**银行网络代收支付平台系统设计与实现**

1. **相关理论和关键技术介绍**

## 2.1 银行网络代收支付平台

银行网络代收支付平台是指具备实力和信誉保障的第三方企业和国内外的各大银行签约，为买方和卖方提供的信用增强。在银行的直接支付环节中增加一中介，在通过第三方支付平台交易时，买方选购商品，将款项不直接打给卖方而是付给中介，中介通知卖家发货；买方收到商品后，通知付款，中介将款项转至卖家账户。

2015年，电子商务交易额翻两番，突破18万亿元。其中，[B2B电子商务](https://www.baidu.com/s?wd=B2B%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YdnjczrAF-nyD3nHRknjT0IAYqnWm3PW64rj0d0AP8IA3qPjfsn1bkrjKxmLKz0ZNzUjdCIZwsrBtEXh9GuA7EQhF9pywdQhPEUiqkIyN1IA-EUBt1PWnkP1Dvn1D1nWcsn16dP1c" \t "https://zhidao.baidu.com/question/_blank)交易规模超过15万亿元，然网络交易支付工具虽多，一个兼具安全性、便利性，且可满足B2B、B2C及C2C的支付服务尚未发展成熟。为掌握法令即将开放可能引发之电子商务金流商机，兰州银行办理网络交易代收代付业务，将可获得网络银行业务、电子化代收业务及信用卡收单业务的拓展，亦可获得代收代付及信用卡收单等手续费与资金池金流的帐款保管运用收益。

如以消费者保护的立场而言，由兰州银行担任网络交易代收代付业者，除可提供消费者与小型卖方价金保管机制等电子商务金流服务，未来可发展网络虚拟账号储值服务，俾强化兰州银行消费金融之深度与广度。同时以兰州银行的稳健形象搭配网络交易代收代付的机制，可有效降低网络交易中的个人资料外泄及诈骗行为，对于我国交易安全及电子商务发展实利多于弊。

**2.2** **银行网络代收支付平台关键技术介绍**

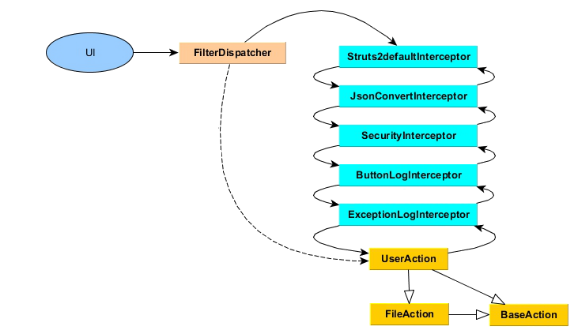
### 2.2.1 Java技术

本系统采用Java语言编写，[Java](http://baike.baidu.com/subview/29/12654100.htm" \t "http://baike.baidu.com/_blank)是一门[面向对象](http://baike.baidu.com/view/125370.htm" \t "http://baike.baidu.com/_blank)编程语言，不仅吸收了[C++](http://baike.baidu.com/view/824.htm" \t "http://baike.baidu.com/_blank)语言的各种优点，还摒弃了C++里难以理解的[多继承](http://baike.baidu.com/view/1160382.htm" \t "http://baike.baidu.com/_blank)、[指针](http://baike.baidu.com/view/159417.htm" \t "http://baike.baidu.com/_blank)等概念，因此Java语言具有功能强大和简单易用两个特征。Java语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程。

Java具有简单性、面向对象、[分布式](http://baike.baidu.com/subview/402382/19718738.htm" \t "http://baike.baidu.com/_blank)、[健壮性](http://baike.baidu.com/view/963152.htm" \t "http://baike.baidu.com/_blank)、[安全性](http://baike.baidu.com/view/421194.htm" \t "http://baike.baidu.com/_blank)、平台独立与可移植性、[多线程](http://baike.baidu.com/view/65706.htm" \t "http://baike.baidu.com/_blank)、动态性等特点。Java可以编写[桌面应用程序](http://baike.baidu.com/view/2251864.htm" \t "http://baike.baidu.com/_blank)、[Web应用程序](http://baike.baidu.com/view/138452.htm" \t "http://baike.baidu.com/_blank)、[分布式系统](http://baike.baidu.com/view/991489.htm" \t "http://baike.baidu.com/_blank)和[嵌入式系统](http://baike.baidu.com/view/6115.htm" \t "http://baike.baidu.com/_blank)应用程序等。

### 2.2.2 Struts2 框架概述

Struts2是一个基于MVC设计模式的Web应用框架，它本质上相当于一个Servlet，在MVC设计模式中，Struts2作为控制器(Controller)来建立模型与视图的数据交互。工作流程如下图：



|  |  |  |  |
| --- | --- | --- | --- |
| **来源** | **元件名称** |  | **功能行为** |
|  |  | UI 发出 request 请求，会被此 filter 拦截，若符合 | |
| Struts2 | FilterDispatcher | 所设定的 URL pattern，如/xxx.action，Strus2 会 | |
|  |  | 依照设定档指派处理此 request 的 Action 物件 | |
| Struts2 | Struts2DefaultInterceptors | 所有 Struts2 预设的 Interceptors，详细请参考 | |
|  |  | Struts2 官方文件 | |
| 自订 | JsonConvertInterceptor | 1. | 负责处理 request JSON 字符串转换至 Action |
|  |  | 2. | VO 物件。 |
|  |  | 统一自动化回复成功信息 |
| 自订 | SecurityInterceptor | 1. | Token 生成与检查 |
|  |  | 2. | IP 来源检查 |
|  |  | 3. | 使用权限限制 |
|  |  | 4. | 注册流程导页控制 |
| 自订 | ButtonLogInterceptor | 自动化网站日志纪录 | |
| 自订 | ExceptionLogInterceptor | 1. | 应用程序 Exception handle |
|  |  | 2. | 统一自动化回复错误信息 |
| 自订 | UserAction | 功能 Action 物件 | |
| 自订 | FileAction | 上传档案 Action 物件 | |
| 自订 | BaseAction | 系统基础 Action 物件 | |

此系统采用 Token 机制，在servlet context建立一个Listener 提供存取 token，配合 SecurityInterceptor 检核 token 的合法性。

1. TokenListener

TokenListener实作javax.servlet.ServletContextListener，藉由此 Interface 来快速建立 Token 机制，当 系统 启动时立即在内存建立一个存放 Token 物件的容器，并同时建立一个 Timer 负责清除 Token。

1. Token

Token 物件属性

|  |  |  |
| --- | --- | --- |
| **属性** | **型态** | **说明** |
| tokenID | String | Token 唯一辨识值 |
| datas | Map | 资料存放物件，让 Token 有 store 资料的功能 |
| effectDate | Date | Token 过期时间 |
| isExpired | Boolean | 是否已过期 |

### 2.2.3 Spring 概述

 Spring框架主要用于与其他技术(struts,hibernate等)进行整合,可将应用程序中的Bean组件实现低耦合关联.最终可以提高系统扩展和维护性，本系统利用Spring框架管理系统的各个组件(Action,Service,DAO)，采用Spring的IOC和AOP机制实现各组件的关联，从而实现了低耦合调用，增强了系统可维护性和扩展性，

Spring主要由以下模块组成：核心容器、Spring 上下文、Spring AOP、Spring DAO，Spring ORM，Spring Web 模块，Spring MVC 框架。



在系统启动时使用Spring提供的org.springframework.web.context.ContextLoaderListener，来建立主要运作物件， ContextLoaderListener 会读取web.xml 环境变量contextConfigLocation 设定的档案位置，将设定档所设定的指定的物件，注册在 Spring Application context 中，成为其中的 Bean Object。Web.xml配置如下图：



为了减少设定档编写的复杂度，使用 Annotation 来协助注册的步骤，要利用 annotation 在设定档设定如下，scan 的范围为 com.neweb package 下所有的物件，配置如下图：

图片21

使用 Annotation 注册物件主要以下5个方面：

1. Spring 提供@Repository/@Service/@Controller/@Component 标签，来进行注册，四种标签皆可注册到 Application context 中，为了物件语意的表达分别使用不同的标签，分配如下：

|  |  |  |
| --- | --- | --- |
| **标签** | **层别** | **对应物件** |
| @Controller | 表现层 | Action |
| @Service | 业务逻辑层 | BO |
| @Repository | 资料存取层 | DAO |
| @Component | anywhere | 共享元件及 Action VO |

1. @Scope("prototype")此标签说明，若向 Spring application context 取得 Bean 时，需要建立一个新的 instance，在预设的情况下注册的 Bean 皆为 singleton，即为单一的 instance。
2. 何时使用 prototype

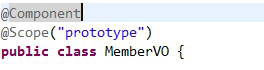
一个 request 如同一个 Thread 依照此系统结构会呼叫到一个 Action Object，并使用到 Action VO，而不同的 request 应该使用各自的 Action 及 Action VO object，才不会造成 non-thread-safe 的状况，所以全部的 Action 及 Action VO 都该设定为prototype。

1. 使用Annotation 注册物件范例

* 范例 Action



* 范例 Action VO



* 范例 BO

图片3

* 范例 DAO

图片4

1. 使用 Annotation 完成 IOC 动作

|  |  |
| --- | --- |
| **标签** | **注入行为** |
| @Resource | 为 javax.annotation.Resource，此标签依名称来找寻相对应的 Application |
|  | context 中的 bean |
| @Autowired | org.springframework.beans.factory.annotation.Autowired，此标签依 type 来 |
|  | 找寻相对应的 Application context 中的 bean |

### 2.2.4 Hibernate 概述

Hibernate是一个开放源代码的对象关系映射框架，它对JDBC进行了非常轻量级的对象封装，它将POJO与数据库表建立映射关系，是一个全自动的orm框架，hibernate可以自动生成SQL语句，自动执行，使得Java程序员可以随心所欲的使用对象编程思维来操纵数据库。 Hibernate可以应用在任何使用JDBC的场合，既可以在Java的客户端程序使用，也可以在Servlet/JSP的Web应用中使用，最具革命意义的是，Hibernate可以在应用EJB的[J2EE](http://baike.baidu.com/view/1507.htm" \t "http://baike.baidu.com/_blank)架构中取代CMP，完成[数据持久化](http://baike.baidu.com/view/4549557.htm" \t "http://baike.baidu.com/_blank)的重任。

Hibernate的核心类和接口一共有6个，分别为:Session、SessionFactory、

Transaction、Query、Criteria和Configuration。这6个核心类和接口在任何开发中都会用到。通过这些[接口](http://baike.baidu.com/view/159864.htm" \t "http://baike.baidu.com/_blank)，不仅可以对持久化对象进行存取，还能够进行事务控制。下面对这6个核心接口和类分别加以介绍。

### Session

[Session](http://baike.baidu.com/view/25258.htm" \t "http://baike.baidu.com/_blank)接口负责执行被持久化对象的[CRUD](http://baike.baidu.com/view/913128.htm" \t "http://baike.baidu.com/_blank)操作(CRUD的任务是完成与数据库的交流，包含了很多常见的SQL语句)。但需要注意的是[Session对象](http://baike.baidu.com/view/2442531.htm" \t "http://baike.baidu.com/_blank)是非[线程安全](http://baike.baidu.com/view/1298606.htm" \t "http://baike.baidu.com/_blank)的。同时，Hibernate的[session](http://baike.baidu.com/view/25258.htm" \t "http://baike.baidu.com/_blank)不同于JSP应用中的HttpSession。当使用session这个术语时，其实指的是Hibernate中的session，而以后会将HttpSession对象称为用户session。

### SessionFactory

SessionFactory接口负责初始化Hibernate。它充当数据存储源的代理，并负责创建Session对象。这里用到了[工厂模式](http://baike.baidu.com/view/1306799.htm" \t "http://baike.baidu.com/_blank)。需要注意的是SessionFactory并不是轻量级的，因为一般情况下，一个项目通常只需要一个SessionFactory就够，当需要操作多个数据库时，可以为每个数据库指定一个SessionFactory。

### Transaction

Transaction 接口是一个可选的API，可以选择不使用这个接口，取而代之的是Hibernate 的设计者自己写的底层事务处理代码。 Transaction 接口是对实际事务实现的一个抽象，这些实现包括JDBC的事务、JTA 中的UserTransaction、甚至可以是CORBA 事务。之所以这样设计是能让开发者能够使用一个统一事务的操作界面，使得自己的项目可以在不同的环境和容器之间方便地移植。

### Query

Query接口方便地对数据库及持久对象进行查询，它可以有两种表达方式：HQL语言或本地数据库的SQL语句。Query经常被用来绑定查询参数、限制查询记录数量，并最终执行查询操作。

### Criteria

Criteria接口与Query接口非常类似，允许创建并执行面向对象的标准化查询。值得注意的是Criteria接口也是轻量级的，它不能在Session之外使用。

### Configuration

Configuration 类的作用是对Hibernate 进行配置，以及对它进行启动。在Hibernate 的启动过程中，Configuration 类的实例首先定位映射文档的位置，读取这些配置，然后创建一个SessionFactory对象。虽然Configuration 类在整个Hibernate 项目中只扮演着一个很小的角色，但它是启动hibernate 时所遇到的第一个对象。

**hibernate工作原理：**

1. 通过Configuration().configure();读取并解析hibernate.cfg.xml[配置文件](http://baike.baidu.com/view/2117618.htm" \t "http://baike.baidu.com/_blank)。

2、由hibernate.cfg.xml中的<mappingresource="com/xx/User.hbm.xml"/>读取解析映射信息。

3、通过config.buildSessionFactory()，得到sessionFactory。

4、sessionFactory.openSession()，得到session。

5、session.beginTransaction()，开启事务。

6、persistent operate。

7、session.getTransaction().commit()，提交事务

8、关闭session。

9、关闭sessionFactory。

**hibernate优点：**

1、封装了jdbc，简化了很多重复性代码。

2、简化了DAO层编码工作，使开发更对象化了。

3、移植性好，支持各种数据库，如果换个数据库只要在[配置文件](http://baike.baidu.com/view/2117618.htm" \t "http://baike.baidu.com/_blank)中变换配置就可以了，不用改变hibernate代码。

4、支持透明持久化，因为hibernate操作的是纯粹的（pojo）java类，没有实现任何接口，没有侵入性。所以说它是一个轻量级框架。

### 2.2.5 DB2数据库概述

IBM DB2 是[美国](http://baike.baidu.com/view/2398.htm" \t "http://baike.baidu.com/_blank)[IBM](http://baike.baidu.com/view/1937.htm" \t "http://baike.baidu.com/_blank)公司开发的一套[关系型数据库管理系统](http://baike.baidu.com/view/1450387.htm" \t "http://baike.baidu.com/_blank)，它主要的运行环境为[UNIX](http://baike.baidu.com/view/8095.htm" \t "http://baike.baidu.com/_blank)（包括IBM自家的[AIX](http://baike.baidu.com/view/349664.htm" \t "http://baike.baidu.com/_blank)）、[Linux](http://baike.baidu.com/view/1634.htm" \t "http://baike.baidu.com/_blank)、IBM i（旧称OS/400）、[z/OS](http://baike.baidu.com/view/1245216.htm" \t "http://baike.baidu.com/_blank)，以及[Windows](http://baike.baidu.com/view/4821.htm" \t "http://baike.baidu.com/_blank)服务器版本。

DB2主要应用于大型应用系统，具有较好的可伸缩性，可支持从大型机到单用户环境，应用于所有常见的服务器操作系统平台下。 DB2提供了高层次的数据利用性、完整性、安全性、可恢复性，以及小规模到大规模应用程序的执行能力，具有与平台无关的基本功能和SQL命令。DB2采用了数据分级技术，能够使大型机数据很方便地下载到LAN数据库服务器，使得客户机/服务器用户和基于LAN的应用程序可以访问大型机数据，并使数据库本地化及远程连接透明化。 DB2以拥有一个非常完备的查询优化器而著称，其外部连接改善了查询性能，并支持多任务并行查询。 DB2具有很好的网络支持能力，每个子系统可以连接十几万个分布式用户，可同时激活上千个活动线程，对大型分布式应用系统尤为适用，DB2在系统中的业务环节如下图：

