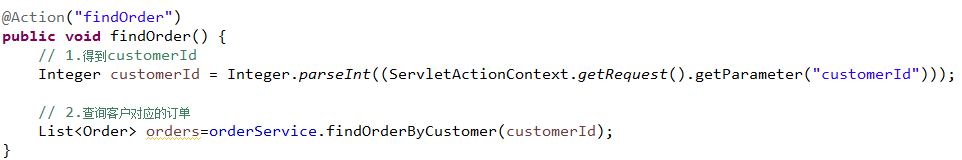


1. 向服务器发送请求，来查询客户的订单信息

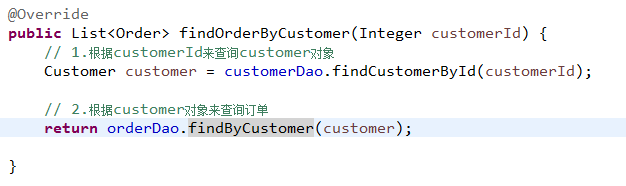
我们通过ajax操作来完成查询订单信息操作，将信息添加到模态框的table中。



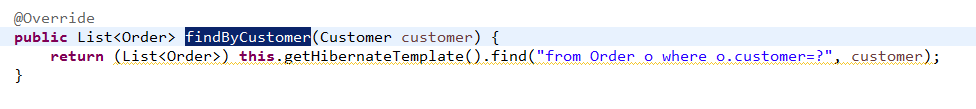
Action操作



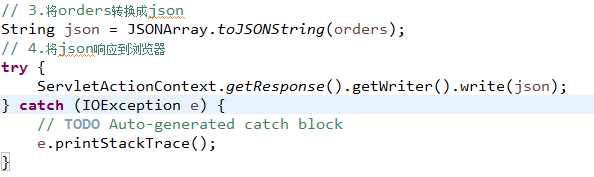
Service操作



Dao操作

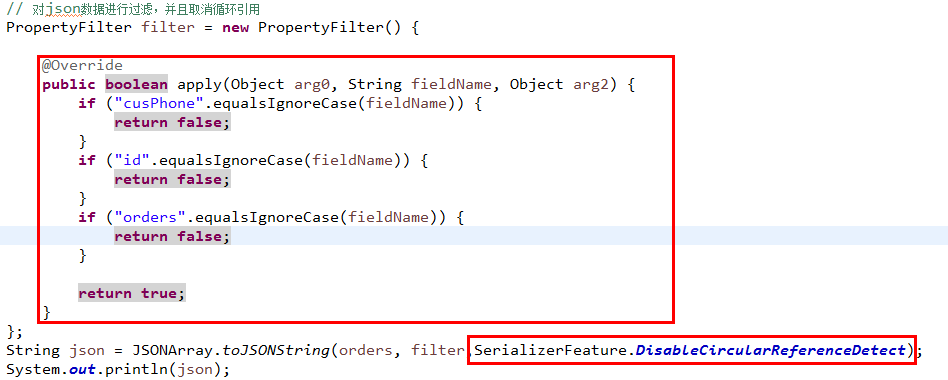


Action完成查询List<order>后操作



Json数据

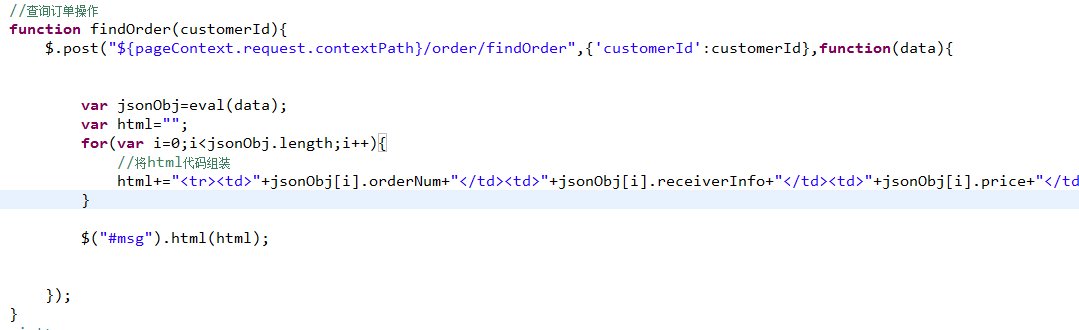
[{"customer":{"cusName":"fox","cusPhone":"13333333333","id":2,"orders":[{"$ref":"$[0]"},{"customer":{"$ref":"$[0].customer"},"orderNum":"def456","price":3000.00,"receiverInfo":"sh"}]},"orderNum":"abc123","price":2000.00,"receiverInfo":"bj"},{"$ref":"$[0].customer.orders[1]"}]



处理后数据

[{"customer":{"cusName":"fox"},"orderNum":"abc123","price":2000.00,"receiverInfo":"bj"},{"customer":{"cusName":"fox"},"orderNum":"def456","price":3000.00,"receiverInfo":"sh"}]

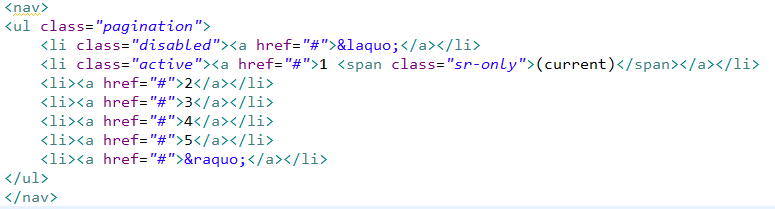
前台展示数据



## 订单信息查询(分页)

1. 在页面上添加分页组件

使用bootstrap提供的分页组件



1. 分析分页显示订单信息

浏览器端传递1.当前页码 2.每页条数

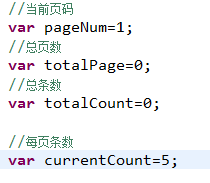
服务器端:

1. 新创建一个PageBean它封装了以下信息
   1. 当前页码
   2. 每一页条数
   3. 总条数
   4. 总页数
   5. 当前页的数据Lists
2. 对于HibernateTemplate如何完成分页

创建PageBean

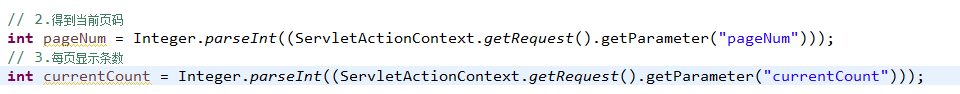


修改页面上提交信息



$.post("${pageContext.request.contextPath}/order/findOrder",{'customerId':customerId,"pageNum":pageNum,"currentCount":currentCount},**function**(data){}

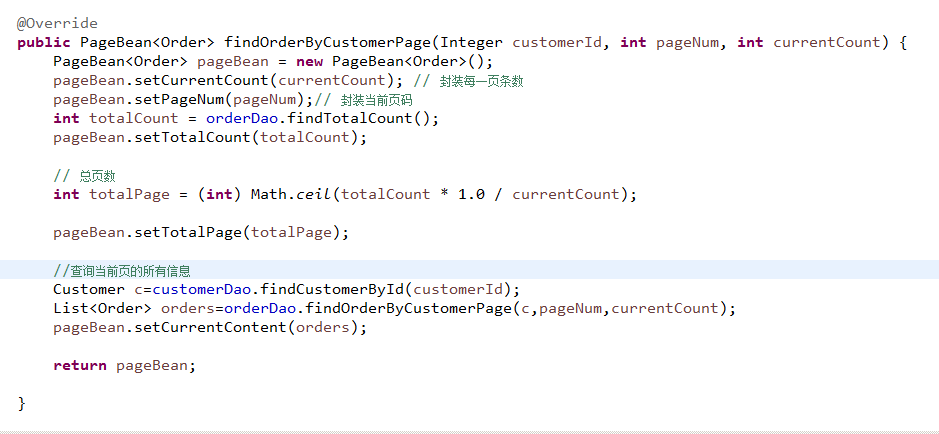
Action中修改



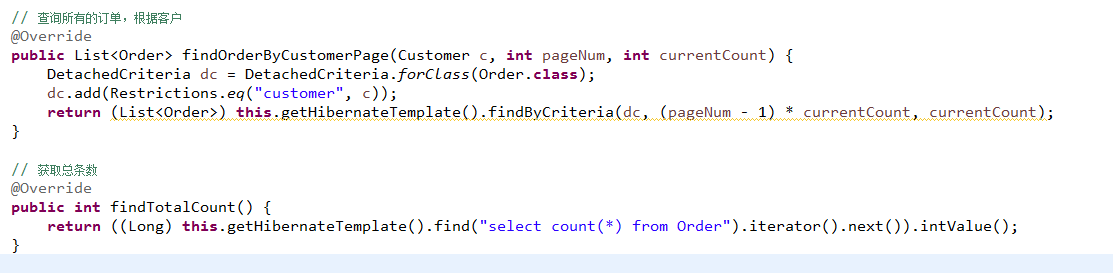
在action中调用service中的分页查询订单操作



Service中操作



Dao中操作

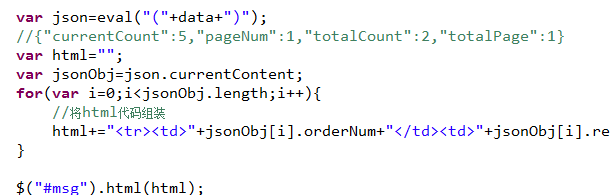


这时响应到浏览器的数据

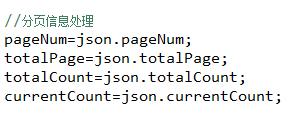
{"currentContent":[{"customer":{"cusName":"fox"},"orderNum":"abc123","price":2000.00,"receiverInfo":"bj"},{"customer":{"cusName":"fox"},"orderNum":"def456","price":3000.00,"receiverInfo":"sh"}],"currentCount":5,"pageNum":1,"totalCount":2,"totalPage":1}

处理方案

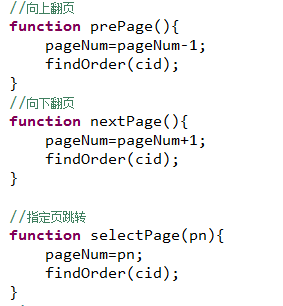
1. 展示数据时不是直接从json中获取，在向下一层查询



1. 关于分页条的处理







## 订单信息删除

课后自己完成，根据id来删除订单。不需要进行级联