Struts 2 + Hibernate integration with "Full Hibernate Plugin"

Written on July 20, 2010 at 4:53 pm by mkyong | Under Struts2 category

Download it – Struts2-Hibernate-FullHibernatePluginExample.zip

In last Struts 2 + Hibernate integration example, it use the servlet context listener to play around with the Hibernate session, and it did very well to integrate the Struts 2 and Hibernate together.

But, there are always something to improve 🙂 In this tutorial, it show a complete example to integrate the Struts 2 and Hibernate by using a Struts2's plugin named "Full Hibernate Plugin", version 2.2GA, created by jyoshiriro.

See the summary of integration steps :

1. Put the "**Full Hibernate Plugin**" jar in the project class path.
2. Use annotations "**@SessionTarget**" to inject the Hibernate session; While "**@TransactionTarget**" to inject the Hibernate transaction.
3. In **struts.xml**, make the package extends "**hibernate-default**", instead of the default stack.

See the relationship :

```
Struts 2 <-- (Full Hibernate Plugin) ---> Hibernate <-----> Database
```

This tutorial is an update version from the previous Struts 2 + Hibernate integration example (servlet context listerner), So the JSP and Hibernate configuration are almost same, just the integration part is a bit different, try to compare both to spot the different.

1. Project structure

See this full project folder structure.

This chrome webpage is not available

The webpage at http://www.facebook.com/plugins/likebox.php?href=http%3A%2F%2Fwww.facebook.com%2Fpages%2FMkyon

Latest Posts

JAX-WS Tutorials March 24, 2011

Hibernate Error – Collection has neither generic type or OneToMany.targetEntity() March 24, 2011

BasicDataSource causing java.util.ConcurrentModificationException in WebSphere March 24, 2011

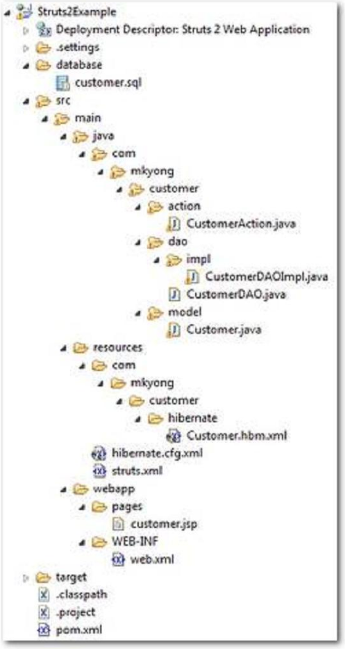
JAX-WS + Spring integration example March 24, 2011

How to enable Subversion (SVN) in Eclipse IDE March 23, 2011

Unable to locate Spring NamespaceHandler for XML schema namespace [http://jax-ws.dev.java.net/spring/servlet] March 23, 2011

Maven : generics are not supported in -source 1.3 March 22, 2011

JAX-WS + Java Web Application Integration Example March 14, 2011



```
Struts2Example
  Deployment Descriptor: Struts 2 Web Application
  .settings
  database
    customer.sql
  src
    main
      java
        com
          mkyong
            customer
              action
                CustomerAction.java
              dao
                impl
                  CustomerDAOImpl.java
                CustomerDAO.java
              model
                Customer.java
      resources
        com
          mkyong
            customer
              hibernate
                Customer.hbm.xml
        hibernate.cfg.xml
        struts.xml
      webapp
        pages
          customer.jsp
        WEB-INF
          web.xml
  target
  .classpath
  .project
  pom.xml
```

2. MySQL table script

Customer's table script.

```
DROP TABLE IF EXISTS `mkyong`.`customer`;
CREATE TABLE  `mkyong`.`customer` (
  `CUSTOMER_ID` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
  `NAME` varchar(45) NOT NULL,
  `ADDRESS` varchar(255) NOT NULL,
  `CREATED_DATE` datetime NOT NULL,
  PRIMARY KEY (`CUSTOMER_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8;
```

3. Get "Full Hibernate Plugin" and Dependencies

Get all the Struts2, Hibernate, "**Full Hibernate Plugin**" and MySQL dependency libraries. Since the "**Full Hibernate Plugin**" is not support Maven, you need to download it from the official website and include into your Maven local repository manually.

1. Download "Full Hibernate Plugin".
2. Put the downloaded jar into c: drive and use following Maven's command to include it into Maven local repository.

```
mvn install:install-file -Dfile=c:\struts2-fullhibernatecore-plugin-2.2-GA.jar
-DgroupId=com.google.code -DartifactId=struts2-fullhibernatecore-plugin
-Dversion=2.2 -Dpackaging=jar
```

3. Link it with the follow Maven coordinate.

```
        <dependency>
                <groupId>com.google.code</groupId>
                <artifactId>struts2-fullhibernatecore-plugin</artifactId>
                <version>2.2</version>
        </dependency>
```

Here's all the dependency libraries in this tutorial :

pom.xml

Remote Debugging with Eclipse + WebSphere 7 March 14, 2011

How to increase WebSphere JVM memory March 8, 2011

Latest Comments

How to verfiy a software or file checksum value (SHA1, MD5, CRC…) - March 30, 2011
Gary1122 : Jacks is excellent. I always knew what it was and saw it, but never could verify before - under the ...

Struts 2 + Spring + Hibernate integration example - March 30, 2011
saleem : Sorry for late reply I don't know where and what are the jar file to be added to this project...

Hibernate – One-to-many relationship example - March 30, 2011
Gregg : In your test code posted above, you do: session.save(stockDailyRecords); session.save(stock); Howeve...

JSF : Page Forward vs Page Redirect - March 29, 2011
Van Long Nguyen : Thanks you very much. My problem solved!...

JSF 2.0 + Spring integration example - March 29, 2011
sapardi : Your error problem is same with me. In my way, i replace application server from tomcat to be glassf...

```
//...
        <!-- Struts 2 -->
        <dependency>
                <groupId>org.apache.struts</groupId>
                <artifactId>struts2-core</artifactId>
                <version>2.1.8</version>
        </dependency>

        <!-- MySQL database driver -->
        <dependency>
                <groupId>mysql</groupId>
                <artifactId>mysql-connector-java</artifactId>
                <version>5.1.9</version>
        </dependency>

        <!-- Struts 2 Hibernate Plugins -->
        <dependency>
                <groupId>com.google.code</groupId>
                <artifactId>struts2-fullhibernatecore-plugin</artifactId>
                <version>2.2</version>
        </dependency>

        <!-- Log4j logging (Struts 2 Hibernate Plugins dependency) -->
        <dependency>
                <groupId>log4j</groupId>
                <artifactId>log4j</artifactId>
                <version>1.2.9</version>
        </dependency>

        <!-- Hibernate validator (Struts 2 Hibernate Plugins dependency) -->
        <dependency>
               <groupId>org.hibernate</groupId>
               <artifactId>hibernate-validator</artifactId>
               <version>3.1.0.GA</version>
        </dependency>

        <!-- slf4j logging (Struts 2 Hibernate Plugins dependency) -->
        <dependency>
               <groupId>org.slf4j</groupId>
               <artifactId>slf4j-api</artifactId>
               <version>1.6.1</version>
        </dependency>

        <!-- Hibernate core -->
        <dependency>
                <groupId>org.hibernate</groupId>
                <artifactId>hibernate</artifactId>
                <version>3.2.7.ga</version>
        </dependency>

        <!-- Hibernate core library dependency start -->
        <dependency>
                <groupId>dom4j</groupId>
                <artifactId>dom4j</artifactId>
                <version>1.6.1</version>
        </dependency>

        <dependency>
                <groupId>commons-logging</groupId>
                <artifactId>commons-logging</artifactId>
                <version>1.1.1</version>
        </dependency>

        <dependency>
                <groupId>commons-collections</groupId>
                <artifactId>commons-collections</artifactId>
                <version>3.2.1</version>
        </dependency>

        <dependency>
                <groupId>cglib</groupId>
                <artifactId>cglib</artifactId>
                <version>2.2</version>
        </dependency>
        <!-- Hibernate core library dependency end -->

        <!-- Hibernate query library dependency start -->
        <dependency>
                <groupId>antlr</groupId>
                <artifactId>antlr</artifactId>
                <version>2.7.7</version>
        </dependency>
        <!-- Hibernate query library dependency end -->
//...
```

The "**Full Hibernate Plugin**" is required the **Hibernate validator** and **SLF4j** dependency, which is not really make sense, as most of the Java developers still do not use it.

4. Hibernate Stuff

All the Hibernate models and configuration stuff.

Customer.java – Create a class for customer table.

```
package com.mkyong.customer.model;

import java.util.Date;

public class Customer implements java.io.Serializable {

        private Long customerId;
        private String name;
        private String address;
        private Date createdDate;

        //getter and setter methods
}
```

Customer.hbm.xml – Hibernate mapping file for customer.

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!-- Generated 20 Julai 2010 11:40:18 AM by Hibernate Tools 3.2.5.Beta -->
<hibernate-mapping>
    <class name="com.mkyong.customer.model.Customer"
                table="customer" catalog="mkyong">
        <id name="customerId" type="java.lang.Long">
            <column name="CUSTOMER_ID" />
            <generator class="identity" />
        </id>
        <property name="name" type="string">
            <column name="NAME" length="45" not-null="true" />
        </property>
        <property name="address" type="string">
            <column name="ADDRESS" not-null="true" />
        </property>
        <property name="createdDate" type="timestamp">
            <column name="CREATED_DATE" length="19" not-null="true" />
        </property>
    </class>
</hibernate-mapping>
```

hibernate.cfg.xml – Hibernate database configuration file.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.bytecode.use_reflection_optimizer">false</property>
    <property name="hibernate.connection.password">password</property>
    <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/mkyong</property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
    <property name="show_sql">true</property>
    <property name="format_sql">true</property>
    <property name="use_sql_comments">false</property>
    <mapping resource="com/mkyong/customer/hibernate/Customer.hbm.xml" />
  </session-factory>
</hibernate-configuration>
```

5. DAO

Implements DAO design pattern to perform the database operation. In the **CustomerDAOImpl** class, declared both Hibernate session and transaction as class members. During the Struts 2 project initialization, "**Full Hibernate Plugin**" will injects the corresponding Hibernate session and transaction into the class members using **@SessionTarget** and **@TransactionTarget** annotation respectively.

CustomerDAO.java

```
package com.mkyong.customer.dao;

import java.util.List;

import com.mkyong.customer.model.Customer;

public interface CustomerDAO{

        void addCustomer(Customer customer);

        List<Customer> listCustomer();

}
```

CustomerDAOImpl.java

```
package com.mkyong.customer.dao.impl;

import java.util.List;

import org.hibernate.Session;
import org.hibernate.Transaction;

import com.googlecode.s2hibernate.struts2.plugin.annotations.SessionTarget;
import com.googlecode.s2hibernate.struts2.plugin.annotations.TransactionTarget;
import com.mkyong.customer.dao.CustomerDAO;
import com.mkyong.customer.model.Customer;

public class CustomerDAOImpl implements CustomerDAO{

        @SessionTarget
        Session session;

        @TransactionTarget
        Transaction transaction;

        //add the customer
        public void addCustomer(Customer customer){

                session.save(customer);

        }

        //return all the customers in list
        public List<Customer> listCustomer(){

                return session.createQuery("from Customer").list();

        }

}
```

6. Action

In Action class, call the DAO class to perform the database operation.

CustomerAction.java

```
package com.mkyong.customer.action;

import java.util.ArrayList;
import java.util.Date;
import java.util.List;

import com.mkyong.customer.dao.CustomerDAO;
import com.mkyong.customer.dao.impl.CustomerDAOImpl;
import com.mkyong.customer.model.Customer;
import com.opensymphony.xwork2.ActionSupport;
import com.opensymphony.xwork2.ModelDriven;

public class CustomerAction extends ActionSupport
        implements ModelDriven{

        Customer customer = new Customer();
        List<Customer> customerList = new ArrayList<Customer>();
        CustomerDAO customerDAO = new CustomerDAOImpl();

        public String execute() throws Exception {
                return SUCCESS;
        }

        public Object getModel() {
                return customer;
        }

        public List<Customer> getCustomerList() {
                return customerList;
        }

        public void setCustomerList(List<Customer> customerList) {
                this.customerList = customerList;
        }

        //save customer
        public String addCustomer() throws Exception{

                //save it
                customer.setCreatedDate(new Date());
                customerDAO.addCustomer(customer);

                //reload the customer list
                customerList = null;
                customerList = customerDAO.listCustomer();

                return SUCCESS;

        }

        //list all customers
        public String listCustomer() throws Exception{

                customerList = customerDAO.listCustomer();

                return SUCCESS;

        }

}
```

7. JSP page

JSP pages to add and list the customer.

customer.jsp

```
<%@ taglib prefix="s" uri="/struts-tags" %>
<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
<head>
</head>

<body>
<h1>Struts 2 Full Hibernate Plugin example</h1>

<h2>Add Customer</h2>
<s:form action="addCustomerAction" >
  <s:textfield name="name" label="Name" value="" />
  <s:textarea name="address" label="Address" value="" cols="50" rows="5" />
  <s:submit />
</s:form>

<h2>All Customers</h2>

<s:if test="customerList.size() > 0">
<table border="1px" cellpadding="8px">
        <tr>
                <th>Customer Id</th>
                <th>Name</th>
                <th>Address</th>
                <th>Created Date</th>
        </tr>
        <s:iterator value="customerList" status="userStatus">
                <tr>
                        <td><s:property value="customerId" /></td>
                        <td><s:property value="name" /></td>
                        <td><s:property value="address" /></td>
                        <td><s:date name="createdDate" format="dd/MM/yyyy" /></td>
                </tr>
        </s:iterator>
</table>
</s:if>
<br/>
<br/>

</body>
</html>
```

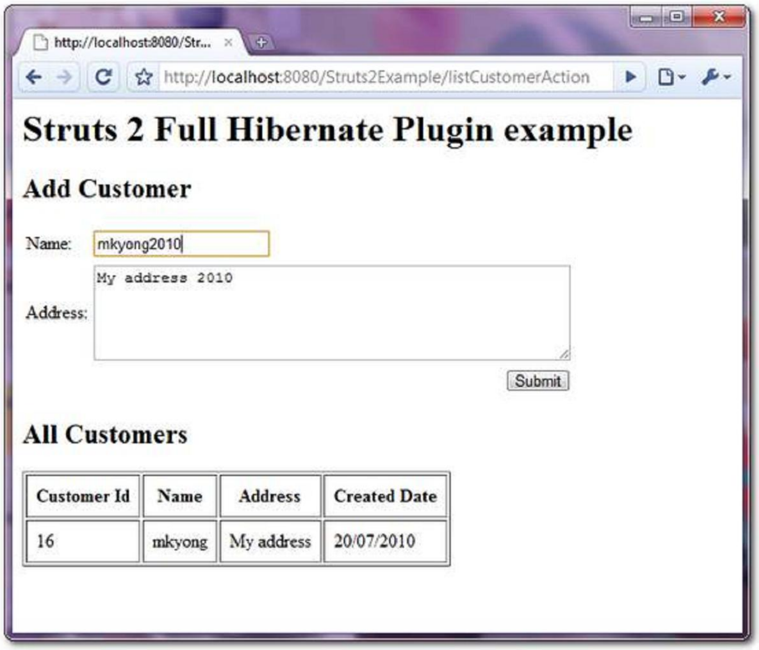
8. struts.xml

Link it all ~ make package extends the "**hibernate-default**" instead of the "**struts-default**".

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
  <constant name="struts.devMode" value="true" />

  <package name="default" namespace="/" extends="hibernate-default">

    <action name="addCustomerAction"
        class="com.mkyong.customer.action.CustomerAction" method="addCustomer" >
       <result name="success">pages/customer.jsp</result>
    </action>

    <action name="listCustomerAction"
        class="com.mkyong.customer.action.CustomerAction" method="listCustomer" >
        <result name="success">pages/customer.jsp</result>
    </action>

</package>
</struts>
```

9. Demo

Access it : http://localhost:8080/Struts2Example/listCustomerAction.action


http://localhost:8080/Str...
http://localhost:8080/Struts2Example/listCustomerAction

Struts 2 Full Hibernate Plugin example

Add Customer

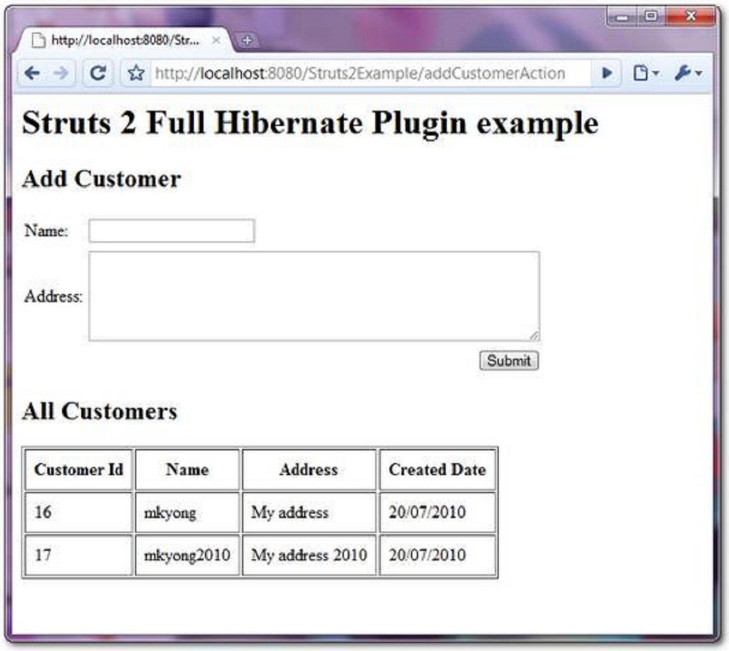
Name: mkyong2010

Address: My address 2010

Submit

All Customers

Customer Id	Name	Address	Created Date
16	mkyong	My address	20/07/2010


http://localhost:8080/Str...
http://localhost:8080/Struts2Example/addCustomerAction

Struts 2 Full Hibernate Plugin example

Add Customer

Name:

Address:

Submit

All Customers

Customer Id	Name	Address	Created Date
16	mkyong	My address	20/07/2010
17	mkyong2010	My address 2010	20/07/2010

Reference

1. Struts 2 Full Hibernate Plugin documentation
2. Struts 2 + Hibernate integration example (servlet context listener)
3. Install library into Maven local repository

Can't find what you're looking for? Try Google Custom Search!

Search

Manage, Audit & Optimize SQL Reporting Services
"Like a flashlight for SSRS"
- testimonial
Download Now!
sqlscrubs

Related Posts

1. Struts 2 + Spring + Hibernate integration example
2. Struts 2 + Hibernate integration example
3. Hibernate Error – Collection has neither generic type or OneToMany.targetEntity()
4. WebSphere 7 & javax/persistence/OneToMany.orphanRemoval() error
5. org.hibernate.AnnotationException: Collection has neither generic type or OneToMany.targetEntity()
6. Hibernate Error : JavaReflectionManager cannot be cast to MetadataProviderInjector
7. java.lang.NoSuchMethodError: org.objectweb.asm.ClassWriter.(Z)V

Popular Posts

1. Top 8 Java people you should know
2. Top 20 Java websites you must visit
3. 10 Java regular expression examples you should know
4. Top 10 open source forums in collection

Popular J2EE Tutorials

JSF2.0	Hibernate Core	JAX-WS 2.x	Spring 2.5.x	Spring MVC
jQuery	Struts 2	Struts 1.x	Java XML	Java RegEx
Java I/O	Maven	jUnit	TestNG	

4 Comments

padmanabha says:
March 22, 2011 at 3:43 pm

Hi …

I get following error when i deploy the application

com.googlecode.s2hibernate.lang.SessionInjectionException: Error! Please, check your JDBC/JDNI Configurations and Database Server avaliability. at /addCustomerAction – Method: com.mkyong.customer.action.CustomerAction.addCustomer() Could not open or inject a Hibernate Session in ValueStack: org.hibernate.HibernateException: Unable to instantiate default tuplizer [org.hibernate.tuple.entity.PojoEntityTuplizer]

```
com.googlecode.s2hibernate.struts2.plugin.interceptors.SessionTransactionInjectorInterceptor.intercept(SessionTransactionInjectorInterceptor.java:177)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
com.googlecode.s2hibernate.struts2.plugin.s2hibernatevalidator.interceptor.HibernateValidatorInterceptor.intercept(HibernateValidatorInterceptor.java:118)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
com.opensymphony.xwork2.interceptor.DefaultWorkflowInterceptor.doIntercept(DefaultWorkflowInterceptor.java:165)
com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:87)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
com.opensymphony.xwork2.interceptor.ConversionErrorInterceptor.intercept(ConversionErrorInterceptor.java:122)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
com.opensymphony.xwork2.interceptor.ParametersInterceptor.doIntercept(ParametersInterceptor.java:195)
com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:87)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
com.opensymphony.xwork2.interceptor.StaticParametersInterceptor.intercept(StaticParametersInterceptor.java:179)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
org.apache.struts2.interceptor.CheckboxInterceptor.intercept(CheckboxInterceptor.java:94)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
org.apache.struts2.interceptor.FileUploadInterceptor.intercept(FileUploadInterceptor.java:235)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
com.opensymphony.xwork2.interceptor.ModelDrivenInterceptor.intercept(ModelDrivenInterceptor.java:89)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
com.opensymphony.xwork2.interceptor.ScopedModelDrivenInterceptor.intercept(ScopedModelDrivenInterceptor.java:130)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
org.apache.struts2.interceptor.ProfilingActivationInterceptor.intercept(ProfilingActivationInterceptor.java:104)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
org.apache.struts2.interceptor.debugging.DebuggingInterceptor.intercept(DebuggingInterceptor.java:267)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
com.opensymphony.xwork2.interceptor.ChainingInterceptor.intercept(ChainingInterceptor.java:126)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
com.opensymphony.xwork2.interceptor.I18nInterceptor.intercept(I18nInterceptor.java:165)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
com.opensymphony.xwork2.interceptor.PrepareInterceptor.doIntercept(PrepareInterceptor.java:138)
com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:87)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
org.apache.struts2.interceptor.ServletConfigInterceptor.intercept(ServletConfigInterceptor.java:164)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
com.opensymphony.xwork2.interceptor.AliasInterceptor.intercept(AliasInterceptor.java:179)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
com.opensymphony.xwork2.interceptor.ExceptionMappingInterceptor.intercept(ExceptionMappingInterceptor.java:176)
com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)
org.apache.struts2.impl.StrutsActionProxy.execute(StrutsActionProxy.java:52)
org.apache.struts2.dispatcher.Dispatcher.serviceAction(Dispatcher.java:488)
org.apache.struts2.dispatcher.ng.ExecuteOperations.executeAction(ExecuteOperations.java:77)
org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter.doFilter(StrutsPrepareAndExecuteFilter.java:91)
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:215)
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:188)
org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:213)
org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:172)
org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:127)
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:117)
org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:108)
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:174)
org.apache.coyote.http11.Http11AprProcessor.process(Http11AprProcessor.java:843)
org.apache.coyote.http11.Http11AprProtocol$Http11ConnectionHandler.process(Http11AprProtocol.java:679)
org.apache.tomcat.util.net.AprEndpoint$Worker.run(AprEndpoint.java:1293)
java.lang.Thread.run(Thread.java:662)
```

Reply

padmanabha says:
March 22, 2011 at 6:08 pm

It got solved…. I added dependency asm to pom.xml.. Thanks for the good example

Reply

ron says:
September 8, 2010 at 9:29 pm

How would you write a JUnit test for the DAO class. I was trying to finf a way to do it but all i can find is JUnit with Spring. I have implemented this hibernate plug-in in my mproject and need to write the Unit test for the DAO classes. thanks in advance.

ron

Reply

Struts 2 + Hibernate integration example says:
July 20, 2010 at 4:54 pm

[...] Struts 2 + Hibernate integration with "Full Hibernate Plugin" [...]

Reply

Leave a Reply

Name (required)

Mail (will not be published) (required)

Website

Submit Comment

About Mkyong.com

Mkyong.com is providing many Java / J2EE web development tutorials, which involve Spring, Hibernate, Struts, Struts2, JSF, Maven, Ant, jQuery, JavaScript...and many other Java de facto technologies.

Subscribe RSS feed

Contact Us

[Advertise With Us]

All Available Tutorials

Java Core Technology - Java RegEx, Java XML, Java I/O, Java Misc
J2EE Frameworks - Hibernate, JSF 2.0, Spring 2.5, Spring MVC, Struts 1.x, Struts 2.x
Web Services - JAX-WS
Build Tools - Maven, Archiva
Unit Test - jUnit, TestNG
Client Scripts - jQuery

Favorites Java Links

Official Java EE 5 Tutorial
Spring 2.5.x documentation
Hibernate core documentation

Friends & Links

JSP Training
JavaScript Training
Best Buy Coupons
China Wholesale
Tomcat Hosting
MySQL Tutorials
PHP Tutorials
TenthOfMarch

Java & Co. Tutorials
Find Free Icons
IT Support Liverpool
China Cell Phone

Java SE 6.0 API documentation
Java EE 6.0 API documentation
Java Secure Socket Extension (JSSE) Reference Guide
JSP home page
JSF home page
Eclipse IDE for Java developer
Maven home page
Ant home page
Struts 1.3 documentation
Struts 2.2 documentation
Maven central repository
Java.Net Maven repository
Martin Fowler