



Kieu Trong Khanh

- Chúng tôi hướng tới việc cung cấp các tutorial và lý thuyết liên quan đến công nghệ Java, đặc biệt là J2EE/JavaEE
- Chúng tôi không chủ trương cung cấp source code, video trên web site này bởi vì chúng tôi mong muốn các bạn làm từng bước một để cảm nhận được kiến thức, hiểu biết và kết quả. Cách tiếp cận của chúng tôi là làm từng bước bằng hướng dẫn - Hướng tiếp cận của chúng tôi theo hướng thể hiện các lý thuyết thông qua ví dụ để giúp tiết kiệm thời gian

Advanced	(18)	Alice	(14)	Android	(8)	Capstones	(69)	EJB	(21)	GoogleAPI	(7)	GoogleWebToolkit	(2)	Hibernate	(8)
JavaFX	(4)	JSF	(5)	Problems&Solution	(7)	Servlet&JSP	(16)	Spring	(3)	Struts	(19)	Video_Android	(7)	Video_EJB2	(2)
Video_EJB3	(7)	Video_JSP	(9)	Video_Servlet	(8)	Video_Struts	(6)	Video_XML	(6)	Web Services	(11)	XML&Java	(35)		

Thứ Tư, 10 tháng 8, 2016

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

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.

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
- 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 (Hướng dẫn cài đặt: [here](#))

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/>

Trang

- [Trang chủ](#)
- [Lời Ngỏ](#)
- [KIỀU TRONG KHÁNH](#)
- [Capstone Projects Topics Phần 2](#)
- [Capstone Project Topics](#)
- [LÊ NGÔ GIANG](#)
- [TRẦN DƯ HÒA BÌNH](#)
- [NGUYỄN ANH KHOA](#)
- [TĂNG HẢI NGỌC SƠN](#)
- [PHẠM HUY HOÀNG](#)
- [PHẠM CÔNG THÀNH](#)

Giới thiệu bản thân

Kieu Trong Khanh

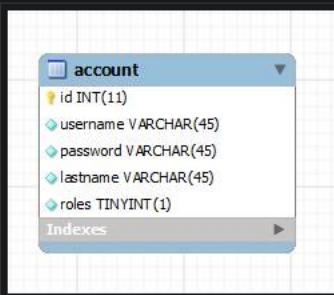
Xem hồ sơ hoàn chỉnh của tôi

Lưu trữ Blog

- ▶ 2019 (7)
- ▶ 2018 (39)
- ▶ 2017 (57)
- ▼ 2016 (190)
 - ▶ tháng mười hai (27)
 - ▶ tháng mười một (21)
 - ▶ tháng mười (10)
 - ▶ tháng chín (25)
 - ▼ tháng tám (107)
 - Sử dụng SQLite Database trong Android trên Eclipse...
 - Làm quen với lập trình android qua ứng dụng game S...
 - Làm quen với lập trình android qua ứng dụng game S...

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).

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

```
C:\Users\khoaan60113\mysql --user=root --password=root
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.5.9 MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved;
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement
mysql>
```

- 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:

```
mysql> DROP SCHEMA IF EXISTS java ;
Query OK, 1 row affected <0.05 sec>
```

Lam quen voi lap trinh Android tren Netbeans bang ...

CÁCH CẤU HÌNH ANDROID VÀ ỨNG DỤNG ĐƠN GIẢN NHẤT SỨ...

Part 3: Đồng bộ lịch của ứng dụng với Google Calen...

Phản 2: Tìm hiểu về Google Calendar API. Hướng dẫn...

Xây dựng ứng dụng quản lý calendar, đồng bộ với Go...
Google API – Các sự kiện của Google Map

Google API – Marker và Info Windows

Google API – Polyline và Polygons

Google API – Tạo một bản đồ trên web với Google Ma...
Giới thiệu về DOM – định nghĩa, DOM API và cách th...

Web Scrapping đơn giản với HTMLUnit – Phần 2

Web Scrapping đơn giản với HTMLUnit XML với MS. Excel – Giới thiệu về POI và chuyển đổi...

Xquery – Giới thiệu các khái niệm về Xquery, cú ph...

Style Sheets – Giới thiệu cách sử dụng và cách viết...

Style Sheets – Giới thiệu cách sử dụng và cách viết...

Giới thiệu về XPath
Giới thiệu về XML Schema – định nghĩa, cách viết X...

Khai báo Entities trong DTD để tăng tính hiệu quả t...

Khai báo Element – Tag và Attributes.
Sử dụng DTD ...

Sử dụng XML với namespace – Sử dụng kết hợp những ...

Giới thiệu về XML – định nghĩa, cách viết XML đúng...

Xây dựng ứng dụng Shopping Cart sử dụng EJB3.1 của...

Xây dựng ứng dụng 3 tiers sử dụng EJB3 với JavaEE ...

About Us

CASE STUDY 1: MINIFACE-MANG XÃ HỘI RÚT GỌN

Spring 3.0 Hello World với MVC Design Pattern

Spring 3.0 Hello World sử dụng tool Eclipse

Hướng dẫn xây dựng ứng dụng bằng GWT

HƯỚNG DẪN LẬP TRÌNH BẰNG GWT

Kết hợp và đồng bộ giữa JavaFX và Web service, JSP...

Kết hợp và đồng bộ giữa JavaFX và Servlet, JSP, Ja...

Chuyển đổi trạng thái trong ứng dụng JavaFX

Kết hợp JavaFX với JSP và JDBC

```

mysql> CREATE SCHEMA IF NOT EXISTS java DEFAULT CHARACTER SET utf8 ;
Query OK, 1 row affected <0.00 sec>

mysql> USE java ;
Database changed
mysql> DROP TABLE IF EXISTS java.account ;
Query OK, 0 rows affected, 1 warning <0.00 sec>

mysql>
mysql> CREATE TABLE IF NOT EXISTS java.account (
    ->     id INT<1> 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);
Query OK, 0 rows affected <0.11 sec>

mysql>
mysql> INSERT INTO account(username,password,lastname,roles) values('John', '123456', 'Longman', '1');
Query OK, 1 row affected <0.03 sec>

mysql> INSERT INTO account(username,password,lastname,roles) values('Nakata', '23456', 'Hoshida', '1');
Query OK, 1 row affected <0.03 sec>

mysql> INSERT INTO account(username,password,lastname,roles) values('Komei', '3456', 'Koba', '0');
Query OK, 1 row affected <0.03 sec>

mysql>

```

Test data:

```

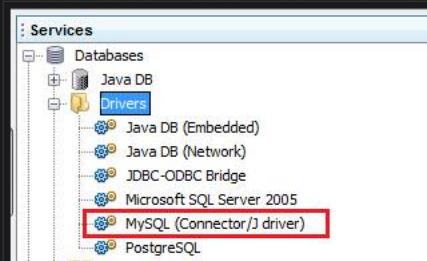
mysql> use java;
Database changed
mysql> select * from account;
+----+-----+-----+-----+-----+
| id | username | password | lastname | roles |
+----+-----+-----+-----+-----+
| 1  | John     | 123456   | Longman  | 1      |
| 2  | Nakata   | 123456   | Hoshida  | 1      |
| 3  | Komei    | 123456   | Koba     | 0      |
+----+-----+-----+-----+-----+
3 rows in set <0.06 sec>

```

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](#).

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.

Sử dụng JSF Framework 2.0 để xây dựng ứng dụng Sho...

JSF Framework – JavaServer Faces Framework, MVC Mo...

Xây dựng ứng dụng Shopping Cart dùng Visual Web JS...

Xây dựng ứng dụng quản lý dùng Visual Web JSF trên...

Cài đặt Visual Web JSF cho Netbeans 6.9.1 và vài Ứ...

Cài đặt và sử dụng Full-text Search trong ứng dụng...

Hướng dẫn thực hiện chức năng login và ứng dụng b...

Mô róng kiến thức về Struts 2 Framework – sử dụng ...

Mô róng kiến thức về Struts 2 Framework – Intercep...
'12 Tổng quan về Struts 2 Framework – một Framework hỗ...

Khái niệm về Dispatch Action trong Struts Framework...

Sử dụng Validation trong Struts Framework

Sử dụng Struts Framework kết hợp với EJB 3 để hỗ t...

Tổng quan về Struts Framework – một Framework hỗ t...

Xây dựng ứng dụng áp dụng RESTful Web Services – s...

Xây dựng ứng dụng áp dụng RESTful Web Services – x...

Giới thiệu về RESTful Web Services – công nghệ tạo...

Xây dựng Web Service sử dụng Web Server Tomcat và ...

Kết hợp Axis2 web service – Java FX – Windows Form...

.NET Windows Form kết hợp sử dụng Axis2 Web servic...

Cấu hình Axis2 trên Eclipse

Tạo web services đơn giản dùng Axis2 trên Eclipse

AXIS2 - OBJECT VÀ MÃNG OBJECT

AXIS2 - Simple Web Services kết hợp JAX-WS

Hibernate Many-to-Many Relationship dùng DB SQL Se...

Hibernate One-to-Many Relationship dùng DB SQL Ser...

Hibernate One-to-One Relationship dùng DB SQL Serv...

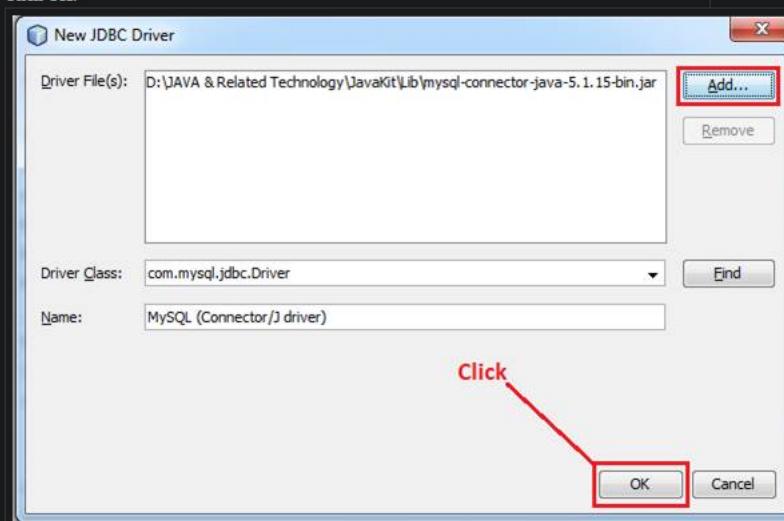
Hibernate One-to-One Relationship dùng DB SQL Serv...

Hibernate One-to-Many Relationship trên MySQL – Mô...

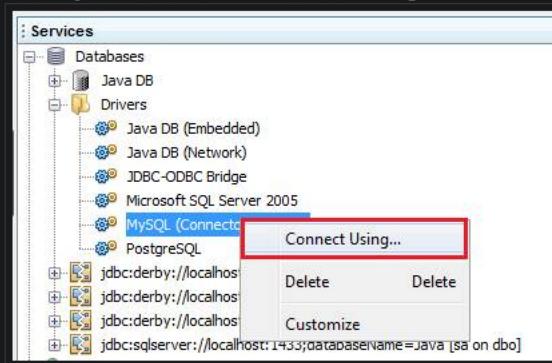
Hibernate One-to-One Relationship – Mỗi quan hệ 1...

Hibernate 3.2.5 in MVC Web Application with NetBea...

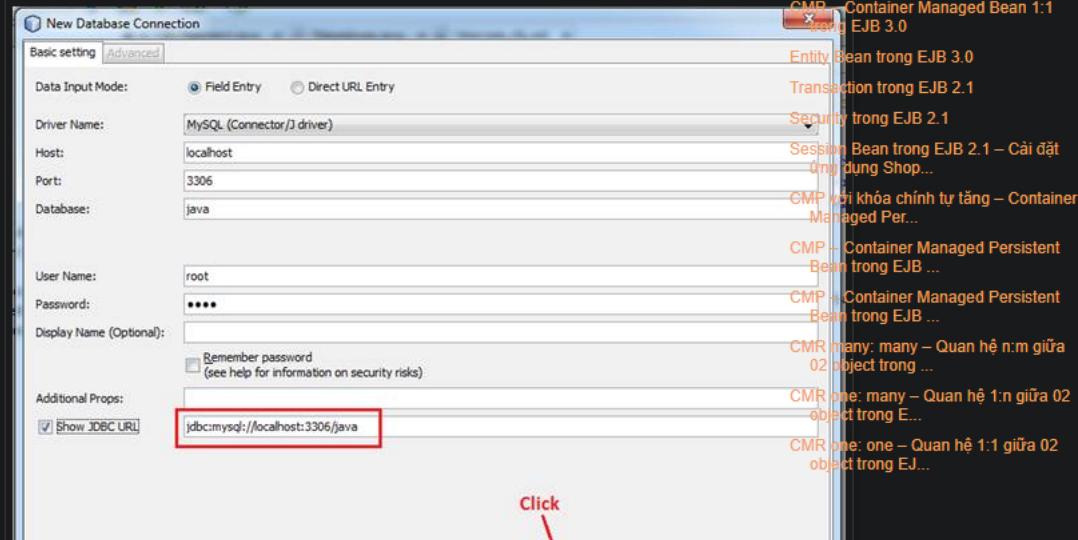
Hibernate 3.2.5 in MVC Web Application with NetBea...



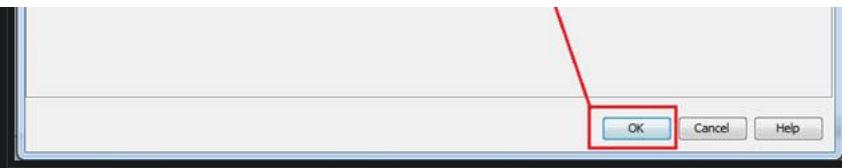
4. Click chuột phải vào driver vừa add và chọn Connect Using



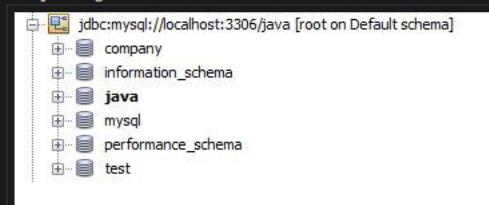
Nhập data cho chuỗi kết nối như hình vẽ dưới đây.
Click OK.



Tổng quan JPA vào MVC2 kết hợp JavaEE6
Sử dụng Request Listener trong J2EE Application
Sử dụng filter trong J2EE application (Phần 2)
Sử dụng filter trong J2EE application (Phần 1)
Xây dựng ứng dụng sử dụng JSP, Servlet kết hợp JQu...
Custom Tag Library – Tạo ra tag library trong JSP
Security Web Application trong Tomcat Web Server
Sử dụng NetBeans 7.2.1, JBoss 7.1.1 Final trên Win...
Sử dụng NetBeans 7.2.1 và Netbeans 7.3.1 với JBoss...
Khắc phục việc deploy ứng dụng trên server J2EE/JA...
Triển khai Deploy ứng dụng Enterprise Application ...
Triển khai Deploy ứng dụng web trên server Tomcat ...
Khắc phục việc port bị chiếm không thể khởi động đ...
Giới thiệu về JMS – Java Message Services – Cài đặt...
Transaction trong EJB 3.0 – cài đặt và vận dụng cá...
Security trong EJB 3
Session Bean trong EJB 3 – Cài đặt ứng dụng Shoppi...
Entity Bean với khóa chính tự tăng – identity tron...
CMR – Container Managed Bean n:m trong EJB 3.0
CMR – Container Managed Bean 1:n trong EJB 3.0
CMP – Container Managed Bean 1:1 trong EJB 3.0
Entity Bean trong EJB 3.0
Transaction trong EJB 2.1
Security trong EJB 2.1
Session Bean trong EJB 2.1 – Cài đặt ứng dụng Shop...
CMP với khóa chính tự tăng – Container Managed Per...
CMP – Container Managed Persistent Bean trong EJB ...
CMP – Container Managed Persistent Bean trong EJB ...
CMR many: many – Quan hệ n:m giữa 02 object trong ...
CMR one: many – Quan hệ 1:n giữa 02 object trong E...
CMR one: one – Quan hệ 1:1 giữa 02 object trong E...

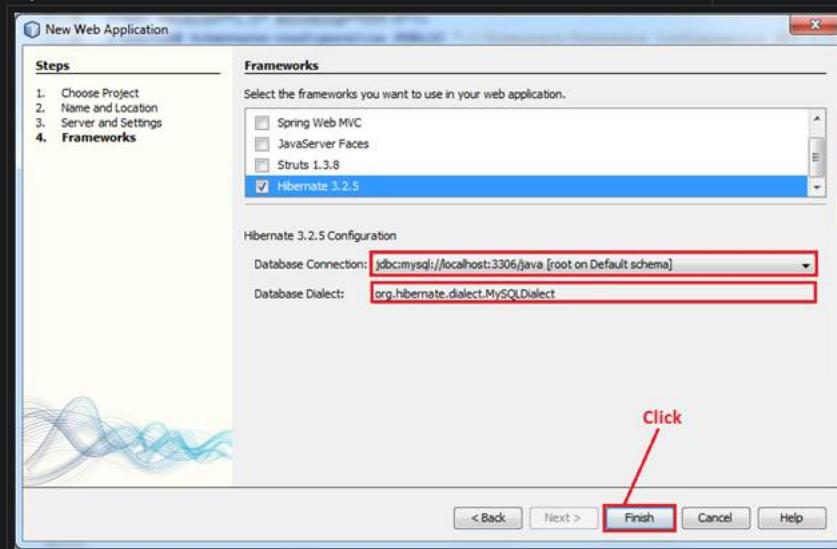


Kết quả trong cửa sổ Service:



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_HelloWorld sẽ thấy các thư viện của Hibernate được NetBean add vào project:





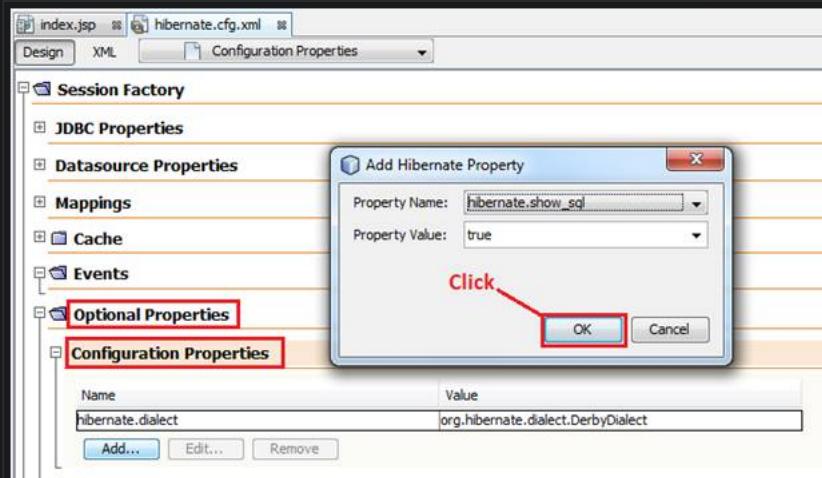
Configure file hibernate.cfg.xml

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



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

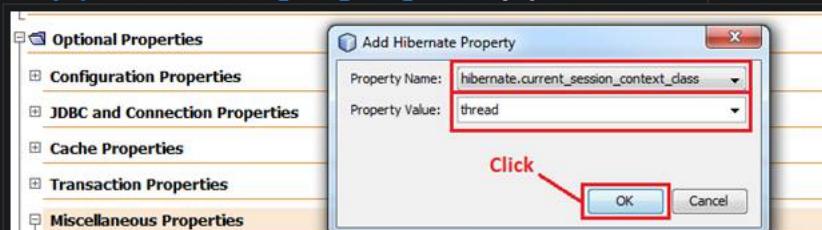


Click OK.

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.

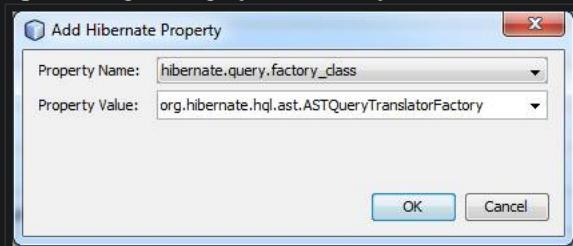
4. Chọn Property Name: hibernate.current_session_context_class và Property value: thread.



Name	Value
Add...	

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:

```

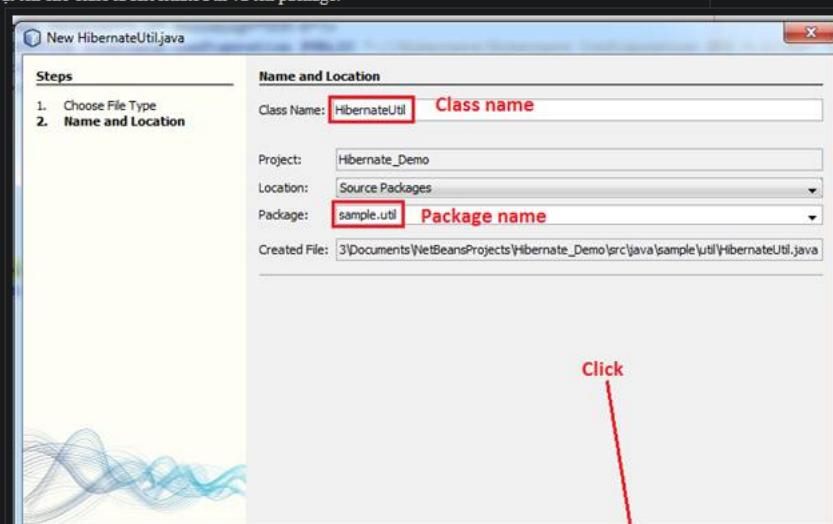
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
3 <hibernate-configuration>
4   <session-factory>
5     <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
6     <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
7     <property name="hibernate.connection.url">dbcmysql://localhost:3306/java</property>
8     <property name="hibernate.connection.username">root</property>
9     <property name="hibernate.connection.password">root</property>
10    <property name="hibernate.show_sql">true</property>
11    <property name="hibernate.current_session_context_class">thread</property>
12    <property name="hibernate.query.factory_class">org.hibernate.hql.ast.ASTQueryTranslatorFactory</property>
13  </session-factory>
14 </hibernate-configuration>
15

```

Tạo file HibernatUtil.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 SessionFactory.

- Chọn File → New File → Other
- Chọn đến Category Hibernate và chọn `HibernateUtil.java`.
- Đặt tên cho class là `HibernateUtil` và tên package.



< Back Next > **Finish** Cancel Help

4. Click Finish.

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

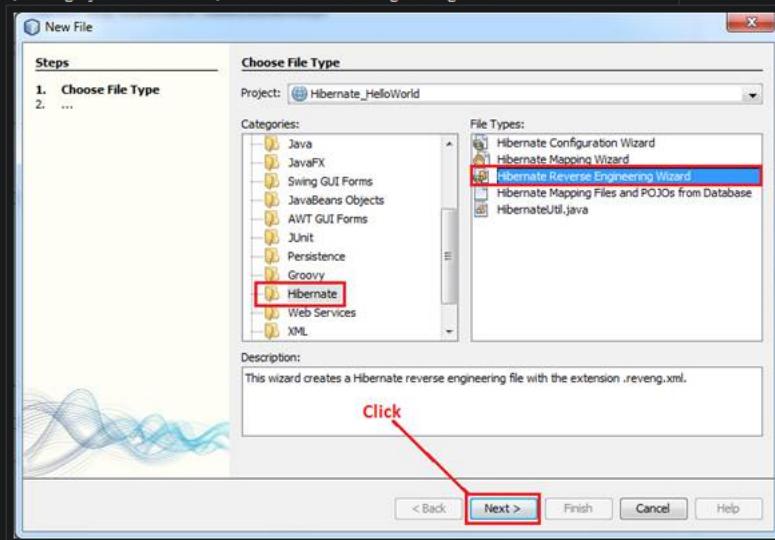
Tạo Hibernate Mapping File và POJO

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.

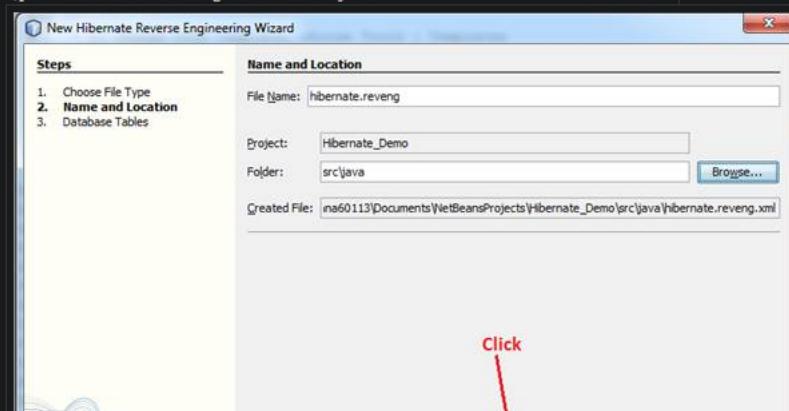
• Tạo Hibernate Reverse Engineering file

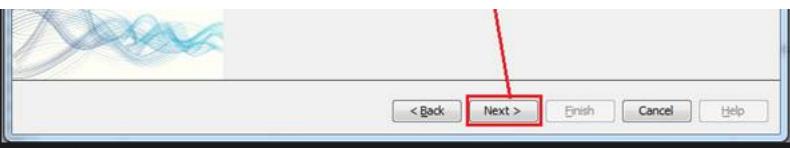
- Chọn File → New File → Other
- Chọn Category Hibernate và chọn Hibernate Reverse Engineering Wizard



Click Next.

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



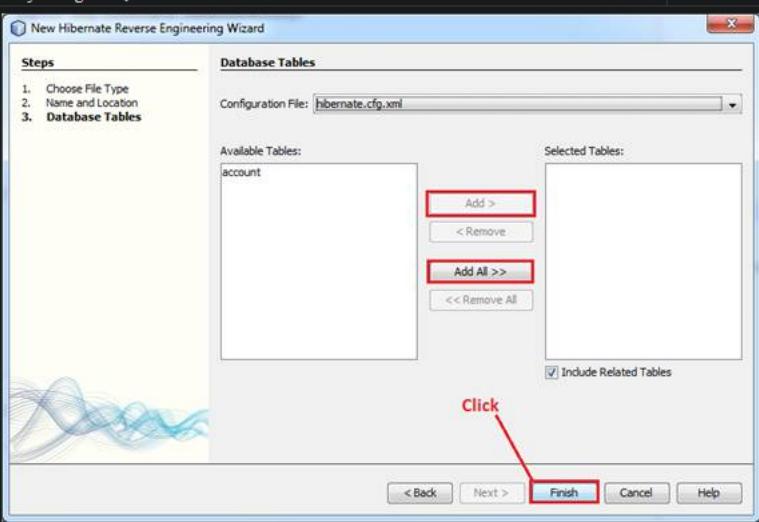


Click Next.

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.

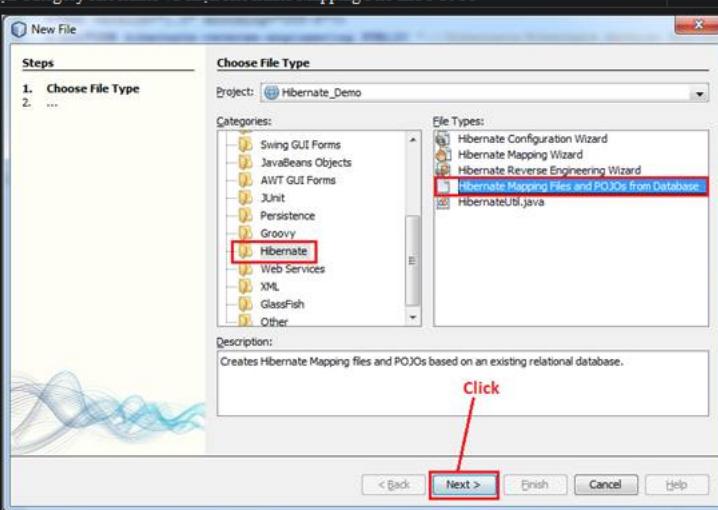


Click Finish.

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

- Chọn File → New File → Other (Ctr + N).
- Chọn Category Hibernate và chọn Hibernate Mapping File and POJOs



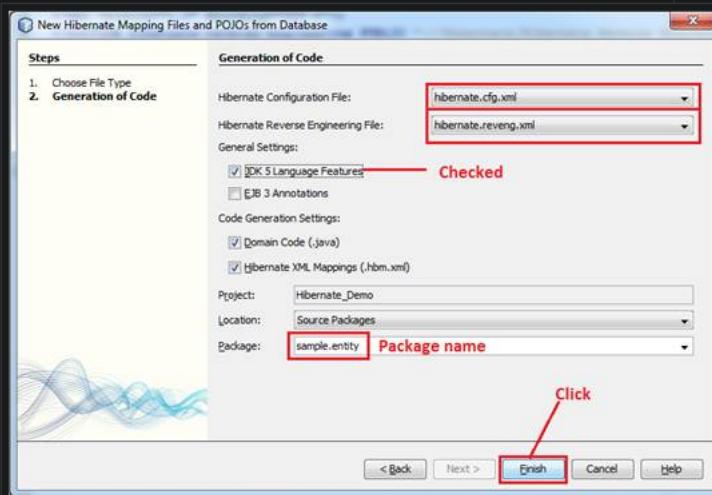
Click Next.

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

Check vào JDK 5 Language Features.

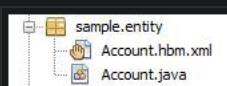
Nhập vào tên package.

Tất cả như hình vẽ bên dưới.



Click Finish.

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



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:

```
public static void main(String[] args) {
    Account acc = new Account("Khoakk", "123456", "Nguyen Anh", true);

    SessionFactory factory = HibernateUtil.getSessionFactory();
    Session session = factory.getCurrentSession();

    try {
        session.beginTransaction();
        session.save(acc);
        session.getTransaction().commit();
    } catch (Exception e) {
        if (session.getTransaction().isActive()) {
            session.getTransaction().rollback();
        }
        e.printStackTrace();
    }
}
```

2. Click chuột phải và chọn Run File.

3. Kết quả trong cửa sổ Console:

```
Apr 24, 2011 12:34:40 AM org.hibernate.cfg.SettingsFactory buildSettings
INFO: Echoing all SQL to stdout
Apr 24, 2011 12:34:40 AM org.hibernate.cfg.SettingsFactory buildSettings
INFO: Statistics: disabled
Apr 24, 2011 12:34:40 AM org.hibernate.cfg.SettingsFactory buildSettings
INFO: Deleted entity synthetic identifier rollback: disabled
Apr 24, 2011 12:34:40 AM org.hibernate.cfg.SettingsFactory buildSettings
INFO: Default entity-mode: pojo
Apr 24, 2011 12:34:40 AM org.hibernate.cfg.SettingsFactory buildSettings
INFO: Named query checking : enabled
Apr 24, 2011 12:34:40 AM org.hibernate.impl.SessionFactoryImpl <init>
INFO: building session factory
Apr 24, 2011 12:34:40 AM org.hibernate.impl.SessionFactoryObjectFactory addInstance
INFO: Not binding factory to JNDI, no JNDI name configured
Hibernate: insert into java.account (username, password, lastname, roles) values (?, ?, ?, ?)
BUILD SUCCESSFUL (total time: 5 seconds)
```

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:

```
mysql> use java;
Database changed
mysql> select * from account;
+----+-----+-----+-----+-----+
| id | username | password | lastname | roles |
+----+-----+-----+-----+-----+
| 1  | John     | 123456   | Longman  | 1      |
| 2  | Nakata   | 123456   | Hoshida   | 1      |
| 3  | Komei    | 123456   | Koha     | 0      |
| 4  | Khoaakk  | 123456   | Nguyen Anh | 1      |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Tạo Web Application với mô hình MVC sử dụng Hibernate

• Create DAO class

DAO(Data Access Object) dùng để implement các business methods. 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 như sau để thực hiện các chức năng thêm, xóa, sửa và tìm kiếm theo Lastname và Id.

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

```
public class AccountDAO {  
    private Session session;  
  
    public AccountDAO() {  
        SessionFactory factory = HibernateUtil.getSessionFactory();  
        this.session = factory.getCurrentSession();  
    }  
}
```

+ Method save() để add thêm một record:

```
public void save(Account account) {  
    try {  
        session.beginTransaction();  
        session.save(account);  
        session.flush();  
        session.getTransaction().commit();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

```
        if(session.getTransaction().isActive()) {
            session.getTransaction().rollback();
        }
        e.printStackTrace();
    }
}
```

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

```
public Account authenticate(String username, String password) {
    try {
        session.beginTransaction();
        String sql = "from Account where username = ? and password = ?";
        Query query = session.createQuery(sql);
        query.setString(0, username);
        query.setString(1, password);

        Account result = (Account)query.uniqueResult();
        session.flush();
        session.getTransaction().commit();
        return result;
    } catch (Exception e) {
        if(session.getTransaction().isActive()) {
            session.getTransaction().rollback();
        }
        e.printStackTrace();
    }
    return null;
}
```

+ Method findByLastName

```
public Account[] findByLastName(String lastname) {

    try {
        session.beginTransaction();
        String sql = "from Account where lastname like ?";
        Query query = session.createQuery(sql);
        query.setString(0, "%" + lastname + "%");
        List resultList = query.list();

        Account[] accountArray = new Account[resultList.size()];
        resultList.toArray(accountArray);
        session.flush();
        session.getTransaction().commit();
        return accountArray;
    } catch (Exception e) {
        if(session.getTransaction().isActive()) {
            session.getTransaction().rollback();
        }
        e.printStackTrace();
    }
    return null;
}
```

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

```
public void update(Account newAcc, int id) {
    try {
        session.beginTransaction();
        Account acc = (Account)session.get(Account.class, id);
        acc.setLastname(newAcc.getLastname());
        acc.setUsername(newAcc.getUsername());
        acc.setPassword(newAcc.getPassword());
        acc.setRoles(newAcc.isRoles());
        session.update(acc);
    }
```

```
        session.flush();
        session.getTransaction().commit();
    } catch (Exception e) {
        if(session.getTransaction().isActive()) {
            session.getTransaction().rollback();
        }
        e.printStackTrace();
    }
}
```

+ Method delete()

```
public void delete(int id) {
    try {
        session.beginTransaction();
        session.delete(session.get(Account.class, id));
        session.flush();
        session.getTransaction().commit();

    } catch (Exception e) {
        if(session.getTransaction().isActive()) {
            session.getTransaction().rollback();
        }
        e.printStackTrace();
    }
}
```

• Sửa file index.jsp để tạo trang Login

Sửa lại file index.jsp như sau:

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Hibernate Hello World</title>
    </head>
    <body>
        <h1>Login</h1>
        <form action="Controller" method="POST">
            Username : <input type="text" name="txtUsername" value="" /><br/>
            Password : <input type="password" name="txtPassword" value="" /><br/>
            <input type="submit" value="Login" name="action" />
        </form>
    </body>
</html>
```

• 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:

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
        String action = request.getParameter("action");
        String urlDispatcher = "";
        if(action.equals("Login")) {
            String username = request.getParameter("txtUsername");
            String password = request.getParameter("txtPassword");

            AccountDAO accountDao = new AccountDAO();
```

```
        Account result = accountDao.authenticate(username, password);

        urlDispatcher = "invalid.jsp";
        if(result != null) {
            HttpSession session = request.getSession(true);
            session.setAttribute("USER",username);
            urlDispatcher = "welcome.jsp";
        }

    }
    RequestDispatcher rd= request.getRequestDispatcher(urlDispatcher);
    rd.forward(request, response);
} catch(Exception e) {
    e.printStackTrace();
}finally {
    out.close();
}
}
```

- **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:

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Invalid Page</title>
    </head>
    <body>
        <h3>Invalid username or password!</h3>
        Click <a href="index.jsp">here</a> to try again!!!
    </body>
</html>
```

- **Tạo file welcome.jsp**

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

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Welcome</title>
    </head>
    <body>
        Welcome,<font color="blue">${sessionScope.USER}</font>
        <h3>Search last name</h3>
        <form action="Controller">
            <input type="text" name="txtSearch" value="" /><br/>
            <input type="submit" value="Search" name="action" />
        </form>
    </body>
</html>
```

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.

```
if(action.equals("Login")) {
    String username = request.getParameter("txtUsername");
    String password = request.getParameter("txtPassword");
}
```

```

AccountDAO accountDao = new AccountDAO();
Account result = accountDao.authenticate(username, password);

urlDispatcher = "invalid.jsp";
if(result != null) {
    HttpSession session = request.getSession(true);
    session.setAttribute("USER",username);
    urlDispatcher = "welcome.jsp";
}

} else if(action.equals("Search")) {
    String lastname = request.getParameter("txtSearch");

    AccountDAO accountDao = new AccountDAO();
    Account[] result = accountDao.findByLastName(lastname);
    request.setAttribute("INFO", result);
    urlDispatcher = "show.jsp";
}

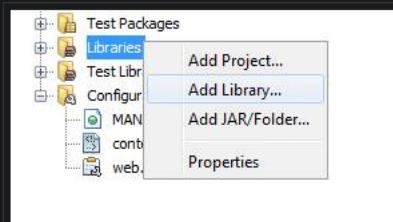
RequestDispatcher rd= request.getRequestDispatcher(urlDispatcher);
rd.forward(request, response);

```

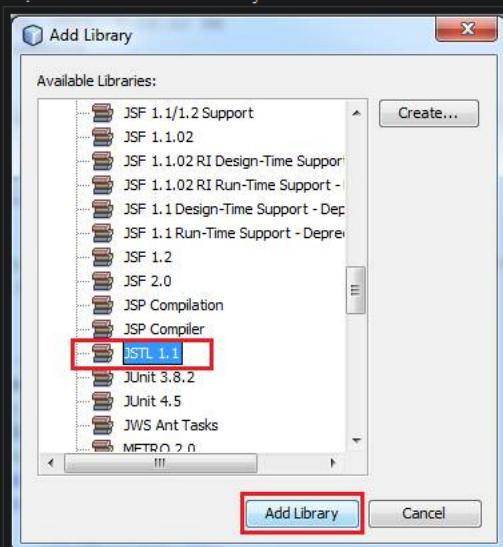
• Tạo file show.jsp

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.

```

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

```

```

<title>Result Page</title>
</head>
<body>
    Welcome,<font color="blue">${sessionScope.USER}</font>
    <h3>Search Result</h3>
    <c:set var="list" value="${requestScope.INFO}" />
    <c:if test="${not empty list}">
        <table border="1">
            <thead>
                <tr>
                    <th>No.</th>
                    <th>Username</th>
                    <th>Password</th>
                    <th>Lastname</th>
                    <th>Admin</th>
                    <th>Action</th>
                </tr>
            </thead>
            <tbody>
                <c:set var="count" value="0"/>
                <c:forEach items="${list}" var="acc">
                    <tr>
                        <form action="Controller">
                            <c:set var="count" value="${count + 1}" />
                            <td>${count}</td>
                            <td><input type="text" name="txtUsername" value="${acc.username}" /></td>
                            <td><input type="text" name="txtPassword" value="${acc.password}" /></td>
                            <td><input type="text" name="txtLastname" value="${acc.lastname}" /></td>
                            <td>
                                <c:if test="${acc.roles eq true}">
                                    <input type="checkbox" name="cbRole" checked/>
                                </c:if>
                                <c:if test="${acc.roles eq false}">
                                    <input type="checkbox" name="cbRole"/>
                                </c:if>
                            </td>
                            <td>
                                <input type="submit" value="Delete" name="action"/>
                                <input type="submit" value="Update" name="action" />
                                <input type="hidden" name="txtId" value="${acc.id}" />
                                <input type="hidden" name="txtSearch" value="${param.txtSearch}" />
                            </td>
                        </form>
                    </tr>
                </c:forEach>
            </tbody>
        </table>
    </c:if>
</body>

```

• 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.

```

} else if(action.equals("Update")){
    String txtSearch = request.getParameter("txtSearch");
    String username = request.getParameter("txtUsername");
    String password = request.getParameter("txtPassword");
    String lastname = request.getParameter("txtLastname");
    String id = request.getParameter("txtId");
    String roles = request.getParameter("cbRole");
    boolean isAdmin = false;
    if(roles != null) {
        isAdmin = true;
    }
    Account account = new Account(username, password, lastname, isAdmin);
    AccountDAO accountDao = new AccountDAO();

```

```
accountDao.update(account, Integer.parseInt(id));

urlDispatcher = "Controller?action=Search&txtSearch=" + txtSearch;

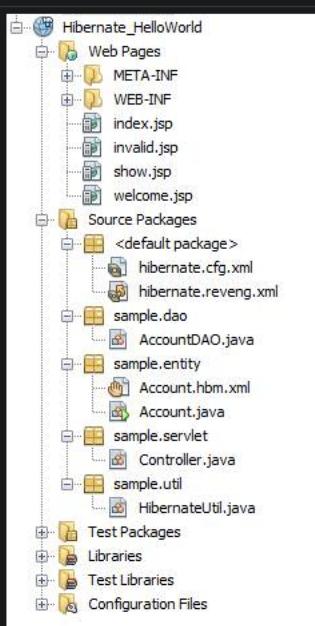
} else if(action.equals("Delete")) {
    String txtSearch = request.getParameter("txtSearch");
    String id = request.getParameter("txtId");
    AccountDAO accountDao = new AccountDAO();
    accountDao.delete(Integer.parseInt(id));

    urlDispatcher = "Controller?action=Search&txtSearch=" + txtSearch;

} else if(action.equals("Add")) {
    String txtSearch = request.getParameter("txtSearch");
    String username = request.getParameter("txtUsername");
    String password = request.getParameter("txtPassword");
    String lastname = request.getParameter("txtLastname");
    String roles = request.getParameter("cbRole");
    boolean isAdmin = false;
    if(roles != null) {
        isAdmin = true;
    }
    Account account = new Account(username, password, lastname, isAdmin);
    AccountDAO accountDao = new AccountDAO();
    accountDao.save(account);

    urlDispatcher = "Controller?action=Search&txtSearch=" + txtSearch;
}
```

Cấu trúc source của project:



Clean and build, run project.

Khi click Add, Delete hoặc 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:

Login

Username :

Password :

Welcome,khoakk

Search last name

2. Kết quả Search

Welcome,khoakk

Search Result

No.	Username	Password	Lastname	Admin	Action
1	John	123456	Longman	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Update"/>
2	Nakata	123456	Hoshida	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Update"/>
3	Komei	123456	Koba	<input type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Update"/>
4	KhoaKK	123456	Nguyen Anh	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Update"/>
New Account	KhoaAA	123456	Duong Le Anh	<input checked="" type="checkbox"/>	<input type="button" value="Add"/>

Nhập thông tin cho user mới để Add.

Kết quả sau khi Add:

Result Page

Result Page - Windows Internet Explorer

http://localhost:8084/Hibernate_HelloWorld/Controller?txtUsername=KhoaAA&txtPassword=123456&txtLastname=Duong+Le+Anh

Favorites Suggested Sites Get More Add-ons

Result Page

Welcome,khoakk

Search Result

No.	Username	Password	Lastname	Admin	Action
1	John	123456	Longman	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Update"/>
2	Nakata	123456	Hoshida	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Update"/>
3	Komei	123456	Koba	<input type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Update"/>
4	KhoaKK	123456	Nguyen Anh	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Update"/>
New Account	KhoaAA	123456	Duong Le Anh	<input checked="" type="checkbox"/>	<input type="button" value="Add"/>

Welcome, khoakk

Search Result

No.	Username	Password	Lastname	Admin	Action
1	John	123456	Longman	<input checked="" type="checkbox"/>	Delete Update
2	Nakata	123456	Hoshida	<input checked="" type="checkbox"/>	Delete Update
3	Komei	123456	Koba	<input type="checkbox"/>	Delete Update
4	Khoakk	123456	Nguyen Anh	<input checked="" type="checkbox"/>	Delete Update
5	KhoaAA	123456	Duong Le Anh	<input checked="" type="checkbox"/>	Delete Update
New Account					

```
Administrator: C:\Windows\system32\cmd.exe - mysql --user=root --password=root
mysql> select * from account;
+----+-----+-----+-----+-----+
| id | username | password | lastname | roles |
+----+-----+-----+-----+-----+
| 1  | John    | 123456  | Longman | 1      |
| 2  | Nakata  | 123456  | Hoshida  | 1      |
| 3  | Komei   | 123456  | Koba    | 0      |
| 4  | Khoakk  | 123456  | Nguyen Anh | 1      |
| 5  | KhoaAA  | 123456  | Duong Le Anh | 1      |
+----+-----+-----+-----+-----+
5 rows in set <0.00 sec>

mysql>
```

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 khác!

Được đăng bởi Kieu Trong Khanh vào lúc 08:23



Nhãn: [Hibernate](#)

1 nhận xét:



Unknown 02:34 6 tháng 7, 2017

Bài viết khá chi tiết. Mình đăng lại và có tham chiếu nguồn của bài viết này trên [Codehub.vn](#) đc ko bạn?

Trả lời

Nhập nhận xét của bạn...



Nhận xét với tên: duonghongthai

Đăng xuất

Xuất bản

Xem trước

Thông báo cho tôi

[Bài đăng Mới hơn](#)

[Trang chủ](#)

[Bài đăng Cũ hơn](#)

Đăng ký: [Đăng Nhận xét \(Atom\)](#)

Chủ đề Đơn giản. Được tạo bởi Blogger.