## 2.4版本完善

1.根据页面特点放置不同文件夹

2.

## 2.4.0版本完善

1.为了实现存储各个操作数据库存储记录，需要改用数据库对存款和取款进行记录。总结有下列方法：

a.操作方法改在数据库---改动较大，效率低，安全性高

1）废除moneybean

2）取款和存款方法参数改变，业务层接口、业务层、持久层接口、持久层、action对象均需改变

3）在持久层方法内容中增加存储记录的操作

4）废除储存方法--此时转账方法后调用存储方法已无意义

b.在业务层取款和存款方法后，调用持久层方法用数据库进行记录---改动较小，效率高，安全性低

1）持久层接口和持久层增加对应方法进行存储记录

2）为了得到用户名，依旧需要改变存取款方法的参数

3）由于使用数据库，moneybean虽然提高效率但是不安全，也仅对存取款有用

b方法同样费力，选用a方法

### 2.4.2更新内容

1.更改查询方法访问数据库

2.更改取款、存款、转账方法

1.更改业务层为直接调用持久层，判断改在持久层相应方法进行判断

2.操作前改调用查询方法进行金额查询及验证

3.注释声明废除原查询及退出方法

4.更改查询的servlet，调用inquiry改为getBalance，返回值由String变为double

### 2.4.3

1.数据库business表，增加transactionType字段存储交易类型；增加

外键userid与用户管联

2.更改业务层，持久层查询余额方法为查询方法，返回值由duoble改为map，并修改调用其的相应语句

3.更改业务层，持久层getBalance方法名为getAccountInformation

4.持久层增加了StoreTransactionRecords方法存储交易记录，并在持久层的取款，存款，转账方法调用其进行存储

5.修改持久层的取得交易记录的方法，并由transactionRecord更名为getTransactionRecord

6.转账对方也存储记录，更改转账交易标记transfer为payment（付款），proceeds（收款）

7.更改business表部分字段名称中间首字母大写，并更改持久层相应代码

8.增加TransactionRecordBean数据bean来封装交易记录

9.数据bean中id类型为Integer，由trb.setBId(Integer.valueOf(rs.getInt("bId")));

trb.setBId(Integer.valueOf(rs.getString("bId")));

trb.setBId(Integer.parseInt(rs.getInt("bId")));

均出空指针异常

原因可能出在TransactionRecordBean中的Integer bId没有设值，为null

int bid=rs.getInt("bId");

trb.setBId(new Integer(bid));依旧空指针异常

在TransactionRecordBean中加上private Integer bId = new Integer(-1);

再trb.setBId(new Integer(bid));还是空指针异常

直接改为trb.setBId(new Integer(1));还是空指针异常

改为

trb.setBId(Integer.valueOf(rs.getInt("bId")));

trb.setBId(Integer.valueOf(rs.getString("bId")));

trb.setBId(Integer.parseInt(rs.getInt("bId")));

均空指针异常

**#####最终发现原来是数据bean忘记了设值，为null**

10.发现private Integer bId;对应的getter、setter方法，首字母小写public Integer getbId() ，不出错，大写public Integer getBId() ,在jsp页面用el表达式时反而出错

# 3.0 struts1

### 需求说明

1.采用dispatchaction对请求进行处理

2.国际化技术，对静态代码和动态代码都进行国际化

3.对异常进行国际化

4.分页技术

### 3.0.0

1.删除2.4.3持久层注释的map及注释的用其插入记录的代码，包com/cx/bank/test和类BankSystem.java及被注释的相应代码，持久层addBank存储方法、moneybean.java，业务层inquiry方法，及由于取款、存款、转账、改数据库操作被注释的代码

### ###3.1.0

1.用dispatchaction登录和注册

login.jsp页面

2.点击注册，出错Invalid path /RegisterLoginAction was requested

3.<form name="loginForm" action="/RegisterLoginAction.do?command=login" method="post">

点击登录，出错：Type Status Report

Message /RegisterLoginAction.do

Description The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.

4.<form name="loginForm" action="RegisterLoginAction.do?command=login" method="post">出错Invalid path /RegisterLoginAction was requested

5.把login.jsp放在根下的bank文件夹，请求带路径，解决，但是并不清楚为什么之前错误 <form name="LoginForm" action="bank/RegisterLoginAction.do?command=login" method="post">

#####6.按之前的，无错。推测是浏览器缓存或者服务器缓存问题，需要牢记。

7.配置文件中<forward name="registerSuccess" path="/registerSuccess.jsp" redirect="true"></forward>，当在RegisterLoginAction中return mapping.findForward("loginSuccess");会出Type Status Report

Description The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.异常。找不到页面。

去掉redirect="true"后，可以找到页面。猜测是因为不是服务器跳转而是客户端跳转，不能访问web-inf下的文件

8.解决system.jsp框架中，框架页面不显示问题

原路径<frame name="top" noresize="noresize" src="/RegisterLoginAction.do?command=show\_top">

出现：404错误

Message /RegisterLoginAction.do

Description The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.

现路径<frame name="top" noresize="noresize" src="RegisterLoginAction.do?command=show\_top">

9.由于在<action path="/BankBusiness/BankBusinessAction" type="com.cx.bank.action.BankBusinessAction"

name="LoginForm" scope="request" parameter="command">中，忘记改表单bean的名字，在BankBusinessAction.java的方法中BusinessForm bf = (BusinessForm) form;会出现java.lang.ClassCastException异常

10.取款、存款、转账页面空白，原因是忘记进页面而直接进方法

11.写完后出现Message /BankBusiness/BankBusiness/BankBusinessAction was requested这样的错误，原因是页面的表单上路径为action="BankBusiness/BankBusinessAction.do?command=withDrawals

重复了。此时的默认路径本来就是BankBusiness下，不用写这个路径，改为action="BankBusinessAction.do?command=withDrawals"后，取款成功但是跳转页面空白。把return mapping.findForward("BankBusiness/BankBusinessAction.do?command=inquiry");改为了ActionForward af = new ActionForward("/BankBusiness/BankBusinessAction.do?command=inquiry", false);return af;成功。原因可能是这个action下没有BankBusiness/BankBusinessAction的跳转路径

#### 12.登录失败跳转错误页面问题

//request.setAttribute("message", "登录");return mapping.findForward("error");无法显示参数，因为 redirect="true"

后改为return mapping.findForward("error?message=登录");根本找不到路径

再改为

##### **方法一：return new ActionForward("error.jsp?message=登录", true);**

成功，但是页面不显示参数。

经查阅发现是EL表达式不能直接获取。

**资料：***在解决这个问题之前，首先要了解el表达式获取值的原理以及范围：第一、el表达式会从小到大去找寻相应的key，如果找到了就取相应的value，这个范围从小到大依次是pageContext、request、session、Application。第二、取值的原理，这几个作用域都有一个相同的属性：attribute。都拥有setAttribute()和getAttribute()方法，el表达式取值就是从这里面取的。*

*搞清楚上面的东西之后再来看之前的问题就好理解了。我们要取的数据是携带在url中的，而获取url中参数的方法是getParameter（）。所以url中的参数并不能直接通过${name}的方式获取。*

*最后，el表示式获取url中参数的方法是：`\*\*\*${param.name}\*\*\*`*

##### 页面中将${ message}改为${param.message}，成功显示

##### 方法二：forward带参数，太麻烦了，现在不用

<https://www.cnblogs.com/wangpei/archive/2012/02/15/2352181.html>

13.将页面放入web-inf下后页面不显示图片

将background-image: *url("images/main04.jpg")*;改为

background-image: *url("****../****images/main04.jpg")*;

#### 说明

完成了页面的跳转和功能的实现

没有进行国际化、没有对异常进行处理

没有分页显示

### 3.2.0分页显示

#### 说明

完成了分页功能和显示

没有进行国际化、没有对异常进行处理

没有对请求字符进行拦截转换

### 3.3.0拦截器

1.拦截器无效，中文依旧是??

查阅后发现**拦截器或者过滤器对表单只是对表单的post方式有效！**

在RegisterLoginAction中**return** **new** ActionForward("error.jsp?message=注册", **true**);是get请求，所以无效

编码解决方式<https://www.cnblogs.com/zengyufei/p/4456472.html>

编码解决麻烦，get请求的字符改用抛出异常的方法处理

#### 说明

加了编码拦截器，但是get请求没用，考虑直接国际化解决，不用拦截器

### 3.4.0静态国际化

实现页面静态国际化，还未实现通过按钮来转换中英文，未实现动态国际化

### 3.5.0动态国际化

实现了国际化自动切换

1、在页面忘记写<%@ taglib prefix=*"html"* uri=*"http://struts.apache.org/tags-html"* %> ，出现

**Message** An exception occurred processing JSP page [/WEB-INF/BankBusiness/queryResults.jsp] at line [48]

**Description** The server encountered an unexpected condition that prevented it from fulfilling the request.

**Root Cause**

javax.servlet.ServletException: javax.servlet.jsp.JspException: Cannot find bean msg in any scope

的错误

3.6.0异常处理

1.在jsp中，输出动态国际化消息，页面出错。message漏写了s

<html:message id=*"msg"* message=*"true"* property=*"message"*>

<bean:write name=*"msg"*/>

</html:message>

**浏览器出错：**

**Type** Exception Report

**Message** /error.jsp (line: [70], column: [0]) No tag [message] defined in tag library imported with prefix [html]

**Description** The server encountered an unexpected condition that prevented it from fulfilling the request.

**Exception**

org.apache.jasper.JasperException: /error.jsp (line: [70], column: [0]) No tag [message] defined in tag library imported with prefix [html]

改为：

<html:messages id=*"msg"* message=*"true"* property=*"message"*>

<bean:write name=*"msg"*/>

</html:messages>

# 4.0 Struts1+Hibernate集成

## 需求说明

1. 将项目升级到Struts+Hibernate集成版本
2. 由于之前的疏忽，更改数据库设计及相应的数据bean
3. 删除此前多余的moneyBean

## 4.1.0

#### 修改UserBean

增加了users表的其他数据的相应属性

#### 修改TransactionRecordBean

修改了数据库结构，复制旧表business创建新表transaction\_record，删除了myName字段。由userId维护关系。**可在其他版本优化为两个外键及鉴别器字段**

#### 3.启动服务读取Hibernate的映射文件出错

严重: Exception starting filter [HibernateFilter]

org.hibernate.InvalidMappingException: Could not parse mapping document from resource com/cx/bank/mode1/TransactionRecordBean.hbm.xml

………………

Caused by: org.hibernate.PropertyNotFoundException: field [id] not found on com.cx.bank.mode1.TransactionRecordBean

…………

原因是映射属性时，属性名错了

实际<id name=*"id"*>并没有id这个属性，而是bId,改成<id name=*"id"*>

### 说明

基本完成了Hibernate的集成，性能有待优化。查询交易记录，出现记录无序错乱的bug。

## 4.2.0

1.发现ManagerImp1类的getTransactionRecord()方法中

list = query.setFirstResult(pageNo).setMaxResults(pageSize).list();

设置错了分页，应为

list = query.setFirstResult((pageNo-1)\*pageSize).setMaxResults(pageSize).list();

### 说明

与Hibernate集成成功，可进行交作业或者性能优化。

## 4.3.0

需求：创建vo类，表单bean不到业务层。

# 5.0 struts1+spring+hibernate

## 5.0.0

1.tomcat启动读取spring配置信息出空指针异常

严重: Exception sending context initialized event to listener instance of class [org.springframework.web.context.ContextLoaderListener]

java.lang.NullPointerException

……………………………………………………………………………………………………………………………………………………

九月 24, 2018 7:56:53 下午 org.apache.catalina.core.StandardContext startInternal

严重: One or more listeners failed to start. Full details will be found in the appropriate container log file

九月 24, 2018 7:56:53 下午 org.apache.catalina.core.StandardContext startInternal

严重: Context [/bank5.0] startup failed due to previous errors

笔记错了，课件的classpath带了\*，实际不能带。但是老师给的代码带了\*

**<context-param>**

**<param-name>contextConfigLocation</param-name>**

**<param-value>classpath\*:applicationContext-\*.xml</param-value>// classpath\*不能加\*，应为classpath:applicationContext-\*.xml**

**</context-param>**

2.一方项目请求就出错，连首页都没进

严重: Servlet.service() for servlet [jsp] in context with path [/bank5.0] threw exception

org.springframework.beans.factory.NoSuchBeanDefinitionException: No bean named 'sessionFactory' is defined

1. 以上都不对

**classpath并不会到web-inf下找文件**

改为<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/applicationContext-\*.xml</param-value>

</context-param>

改为 <!-- 配置sessionFactory -->

<bean id=*"sessionFactory"* class=*"org.springframework.orm.hibernate3.LocalSessionFactoryBean"*>

<property name=*"configLocation"*>

<value>/WEB-INF/hibernate.cfg.xml</value>

</property>

</bean>

### 说明：

三大框架集成，ssh。未对性能进行优化。未增加管理员权限验证，冻结功能和日志功能

## 5.1.0

### 需求：

1. 增加身份验证，增加管理员角色，并增加相应页面。
2. 修改业务层的银行业务方法参数为原来参数类型，不用vo类
3. 用户被冻结将无法进行取款存款及转账操作

### 5.1.1

1.增加了账户状态字段及相应属性及方法

2.将管理员的相应类独立为一个模块

3.管理员模块实现了查询所有用户信息

### 5.1.2

1.实现翻页

2.实现查询

### 5.1.3

1. 完成冻结功能
2. 页面设计完毕
3. 未实现身份验证

#### 问题

##### el表达式比较字符串没有用【重要】

**原因：el表达式和双引号间有空格**

将<c:if test=*"*${ub.conditions **eq** 'activation'} *"*>改为<c:if test=*"*${ub.conditions **eq** 'activation'}*"*>

2

### 5.1.final

完成用户权限验证，用户账户被冻结后将无法进行取款存款及转账操作

## 5.2.0

增加日志存储

## 5.3.0

实现日志功能

### 问题【重要】

#### 1.由于状态信息设在状态拦截器里，为静态属性。当一个冻结用户登录后，其他没有被冻结的用户也无法进行相关被冻结的操作。相关类：com.cx.bank.filter.StateFilter com.cx.bank.aspect.AspectHandler com.cx.bank.manager.ManagerImp1

#### 解决：

去除了用户状态拦截器，在每次调用相应方法时，在切面类的相关方法中对用户状态进行判断

空指针是System.***out***.println("this.getHibernateTemplate():"+**this**.getHibernateTemplate());

getHibernateTemplate()得到的对象为null

1. 通过注入业务对象，不行，业务对象要被织入advice，造成循环注入
2. 在切面类中通过单例得到业务对象，调用查询方法。不行，出现空指针异常
3. 把业务类改为公有构造方法，调用查询方法。不行，依旧出现空指针异常
4. **通过joinPoint得到代理对象或者目标对象，调用其相应的方法得到状态信息，方法可行**

System.***out***.println("getTarget() ="+**joinPoint. getTarget()** );

System.***out***.println("getThis() ="+**joinPoint. getThis()** );

1. 直接在切面类中访问数据库，理论可行

#### 问题原因：

推测是由于在切面类中得到的业务层对象不是通过spring注入的，所以没有注入sessionFactory，得到的Session为null

# 6.0 SpringMVC+Spring+Hibernate

#### 导入的架包：【重要】

**spring-framework-3.0.0.RELEASE\dist\\*.jar**

**hibernate-3.2.6\lib\\*.jar**

**mysql-connector-java-5.1.37-bin.jar**

**jstl.jar和standard.jar**

**spring-framework-2.0-with-dependencies文件夹下**

**SPRING\_HOME/lib/jakarta-commons/commons-logging.jar**

**SPRING\_HOME/lib/aspectj/\*.jar 切面类相关的库**

**SPRING\_HOME/cglib/\*.jar CGLIB库**

**SPRING\_HOME/lib/aopallianceaopalliance.jar 出错后发现要导这个包**

### 6.0.1

说明：导入了架包

导入了架包，修改了控制类，除了异常完成了配置。没有对身份进行验证。正在给银行业务控制类修改路径配置

## 6.1.0

### 说明

1. 修改了所有的action类为spring mvc的控制类
2. 给所有的action类及其方法映射了注解
3. 删除了com.cx.bank.form整个包及其类，并删除或注释了多余的类
4. 修改了UserBean，用来接收注册和登录的请求参数
5. 增加BusinessBean用来接收用户的业务请求参数
6. 增加AdministratorBean用来接收管理员的请求参数
7. 完成了所有页面的标签设置：国际化标签从struts1标签变为了spring mvc标签
8. 把动态国际化消息转换成了字符串放在错误页面（error.jsp）及查询余额页面（）显示
9. 修改了com.cx.bank.mode1包名为com.cx.bank.model，并在需要导入这个包的相应地方做了修改
10. 完善了配置文件，删除了多余的配置文件

### 未实现的功能

1. 还未调试各个功能可能出现的bug
2. 在一些地方带路径的请求和不带路径的请求混用，以求测试bug，但还并未调试，及修改为正确路径
3. 动态转换国际化：用户通过请求切换中英文
4. 异常的处理，没有修改异常，依旧是上个版本的异常处理
5. 没有对未登陆的请求进行验证，预测用拦截器做

### 6.1.X bug日志

#### 第一次启动报错

[**6.1.x启动报错的全部信息01.txt**](错误日志/6.1.x启动报错的全部信息01.txt)

##### 部分错误信息：

十月 04, 2018 8:48:48 下午 org.apache.catalina.core.ApplicationContext log

信息: Initializing Spring root WebApplicationContext

20:48:49,633 ERROR ContextLoader:208 - Context initialization failed

**org.springframework.beans.factory.BeanDefinitionStoreException: Unexpected exception parsing XML document from ServletContext resource [/WEB-INF/applicationContext-common.xml]; nested exception is java.lang.NoClassDefFoundError:** org/aopalliance/intercept/MethodInterceptor

…………………………………………………………

##### 解决

导入spring-framework-2.0-with-dependencies\lib\**aopallianceaopalliance.jar**

#### 第二次启动

产生的错误为：[**错误日志\6.1.x启动报错的全部信息02.txt**](错误日志/6.1.x启动报错的全部信息02.txt)

##### 部分错误信息：

21:05:14,514 INFO Configuration:559 - Reading mappings from resource : com/cx/bank/mode1/UserBean.hbm.xml

21:05:14,530 ERROR ContextLoader:208 - Context initialization failed

org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'administratorController': Injection of resource dependencies failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'administratorManagerImp1' defined in ServletContext resource [/WEB-INF/applicationContext-beans.xml]: Cannot resolve reference to bean 'sessionFactory' while setting bean property 'sessionFactory'; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'sessionFactory' defined in ServletContext resource [**/WEB-INF/applicationContext-common.xml**]: Invocation of init method failed; nested exception is org.hibernate.MappingNotFoundException: resource: **com/cx/bank/mode1/UserBean.hbm.xml not found**

at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.postProcessPropertyValues(CommonAnnotationBeanPostProcessor.java:297)

……………………

##### 解决：

**在/bank6.0/WebRoot/WEB-INF/hibernate.cfg.xml文件中**

修改：把标红的数字**1**，修改为字母**l**，因为此前修改了包名

<!-- 引入映射文件 -->

<mapping resource=*"com/cx/bank/mode****1****/UserBean.hbm.xml"*/>

<mapping resource=*"com/cx/bank/mode****1****/TransactionRecordBean.hbm.xml"*/>

<mapping resource=*"com/cx/bank/mode****1****/LogBean.hbm.xml"*/>

#### 第三次启动

错误：[**错误日志\6.1.x启动报错的全部信息03.txt**](错误日志/6.1.x启动报错的全部信息03.txt)

**发现和上次差不多的错误，仔细查看和思考后，发现应该是映射文件忘记了修改包名。**

##### 解决

把

/bank6.0/src/com/cx/bank/model/**LogBean.hbm.xml**

/bank6.0/src/com/cx/bank/model/**TransactionRecordBean.hbm.xml**

/bank6.0/src/com/cx/bank/model/**UserBean.hbm.xml**

三个文件中的<hibernate-mapping package=*"com.cx.bank.mode****1****"*>

改为<hibernate-mapping package=*"com.cx.bank.mode****l****"*>

#### 第四次启动：无错

#### 5.进入项目出错：推测是没有导入国际化资源文件的配置。

[**6.1.x发请求出错01.txt**](错误日志/6.1.x发请求出错01.txt)

##### 部分错误信息：

21:40:47,513 ERROR MessageTag:81 - No message found under code **'bank.index.text.welcome' for locale 'zh\_CN'.**

javax.servlet.jsp.JspTagException: No message found under code **'bank.index.text.welcome' for locale 'zh\_CN'.**

………………

##### 解决

在/bank6.0/WebRoot/WEB-INF/**applicationContext-beans.xml**文件中加入以下内容：

<bean id=*"messageSource"* class=*"org.springframework.context.support.ResourceBundleMessageSource"*>

<property name=*"basename"* value=*"res/messages"* /><!-- 在bean工厂中通过配置注入国际化资源文件的基名 -->

</bean>

并修改国际化资源文件的名字，因为Spring MVC读取国际化资源文件的名字和Struts1不一样。

把： **MessageResources\_zh\_CN.properties MessageResources\_en\_US.properties MessageResources.properties**

修改为： **messages\_en\_US.properties messages\_zh\_CN.properties messages.properties**

## 6.2.0 调完bug，国际化切换成功

### 说明

在6.1.0的基础上调试完了bug，其他如6.1.0

### bug日志

#### 1、管理员页面冻结操作出错

[**6.2.x管理员页面冻结操作出错.txt**](错误日志/6.2.x管理员页面冻结操作出错.txt)

##### 部分错误信息：

十月 05, 2018 5:06:46 下午 org.apache.catalina.core.ApplicationDispatcher invoke

严重: Servlet.service() for servlet [spring] threw exception

org.springframework.validation.BindException: org.springframework.validation.BeanPropertyBindingResult: 2 errors

Field error in object 'administratorBean' on field 'pageNo': rejected value [1,1]; codes [typeMismatch.administratorBean.pageNo,typeMismatch.pageNo,typeMismatch.int,typeMismatch]; arguments [org.springframework.context.support.DefaultMessageSourceResolvable: codes [administratorBean.pageNo,pageNo]; arguments []; default message [pageNo]]; default message [Failed to convert property value of type 'java.lang.String[]' to required type 'int' for property 'pageNo'; nested exception is java.lang.NumberFormatException: For input string: "1,1"]

Field error in object 'administratorBean' on field 'pageSize': rejected value [10,10]; codes [typeMismatch.administratorBean.pageSize,typeMismatch.pageSize,typeMismatch.int,typeMismatch]; arguments [org.springframework.context.support.DefaultMessageSourceResolvable: codes [administratorBean.pageSize,pageSize]; arguments []; default message [pageSize]]; default message [Failed to convert property value of type 'java.lang.String[]' to required type 'int' for property 'pageSize'; nested exception is java.lang.NumberFormatException: For input string: "10,10"]

……………………一堆看不懂的东西！！！！！实际好像没啥用的提示

##### 解决【有疑惑】【重要】

将返回信息从**return** **new** ModelAndView("AdministratorController.do?method=userManagementInfo&pageNo="+ab.getPageNo()+ "&pageSize=" + ab.getPageSize()+ "&queryStr=" + ab.getQueryStr());

改为：**return** **new** ModelAndView("AdministratorController.do?method=userManagementInfo");

猜测是由于和struts1不一样，不能重复给bean赋值？？？

**spring MVC接收参数是字符串数组对象的形式**

#### 2、首页（不经过spring MVC）的页面国际化切换无效的问题

##### 解决

1. 采用双首页，默认的首页放在项目根下，国际化切换后的首页在WEB-INF的bank文件夹下，其他页面也都放入WEB-INF的bank文件夹下。并增加或修改相应的action类的方法，以实现页面跳转。
2. 运行发现出错：

**Type** Status Report

**Description** The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.

发现是由于在后台使用了重定向redirct，而因此找不到WEB-INF下的文件

修改后无错。

1. 修改语言改变的action类，在登录页面也增加转换方法，跳转到登录页面，

因为退出后跳转到登录页面为默认语言。

## 6.3.0 实现登录验证

1.想用排除相应路径方法做，结果不管是标签配置还是在验证的action类中设置list集合，从配置中读取排除路径的方法都不行。经过测试发现截取的路径只有到.do（如：http://127.0.0.1:8080/bank6.0/RegisterLoginController.do），而没有方法，即没有method=方法名。思考后发现原来后面是参数，所以截取不到。





2.将RegisterLoginController类拆分为两个类：RegisterLoginController和Page Controller。

RegisterLoginController用来实现登录注册退出及相应页面的方法，而PageController来实现一些敏感页面的跳转实现。

## 6.4.0 异常处理

#### 1．异常配置里后面跟的是页面【重要】

<bean

class=*"org.springframework.web.servlet.handler.SimpleMappingExceptionResolver"*>

**<!-- 定义默认的异常处理页面，当该异常类型的注册时使用 -->**

<property name=*"defaultErrorView"* value=*"/WEB-INF/bank/error.jsp"*></property>

**<!-- 定义异常处理页面用来获取异常信息的变量名，默认名为exception -->**

<property name=*"exceptionAttribute"* value=*"ex"*></property>

**<!-- 定义需要特殊处理的异常，用类名或完全路径名作为key，异常页面作为值 -->**

<property name=*"exceptionMappings"*>

<props>

<prop key=*"com.cx.bank.util.AccountOverDrawnException"*>/WEB-INF/bank/error.jsp</prop>

<prop key=*"com.cx.bank.util.InvalidDepositException"*>/WEB-INF/bank/error.jsp</prop>

<prop key=*"com.cx.bank.util.InvalidWithDrawalstException"*>/WEB-INF/bank/error.jsp</prop>

<prop key=*"com.cx.bank.util.BankPublicException"*>/WEB-INF/bank/error.jsp</prop>

</props>

</property>

</bean>

#### 2.将抛出的异常里的信息由国际化资源文件的key改写成中文信息

#### 3.SQLFileDaoImp类登录时，如果用户不存在出现下标越界异常

UserBean ub = (UserBean)**this**.getHibernateTemplate().executeFind(**new** HibernateCallback() {

**public** Object doInHibernate(Session session) **throws** HibernateException, SQLException {

**return** session.createQuery("FROM UserBean u WHERE u.username = ?").setParameter(0, username).list();

}

}).get(0); **//当数据库不存在该用户时，出现数组下标越界异常**

**改为：**

// 数据库操作

List list = **null**;

list = **this**.getHibernateTemplate().executeFind(**new** HibernateCallback() {

**public** Object doInHibernate(Session session) **throws** HibernateException, SQLException {

**return** session.createQuery("FROM UserBean u WHERE u.username = ?").setParameter(0, username).list();

}

});

**由判断UserBean改为判断List**

#### 4.注册页面增加密码不得小于6位的js判断

#### 5.文件存储优化

由于properties文件共用一个文件夹，造成用户数据和系统数据混淆。删掉了JDBC连接池及其配置文件，由于早已不在使用持久层工厂并早已删除了工厂类，删除了工厂的配置文件classinfo.properties。并将注册的模板文件放在了system子文件夹下：FileInputStream in = **new** FileInputStream(purl + "\\system\\Bank.properties");

# 7.0 SpringMVC+Spring3+MyBatis

## 导入的架包：【重要】

### spring的架包：

**spring-framework-3.0.0.RELEASE\dist\\*.jar**

**spring-framework-2.0-with-dependencies文件夹下**

**SPRING\_HOME/lib/jakarta-commons下commons-logging.jar**

**SPRING\_HOME/lib/aspectj/\*.jar 切面类相关的库**

**SPRING\_HOME/lib/cglib/\*.jar CGLIB库**

**SPRING\_HOME/lib/aopalliance/aopalliance.jar**

### 数据库：

**mysql-connector-java-5.1.37-bin.jar**

### 标签：

**jstl.jar和standard.jar**

### MyBatis的架包

导入Apache Commons几个jar包

**spring-framework-2.0-with-dependencies\lib\jakarta-commons文件夹下**

**commons-dbcp-1.4.jar**

**commons-pool-1.5.6.jar**

2. 导入日志系统门面几个jar包

mybatis-eng\mybatis-3.2.2\lib下：

slf4j-api-1.6.1.jar :核心API

slf4j-log4j12-1.6.1.jar :log4j实现

log4j-1.2.15.jar

3.导入支持注解的几个jar包---找不到从老师的程序中取得

• persistence.jar

• hibernate-commons-annotations.jar -----找不到，没有导入，实际项目运行也无错。

spring-framework-2.0-with-dependencies\lib\hibernate文件夹下

• hibernate-annotations.jar

• hibernate-entitymanager.jar包

4.导入Mybatis几个jar包

mybatis的jar包

1)**mybatis-3.2.2.jar**

2)mybatis-spring-1.2.1.jar(和spring的集成包）

### 7.0.1

#### 1.配置*mapperLocations*扫描多个路径，使用list标签【重要】

<property name=*"mapperLocations"*>

<list>

<value>classpath:com/cx/bank/model/\*.xml</value>

<value>classpath:com/cx/administrator/model/\*.xml</value>

</list>

</property>

## 7.1.0 完成了集成

### 说明

1. 完成了集成
2. 修改并完善了注释

### 待完善

1. 去除数据库密码字段
2. 增加注册页面并国际化

### 错误日志

#### 第一次启动报错

[**7.1.x启动报错的全部信息01.txt**](错误日志/7.1.x启动报错的全部信息01.txt)

##### 部分错误信息：

十月 08, 2018 10:09:50 下午 org.apache.jasper.servlet.TldScanner scanJars

信息: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation time.

十月 08, 2018 10:09:50 下午 org.apache.catalina.core.ApplicationContext log

信息: Initializing Spring root WebApplicationContext

22:09:52,308 ERROR ContextLoader:208 - Context initialization failed

org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'administratorController': Injection of resource dependencies failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'administratorManagerImp1': Injection of resource dependencies failed; nested exception is **java.lang.NoClassDefFoundError: org/apache/ibatis/session/SqlSession**

………………………………

##### 解决

发现忘记导**mybatis-3.2.2.jar**

#### 第二次启动报错

[**7.1.x启动报错的全部信息02.txt**](错误日志/7.1.x启动报错的全部信息02.txt)

##### 部分错误信息：

十月 08, 2018 10:20:57 下午 org.apache.jasper.servlet.TldScanner scanJars

信息: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation time.

十月 08, 2018 10:20:57 下午 org.apache.catalina.core.ApplicationContext log

信息: Initializing Spring root WebApplicationContext

22:20:59,743 ERROR ContextLoader:208 - Context initialization failed

org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'administratorController': Injection of resource dependencies failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'administratorManagerImp1': Injection of resource dependencies failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'mapper' defined in ServletContext resource [/WEB-INF/applicationContext-common.xml]: Cannot resolve reference to bean 'sqlSessionFactory' while setting bean property 'sqlSessionFactory'; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'sqlSessionFactory' defined in ServletContext resource [/WEB-INF/applicationContext-common.xml]: Invocation of init method failed; nested exception is org.springframework.core.NestedIOException: Failed to parse mapping resource: '**file [D:\Tomcat 8.5\webapps\bank7.0\WEB-INF\classes\com\cx\bank\model\LogBean.xml]**'; nested exception is java.lang.RuntimeException: Error parsing Mapper XML. Cause: org.apache.ibatis.builder.BuilderException: Error resolving class. Cause: org.apache.ibatis.type.TypeException: **Could not resolve type alias 'LogBean'**. Cause: java.lang.ClassNotFoundException: Cannot find class: LogBean

at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.postProcessPropertyValues(CommonAnnotationBeanPostProcessor.java:297)

##### 解决

1.发现此语句WHERE userName='#{queryStr}' AND logType='{type}'的logType='{type}'少了“#”，加上后错误不变。

2.发现错误信息中有一句**Could not resolve type alias 'LogBean'，于是将配置中的LogBean改为*com.cx.bank.model.LogBean。***错误改变，下一个类QueryObject类报同样的错误。应该是没有找到类引起的，所有的类都要带路径全称。

#### 第三次启动报错：和之前同样的错误，只是是配置文件中的东西找不到

[**7.1.x启动报错的全部信息03.txt**](错误日志/7.1.x启动报错的全部信息03.txt)

##### 部分错误信息：

org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'administratorController': Injection of resource dependencies failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'administratorManagerImp1': Injection of resource dependencies failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'mapper' defined in ServletContext resource [/WEB-INF/applicationContext-common.xml]: Cannot resolve reference to bean 'sqlSessionFactory' while setting bean property 'sqlSessionFactory'; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'sqlSessionFactory' defined in ServletContext resource [/WEB-INF/applicationContext-common.xml]: Invocation of init method failed; nested exception is org.springframework.core.NestedIOException: Failed to parse mapping resource: 'file [D:\Tomcat 8.5\webapps\bank7.0\WEB-INF\classes\com\cx\bank\model\LogBean.xml]'; nested exception is java.lang.RuntimeException: Error parsing Mapper XML. Cause: org.apache.ibatis.builder.BuilderException: Error resolving class. Cause: org.apache.ibatis.type.TypeException: Could not resolve type alias **'resultListLogs'**. Cause: java.lang.ClassNotFoundException: Cannot find class: **resultListLogs**

……………………

##### 解决：【重要】

1.将**resultListLogs变为*com.cx.bank.model.*resultListLogs，依旧报同样的错误**

2.将<select id=*"selectAllLogInformation"* parameterType=*"com.cx.bank.model.QueryObject"* resultType=*"resultListLogs"*>**的resultType改为resultMap，异常改变，为其他的类报同样的异常。其他的地方同样操作即可。**

#### 第四次启动报错：

[**7.1.x启动报错的全部信息04.txt**](错误日志/7.1.x启动报错的全部信息04.txt)

##### 部分错误信息：

十月 08, 2018 11:18:59 下午 org.apache.catalina.core.StandardContext filterStart

严重: Exception starting filter [OpenSessionInViewFilter]

java.lang.NoClassDefFoundError: org/hibernate/FlushMode

at org.springframework.orm.hibernate3.support.OpenSessionInViewFilter.<init>(OpenSessionInViewFilter.java:102)

…………………………

at java.lang.Thread.run(Thread.java:748)

Caused by: java.lang.ClassNotFoundException: org.hibernate.FlushMode

…………………………

##### 解决：导入hibernate3.jar。发现是因为OpenSessionInViewFilter需要，但是升级到mybatis后，并不需要对Hibernate的Session进行管理，所以正确的解决方式是将这个拦截器注释掉，不导hibernate3.jar，再次启动也无异常。

#### 页面无图片（发现是忘记复制图片文件夹），登录出异常，跳转到错误页面，错误信息为：

### Error querying database. Cause: java.sql.SQLException: Parameter index out of range (1 > number of parameters, which is 0). ### The error may exist in file [D:\Tomcat 8.5\webapps\bank7.0\WEB-INF\classes\com\cx\bank\model\UserBean.xml] ### The error may involve com.cx.bank.dao.OperationMapper.selectUserByName-Inline ### The error occurred while setting parameters ### SQL: SELECT \* FROM `users` WHERE username = '?' ### Cause: java.sql.SQLException: Parameter index out of range (1 > number of parameters, which is 0). ; SQL []; Parameter index out of range (1 > number of parameters, which is 0).; nested exception is java.sql.SQLException: Parameter index out of range (1 > number of parameters, which is 0).

##### 解决

<!-- 根据用户名查询数据的SQL语句 -->

<select id=*"selectUserByName"* parameterType=*"String"* resultType=*"com.cx.bank.model.UserBean"*>

SELECT \* FROM `users` WHERE username = '#{username}'

</select>

将'#{username}'的单引号去掉后，没有异常信息，依旧失败。

经过排查，应该是在logManager.storageLog(log);存储日志的配置里有单引号或者其他错误。发现是LogManagerImp1的mapper属性没有加注解@Resource(name="mapper")

#### 一进交易记录就跳转到错误页面，没有错误信息。

最终发现是没有创建查询对象，空指针异常。QueryObject qo = **null**; qo.setId(userId);。所有的QueryObject qo全是null，简直吐血。

错误页面没有提示是因为在错误页面中仅仅输出了异常消息${ex.message }。可在调试的时候为了方便调试，改为${ex.message } ${ex }

#### 一进交易记录报错：

[**mybatis xml文件中的大于、小于、及like模糊查询的写法**](https://blog.csdn.net/wangshuang1631/article/details/53488188)

[**关于mybatis中llike模糊查询中#和$的使用**](https://blog.csdn.net/u010398771/article/details/70768280)

错误记录： [**错误日志\交易记录语句错误日志（全）.txt**](错误日志/交易记录语句错误日志（全）.txt)

##### 报错语句为：AND (toName LIKE '%#{queryStr}%' OR transactionType LIKE '%#{queryStr}%')，部分错误信息如下：

………… (toName LIKE '%?%' OR transactionType LIKE '%?%') …………### Cause: java.sql.SQLException: Parameter index out of range (4 > number of parameters, which is 3). ; SQL []; Parameter index out of range (4 > number of parameters, which is 3).; nested exception is java.sql.SQLException: Parameter index out of range (4 > number of parameters, which is 3).

##### 2.去掉单引号后，报错语句为：AND (toName LIKE %#{queryStr}% OR transactionType LIKE %#{queryStr}%)，部分错误信息如下：

…………(toName LIKE %?% OR transactionType LIKE %?%) …………### Cause: com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'WHERE (toName LIKE %''% OR transactionType LIKE %''%) ORDER BY transac' at line 7 ; bad SQL grammar []; nested exception is com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'WHERE (toName LIKE %''% OR transactionType LIKE %''%) ORDER BY transac' at line 7

##### 3.改为：AND (toName LIKE "%"#{queryStr}"%" OR transactionType LIKE "%"#{queryStr}"%")，部分错误信息如下：

check the manual that corresponds to your MySQL server version for the right syntax to use near 'WHERE (toName LIKE "%"''"%" OR transactionType LIKE "%"''"%") ORDER BY' at line 7

##### 4.改为：AND (toName LIKE "%"'#{queryStr}'"%" OR transactionType LIKE "%"'#{queryStr}'"%")，部分错误信息如下：

### Cause: java.sql.SQLException: Parameter index out of range (4 > number of parameters, which is 3). ; SQL []; Parameter index out of range (4 > number of parameters, which is 3).; nested exception is java.sql.SQLException: Parameter index out of range (4 > number of parameters, which is 3).

##### 5.改为：AND (toName LIKE %${queryStr}% OR transactionType LIKE %${queryStr}%)，部分错误信息如下：

check the manual that corresponds to your MySQL server version for the right syntax to use near 'WHERE (toName LIKE %% OR transactionType LIKE %%) ORDER BY transaction' at line 7

##### 6.改为：AND (toName LIKE '%${queryStr}%' OR transactionType LIKE '%${queryStr}%')，部分错误信息如下：

(toName LIKE '%%' OR transactionType LIKE '%%') ORDER BY transac' at line 7

##### 7.改为：AND (toName LIKE concat('%',#{queryStr},'%') OR transactionType LIKE concat('%',#{queryStr},'%'))，依旧不行

##### 8.不想再试，直接在参数中加入%。不管加不加单引号，依旧错…………

##### 9.以上错误推翻部分，在xml文件中去掉<where>标签后成功，因为<where>自动加上了一个where

#### 一进交易记录报错的总结：【重要】

##### 可用的方法

**CONCAT('%','${queryStr}','%') 最好用这种**

**'%${queryStr}%'**

**'%'#{queryStr}'%' 这种方法参数为空时，查不到记录**

**直接在参数中加%再传到sql语句中**

##### <where>标签相当于加where，在有where时不要用<where>标签。并且最好把语句写在<![CDATA[]]>里

#### 管理员没带参数查所有用户信息，没有记录。

在xml中增加判断字符串为空串的操作。<if test=*"queryStr != null and queryStr != ''"*>。**依旧没有记录。**

因为list为null，把mapper.selectAllUserInformation(qo);改为list = mapper.selectAllUserInformation(qo);

[**spring集成mybatis后，打印SQL语句**](https://www.cnblogs.com/qlong8807/p/5580424.html)

之后查日志也是没有记录，但是是因为没有在xml中增加判断字符串为空串的操作。

#### 数据时间格式问题：

此前显示的时间是英文格式的，把时间属性的类型从java.util.Date改为java.sql.Timestamp后，显示为相要的格式：2018-07-15 18:44:13.0。但是最后多了一个0。

创建Timestamp对象时从Timestamp time2 = new Timestamp(**new Date).getTime()**);改为new Timestamp(**System.currentTimeMillis()**);，提高效率

## 7.2.0 系统完善

1. 去除数据库密码字段
2. 修改查询记录可查询金额
3. 增加了注册页面，给用户增加真实姓名、性别、邮箱地址属性。并完善美化了页面。
4. 给图片按页面重命名
5. 启用视图定位器，发现对方法跳方法的找不到路径，于是又回复原来不启用的状态

## 7.3.0 后期完善

### 1.在web.xml中增加Session的时间限制，3分钟无操作将自动销毁Session

<!-- 设置3分钟没有操作自动销毁Session -->

<session-config>

<session-timeout>3</session-timeout>

</session-config>

### 2.在登录验证拦截器（com.cx.bank.filter.IdentityValidateFilter）中将跳转优化为跳出框架的跳转。【重要】

//response.sendRedirect(request.getContextPath()+"/index.jsp");//出现页面重复

//后台跳出框架，避免页面重复

**response.getWriter().print("<script>parent.window.location.href='"+request.getContextPath()+"/index.jsp"+"'</script>");**

### 3.其他的一些细节优化，如登录的角色信息国际化、银行更名为简约银行系统等。