IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727,
PP 63-68
www.iosrjournals.org

Integration Of Struts2 And Hibernate Frameworks

Remya P V[1], Aswathi R S[1], Swetha M[1], Sijil Sasidharan[1], Sruthi E[1], Vipin Kumar N[1]
[1](Department of MCA,NMAMIT Nitte/ Autonomous under VTU, India)

ABSTRACT: *The struts2 framework is a brand new framework that introduces many architectural refinements over the existing ones to make the web based system more efficient to use.The major features of Struts 2 are Interceptor, user interface tags, result and validation. Now a days we have a lot of data access frameworks like Hibernate, JPAs, JDBC, etc. Struts can integrate easily with the popularly used data access frameworks.Hibernate is one of them that easily integrates with other Java EE frameworks.Hibernate ORM helps your application to achieve persistence.In this paper we describe and analyze the modern frameworks such as struts2 and hibernate.Also integerating them.Here we used framework Struts2 as frontend and hibernate as backend. The integrated framework will provide a better database access with better frontend design.*
Keywords- *Hibernate,Interceptors,Model View Controller (MVC),Object Relational Mapping(ORM),Struts2.*

I. Introduction

Web application development has been around long enough that the term Web 2.0 is being introduced to describe the next generation of web application.Web 2.0 is an intersection of new business models, new ideas, and multifaceted sharing and collaboration.Along with Web 2.0 came a revival of scripting languages (and even a few new ones), all dynamic and supporting fast-paced and highly productive development environments.

Struts is the first, and most popular Java web application framework ever. By its second major release it achieves important milestone for the framework in terms of functionality but also for the improvements made to increase developer productivity. By decreasing coupling within the framework, reducing configuration, providing default and different configuration options (via annotations), and providing a plug-in mechanism to easily extend the base features, Struts2 is providing a platform that can be built upon for the next generation of web applications.

The Hibernate framework is an object/relational mapping tool for Java environments. The term object/relational mapping refers to the technique of mapping a data representation from an object model to a relational data model with a SQL-based schema. Hibernate not only takes care of the mapping from Java classes to database tables (and from Java data types to SQL data types), but also provides data query and retrieval facilities and can significantly reduce development time otherwise spent with manual data handling in SQL and JDBC. Hibernates goal is to relieve the developer from 95 percent of common data persistence related programming tasks.

Integration of Struts2 framework and Hibernate3 framework will reduce the developer's implementation complexities. Developer can develop and maintain the system by giving more importance to the software quality attributes.

II. Hibernate

It's a good choice to implement the fine data persistence layer by the Hibernate framework, which enables developers to interact with a relational database by an object-oriented programming surface. Hibernate is a powerful, high performance Object/Relational persistence. Its primary feature is mapping from Java classes to database tables (and from Java data types to SQL data types), which also provides data query and retrieval facilities. Hibernate generates the SQL calls and relieves the developer from manual result set handling and object conversion, keeping the application portable to all supported SQL databases, with database portability delivered at very little performance overhead[3][4].

III. Interceptors

Interceptors allow for crosscutting functionality to be implemented separately from the action as well as the framework. Web application developers can achieve the following using interceptors: Providing preprocessing logic before the action is called, providing post processing logic after the action is called and catching exceptions so that alternate processing can be performed. Many of the features provided in the Struts2 framework are implemented using interceptors; examples include exception handling, file uploading, lifecycle callbacks and validation etc.

IV. MODEL VIEW CONTROLLER

Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts: Models, View, Controller.

- Model is the lowest level of the pattern which is responsible for maintaining data.
- View is responsible for displaying all or a portion of the data to the user.
- Controller is software Code that controls the interactions between the Model and View.

MVC is popular because it isolates the application logic from the user interface layer. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. A final presentable response is generated by the View that uses the data prepared by the Controller.
Struts 2 is a second-generation web application framework that implements the Model-View-Controller (MVC) design pattern. Struts 2 framework can be also called _MVC from 30,000 feet‘ or _pull-MVC‘ as it provides a cleaner implementation of MVC.

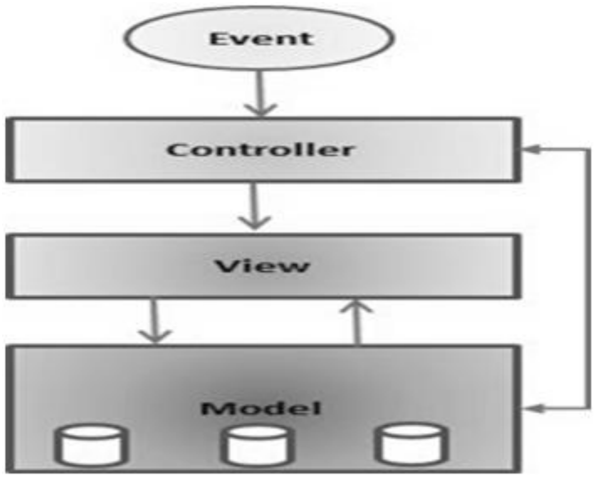

Fig.1. Model View Controller

V. OBJECT RELATIONAL MAPPING

Object Relational Mapping is a layer that is responsible to bridge the gap between object-oriented programs and relational database. In fact which provides the objects with persistent services which means that ability to read data from and write data to, and delete data from data sources.

In other words, object relational mapping is a technology that is employed to bridge the impedance mismatch between object- oriented programs and relational database. It tries to eliminate the duplication of data and maintenance cost and susceptibility to error(s) associated with it [5].

VI. STRUTS2

A web application framework is apiece of structural software that provides automation of common tasks of the domain as well as a built-in architectural solution that can be easily inherited by applications implemented on the framework.Struts2 is popular and mature web application framework based on the MVC[1]. Struts 2 is slightly different from a traditional MVC framework in that the action takes the role of the model rather than the controller,although there is some overlap. The controller is implemented with a struts2 dispatch servlet filter as well as interceptors,the model is implemented with actions, and the view as a combination of result types and results.

VII. PROPOSED SYSTEM

The Struts2 and Hibernate3 framework can be treated as frameworks extending front-end and back-end features respectively. The integration is done using the JavaBeans(The POJO files) created by the Hibernate and the action class of the Struts framework. The two completely different frameworks will be communicated only for the transfer of JavaBeans. The action class will create the JavaBean object, which will be populated using

the getter-setter methods by some user or by the system itself. The populated bean can be called as transient object.
These are transient because they are not connected with any session in the application. Such objects are made persistent by associating the object with a session. This is the place where the actual integration of Struts and Hibernate occurs. A persistent object is a short-lived, single threaded object containing persistent state and business function. These are JavaBeans/POJOs associated with a session. The session is a single threaded short-lived object representing conversation between application and persistent store. These objects can be modified and inserted into the databaseusing Hibernate Query Language (HQL).Fig 1Considering the Architecture of frameworks The Struts2 Architecture and Hibernate3 Architecture can be integrated at the point where conversion of the transient object to persistent object occurs. Instead of that we can consider the Struts2 as an Application part in the Hibernate2 Architecture, which provides a transient object to the Hibernate Architecture.There are no official plugins used to integrate the Hibernate framework.So we have to follow some steps such as:-

- Register a user class and its implementation class as UserDao.
- In the UserDao class,initialize the Hibernate session andand perform desired tasks.
- In action class, initialize the object of UserDao class and call the methods according to the actions.

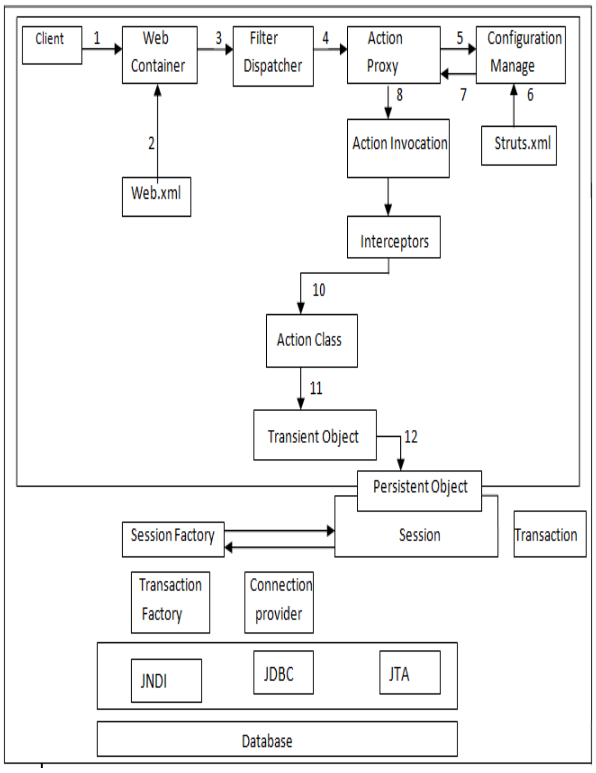

Fig.2. Integration of Struts2 and Hibernate frameworks

Example:-
It shows a simple user module(add and list function), developed in Struts 2, and perform the database operation with Hibernate3.

1.Creating a user table in SQL table script

```
CREATE TABLE user' sampledb'.'user' (
'id'  VARCHAR(10),
'name'  VARCHAR(20),
'password'  VARCHAR(20),
PRIMARYKEY('id')
);
```

2.Hibernate model and Configuration.

Creating class for user :

```
package dao;
public class User  implements java.io.Serializable {
    private String id;
    private String name;
    private String password;

   public User() {
   }
   public User(String id) {
      this.id = id;
   }
   public User(String id, String name, String
password) {
      this.id = id;
      this.name = name;
      this.password = password;
   }

   public String getId() {
      return this.id;
   }

   public void setId(String id) {
      this.id = id;
   }
   public String getName() {
      return this.name;
   }

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

   public void setPassword(String password) {
      this.password = password;
   }
}
```

3.Hibernate mapping file for customer
User.hbm.xml:

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-
//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-
mapping-3.0.dtd">
<hibernate-mapping>
<class name="dao.User" table="user"
catalog="sampledb">
<id name="id" type="string">
<column name="id" length="10" />
<generator class="assigned" />
</id>
<property name="name" type="string">
<column name="name" length="20" />
</property>
<property name="password" type="string">
<column name="password" length="20" />
</property>
</class>
</hibernate-mapping>
```

4.Hibernate database configuration file
hibernate.cfg.xml:

```
<?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.dialect">org.hibernate.dialect.My
SQLDialect</property>
<property
name="hibernate.connection.driver_class">com.m
ysql.jdbc.Driver</property>
<property
name="hibernate.connection.url">jdbc:mysql://loc
alhost:3306/sampledb</property>
<property
name="hibernate.connection.username">root</pro
perty>
<property
name="hibernate.connection.password">sijil</prop
erty>
<mapping resource="dao/User.hbm.xml"/>
</session-factory>
</hibernate-configuration>
```

Hibernate.reveng.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-reverse-engineering
PUBLIC "-//Hibernate/Hibernate Reverse
Engineering DTD
3.0//EN""http://hibernate.sourceforge.net/hibernate
-reverse-engineering-3.0.dtd">
<hibernate-reverse-engineering>
<schema-selection match-catalog="sampledb"/>
<table-filter match-name="user"/>
</hibernate-reverse-engineering>
```

5. Create a UserDao class for implementing operations on User class

```
package dao;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import java.util.List;
```

```
ublic class UserDao {
List<User> userList=null;
public void addUser(String id,String name,String
password)
{
User u=new User(id,name,password);
SessionFactory                          sf=new
Configuration().configure().buildSessionFactory();
Session s=sf.openSession();
Transaction t=s.beginTransaction();
s.save(u);
t.commit();
s.close();
}
public List<User> listUser()
{
SessionFactory                          sf=new
Configuration().configure().buildSessionFactory();
Session s=sf.openSession();
//Transaction t=s.beginTransaction();
userList=s.createQuery("from User").list();
s.close();
return userList;
}
}
```

6 .Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app                          version="3.0"
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/ja
vaee         http://java.sun.com/xml/ns/javaee/web-
app_3_0.xsd">
<filter>
<filter-name>struts2</filter-name>
<filterclass>org.apache.struts2.dispatcher.FilterDis
patcher
</filter-class>
</filter>
<filter-mapping>
<filter-name>struts2</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>
<session-config>
<session-timeout>
30
</session-timeout>
</session-config>
<welcome-file-list>
<welcome-file>index.jsp</welcome-file>
</welcome-file-list>
</web-app>
```

7.Action class

UserAction.java:

```
import com.opensymphony.xwork2.ActionSupport;
import dao.User;
import dao.UserDao;
import java.util.List;
public class UserAction extends ActionSupport {

  private String id,name,password;
  List<User> userList=null;

  public List<User> getUserList() {
    return userList;
  }

  public void setUserList(List<User> userList) {
    this.userList = userList;
  }

  public String getId() {
    return id;
  }

  public void setId(String id) {
    this.id = id;
  }

  public String getName() {
    return name;
  }

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

  public String getPassword() {
    return password;
  }

  public void setPassword(String password) {
    this.password = password;
  }


@Override
public String execute() throws Exception {
try
{
UserDao ud=new UserDao();
ud.addUser(id, name, password);
userList=ud.listUser();
return "success";
}
catch(Exception e)
{
return "error";
}
}
```

8. index.jsp

```
<%--
Document   : index
Created on : Mar 5, 2015, 5:51:52 PM
Author     : si
--%>
<%@taglib prefix="s" uri="/struts-tags" %>
<%@page contentType="text/html"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<s:form action="Register">
<s:textfield name="id" label="ID" />
<s:textfield name="name" label="Name" />
<s:textfield name="password" label="Password" />
<s:submit label="Submit" />
</s:form>
<s:if test="userList.size()>0">
<table>
<tr><th>ID</th><th>Name</th></tr>
<s:iterator value="#request.userList">
<tr><td><s:property
value="id"></s:property></td><td><s:property
value="name"></s:property></td></tr>
</s:iterator>
</table>
</s:if>
</body>
</html>\
```

9.struts.xml

```
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts
Configuration 2.0//EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
<!-- Configuration for the default package. -->
<package name="default" extends="struts-
default">
<action name="Register" class="UserAction">
<result name="success">index.jsp</result>
</action>
</package>
</struts>
```

VIII. CONCLUSION

The integration of struts2 and hibernate3 framework will provide the developer an experience of developing web applications in a well structured manner, which can be designed in multiple ways. According to the customers requirement it is flexible to change, add, edit the integration architecture. This architecture provide better support for web application development as like the PHP Codignator framework. Codignator is a well known framework provide MVC support for PHP. The integrated Struts2 and Hibernate3 framework will provide a better database access mechanism and understandability than the codignator framework. The integrated framework will provide a better database access with better frontend design.

REFERENCES

[1]. Zhou Dongxing, Li Xinke, "Design and application of a web development model based on MVC and AJAX",JOURNAL OF HEFEI UNIVERSITY OF TECHNOLOGY, Vol 31 No. 9 Sept.2008.
[2] Nielet D'mello1 , Larkins Carvalho2, "- The modern web application framework" Vol. 3, Issue. 3, May - June 2013
[3] G.King.,C.Bauer, "Hibernate in Action", Manning Publication Co, 2004.
[4] C.Bauer, G.King, "Java Persistence with Hibernate",Manning Publication Co, 2007
[5] A. Keller and C. Keene (1993), Persistence software: bridging object-Oriented programming and relational databases, In Proceedings of the ACM SIGMOD Int'' l Conference on Management of Data, pp. 540-541.