

CD Shop 详细设计规约说明书

ID: CDS-DDD

目录

目录

- 1. 引言 5
 - 1.1. 编写目的 5
 - 1.2. 详细设计背景和依据 5
- 2. 系统软件结构 5
 - 2.1. 系统总体架构设计 5
 - 2.1.1. 基础层 6
 - 2.1.2. 数据访问层 6
 - 2.1.3. 服务层(业务层)..... 7
 - 2.1.4. 视图层 7
- 3. 程序设计说明 7
 - 3.1 ShowAction 类 7
 - 3.1.1. 功能 7
 - 3.1.2. 性能 8
 - 3.1.3. 输入项（输入接口） 8
 - 3.1.4. 输出项（输出接口） 9
 - 3.1.5. 属性详细描述 9
 - 3.1.6. 类内、类外方法调用状态图 10
 - 3.2 SaleAction 类 13
 - 3.2.1. 功能 13
 - 3.2.2. 性能 13
 - 3.2.3. 输入项（输入接口） 13
 - 3.2.4. 输出项（输出接口） 14
 - 3.2.5. 属性详细描述 15

3.2.6. 方法详细描述及实现流程	16
3.2.7. 类内、类外方法调用状态图	16
3.3 AdminAction 类	19
3.3.1. 功能	19
3.3.2. 性能	19
3.3.3. 输入项（输入接口）	19
3.3.4. 输出项（输出接口）	21
3.3.5. 属性详细描述	22
3.3.6. 方法详细描述及实现流程	23
3.3.7. 类内、类外方法调用状态图	23
3.4 所有的 model 类	28
3.4.1. 功能	28
3.4.2. 性能	28
3.4.3. 输入项（输入接口）	28
3.4.4. 输出项（输出接口）	29
3.4.6. 方法详细描述及实现流程	29
3.4.7. 类内、类外方法调用状态图	29
3.5 BaseDao 类及 IBaseDao	29
3.5.1. 功能	29
3.5.2. 性能	30
3.5.3. 输入项（输入接口）	30
3.5.4. 输出项（输出接口）	31
3.5.5. 属性详细描述	31

- 3.5.6. 方法详细描述及实现流程 32
 - 3.5.7. 类内、类外方法调用状态图 32
- 3.6 Service 类 33
 - 3.6.1. 功能 33
 - 3.6.2. 性能 34
 - 3.6.3. 输入项（输入接口） 34
 - 3.6.4. 输出项（输出接口） 35
 - 3.6.5. 属性详细描述 35
 - 3.6.6. 方法详细描述及实现流程 36
 - 3.6.7. 类内、类外方法调用状态图 36

1. 引言

1.1. 编写目的

本文档旨在说明软件系统各个层次中的每个程序（每个模块或子程序）和系统软件结构的设计考虑，为程序员编码提供详细依据。本文案的重点是描述模块的执行流程。方便程序员进一步考虑每个类和模块的具体实现思路，并提高调试和测试的执行效率。

本文档的目标读者为本软件的开发人员和其他相关人员。

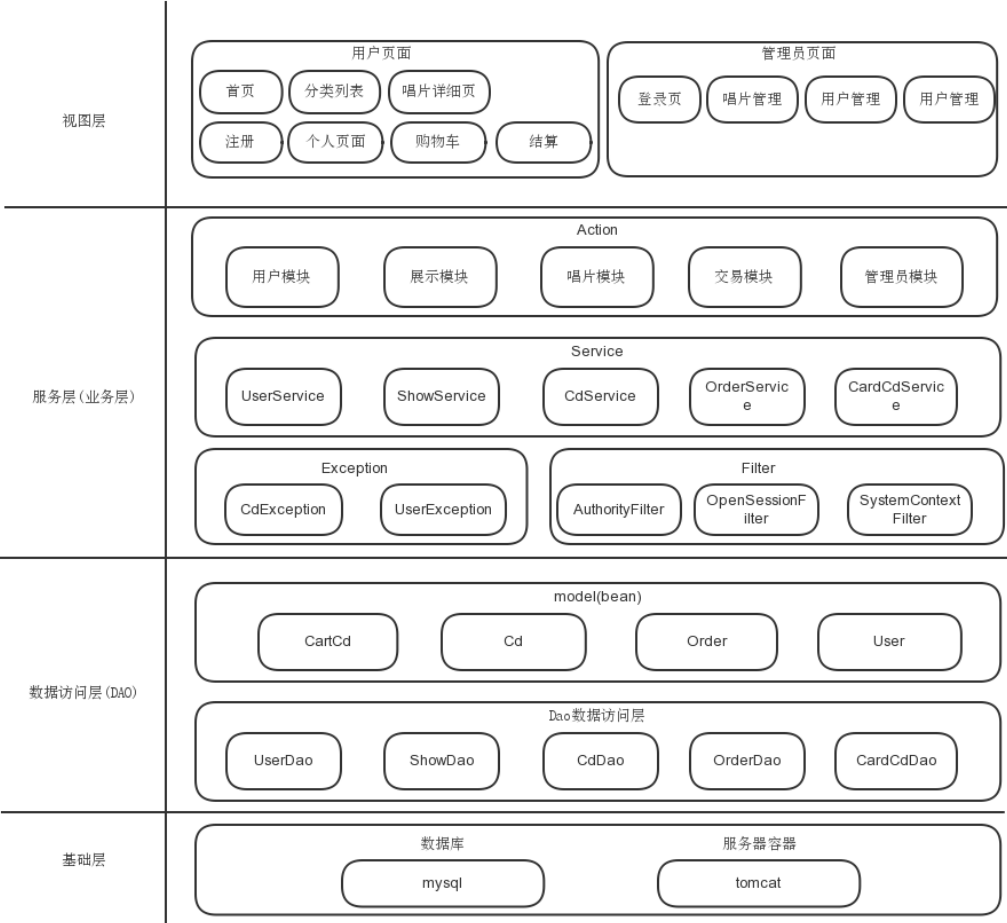
1.2. 详细设计背景和依据

本文档依据概要设计规约（CDS-HLD）编写，参考《计算机软件产品开发文件编制指南（GB8567-88）》。

2. 系统软件结构

2.1. 系统总体架构设计

以概要设计规约为依据，有如下架构设计图



整个系统划分为四层，分别是视图层，服务层，数据访问层，数据库层
整个系统运用了 j2e 开发框架 Struts2+Spring+Hibernate(SSH)为基础进行开发。以最大化简化开发量。

2.1.1. 基础层

本系统为 B/S 架构的网络应用。其系统部署的服务器要求具有 java 虚拟机。部署的系统为 linux/unix 或 windows 系统。所部署的服务器要求具有 mysql 数据库，及 tomcat 的服务器容器。

2.1.2. 数据访问层

该层为对数据库访问的中间层，用于连接服务层(业务层)与数据库间的 DAO(Data Access Object)。其中的 model 是数据访问中的实体(java bean)，其中的实体类(bean)应与数据库中的表(table)一一对应。故数据库中的每一个表都在 model 中有一个实体。而 DAO 数据访问层为操作数据库映射的实体类(bean)对数据库进行 CRUD(增删差改)操作。该层中运

用了数据访问中间件 **Hibernate**，在 **Hibernate** 中间件的作用下大大简化了对数据库的操作。再依据 **Spring** 对 **Hibernate** 的注入，将 **Hibernate** 的操作与服务层实现结合。

2.1.3. 服务层(业务层)

该层为整个系统的业务逻辑部分，是最核心的部分。该层负责处理浏览器传送过来的所有请求，并决定要将什么内容响应给浏览器。该层依据业务逻辑分为用户模块，展示模块，唱片模块，交易模块，管理员模块 5 个主要部分。依据功能架构可分为处理业务的 **Action** 部分，负责业务与 **DAO** 层调用的 **Service** 部分，处理异常的 **Exception** 层，及 **struts2** 实现部分功能的过滤器 **Filter** 部分。

2.1.4. 视图层

该层为整个系统与外界交互的部分。在该应用中，视图 (**View**) 由 **Java Server Page(JSP)** 承担。生成视图的代码则可能是一个 **servlet** 的一部分，特别是在客户端服务端交互的时候。视图层分为 2 类，一类为用户服务，一类为唱片系统管理的管理员服务。两类页面覆盖系统的所有功能。

3. 程序设计说明

应用采用了 **Struts2+Spring+Hibernate(SSH)** 的开发框架，故在业务层中，应结合 **Struts2** 与 **Spring** 的特点，将功能部署于 **Struts2** 的 **action** 类的方法中，并利用 **Spring** 实现注入。

3.1 ShowAction 类

3.1.1. 功能

该类主要实现与唱片有关的页面的展示及逻辑控制。

3.1.1.1. 主页显示

3.1.1.2. 唱片详情

3.1.1.3. 购物车页

3.1.1.4. 进入商品页

3.1.1.5. 登录注册注销

3.1.2. 性能

3.1.3. 输入项（输入接口）

在 `strut2` 中 `setter` 方法由容器获取得到对类中的私有变量进行设置。

3.1.3.1. 无

3.1.3.2. 唱片的 id

由容器调用 `public void setCid(int cid)` 设置

3.1.3.3. 删除唱片的 id，更改唱片的 id

3.1.3.4. 用户提交的商品分类查询条件

`Searchkey`, `gid`, `cid`

3.1.3.5. 用户输入的用户名，密码，注册的用户名，密码，电话，邮箱(传真)

`getRequest().getParameter(“string”)`
或者管理员输入的用户名及密码

3.1.4. 输出项（输出接口）

在网络应用中，将输出的内容加入 context 对象中，为键值对，在页面中调用。

3.1.4.1. 放入最新专辑的列表和热门专辑的列表

`Context().put("newlist")`
`Context().put("hotlist")`

3.1.4.2. 放入唱片的相关信息

`Context().put("cd")`
`Context().put("relatedlist")`

3.1.4.3. 放入购物车中已经更改过的信息

`cart.add(cdId)`

3.1.4.4. 更具提交分类条件筛选过后的商品信息

`Context().put("cdlist")`

3.1.4.5. 用户添加或更改的结果

根据结果进行链接跳转。

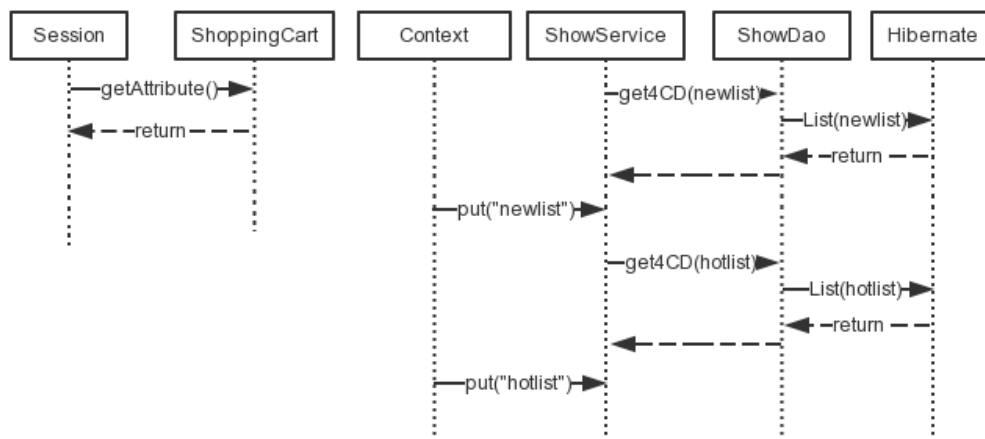
3.1.5. 属性详细描述

/**对用户操作的 Service**/

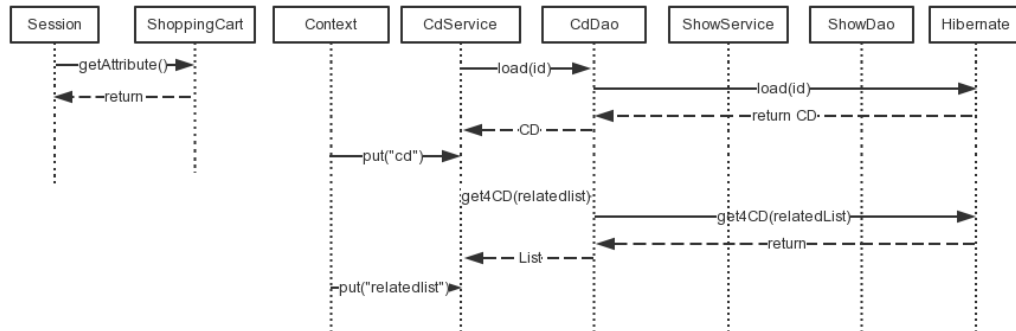
```
private IUserService userService;  
/**对界面操作的 Service**/  
private IShowService showService;  
/**对唱片操作的 Service**/  
private ICdService cdService;  
/**查询的关键字**/  
private String searchKey;  
/**查询的唱片的 id**/  
private int id;
```

3.1.6. 类内、类外方法调用状态图

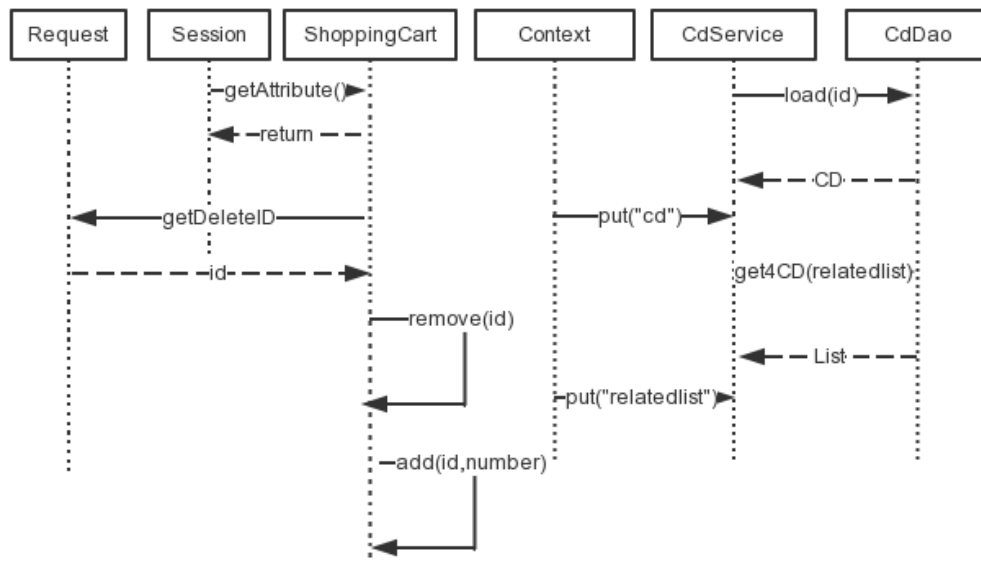
3.1.6.1. 主页显示



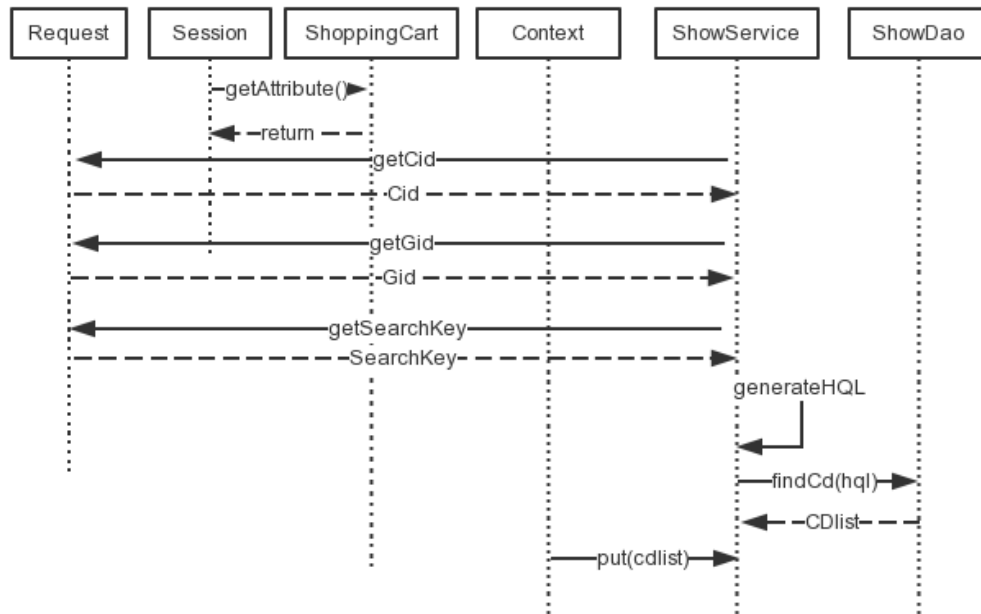
3.1.6.2. 唱片详情



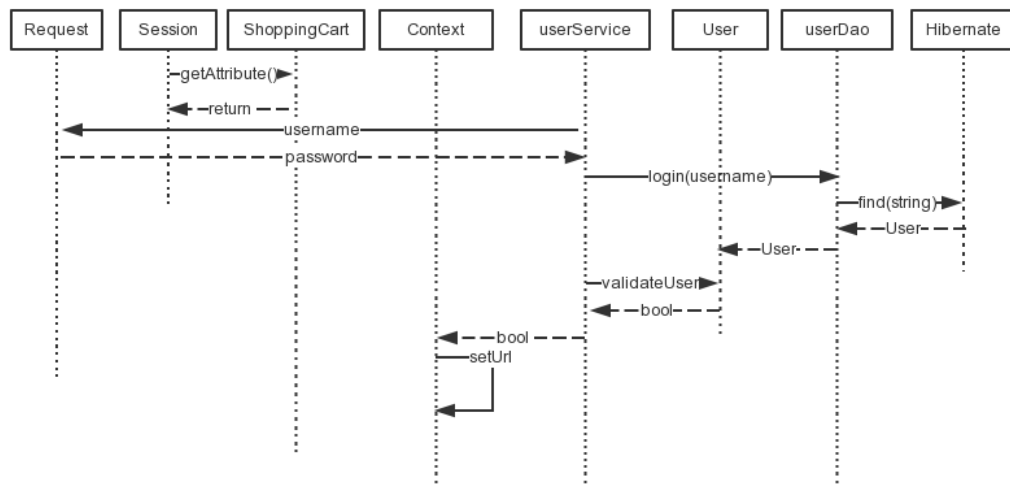
3.1.6.3. 购物车页



3.1.6.4. 进入商品页



3.1.6.5. 登录注册注销



3.2 SaleAction 类

3.2.1. 功能

该类主要控制订单信息，购物车结算及付款的相关逻辑

3.2.1.1. 查看当前所有订单信息

3.2.1.2. 改变密码

3.2.1.3. 修改送货的相关信息

3.2.1.4. 查询订单的信息

3.2.1.5. 确认支付

3.2.2. 性能

3.2.3. 输入项（输入接口）

在 struts2 中 setter 方法由容器获取得到对类中的私有变量进行设置。

3.2.3.1. 无

3.2.3.2. 用户输入的密码字符串

```
/**更改的密码**/  
    password1  
/**更改的密码再次确认**/  
    password2  
getRequest().getParameter( "string" )
```

3.2.3.3. 送货相关信息

```
recipient/**姓名**/  
address/**地址**/  
email/**邮箱**/  
phone/**电话**/  
fax/**传真**/  
getRequest().getParameter( "string" )
```

3.2.3.4. 查询的订单号的 id

```
public void setOrderId(int orderId)
```

3.2.3.5. 购物车的所有信息

```
session.getAttribute()
```

3.2.4. 输出项（输出接口）

在网络应用中，将输出的内容加入 context 对象中，为键值对，在页面中调用。

3.2.4.1. 放入当前用户已有的订单信息，热门列表

```
Context().put("orderList")  
Context().put("hotlist")
```

3.2.4.2. 根据密码的比较结果进行连接跳转

3.2.4.3. 跳转到当前页面

3.2.4.4. 根据订单 id 得到的订单列表

```
Context().put("cartCdList")
```

3.2.4.5.根据支付结果进行链接跳转

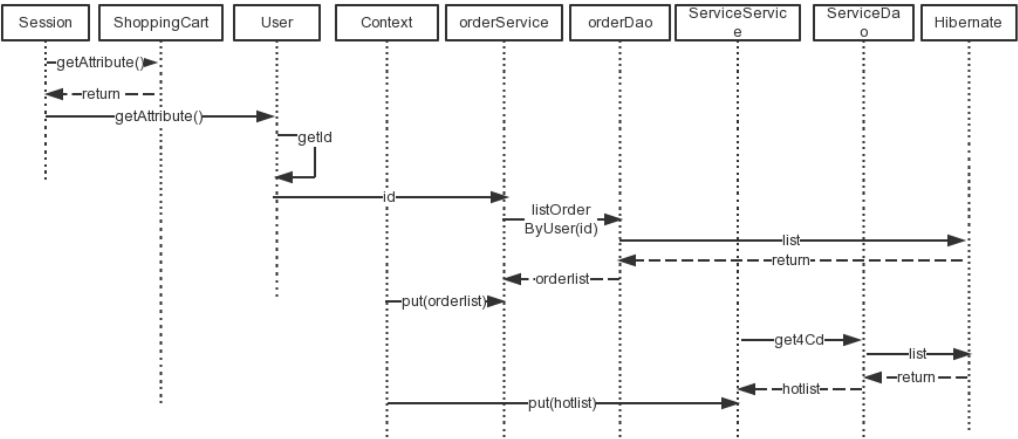
3.2.5. 属性详细描述

```
/**对应订单的 Service**/  
    private IOrderService orderService;  
/**对用户操作的 Service**/  
    private IUserService userService;  
/**对购物车操作的 Service**/  
    private ICartService cartService;  
/**对唱片操作的 Service**/  
    private ICdService cdService;  
/**查询的订单的 id**/  
    private int orderId;
```

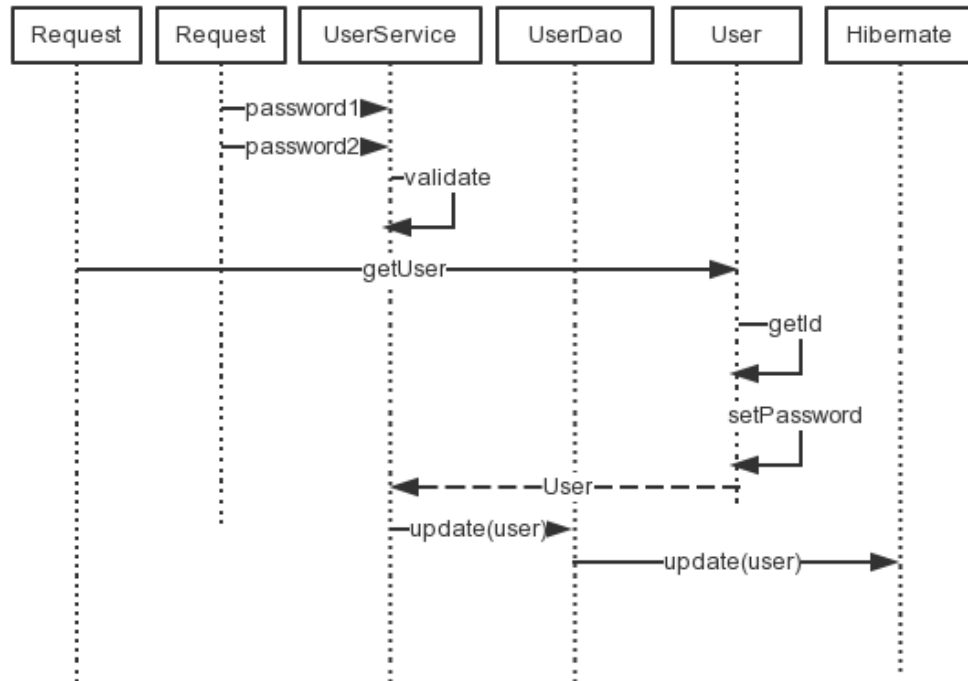
3.2.6. 方法详细描述及实现流程

3.2.7. 类内、类外方法调用状态图

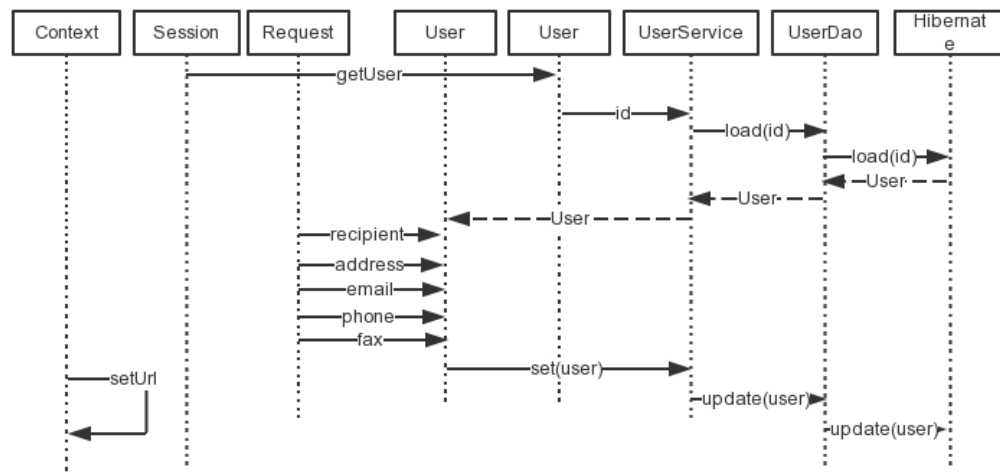
3.2.7.1. 查看当前所有订单信息



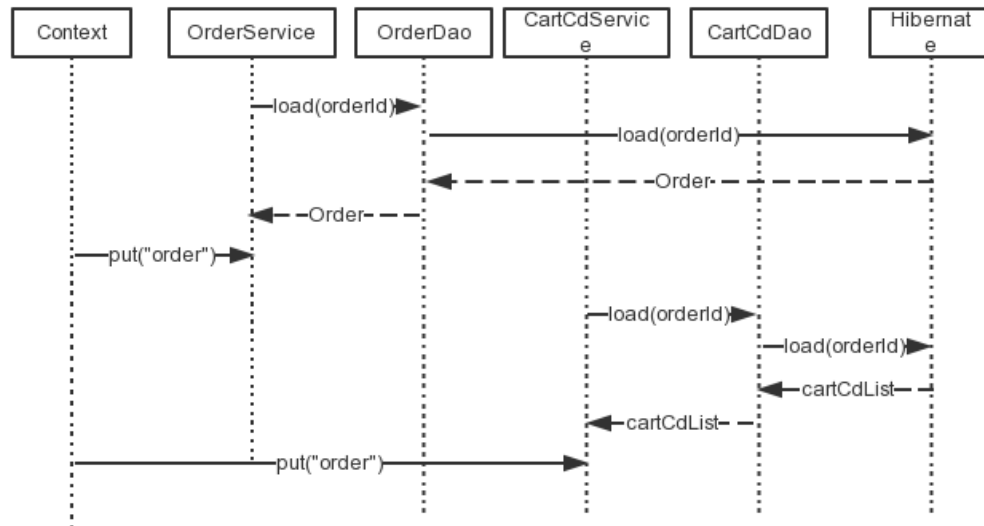
3.2.7.2. 改变密码



3.2.7.3. 修改送货的相关信息

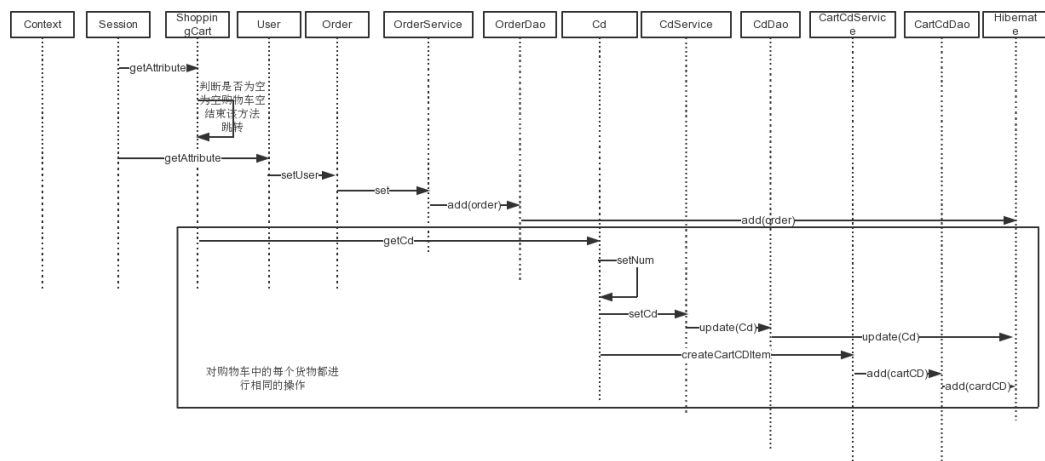


3.2.7.4. 查看订单的信息



3.2.7.5. 确认支付

把购物车中的信息做汇总 调用外部付款接口，进行付款。



3.3 AdminAction 类

3.3.1. 功能

该类主要是管理员对当前系统的订单操作，用户权限操作，唱片操作。

3.3.1.1. 查看当前所有订单信息

3.3.1.2. 修改送货的相关信息

3.3.1.3. 查看订单的信息

3.3.1.4. 用户信息管理

3.3.1.5. 删除用户

3.3.1.6. 唱片信息管理

3.3.1.7. 添加唱片

3.3.2. 性能

3.3.3. 输入项（输入接口）

在 struts2 中 setter 方法由容器获取得到对类中的私有变量进行设置。

3.3.3.1. 无

3.3.3.2. 修改送货的相关信息

修改送货订单的 id

`Context.getRequest().getParameter("id")`

3.3.3.3. 查看订单的信息

被查询订单的 id

`Context.getRequest().getParameter("id")`

3.3.3.4. 用户信息管理

输入用户的 id

管理员/用户

密码/邮箱

3.3.3.5. 删除用户

删除用户的 id

3.3.3.6. 唱片信息管理

唱片的 id

唱片名

歌手

发行时间

价格

库存

封面图片

图片 1-6

音频链接

流派

歌手类型

简介

3.3.3.7. 添加唱片

唱片名
歌手
发行时间
价格
库存
封面图片
图片 1-6
音频链接
流派
歌手类型
简介

3.3.4. 输出项（输出接口）

在网络应用中，将输出的内容加入 context 对象中，为键值对，在页面中调用。

3.3.4.1. 放入当前用户已有的订单信息，热门列表

```
Context().put("orderList")  
Context().put("hotlist")
```

3.3.4.2. 修改送货的相关信息

链接跳转到列出订单情况

3.3.4.3. 查看订单的信息

```
Context().put("order")  
Context().put("cartCdList")
```

3.3.4.4. 用户信息管理

跳转到用户列表

3.3.4.5. 删除用户

跳转到用户列表

3.3.4.6. 唱片信息管理

跳转到唱片列表

3.3.4.7. 添加唱片

跳转到唱片列表

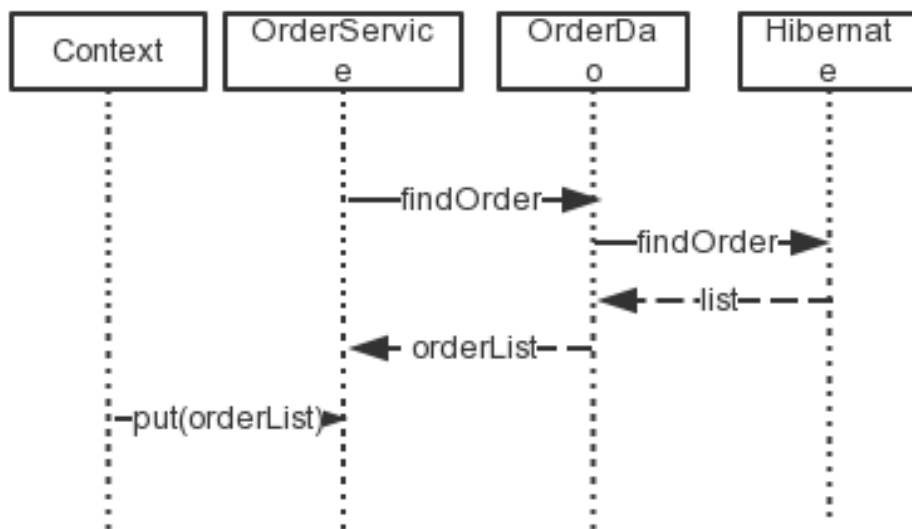
3.3.5. 属性详细描述

```
/**对应订单的 Service**/  
    private IOrderService orderService;  
/**对用户操作的 Service**/  
    private IUserService userService;  
/**对购物车操作的 Service**/  
    private ICartService cartService;  
/**对唱片操作的 Service**/  
    private ICdService cdService;  
/**对界面操作的 Service**/  
    private IShowService showService;  
/**设置唱片的日期**/  
    private String sdate;  
/**设置唱片的图片集合**/  
    private File[] photo;  
/**设置上传唱片的图片文件名集合**/  
    private String[] photoFileName;
```

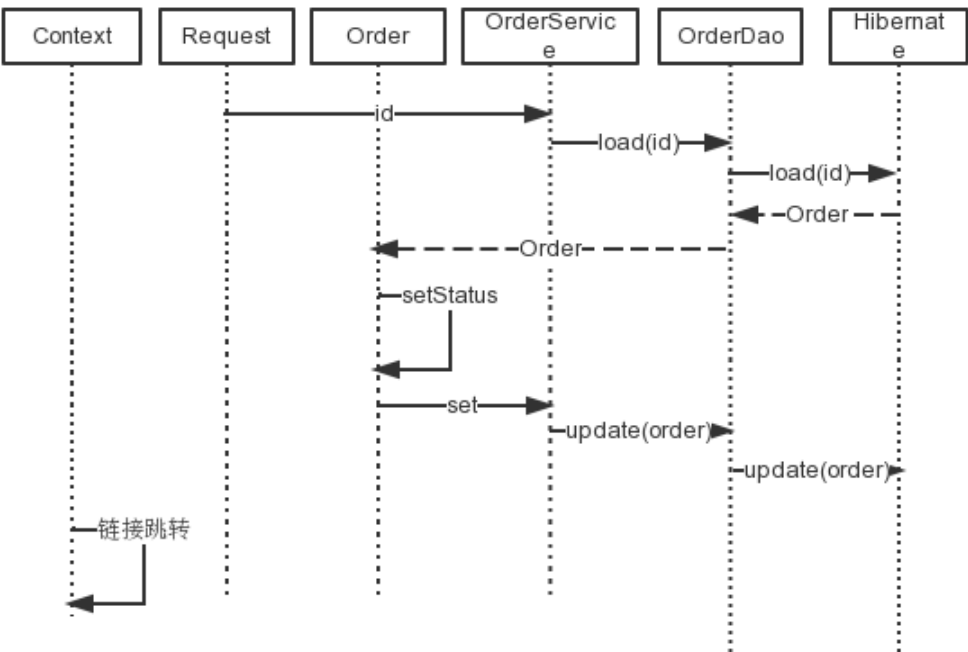
3.3.6. 方法详细描述及实现流程

3.3.7. 类内、类外方法调用状态图

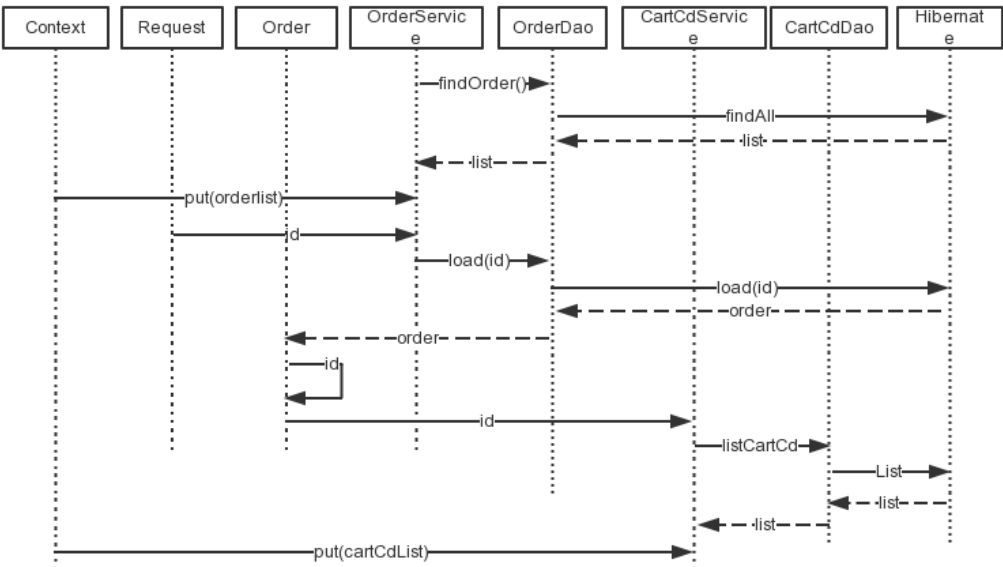
3.3.7.1. 查看当前所有订单信息



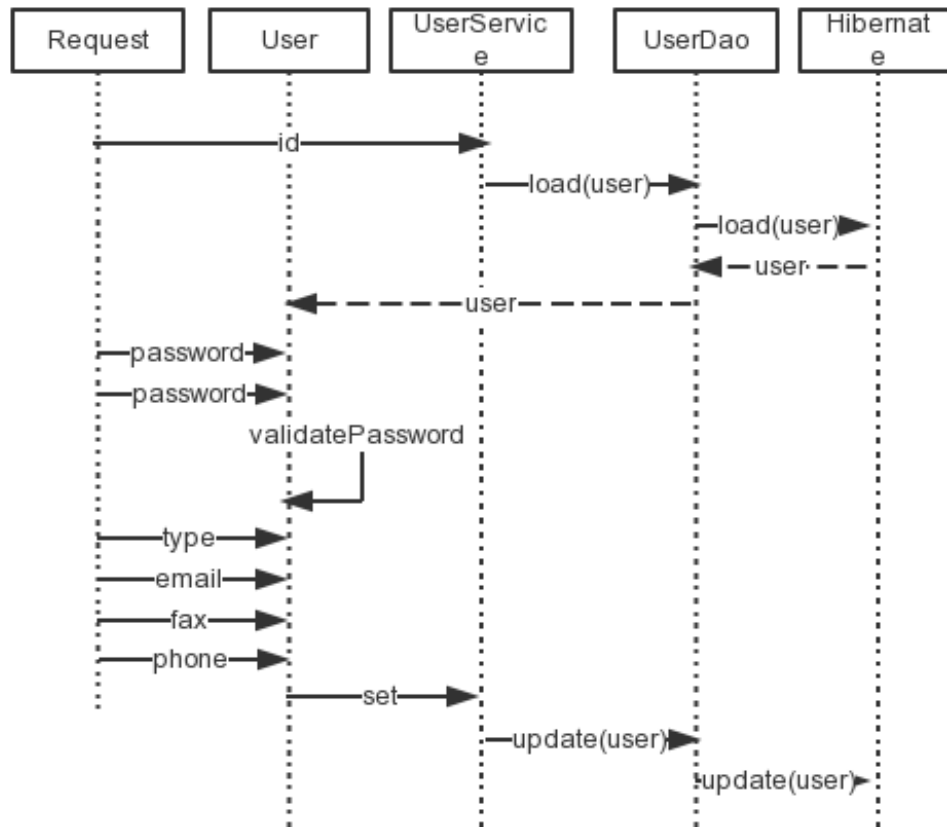
3.3.7.2. 修改送货的相关信息



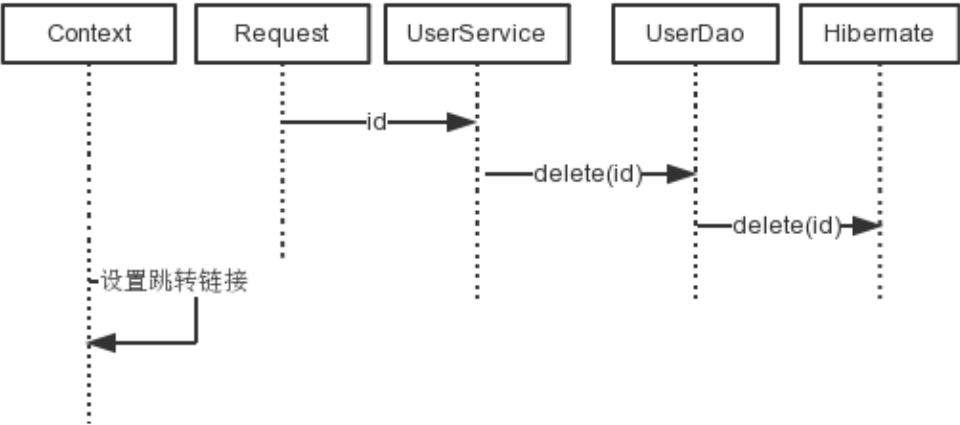
3.3.7.3. 查看订单的信息



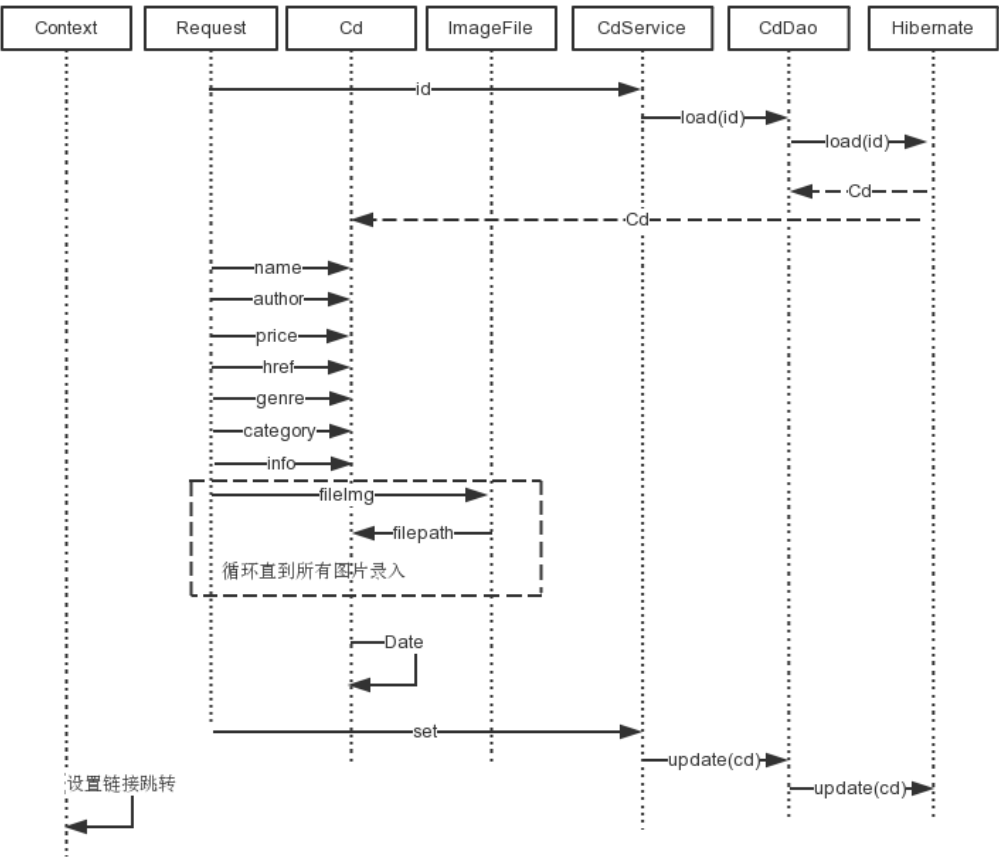
3.3.7.4. 用户信息管理



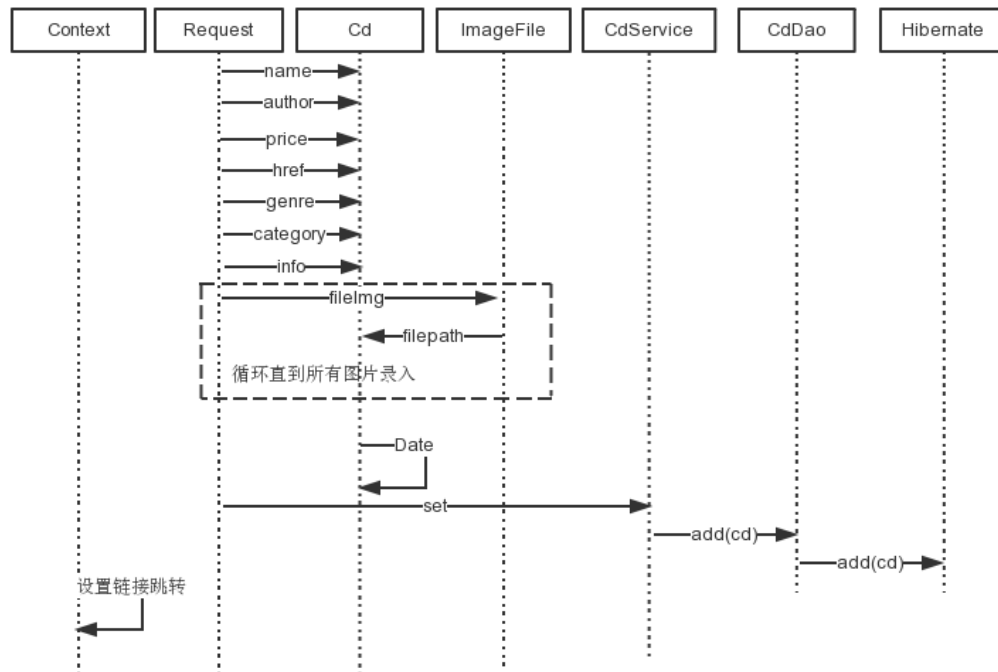
3.3.7.5. 删除用户



3.3.7.6. 唱片信息管理



3.3.7.7. 添加唱片



3.4 所有的 model 类

3.4.1. 功能

对数据库表的映射实体类

3.4.2. 性能

3.4.3. 输入项（输入接口）

在 java bean 中输入项为所有私有属性的 set 方法
SetXXX()

3.4.4. 输出项（输出接口）

在 Java bean 中输出项为所有私有属性的 get 方法
GetXXX()

3.4.5. 属性详细描述

所有 model 的属性为映射的数据表中的字段。

3.4.6. 方法详细描述及实现流程

3.4.7. 类内、类外方法调用状态图

Java bean为Service所操纵的对象。

3.5 BaseDao 类及 IBaseDao

3.5.1. 功能

被Service调用的操作java bean(model)的数据库访问层。

IBaseDao为BaseDao的接口设计

BaseDao为IbaseDao的实现。

其他的dao都继承于BaseDao类

3.5.1.1. 增单个对象

3.5.1.2. 删单个对象

3.5.1.3. 查单个对象

3.5.1.4. 改单个对象

3.5.1.5. 查询一系列对象

3.5.2. 性能

3.5.3. 输入项（输入接口）

3.5.1.1. 增单个对象

Add(T) T 为对应的 Javabean

3.5.1.2. 删单个对象

delete(T) T 为对应的 Javabean

3.5.1.3. 查单个对象

load(id) id 为对应的 java bean 的 id 属性

3.5.1.4. 改单个对象

update(T) T 为对应的 Javabean

3.5.1.5. 查询一列对象

List(String) String 为 hql

3.5.4. 输出项（输出接口）

3.5.1.1. 增单个对象

无

3.5.1.2. 删单个对象

无

3.5.1.3. 查单个对象

T T 为对应的 Javabean

3.5.1.4. 改单个对象

无

3.5.1.5. 查询一列对象

List(T) T 为对应的 Javabean

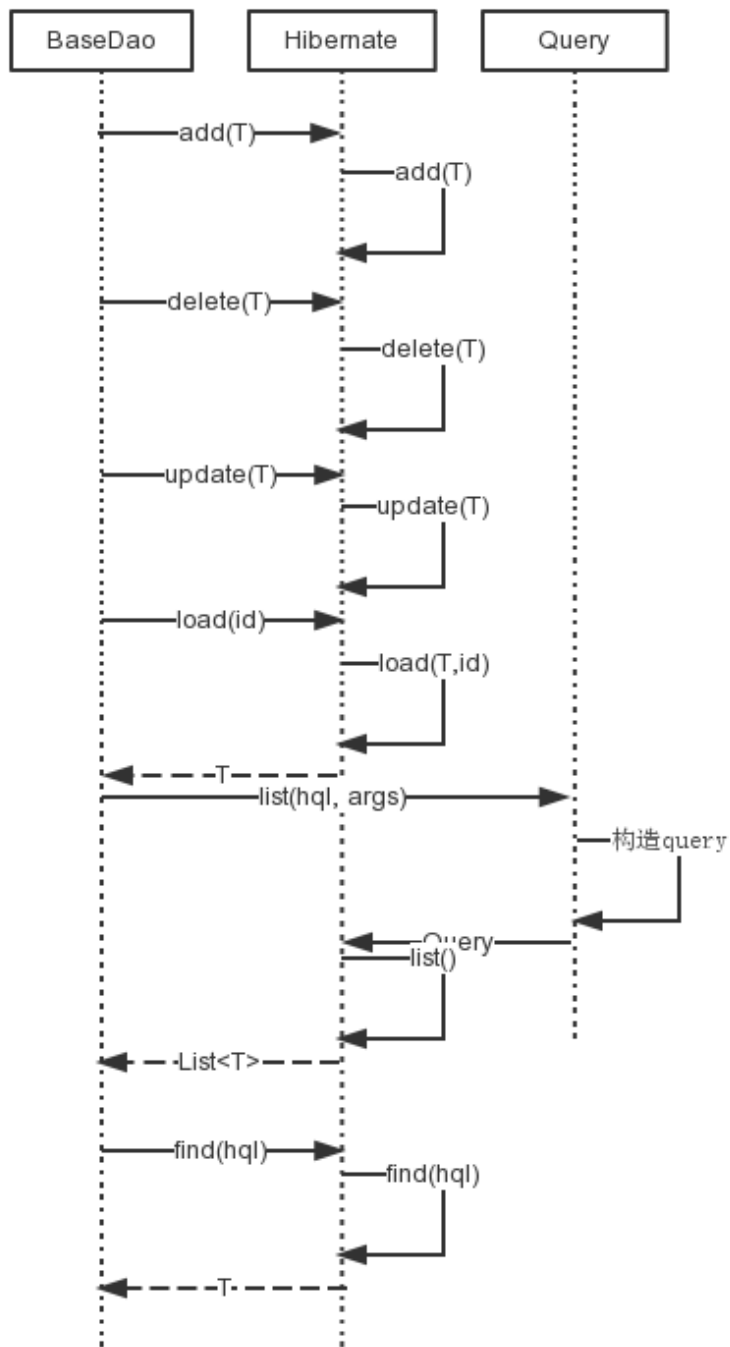
3.5.5. 属性详细描述

无

3.5.6. 方法详细描述及实现流程

3.5.7. 类内、类外方法调用状态图

使用hibernate的方法操作数据库



3.6 Service 类

3.6.1. 功能

该类介于 action 逻辑层与 dao 数据访问层之间,作为 C (controller) 与 M (model) 的桥梁,

提供一定扩展性

3.6.1.1. 增

3.6.1.2. 删

3.6.1.3. 查

3.6.1.4. 改

3.6.1.5. 查询列表

3.6.2. 性能

3.6.3. 输入项（输入接口）

3.6.1.1. 增单个对象

Add(T) T 为该 Service 对应注入的 Javabean

3.6.1.2. 删单个对象

delete(id) id 为该 Service 对应注入的 Javabean 的 id 字段

3.6.1.3. 查单个对象

load(id) id 为该 Service 对应注入的 Javabean 的 id 字段

3.6.1.4. 改单个对象

update(T) T 为该 Service 对应注入的 Javabean

3.6.1.5. 查询一列对象

List(String) String 为 hql

3.6.4. 输出项（输出接口）

3.6.1.1. 增单个对象

无

3.6.1.2. 删单个对象

无

3.6.1.3. 查单个对象

T T 为对应的 Javabean

3.6.1.4. 改单个对象

无

3.6.1.5. 查询一列对象

List(T) T 为从相应 dao 返回的对应的 Javabean

3.6.5. 属性详细描述

该 Service 对应的 bean 的 dao 层的接口 dao 的属性变量
IXXXDao xxxDao;

3.6.6. 方法详细描述及实现流程

3.6.7. 类内、类外方法调用状态图

相应方法内调用了相应的dao的方法。

