

[JAVA TUTORIAL](#)[#INDEX POSTS](#)[#INTERVIEW QUESTIONS](#)[RESOURCES](#)[HIRE ME](#)[DOWNLOAD ANDROID APP](#)[CONTRIBUTE](#)**Subscribe to Download Java Design Patterns eBook****DOWNLOAD NOW**[HOME](#) » [STRUTS 2](#) » STRUTS2 HIBERNATE INTEGRATION EXAMPLE TUTORIAL

Struts2 Hibernate Integration Example Tutorial

APRIL 2, 2018 BY [PANKAJ](#) — [8 COMMENTS](#)

Struts2 and **Hibernate** both are widely used frameworks in their respective area. Today we will learn how to integrate Struts2 Web application framework with Hibernate ORM framework.

Table of Contents [\[hide\]](#)

1 Struts2 Hibernate

- [1.1 Struts2 Hibernate Example Database Setup](#)
- [1.2 Struts2, Hibernate and MySQL Maven Dependencies](#)
- [1.3 Struts2 Hibernate Example Deployment Descriptor](#)
- [1.4 Entity Bean](#)
- [1.5 Hibernate DAO Implementation Classes](#)
- [1.6 Servlet Context Listener](#)
- [1.7 Struts2 Action Class](#)
- [1.8 Struts2, Hibernate Configuration and Hibernate Mapping XML](#)
- [1.9 Struts2 View Pages](#)
- [1.10 Struts2 Hibernate Application Test](#)
 - [1.10.1 Struts2 Hibernate Login Page](#)
 - [1.10.2 Struts2 Hibernate Welcome Page](#)
 - [1.10.3 Struts2 Hibernate Login Error Page](#)

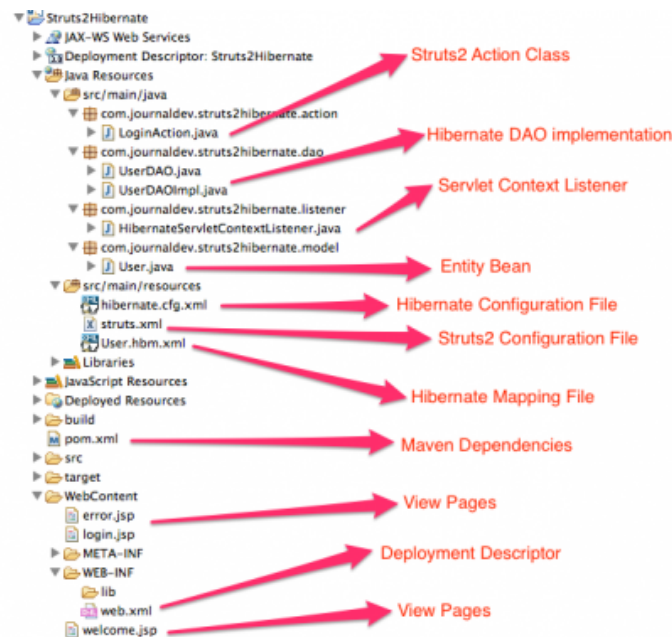
1.11 Struts2 Hibernate Integration Summary

Struts2 Hibernate

Unlike [Spring Hibernate Integration](#), there are no plugins provided by Struts2 framework that we can use. So we would need to use the [Servlet Listener](#) to manage the [Hibernate SessionFactory](#).

Let's see how to integrate Struts2 with Hibernate with a simple [web application example](#).

Our final project structure will look like below image, we will look into each of the component one by one. Start with creating a "Dynamic Web Project" in Eclipse and then convert it to maven project to get the basic skeleton ready for our web application.



Struts2 Hibernate Example Database Setup

We will create a login application, after authentication user will be present to the welcome page where all his information will be displayed. Below SQL script will create the required table and insert data that we will use for authentication.

setup.sql

```
CREATE TABLE `User` (
  `user_id` varchar(10) NOT NULL DEFAULT '',
  `password` varchar(40) NOT NULL DEFAULT '',
  `name` varchar(40) NOT NULL DEFAULT '',
  `email` varchar(40) NOT NULL DEFAULT '',
  PRIMARY KEY (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
INSERT INTO `User` (`user_id`, `password`, `name`, `email`)
VALUES
('pankaj', 'pankaj123', 'Pankaj Kumar', 'pankaj@journaldev.com');
```

Struts2, Hibernate and MySQL Maven Dependencies

We need to add Struts2, Hibernate and MySQL maven dependencies in pom.xml file, final pom.xml file looks like below.

```
<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>Struts2Hibernate</groupId>
  <artifactId>Struts2Hibernate</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>

  <dependencies>
    <dependency>
      <groupId>org.apache.struts</groupId>
      <artifactId>struts2-core</artifactId>
      <version>2.3.15.1</version>
    </dependency>
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>4.3.5.Final</version>
    </dependency>
    <dependency>
```

We are using Hibernate 4 for our example, however same code will work with Hibernate 3 too. Although you might need to do some configuration hibernate 3 specific changes.

Struts2 Hibernate Example Deployment Descriptor

We need to plugin Struts2 framework in our web application by applying `StrutsPrepareAndExecuteFilter` in deployment descriptor.

web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
    <display-name>Struts2Hibernate</display-name>

    <filter>
        <filter-name>struts2</filter-name>
        <filter-
class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-
class>
    </filter>

    <filter-mapping>
        <filter-name>struts2</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>

    <listener>
        <listener-
class>com.journaldev.struts2hibernate.listener.HibernateServletContextListener</listener-

```

Notice that `HibernateServletContextListener` will be used to manage `Hibernate SessionFactory`, we will look it's code later on.

Entity Bean

Our entity bean will also work as the struts2 action class model bean, it looks like below.

User.java

```

package com.journaldev.struts2hibernate.model;

public class User {

    private String id;
    private String name;
    private String pwd;
    private String email;

    public String getName() {
        return name;
    }

```

```
}

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

    public String getPwd() {
        return pwd;
    }

    public void setPwd(String pwd) {
```

I am using XML based mappings, but we can also use JPA annotations based mapping for our entity bean.

Hibernate DAO Implementation Classes

Our UserDao interface declares the method that will be exposed for User entity bean.

UserDAO.java

```
package com.journaldev.struts2hibernate.dao;

import com.journaldev.struts2hibernate.model.User;

public interface UserDao {

    User getUserByCredentials(String userId, String password);
}
```

Hibernate specific DAO implementation looks like below.

UserDAOImpl.java

```
package com.journaldev.struts2hibernate.dao;

import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;

import com.journaldev.struts2hibernate.model.User;

public class UserDaoImpl implements UserDao {
```

```

private SessionFactory sf;

public UserDAOImpl(SessionFactory sf){
    this.sf = sf;
}

@Override
public User getUserByCredentials(String userId, String password) {
    Session session = sf.openSession();
    Transaction tx = session.beginTransaction();
    Query query = session.createQuery("from User where id=:id and

```

Servlet Context Listener

We will initialize Hibernate SessionFactory in the ServletContextListener implementation, when application will be stopped, we will destroy the SessionFactory.

HibernateServletContextListener.java

```

package com.journaldev.struts2hibernate.listener;

import java.net.URL;

import javax.servlet.ServletContextEvent;
import javax.servlet.ServletContextListener;

import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;

public class HibernateServletContextListener implements ServletContextListener {

    @Override
    public void contextDestroyed(ServletContextEvent sce) {
        SessionFactory sf = (SessionFactory)
        sce.getServletContext().getAttribute("SessionFactory");
        sf.close();
    }

    @Override

```

Struts2 Action Class

Our hibernate specific setup is ready, let's move on to Struts2 action class.

LoginAction.java

```
package com.journaldev.struts2hibernate.action;

import javax.servlet.ServletContext;

import org.apache.struts2.util.ServletContextAware;
import org.hibernate.SessionFactory;

import com.journaldev.struts2hibernate.dao.UserDAO;
import com.journaldev.struts2hibernate.dao.UserDAOImpl;
import com.journaldev.struts2hibernate.model.User;
import com.opensymphony.xwork2.Action;
import com.opensymphony.xwork2.ModelDriven;

public class LoginAction implements Action, ModelDriven<User>, ServletContextAware {

    @Override
    public String execute() throws Exception {

        SessionFactory sf = (SessionFactory)
ctx.getAttribute("SessionFactory");
        UserDAO userDAO = new UserDAOImpl(sf);
        User userDB = userDAO.getUserByCredentials(user.getId()),
```

Struts2, Hibernate Configuration and Hibernate Mapping XML

We have following URI configurations in Struts2 configuration file.

struts.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
    "http://struts.apache.org/dtds/struts-2.3.dtd">

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

    <constant name="struts.convention.result.path" value="/"></constant>
```

```

<package name="user" namespace="/User" extends="struts-default">
    <action name="home">
        <result>/login.jsp</result>
    </action>
    <action name="login"
class="com.journaldev.struts2hibernate.action.LoginAction">
        <result name="success">/welcome.jsp</result>
        <result name="error">/error.jsp</result>
    </action>

</package>

```

Hibernate Configuration File looks like below.

hibernate.cfg.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <!-- Database connection properties - Driver, URL, user, password -->
        <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
        <property
name="hibernate.connection.url">jdbc:mysql://localhost/TestDB</property>
        <property name="hibernate.connection.username">pankaj</property>
        <property name="hibernate.connection.password">pankaj123</property>
        <!-- Connection Pool Size -->
        <property name="hibernate.connection.pool_size">1</property>

        <!-- org.hibernate.HibernateException: No CurrentSessionContext
configured! -->
        <property
name="hibernate.current_session_context_class">thread</property>

```

Hibernate Mapping XML file looks like below.

User.hbm.xml


```

<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
  <class name="com.journaldev.struts2hibernate.model.User" table="USER">
    <id name="id" type="java.lang.String">
      <column name="USER_ID" />
      <generator class="assigned" />
    </id>
    <property name="name" type="java.lang.String">
      <column name="NAME" />
    </property>
    <property name="pwd" type="java.lang.String">
      <column name="PASSWORD" />
    </property>
    <property name="email" type="java.lang.String">
      <column name="EMAIL" />
    </property>
  </class>
</hibernate-mapping>

```

Struts2 View Pages

As you can see from struts2 configuration file, we have three view pages. They are simple pages and looks like below.

login.jsp

```

<%@ page language="java" contentType="text/html; charset=US-ASCII"
    pageEncoding="US-ASCII"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<!-- Using Struts2 Tags in JSP -->
<%@ taglib uri="/struts-tags" prefix="s"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=US-ASCII">
<title>Login Page</title>
</head>
<body>
<h3>Welcome User, please login below</h3>
<s:form action="login">

```

```
<s:textfield name="id" label="User ID"></s:textfield>
<s:textfield name="pwd" label="Password" type="password"></s:textfield>
<s:submit value="Login"></s:submit>
</s:form>
</body>
</html>
```

welcome.jsp

```
<%@ page language="java" contentType="text/html; charset=US-ASCII"
    pageEncoding="US-ASCII"%>
<%@ taglib uri="/struts-tags" prefix="s"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=US-ASCII">
<title>Welcome Page</title>
</head>
<body>
<h3>Welcome <s:property value="name"></s:property></h3>

<h3>Your id is <s:property value="id"></s:property></h3>

<h3>Your password is <s:property value="pwd"></s:property></h3>

<h3>Your email id is <s:property value="email"></s:property></h3>

</body>
</html>
```

error.jsp

```
<%@ page language="java" contentType="text/html; charset=US-ASCII"
    pageEncoding="US-ASCII"%>
<%@ taglib uri="/struts-tags" prefix="s"%>

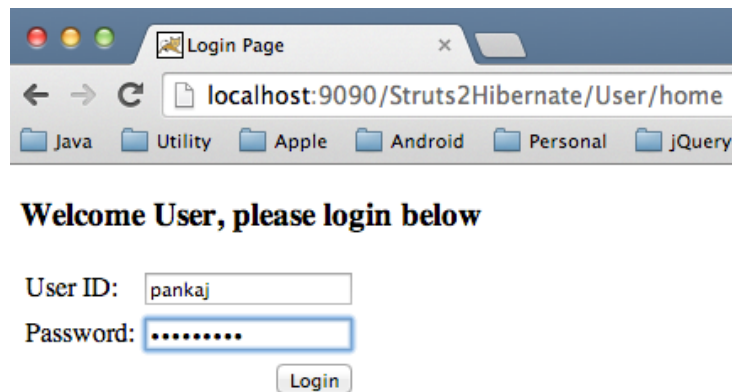
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=US-ASCII">
```

```
<title>Error Page</title>
</head>
<body>
<h4>User Name or Password is wrong</h4>
<s:include value="login.jsp"></s:include>
</body>
</html>
```

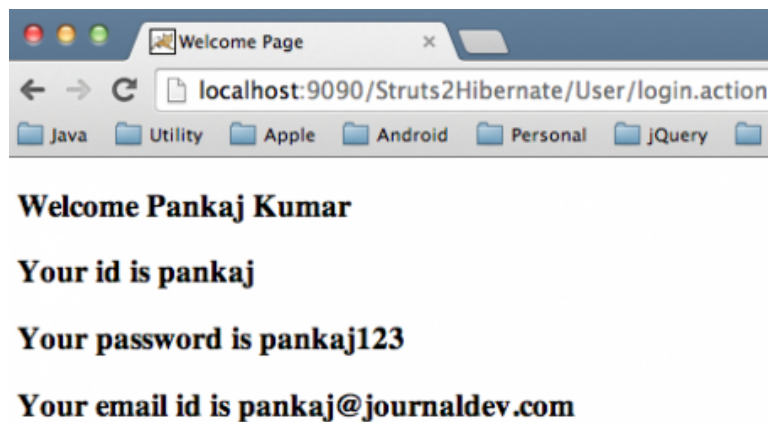
Struts2 Hibernate Application Test

Our application is ready, build and deploy it into your favorite servlet container and you should get response pages like below images.

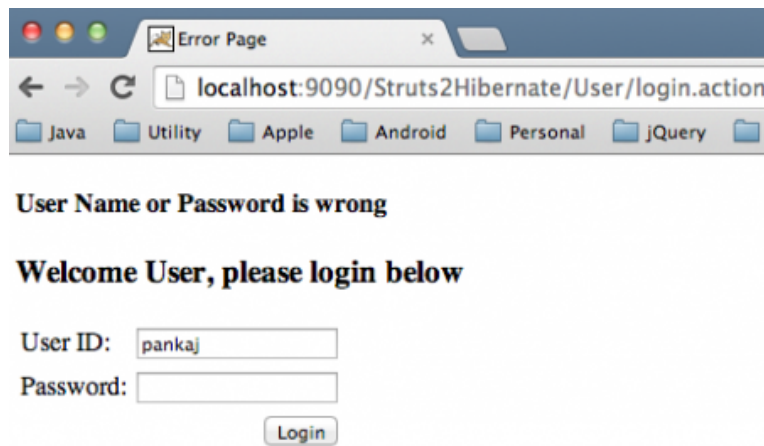
Struts2 Hibernate Login Page



Struts2 Hibernate Welcome Page



Struts2 Hibernate Login Error Page



The screenshot shows a web browser window with the title "Error Page". The address bar displays "localhost:9090/Struts2Hibernate/User/login.action". Below the address bar, there are several folder icons: Java, Utility, Apple, Android, Personal, and jQuery. The main content area of the browser displays the following text:

User Name or Password is wrong

Welcome User, please login below

User ID:

Password:

You will also notice hibernate queries and other logs in the server log file.

```
Hibernate: select user0_.USER_ID as USER_ID1_0_, user0_.NAME as NAME2_0_,  
user0_.PASSWORD as PASSWORD3_0_, user0_.EMAIL as EMAIL4_0_ from USER user0_ where  
user0_.USER_ID=? and user0_.PASSWORD=?  
User Retrieved from DB::Name= Pankaj Kumar, Email= pankaj@journaldev.com
```

Struts2 Hibernate Integration Summary

This tutorial was aimed to provide details about **Struts2 and Hibernate integration**, so that you can get started. As you can see that Servlet Context Listener makes it very easy to integrate. You can download the final project from below link.

[Download Struts2 Hibernate Example Project](#)

« PREVIOUS

Spring MVC Hibernate MySQL Integration CRUD
Example Tutorial

NEXT »

Hibernate Validator JSR303 Example Tutorial

About Pankaj

If you have come this far, it means that you liked what you are reading. Why not reach little more and connect with me directly on [Google Plus](#), [Facebook](#) or [Twitter](#). I would love to hear your thoughts and opinions on my articles directly.

Recently I started creating video tutorials too, so do check out my videos on [Youtube](#).

FILED UNDER: [HIBERNATE](#), [STRUTS 2](#)

Comments

Sarat Soubam says

[JANUARY 29, 2017 AT 10:51 PM](#)

Yeah, great tutorial for hibernate beginners like me.

Thank a lot journaldev

[Reply](#)

Sagar says

DECEMBER 11, 2016 AT 10:52 PM

i am getting java.lang.NullPointerException for servletcontext

[Reply](#)

angel says

JANUARY 31, 2017 AT 1:29 AM

same here I am also getting the same error can u please help

[Reply](#)

Sarit Gupta says

MAY 27, 2016 AT 5:50 AM

I am getting 404 error while running this code

[Reply](#)

mindy says

NOVEMBER 12, 2014 AT 6:28 PM

good example! thanks

[Reply](#)

Helen says

SEPTEMBER 30, 2014 AT 11:43 AM

I'm getting these errors below for the User.hbm.xml property:

Multiple annotations found at this line:

– The content of element type "property" must match

"null".

– Attribute "type" must be declared for element type

"property".

[Reply](#)

Mahesh says

SEPTEMBER 24, 2014 AT 4:45 AM

This site and your work is Excellent ..!

Please include all jar files of corresponding applications .

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

☐

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

[DOWNLOAD ANDROID APP](#)



HIBERNATE FRAMEWORK

Hibernate Tutorial

- > [Hibernate Example](#)
- > [Hibernate SessionFactory](#)
- > [Hibernate Session get load](#)
- > [Hibernate Session save](#)
- > [HQL Example](#)
- > [Hibernate Criteria](#)
- > [Hibernate SQL](#)
- > [Hibernate Named Query](#)
- > [Hibernate Log4J](#)
- > [Hibernate Validator](#)
- > [Hibernate Tomcat DataSource](#)

Hibernate Mapping

- > [Hibernate One to One Mapping](#)
- > [Hibernate One to Many Mapping](#)
- > [Hibernate Many to Many Join Tables](#)

Hibernate Caching

- > [Hibernate Cache](#)
- > [Hibernate EHCache](#)

Hibernate Integrations

- > [Hibernate Spring](#)
- > [Hibernate Spring MVC](#)
- > [Hibernate Struts 2](#)
- > [Hibernate Primefaces](#)
- > [Hibernate Primefaces Spring](#)
- > [Hibernate SpringRoo Primefaces](#)
- > [Hibernate JSF Spring](#)

Miscellaneous

- > [Hibernate Tools Eclipse Plugin](#)
- > [Hibernate Configuration Offline](#)
- > [\[Solved\] No identifier specified](#)
- > [Hibernate Program Not Terminating](#)
- > [Access to DialectResolutionInfo](#)
- > [get is not valid](#)
- > [No CurrentSessionContext configured](#)
- > [Hibernate Interview Questions](#)

RECOMMENDED TUTORIALS

Java Tutorials

- > [Java IO](#)
- > [Java Regular Expressions](#)
- > [Multithreading in Java](#)
- > [Java Logging](#)
- > [Java Annotations](#)
- > [Java XML](#)
- > [Collections in Java](#)
- > [Java Generics](#)
- > [Exception Handling in Java](#)
- > [Java Reflection](#)
- > [Java Design Patterns](#)
- > [JDBC Tutorial](#)

Java EE Tutorials

- > [Servlet JSP Tutorial](#)
- > [Struts2 Tutorial](#)
- > [Spring Tutorial](#)
- > [Hibernate Tutorial](#)
- > [Primefaces Tutorial](#)
- > [Apache Axis 2](#)
- > [JAX-RS](#)
- > [Memcached Tutorial](#)

© 2018 · Privacy Policy · Don't copy, it's Bad Karma · P

Open Source Code

Medical Coding Company

Medical Coding Outsourcing

Database

Download SQL Server

Learn SQL

MySQL Server

SQL Tutorial

