Web Application with Hibernate

Here, we are going to create a web application with hibernate. For creating the web application, we are using JSP for presentation logic, Bean class for representing data and DAO class for database codes.

As we create the simple application in hibernate, we don't need to perform any extra operations in hibernate for creating web application. In such case, we are getting the value from the user using the JSP file.

Example to create web application using hibernate

In this example, we are going to insert the record of the user in the database. It is simply a registration form.

index.jsp

This page gets input from the user and sends it to the register.jsp file using post method.

```
<form action="register.jsp" method="post">
Name:<input type="text" name="name"/><br><br/>
Password:<input type="password" name="password"/><br><br/>
Email ID:<input type="text" name="email"/><br><br/>
<input type="submit" value="register"/>"

</form>
```

register.jsp

This file gets all request parameters and stores this information into an object of User class. Further, it calls the register method of UserDao class passing the User class object.

```
<%@page import="com.javatpoint.mypack.UserDao"%>
<jsp:useBean id="obj" class="com.javatpoint.mypack.User">
</jsp:useBean>
<jsp:setProperty property="*" name="obj"/>
```

```
<%
int i=UserDao.register(obj);
if(i>0)
out.print("You are successfully registered");

%>
```

User.java

It is the simple bean class representing the Persistent class in hibernate.

```
package com.javatpoint.mypack;

public class User {
private int id;
private String name,password,email;

//getters and setters

}
```

user.hbm.xml

It maps the User class with the table of the database.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-mapping PUBLIC
 "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
 "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

 <hibernate-mapping>
 <class name="com.javatpoint.mypack.User" table="u400">
 <id name="id">
 <generator class="increment"></generator>
 </id>
 <property name="name"></property>
```

```
<property name="password"></property>
<property name="email"></property>
</class>

</hibernate-mapping>
```

UserDao.java

A Dao class, containing method to store the instance of User class.

```
package com.javatpoint.mypack;



import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

public class UserDao {

public static int register(User u){
 int i=0;
 Session session=new Configuration().
  configure().buildSessionFactory().openSession();

  Transaction t=session.beginTransaction();
  t.begin();

  i=(Integer)session.save(u);

  t.commit();
  session.close();

  return i;
 }
}
```

hibernate.cfg.xml

It is a configuration file, containing informations about the database and mapping 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="hbm2ddl.auto">create</property>
 <property name="dialect">org.hibernate.dialect.Oracle9Dialect</property>
 <property name="connection.url">jdbc:oracle:thin:@localhost:1521:xe</property>
 <property name="connection.username">system</property>
 <property name="connection.password">oracle</property>
 <property name="connection.driver_class">oracle.jdbc.driver.OracleDriver</property>

 <mapping resource="user.hbm.xml"/>
 </session-factory>

</hibernate-configuration>
```

download this hibernate example (developed using Myeclipse IDE)

← prev next →

Share 1