**Hibernate 3.2.5 in MVC Web Application with NetBean 6.9.1 and MySQL Server.**

**Tác giả: Nguyễn Anh Khoa**

Mục đích: Trong chủ đề, chúng ta sẽ thực hiện một Web Application với NetBean 6.9.1 có chức năng thêm xóa sửa và lấy dữ liệu từ database với mô hình MVC, sử dụng Hibernate cho lớp Model.

Để thực hiện được bài demo này, các bạn cần những tool sau đây:

         [NetBean IDE 1.9.1](http://netbeans.org/downloads/index.html)

         Apache Tomcat v6 (package này có sẵn trong NetBean nên khi cài đặt chỉ cần chọn **Customize**, check vào Apache Tomcat).

         [My SQL Database Server v5.x](http://www.mysql.com/downloads/mysql/5.5.html) (Hướng dẫn cài đặt: [here](http://www.reaper-x.com/2007/06/27/mysql-installation-and-configuration-setting-up-private-wow-server/))

Với ứng dụng này, user sẽ login và search các user khác theo tên và có thể thêm xóa sửa thông tin của các user khác. Thông tin của user được lưu trong table Account.

**Hibernate là gì?**

Các bạn có thể tham khảo tutorial tại:

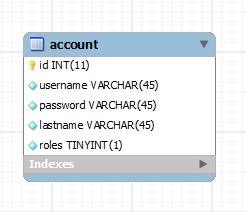
         <http://www.hibernatetutorial.com/index.html>

         <http://www.roseindia.net/hibernate/>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

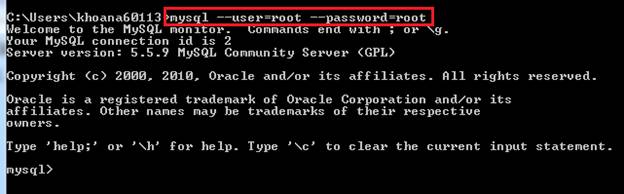
**Tạo database**

Chúng ta cần tạo table Account như sau vào MySQL Server:



Primary key là **id**và có thuộc tính là AUTO\_INCREMENT (tương tự như **Identity**trong SQL Server).

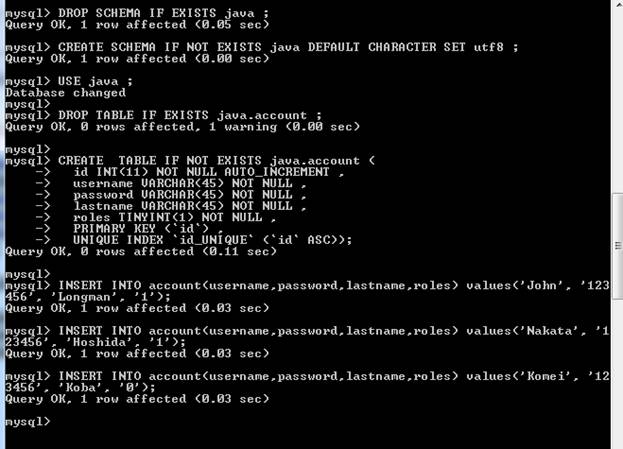
1.      Sau khi cài đặt MySQL Server, mở command line  để login vào MySQL Server với username và password như lúc cài đặt:



2.      Copy đoạn SQL script này vào cửa sổ command line:

|  |
| --- |
| DROP SCHEMA IF EXISTS java ;  CREATE SCHEMA IF NOT EXISTS java DEFAULT CHARACTER SET utf8 ;  USE java ;    DROP TABLE IF EXISTS java.account ;    CREATE  TABLE IF NOT EXISTS java.account (    id INT(11) NOT NULL AUTO\_INCREMENT ,    username VARCHAR(45) NOT NULL ,    password VARCHAR(45) NOT NULL ,    lastname VARCHAR(45) NOT NULL ,    roles TINYINT(1) NOT NULL ,    PRIMARY KEY (`id`) ,    UNIQUE INDEX `id\_UNIQUE` (`id` ASC));    INSERT INTO account(username,password,lastname,roles) values('John', '123456', 'Longman', '1');  INSERT INTO account(username,password,lastname,roles) values('Nakata', '123456', 'Hoshida', '1');                INSERT INTO account(username,password,lastname,roles) values('Komei', '123456', 'Koba', '0'); |

Kết quả là table **account** được tạo:



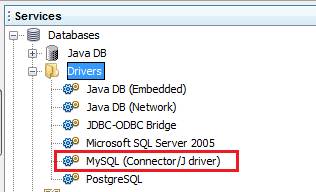
Test data:



3.      Khởi động NetBean.

Trong cửa sổ Services chọn Database  Drivers.

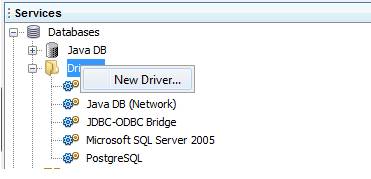
Xem đã có driver cho MySQL như hình vẽ bên dưới chưa:



Nếu chưa có, download driver [tại đây](http://ftp.jaist.ac.jp/pub/mysql/Downloads/Connector-J/mysql-connector-java-5.1.15.zip).

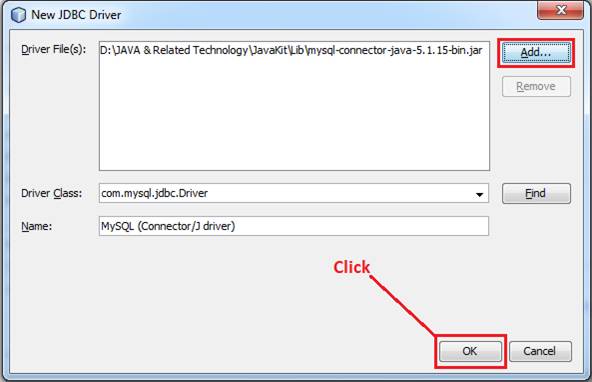
Sau khi download, giải nén để lấy file **mysql-connector-java-5.x-bin.jar.**

Click chuột phải vào Drivers và chọn New Driver:

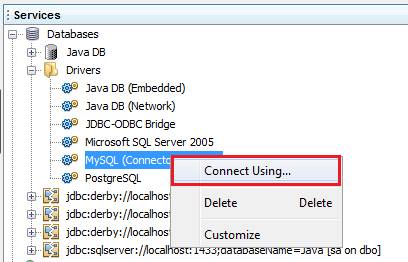


Click Add và chọn đến file **mysql-connector-java-5.x-bin.jar.**

Click OK.

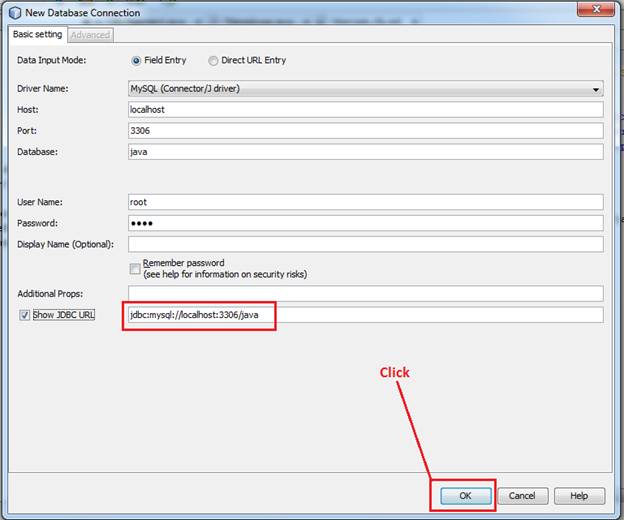


4.      Click chuột phải vào driver vừa add và chọn **Connect Using**

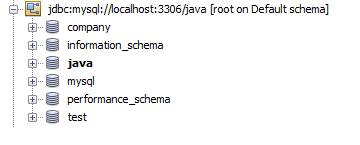


Nhập data cho chuỗi kết nối như hình vẽ dưới đây .

Click OK.



Kết quả trong cửa sổ Service:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Tạo Web Application Project**

1.      Click File  New  Project. Chọn Web Application trong mục Java Web.

Click Next.

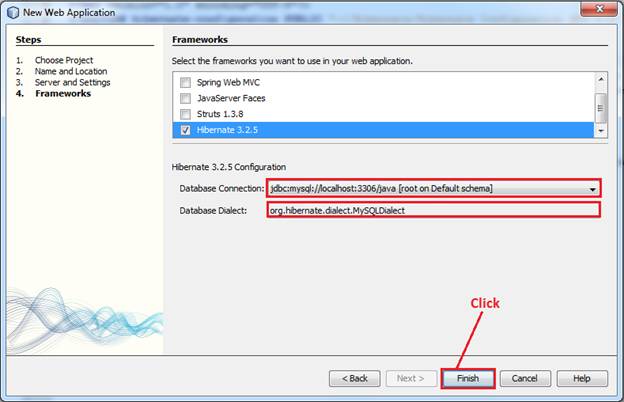
2.      Nhập tên project là Hibernate\_HelloWorld. Click Next.

3.      Chọn Server là Apache Tomcat 6.0.26 và Java EE version là Java EE5.

Click Next.

4.      Chọn framework cho project là Hibernate 3.2.5

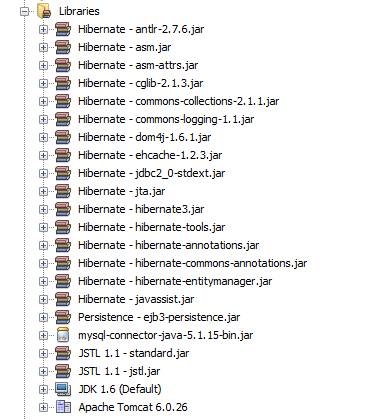
Chọn Database Connection và Database Dialect như hình vẽ:



Click Finish.

NetBean tạo project mới và mở file hibernate.cfg.xml trong editor.

Trong cửa sổ Project Explorer, click Libraries của project Hibernate\_HeloWorld sẽ thấy các thư viện của Hibernate được NetBean add vào project:



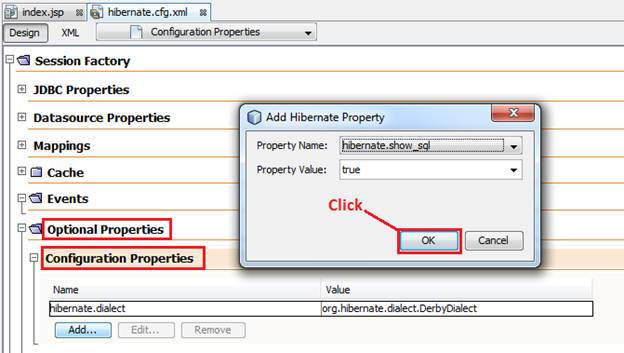
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Configure file hibernate.cfg.****xml**

1.      Mở file hibernate.cfg.xml và chuyển sang mode design.

12.png

2.      Trong node SessionFactory  Optional Properties  Configuration Properties, click Add để mở dialog box Add Hibernate Property.

Chọn Property Name: **hibernate.show\_sql** và Property value: **true**

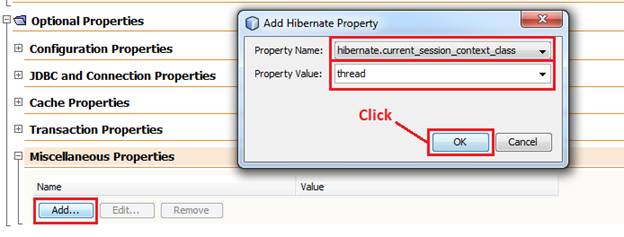


Click OK.

Property này sẽ enable chế độ debug các câu lệnh SQL khi thao tác với database. Khi run project, các câu lệnh này sẽ hiển thị trong cửa sổ Console.

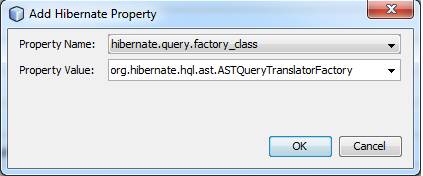
3.      Trong node SessionFactory  Optional Properties  Miscellaneous Properties, click Add.

4.      Chọn Property Name:**hibernate.current\_session\_context\_class** và Property value: **thread.**



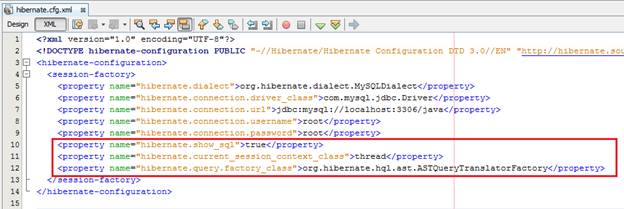
Click OK.

5.      Click Add và chọn Property Name:**hibernate.query.factory\_class** và Property value:**org.hibernate.hql.ast.ASTQueryTranslatorFactory**



Click OK.

6.      Lưu lại file và đổi sang mode XML, bạn sẽ thấy các property vừa mới được add:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

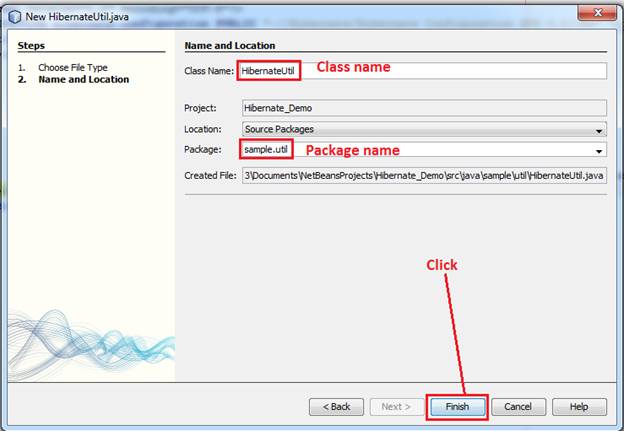
**Tạo file HiberntaeUtil.java - Helper file**

Để sử dụng hibernate, chúng ta cần tạo một helper class để access Session Factory cho việc lấy đối tượng Session. Class này gọi hàm configure() để load file hibernate.cfg.xml và build đối tượng SesionFactory.

1.      Chọn File  New File Other

2.      Chọn đến Category Hibernate và chọn HibernateUtil,java.

3.      Đặt tên cho class là HibernateUtil và tên package.



4.      Click Finish.

File này chúng ta không cần chỉnh sửa gì vì NetBean đã generate ra method mà ta cần dùng.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Tạo Hibernate Mapping File và POJO**

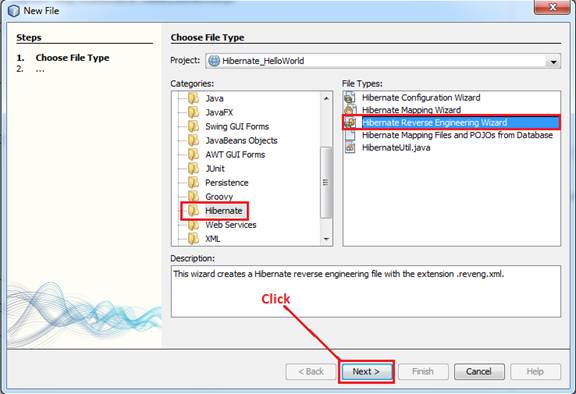
Hibernate dùng POJO (Plain Old Java Object)  để mô tả dữ liệu của table trong database. Mỗi thuộc tính của object tương ứng với một field trong table và chúng được mapping thông qua hibernate mapping file hoặc dùng annotation trong class. Trong bài này chúng ta sử dụng mapping file. Cách sử dụng annotation để mapping giống như EJB3(các bạn có thể refer các bài hướng dẫn rất chi tiết về EJB3 trong blog này). POJO cũng tương tự như JavaBean, chứa các getter và setter.

Để sử dụng Hibernate Mapping file và POJO, trước hết cần tạo file hibernate.reveng.xml (reverse engineering file ) để hỗ trợ chúng ta lựa chọn table cần dùng trong database dễ dàng hơn.

        **Tạo Hibernate Reverse Engineering file**

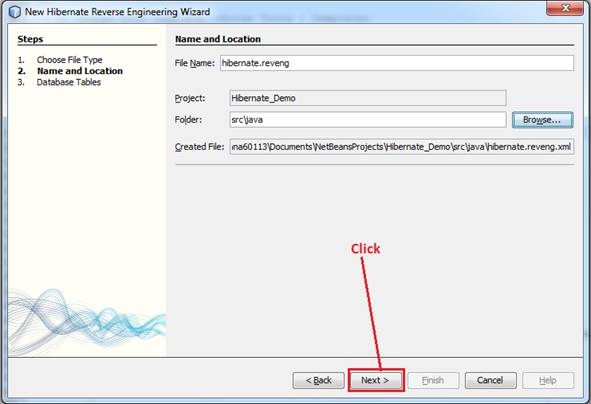
1.      Chọn File  New File Other

2.      Chọn Category Hibernate và chọn Hibernate Reverse Engineering Wizard



Click Next.

3.      Nhập tên file là hibernate.reveng và Folder là src/java.

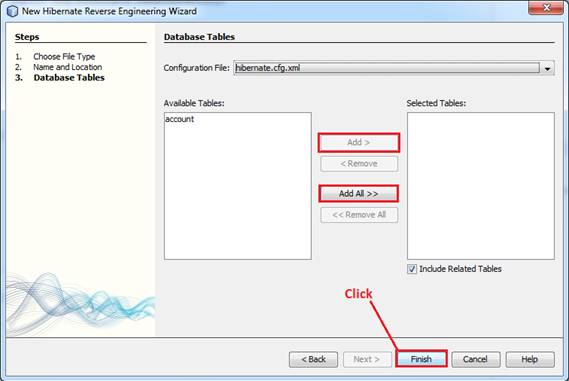


Click Next.

4.      Available Tables chứa tất cả các table có trong database mà ta đang kết nối.

Chọn table muốn sử dụng và click Add hoặc chọn tất cả các table Add all.

Ở đây chúng ta chọn account.



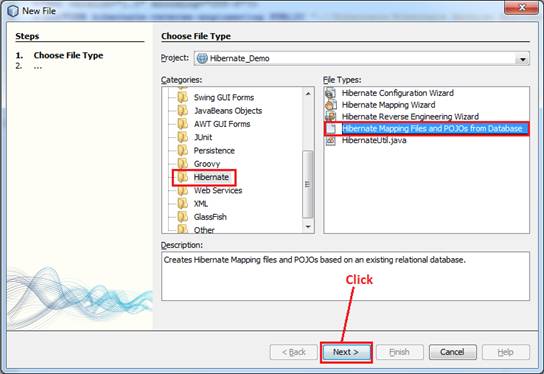
Click Finish.

File này chúng ta không làm gì với nó cả. NetBean dùng file này để xác định những table cần generate mapping file và POJO.

        **Tạo Hibernate Mapping File và POJO**

1.      Chọn File  New File  Other (Ctr + N).

2.      Chọn Category Hibernate và chọn Hibernate Mapping File and POJOs



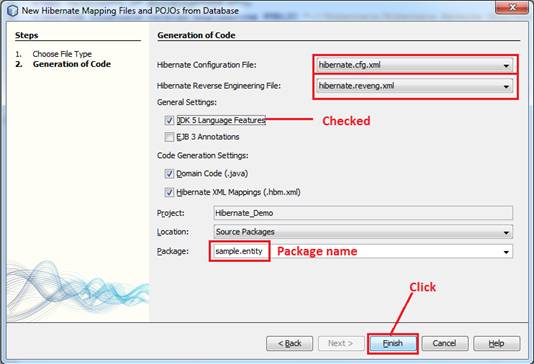
Click Next.

3.      Tên file config và file reverse engineering phải được load trong combo box.

Check vào JDK 5 Language Features.

Nhập vào tên package.

Tất cả như hình vẽ bên dưới.



            Click Finish.

            Trong package sample.entity vừa được tool generate có chứa file mapping và file java.

23.png

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Add một record vào database**

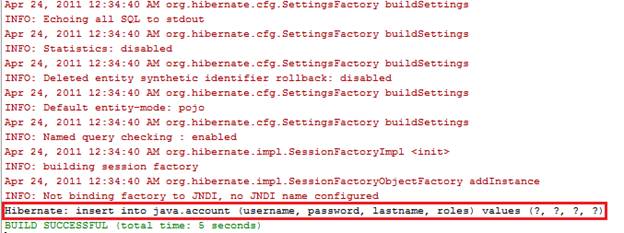
Trước khi tạo web application với mô hình MVC, chúng ta hãy tận hưởng thành quả từ những gì chúng ta đã làm.

1.      Mở file Account.java trong editor và thêm hàm main như sau:



2.      Click chuột phải và chọn Run File.

3.      Kết quả trong cửa sổ Console:



            Câu lệnh SQL xuất hiện được trong Console là do chúng ta đã set giá trị true cho property hibernate.show\_sql trong file hibernate.cfg.xml.

            Kết quả trong database:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Tạo Web Application với mô hình MVC sử dụng Hibernate**

        **Create DAO class**

**DAO(Data Access Object)**dùng để implement các business methos. Ý tưởng cũng tương tự như khi dùng Session Bean để thao tác Entity Bean trong EJB.

1.      Chọn File  New File  Other (Ctr + N)

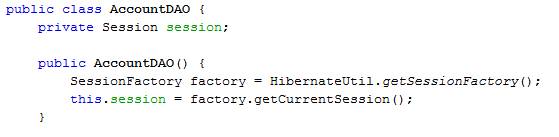
2.      Chọn Category Java và chọn Java Class. Click Next.

3.      Nhập tên cho Class là AccountDAO và package là sample.dao

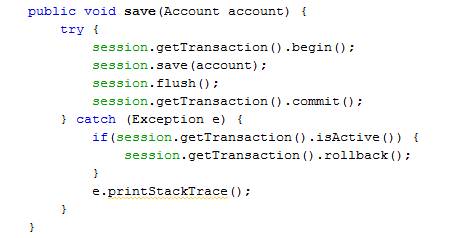
Click Finish.

4.      Implement class AccountDAO nhu sau để thực hiện các chức năng thêm, xóa, sửa và tiềm kiếm theo Lastname và Id.

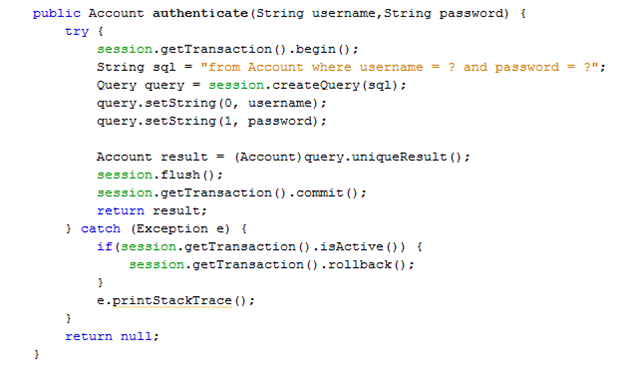
+ Thuộc tính Session để lưu đối tượng Session khi thao tác với database.



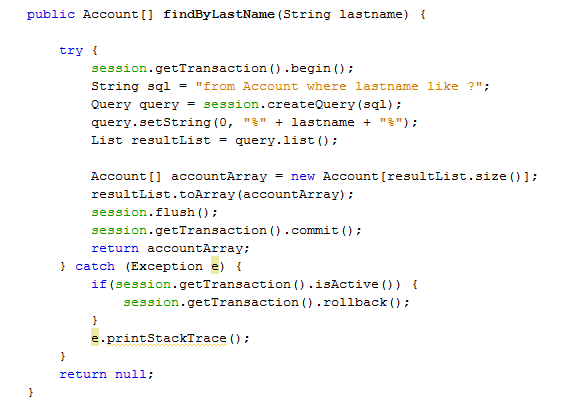
+ Method save() để add thêm một record:



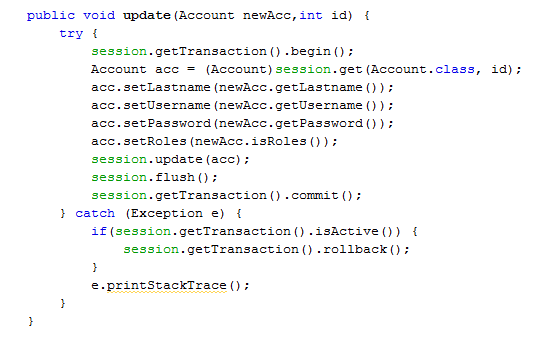
+ Method authenticate() để check username and password cho chức năng Login.



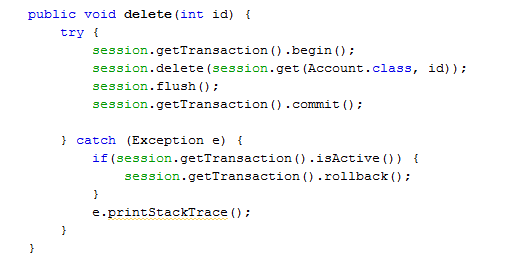
+ Method findByLastName



+ Method update() dùng để sửa thong tin của account



+ Method delete()



        **Sửa file index.jsp để tạo trang Login**

Sửa lại file index.jsp như sau:



        **Tạo Servlet và tạo controller cho chức năng login**

1.      Chọn File  New File  Other (Ctr + N)

2.      Chọn Category Web và chọn Servlet. Click Next.

3.      Đặt tên cho servlet là Controller và package là sample.servlet. Click Next.

4.      Click Finish.

5.      Sửa lại file Controller.java như sau:



        **Tạo file invalid.jsp**

1.      Chọn File  New file  Other (Ctr + N)

2.      Chọn Category Web và chọn JSP. Click Next.

3.      Nhập tên file là invalid và click Finish.

4.      Sửa lại nội dung cho file invalid.jsp như sau:



        **Tạo file welcome.jsp**

Tạo file JSP tương tự như trên và chỉnh sửa nội dung file welcome.jsp như sau:



Đến đây, bạn có thể start Tomcat, Clean and Build project , Deploy và Run.

        **Edit file Controller.java để implement chức năng Search:**

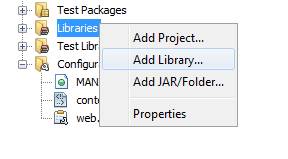
Mở file Controller.java và viết tiếp vào điều kiện if cho giá trị của action.



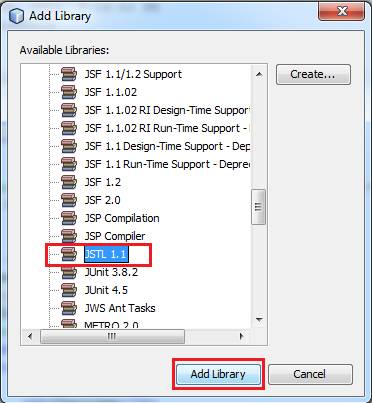
        **Tạo file show.jsp**

1.      Add thư viện JSTL

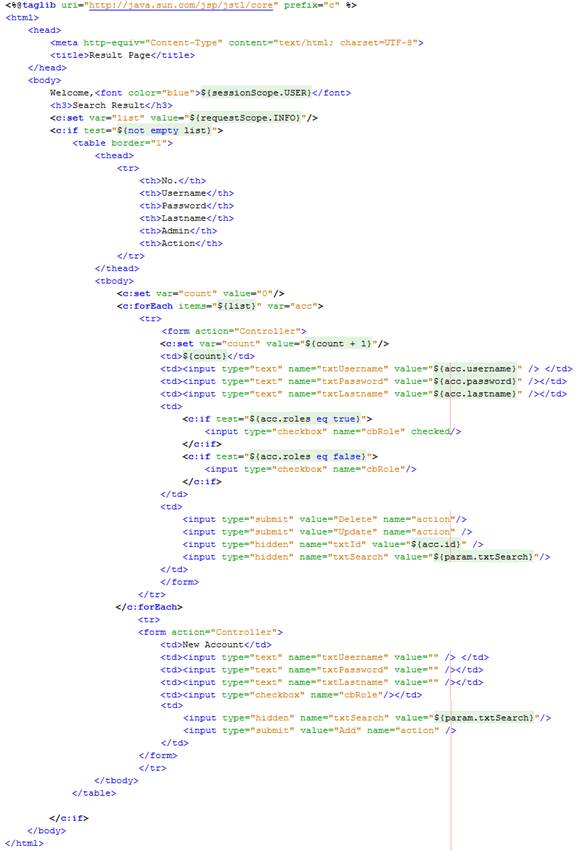
Click chuột phải vào Libraries và chọn Add Library.



Chọn JSTL 1.1 và click Add Library.

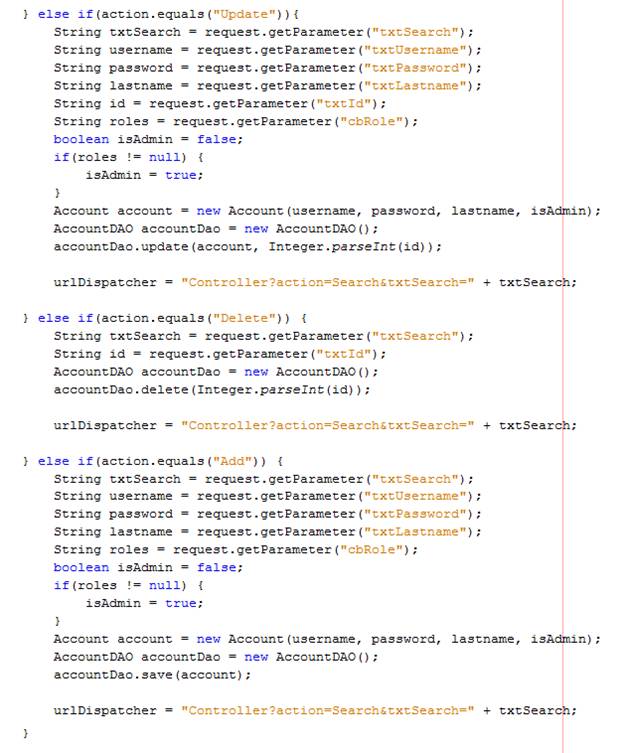


2.      Tạo file show.jsp để hiển thị kết quả search và tạo giao diện thêm xóa sửa.

em

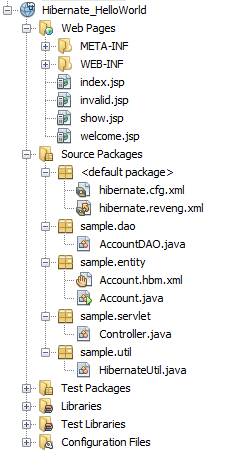
        **Edit file Controller.java để implement chức năng thêm, xóa, sửa**

Mở file Controller.java và viết tiếp vào điều kiện if cho giá trị của action.

****

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Cấu trúc source của project:

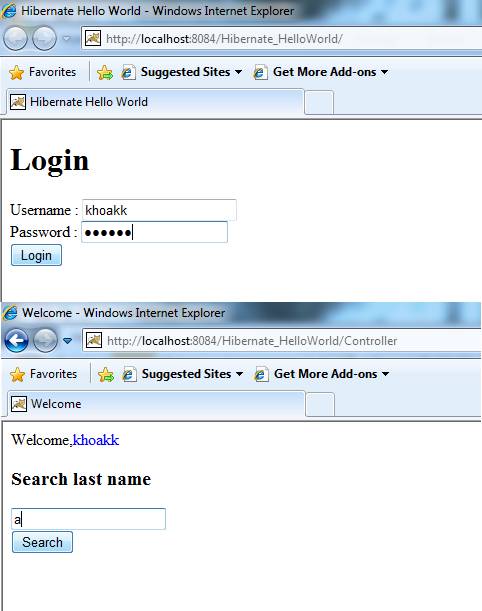


Clean and build, run project.

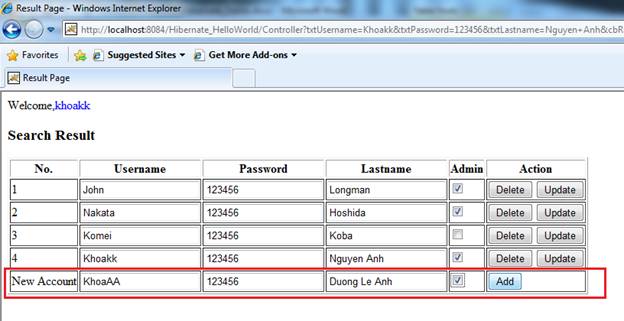
Khi click Add, Delete hoac Update trên trang show.jsp, trang show.jsp sẽ load lại và hiển thị kết quả vừa được sửa.

Kết quả khi run như sau:

1.      Login:

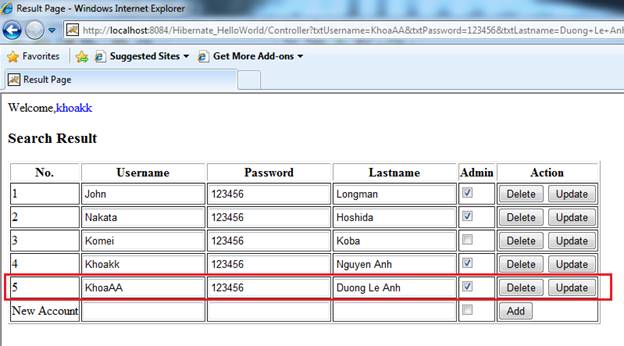


2.      Kết quả Search



Nhập thong tin cho user mới để Add.

Kết quả sau khi Add:





\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Chúc mừng các bạn đã vừa hoàn thành xong bài Demo Hibernate cơ bản với NetBean 6.9.1!

Hẹn gặp lại ở các chủ đề các!