Integration of Hibernate with Spring

Pom.xml:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>SpringHibernateWeb</groupId>
<artifactId>SpringHibernateWeb</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>war</packaging>
<name>SpringHibernateWeb Maven Webapp</name>
<!-- FIXME change it to the project's website -->
<url>http://www.example.com</url>
<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<maven.compiler.source>1.7</maven.compiler.source>
<maven.compiler.target>1.7</maven.compiler.target>
<org.springframework.version>3.0.5.RELEASE</org.springframework.version>
</properties>
<repositories>
<repository>
<id>JBoss repository</id>
<url>http://repository.jboss.org/nexus/content/groups/public/</url>
</repository>
</repositories>
<dependencies>
<dependency>
```

```xml
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>4.11</version>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.hibernate.javax.persistence</groupId>
<artifactId>hibernate-jpa-2.1-api</artifactId>
<version>1.0.0.Final</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-core</artifactId>
<version>${org.springframework.version}</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-expression</artifactId>
<version>${org.springframework.version}</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-beans</artifactId>
<version>${org.springframework.version}</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-context</artifactId>
<version>${org.springframework.version}</version>
</dependency>
<dependency>
```

```xml
<groupId>org.springframework</groupId>
<artifactId>spring-context-support</artifactId>
<version>${org.springframework.version}</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-jdbc</artifactId>
<version>${org.springframework.version}</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-orm</artifactId>
<version>${org.springframework.version}</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-web</artifactId>
<version>${org.springframework.version}</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-webmvc</artifactId>
<version>${org.springframework.version}</version>
</dependency>
<dependency>
<groupId>log4j</groupId>
<artifactId>log4j</artifactId>
<version>1.2.16</version>
<scope>runtime</scope>
</dependency>
<dependency>
```

```xml
<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>3.6.3.Final</version>

</dependency>

<dependency>

<groupId>javassist</groupId>

<artifactId>javassist</artifactId>

<version>3.12.1.GA</version>

</dependency>

<dependency>

<groupId>taglibs</groupId>

<artifactId>standard</artifactId>

<version>1.1.2</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>commons-dbcp</groupId>

<artifactId>commons-dbcp</artifactId>

<version>1.4</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.9</version>

</dependency>

<dependency>

<groupId>commons-configuration</groupId>

<artifactId>commons-configuration</artifactId>

<version>1.9</version>

</dependency>

</dependencies>
```

```xml
<build>
<finalName>SpringHibernateWeb</finalName>
<pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be
moved to parent pom) -->
<plugins>
<plugin>
<artifactId>maven-clean-plugin</artifactId>
<version>3.1.0</version>
</plugin>
<!-- see http://maven.apache.org/ref/current/maven-
core/defaultbindings.html#Plugin_bindings_for_war_packaging -->
<plugin>
<artifactId>maven-resources-plugin</artifactId>
<version>3.0.2</version>
</plugin>
<plugin>
<artifactId>maven-compiler-plugin</artifactId>
<version>3.8.0</version>
</plugin>
<plugin>
<artifactId>maven-surefire-plugin</artifactId>
<version>2.22.1</version>
</plugin>
<plugin>
<artifactId>maven-war-plugin</artifactId>
<version>3.2.2</version>
</plugin>
<plugin>
<artifactId>maven-install-plugin</artifactId>
<version>2.5.2</version>
</plugin>
```

```
<plugin>

<artifactId>maven-deploy-plugin</artifactId>

<version>2.8.2</version>

</plugin>

</plugins>

</pluginManagement>

</build>

</project>
```

Web.xml:

```
<web-app

xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"

version="3.0">

<display-name>Archetype Created Web Application</display-name>

<display-name>Archetype Created Web Application</display-name>

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>eproduct</servlet-name>

<servlet-class>

org.springframework.web.servlet.DispatcherServlet

</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>eproduct</servlet-name>

<url-pattern>/</url-pattern>
```

```xml
</servlet-mapping>
<context-param>
<param-name>contextConfigLocation</param-name>
<param-value>/WEB-INF/eproduct-servlet.xml</param-value>
</context-param>
<listener>
<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
</web-app>
```

eproduct-servlet.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:p="http://www.springframework.org/schema/p"
xmlns:tx="http://www.springframework.org/schema/tx"
xsi:schemaLocation="
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.0.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.0.xsd">
<context:annotation-config/>
<context:component-scan base-package="com.ecommerce.controller" />
<bean id="jspViewResolver"
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
<property name="viewClass"
```

```xml
        value="org.springframework.web.servlet.view.JstlView"></property>
        <property name="prefix" value="/WEB-INF/view/"></property>
        <property name="suffix" value=".jsp"></property>
    </bean>
    <bean id="messageSource"
        class="org.springframework.context.support.ReloadableResourceBundleMessageSource">
        <property name="basename" value="classpath:messages"></property>
        <property name="defaultEncoding" value="UTF-8"></property>
    </bean>
    <bean id="propertyConfigurer"
        class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"
        p:location="/WEB-INF/jdbc.properties"></bean>
    <bean id="dataSource"
        class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close"
        p:driverClassName="${jdbc.driverClassName}"
        p:url="${jdbc.databaseurl}" p:username="${jdbc.username}"
        p:password="${jdbc.password}"></bean>
    <bean id="sessionFactory"
        class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
        <property name="dataSource" ref="dataSource"></property>
        <property name="configLocation">
        <value>classpath:hibernate.cfg.xml</value>
        </property>
        <property name="configurationClass">
        <value>org.hibernate.cfg.AnnotationConfiguration</value>
        </property>
        <property name="hibernateProperties">
        <props>
        <prop key="hibernate.dialect">${jdbc.dialect}</prop>
        <prop key="hibernate.show_sql">true</prop>
        </props>
```

```xml
    </property>

  </bean>

  <bean id="eproductDAO" class="com.ecommerce.dao.EProductDAO"></bean>

  <tx:annotation-driven />

  <bean id="transactionManager"

    class="org.springframework.orm.hibernate3.HibernateTransactionManager">

    <property name="sessionFactory" ref="sessionFactory"></property>

  </bean>

</beans>
```

jdbc.properties:

```properties
jdbc.driverClassName=com.mysql.jdbc.Driver

jdbc.dialect=org.hibernate.dialect.MySQLDialect

jdbc.databaseurl=jdbc:mysql://127.0.0.1:3306/ecommerce

jdbc.username=root

jdbc.password=Nithanya@1
```

Index.jsp:

```html
<html>

<body>

<h2>Spring With Hibernate</h2>

<a href="productList.jsp">Product List</a>

</body>

</html>
```

ProductList.jsp:

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<%@taglib uri="http://www.springframework.org/tags" prefix="spring"%>

<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>

<html>
```

```
<head>
<title>Spring With Hibernate</title>
</head>
<body>
<h3>Product List </h3>
<c:if test="${!empty productList}">
<table class="data">
<tr>
<th>Name</th>
<th>Price</th>
<th>Date Added</th>
</tr>
<c:forEach items="${productList}" var="product">
<tr>
<td>${product.name} </td>
<td>${product.price}</td>
<td>${product.date_added}</td>
</tr>
</c:forEach>
</table>
</c:if>
</body>
</html>
```

Hibernate.cfg.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<mapping class="com.ecommerce.entity.EProductEntity"></mapping>
```

</session-factory>

</hibernate-configuration>

EProductEntity.java:

```java
package com.ecommerce.entity;

import java.math.BigDecimal;

import java.util.Date;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity
@Table(name= "eproduct")
public class EProductEntity {

 @Id @GeneratedValue
 @Column(name = "ID")
 private long ID;


 @Column(name = "name")
 private String name;


 @Column(name = "price")
 private BigDecimal price;


 @Column(name = "date_added")
 private Date dateAdded;


 public long getID() {return this.ID; }

 public String getName() { return this.name;}

 public BigDecimal getPrice() { return this.price;}
```

public Date getDateAdded() { return this.dateAdded;}

public void setID(long id) { this.ID = id;}

public void setName(String name) { this.name = name;}

public void setPrice(BigDecimal price) { this.price = price;}

public void setDateAdded(Date date) { this.dateAdded = date;}

}

EProductDAO.java:

```
package com.ecommerce.dao;

import java.util.List;

import org.hibernate.SessionFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

import com.ecommerce.entity.EProductEntity;

@Repository

public class EProductDAO {

@Autowired

private SessionFactory sessionFactory;

@SuppressWarnings("unchecked")

public List<EProductEntity> getAllProducts() {

return this.sessionFactory.getCurrentSession().createQuery("from EProducts").list();

}

}
```

EProductController.java:

```
package com.ecommerce.Controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;
```

```java
import com.ecommerce.dao.EProductDAO;

import com.ecommerce.entity.EProductEntity;

@Controller

public class EProductController {

@Autowired

private EProductDAO eproductService;

@RequestMapping(value = "/productList", method = RequestMethod.GET)

public String listProducts(ModelMap map)

{

map.addAttribute("eproduct", new EProductEntity());

map.addAttribute("productList", eproductService.getAllProducts());

return "productList";

}

}
```