

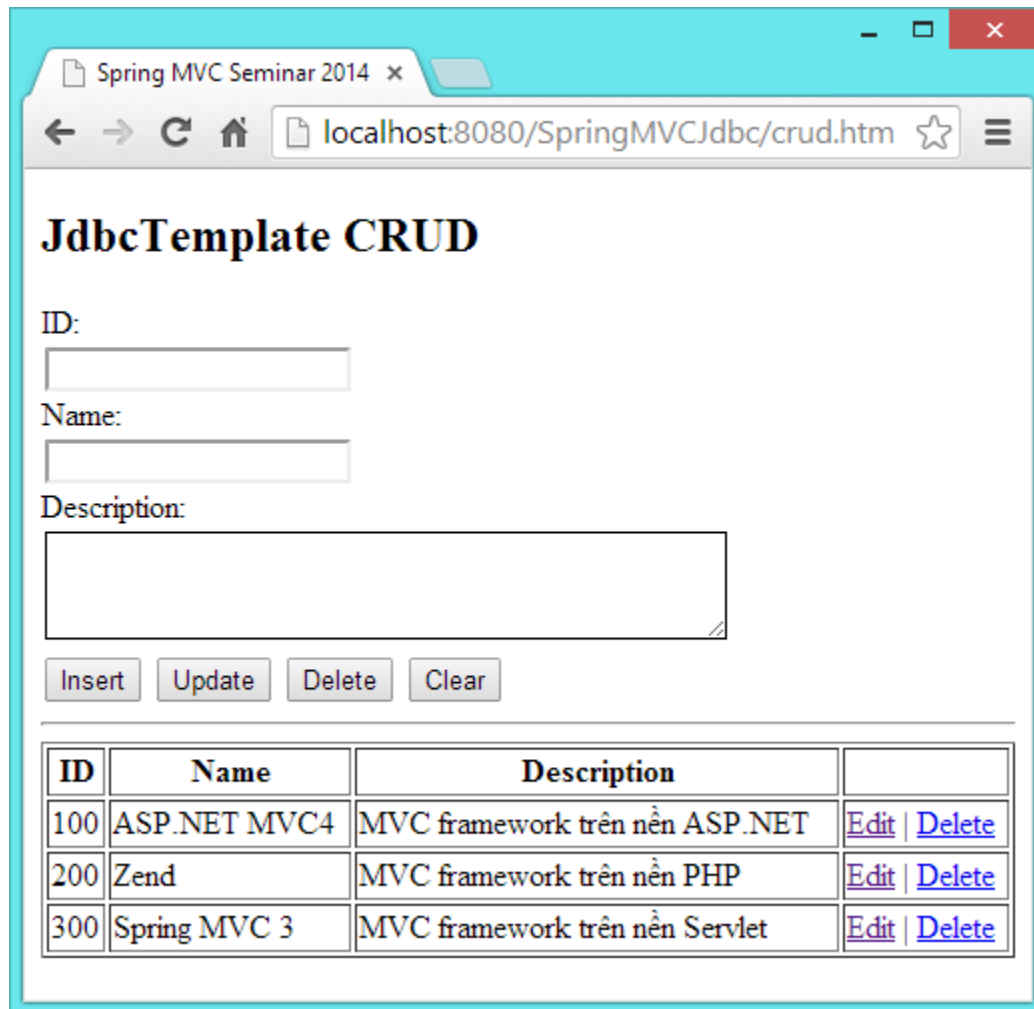
## MỤC TIÊU

Kết thúc bài thực hành này, bạn có khả năng

- ✓ Xây dựng một trang web quản lý 1 bảng trong CSDL sử dụng JdbcTemplate.
- ✓ Biết cách tổ chức và phân chia các thành phần DAO, Entity, cấu hình...

## MÔ TẢ

Trong bài này bạn phải xây dựng một ứng dụng CRUD (quản lý CSDL) như hình sau:



ID	Name	Description	
100	ASP.NET MVC4	MVC framework trên nền ASP.NET	<a href="#">Edit</a>   <a href="#">Delete</a>
200	Zend	MVC framework trên nền PHP	<a href="#">Edit</a>   <a href="#">Delete</a>
300	Spring MVC 3	MVC framework trên nền Servlet	<a href="#">Edit</a>   <a href="#">Delete</a>

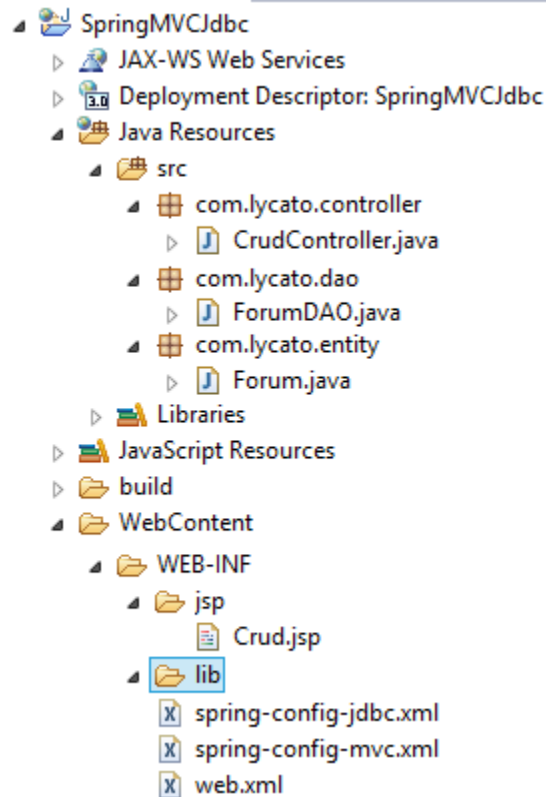
Hoạt động:

- ✓ Chạy crud.htm hiển thị giao diện gồm form nhập và lưới dữ liệu chứa danh sách các bản ghi
- ✓ Nhấp [Insert] sẽ thực hiện thêm mới 1 bản ghi vào CSDL
- ✓ Nhấp [Edit] sẽ hiển thị dữ liệu của bản ghi được chọn lên form
- ✓ Nhấp [Update] sẽ thực hiện cập nhật thông tin đã sửa đổi trên form vào CSDL

- ✓ Nhấp [Delete] sẽ thực hiện bản ghi đang xem trên form hoặc hàng trên lưới khỏi CSDL
- ✓ Nhấp [Clear] sẽ thực hiện xóa trắng form nhập

### THỰC HIỆN

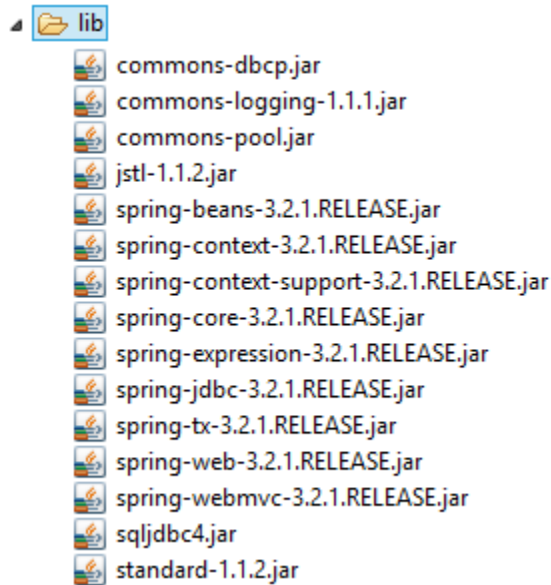
Trong bài này, bạn sẽ phải tạo một project có tổ chức như sau:



- ✓ Bước 1: Thư viện và cấu hình project
- ✓ Bước 2: CSDL
- ✓ Bước 4: Tạo lớp DAO và Entity
- ✓ Bước 5: Tạo giao diện
- ✓ Bước 6: Tạo Controller
- ✓ Bước 7: Chạy

## Bước 1: Thư viện và cấu hình project

### Thư viện



Bên cạnh các thư viện của Thư viện cần thiết cho ứng dụng

- ✓ SQLServerDriver
  - sqljdbc4.jar
- ✓ JdbcTemplate
  - commons-dbc.jar
  - spring-jdbc-3.2.1.RELEASE.jar
  - spring-tx-3.2.1.RELEASE.jar

### Cấu hình

#### ❖ 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>SpringMVCEmail</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>dispatcher</servlet-name>
    <servlet-class>
      org.springframework.web.servlet.DispatcherServlet
    </servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>/WEB-INF/spring-config-*.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>
</web-app>
```

```
</servlet>
<servlet-mapping>
    <servlet-name>dispatcher</servlet-name>
    <url-pattern>*.htm</url-pattern>
</servlet-mapping>
</web-app>
```

Cấu hình để Spring MVC nạp nhiều file cấu hình: spring-config-\*.xml. Dấu \* sẽ đại diện cho nhóm ký tự bất kỳ. Cụ thể ở bài này là mvc, gmail và upload

#### ❖ spring-config-mvc.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-3.2.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context-3.2.xsd
        http://www.springframework.org/schema/tx
        http://www.springframework.org/schema/tx/spring-tx-3.2.xsd
        http://www.springframework.org/schema/mvc
        http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd">

    <!-- Declare a view resolver -->
    <bean id="viewResolver" p:prefix="/WEB-INF/jsp/" p:suffix=".jsp"
        class="org.springframework.web.servlet.view.InternalResourceViewResolver"/>

    <!-- Spring MVC Annotation -->
    <mvc:annotation-driven />
    <context:annotation-config />

    <!-- Where to find component -->
    <context:component-scan base-package="com.lycato" />
</beans>
```

- ✓ Khai báo bean InternalResourceViewResolver để xử lý view
- ✓ Chỉ rõ package tìm kiếm các component là com.lycato
- ✓ Chỉ rõ ứng dụng Spring này được phép sử dụng annotation

#### ❖ spring-config-jdbc.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-3.2.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context-3.2.xsd
        http://www.springframework.org/schema/tx
        http://www.springframework.org/schema/tx/spring-tx-3.2.xsd
```

<http://www.springframework.org/schema/mvc>  
<http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd>

```
<bean id="dataSource" destroy-method="close"
      class="org.apache.commons.dbcp.BasicDataSource"
      p:driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
      p:url="jdbc:sqlserver://localhost:1433;DatabaseName=Seminar"
      p:username="sa"
      p:password="songlong"/>

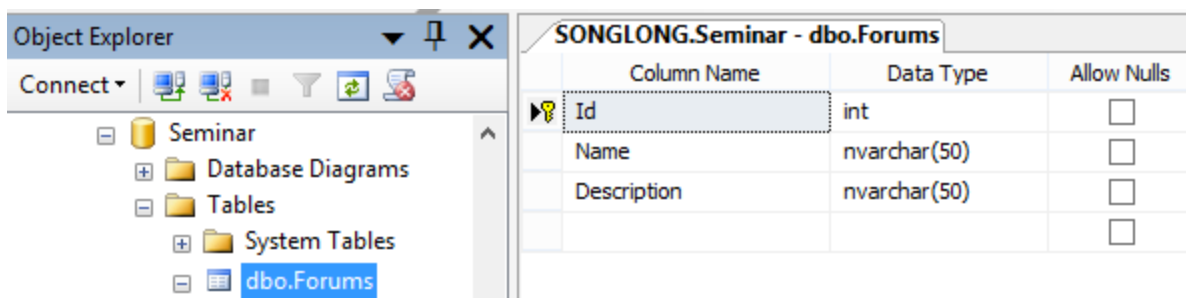
<bean id="jdbcTemplate"
      class="org.springframework.jdbc.core.JdbcTemplate">
  <constructor-arg ref="dataSource"/>
</bean>

</beans>
```

Trong file cấu hình này bạn phải khai báo 2 bean.

- ✓ BasicDataSource: bean này cấu hình các thông số kết nối CSDL
- ✓ JdbcTemplate: bean này được khai báo để làm việc với CSDL được tiêm vào và sử dụng sau này trong ứng dụng

### Bước 2: CSDL



Column Name	Data Type	Allow Nulls
<b>Id</b>	int	<input type="checkbox"/>
Name	nvarchar(50)	<input type="checkbox"/>
Description	nvarchar(50)	<input type="checkbox"/>
		<input type="checkbox"/>

Hình: CSDL Seminar chứa bảng Forums

### Bước 4: Tạo lớp mô tả và truy xuất dữ liệu

#### Lớp mô tả dữ liệu (Entity)

Lớp này mô tả cấu trúc bảng. Mục đích là để chứa một bản ghi dữ liệu thao tác với CSDL. Nó cũng được kết nối với các trường form để hiển thị dữ liệu đọc được cho người dùng xem.

```
package com.lycato.entity;

public class Forum {

    private Integer id;
    private String name, description;

    public Integer getId() {
        return id;
    }

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

```
public String getName() {  
    return name;  
}  
public void setName(String name) {  
    this.name = name;  
}  
  
public String getDescription() {  
    return description;  
}  
public void setDescription(String description) {  
    this.description = description;  
}  
}
```

### Lớp truy xuất dữ liệu (DAO)

Lớp này chứa các phương thức thao tác dữ liệu (thêm, sửa, xóa) và truy vấn dữ liệu.

- ✓ Insert(): thêm
- ✓ Update(): sửa
- ✓ Delete(): xóa
- ✓ getXyz(): truy vấn

Lớp nào được chú thích bởi `@Repository` để có thể tiêm vào `CrudController` trong ứng dụng bởi `@Autowired` để sử dụng sau này.

```
package com.lycato.dao;  
  
import java.io.Serializable;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.jdbc.core.BeanPropertyRowMapper;  
import org.springframework.jdbc.core.JdbcTemplate;  
import org.springframework.jdbc.core.RowMapper;  
import org.springframework.stereotype.Repository;  
  
import com.lycato.entity.Forum;  
  
@Repository  
public class ForumDAO {  
  
    /**  
     * Inject từ <bean class="...JdbcTemplate">  
     */  
    @Autowired  
    protected JdbcTemplate jdbc;  
  
    /**  
     * Thêm mới 1 thực thể  
     * @param entity là thực thể mới  
     */  
    public void insert(Forum forum) {  
        String sql = "INSERT INTO Forums (Id, Name, Description) VALUES (?, ?, ?)";  
        jdbc.update(sql, forum.getId(), forum.getName(), forum.getDescription());  
    }  
}
```

```
/**
 * Cập nhật thực thể
 * @param entity là thực thể cần cập nhật
 */
public void update(Forum forum) {
    String sql = "UPDATE Forums SET name=?, Description=? WHERE Id=?";
    jdbc.update(sql, forum.getName(), forum.getDescription(), forum.getId());
}

/**
 * Xóa thực thể theo mã
 * @param id mã thực thể cần xóa
 */
public void delete(Serializable id) {
    String sql = "DELETE FROM Forums WHERE Id=?";
    jdbc.update(sql, id);
}

/**
 * Truy vấn 1 thực thể theo mã
 * @param id mã thực thể cần truy vấn
 * @return thực thể truy vấn được
 */
public Forum getById(Serializable id) {
    String sql = "SELECT * FROM Forums WHERE Id=?";
    return jdbc.queryForObject(sql, getRowMapper(), id);
}

/**
 * Truy vấn tất cả các thực thể
 * @return danh sách thực thể truy vấn được
 */
public List<Forum> getAll() {
    String sql = "SELECT * FROM Forums";
    return jdbc.query(sql, getRowMapper());
}

/**
 * Truy vấn các thực thể theo câu lệnh sql
 * @param sql câu lệnh truy vấn
 * @return danh sách thực thể truy vấn được
 */
protected List<Forum> getBySql(String sql) {
    return jdbc.query(sql, getRowMapper());
}

/**
 * Truy vấn thực thể theo tên
 * @param name tên của thực thể cần truy vấn
 * @return danh sách thực thể truy vấn được
 */
public List<Forum> getName(String name) {
    String sql = "SELECT * FROM Forums WHERE Name LIKE ?";
    return jdbc.query(sql, getRowMapper(), "%" + name + "%");
}

/**
 * Ánh xạ cấu trúc bản ghi theo thuộc tính của bean
 * @return ánh xạ bản ghi theo thuộc tính bean
 */
private RowMapper<Forum> getRowMapper() {
    return new BeanPropertyRowMapper<Forum>(Forum.class);
}
```

### Bước 5: Tạo giao diện

Giao diện cho thấy 2 thuộc tính model cần thiết là forum (dùng để hiển thị dữ liệu của bản ghi được chọn edit lên form) và forums (là danh sách cần thiết để hiển thị dữ liệu cho lưới phía dưới).

Phản tương tác người dùng: chúng ta thấy có 7 hành động tương tác với trang web này:

1. Chạy trang web này với crud.htm (không có tham số nào)
2. Nhấp nút [insert] (có tham số insert)
3. Nhấp nút [update] (có tham số update)
4. Nhấp nút [delete] (có tham số delete)
5. Nhấp nút [clear] (không có tham số nào)
6. Nhấp liên kết [edit] (có tham số edit)
7. Nhấp liên kết [delete] (có tham số delete)

```
<%@page pageEncoding="utf-8" contentType="text/html; charset=utf-8" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Spring MVC Seminar 2014</title>
</head>
<body>
    <h2>JdbcTemplate CRUD</h2>

    <form:form action="crud.htm"
        method="post" modelAttribute="forum">
        <div>ID:</div>
        <form:input path="id"/>

        <div>Name:</div>
        <form:input path="name"/>

        <div>Description:</div>
        <form:textarea path="description" rows="3" cols="40"/>
        <br>
        <input type="submit" value="Insert" name="insert">
        <input type="submit" value="Update" name="update">
        <input type="submit" value="Delete" name="delete">
        <input type="submit" value="Clear">
    </form:form>
    <hr>
    <table border="1" style="width:100%">
    <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Description</th>
        <th>&nbsp;</th>
    </tr>
    <c:forEach var="f" items="${forums}">
    <tr>
        <td>${f.id}</td>
        <td>${f.name}</td>
```



```
<td>${f.description}</td>
<td>
    <a href="?edit&id=${f.id}">Edit</a> |
    <a href="?delete&id=${f.id}">Delete</a>
</td>
</tr>
</c:forEach>
</table>
</body>
</html>
```

## Bước 6: Tạo Controller

```
package com.lycato.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;

import com.lycato.dao.ForumDAO;
import com.lycato.entity.Forum;

@Controller
@RequestMapping(value="crud")
public class CrudController {

    /**
     * Inject từ @Repository ForumDAO
     */
    @Autowired
    private ForumDAO dao;

    /**
     * GET|POST: crud.htm
     */
    @RequestMapping()
    public String initiate(ModelMap model) {
        model.addAttribute("forum", new Forum());
        model.addAttribute("forums", dao.getAll());
        return "Crud";
    }

    /**
     * POST: crud.htm?insert
     */
    @RequestMapping(params="insert", method = RequestMethod.POST)
    public String insert(@ModelAttribute("forum") Forum forum, ModelMap model) {
        dao.insert(forum);
        return initiate(model);
    }

    /**
     * POST: crud.htm?update
     */
    @RequestMapping(params="update", method = RequestMethod.POST)
    public String update(@ModelAttribute("forum") Forum forum, ModelMap model) {
        dao.update(forum);
        model.addAttribute("forums", dao.getAll());
    }
}
```

```

        return "Crud";
    }

    /**
     * GET|POST: crud.htm?delete
     */
    @RequestMapping(params="delete")
    public String delete(@RequestParam("id") Integer id, ModelMap model) {
        dao.delete(id);
        return initiate(model);
    }

    /**
     * GET: crud.htm?edit
     */
    @RequestMapping(params="edit", method = RequestMethod.GET)
    public String edit(@RequestParam("id") Integer id, ModelMap model) {
        Forum f = dao.getById(id);
        model.addAttribute("forum", f);
        model.addAttribute("forums", dao.getAll());
        return "Crud";
    }
}

```

Khi bạn tương tác vào chương trình tương ứng với 7 hành động đã được phân tích trong phần giao diện thì các phương thức cài đặt mã chuyên dụng sẽ được thực hiện như sau:

STT	Hành động	Phương thức	Mô tả
1	Chạy crud.htm	Initiate()	Chuẩn bị dữ liệu hiển thị lên form và lưới
2	Nhấp [Insert]	Insert()	Thêm đối tượng forum vào CSDL (đối tượng forum chứa toàn bộ dữ liệu form nhập). Thực hiện lại (1)
3	Nhấp [Update]	Update()	Cập nhật đối tượng forum vào CSDL
4	Nhấp [Delete]	Delete()	Xóa bản ghi theo id (nhận từ form). Thực hiện lại (1)
5	Nhấp [Clear]	Initiate()	Chuẩn bị dữ liệu hiển thị lên form và lưới
6	Nhấp liên kết [Edit]	Edit()	Truy vấn bản ghi từ CSDL theo id và thay thế thuộc tính forum trong model để hiển thị thông tin lên form
7	Nhấp liên kết [Delete]	Delete()	Xóa bản ghi theo id (nhận từ form)

### Bước 7: Chạy

<http://localhost:8080/SpringMVCJdbc/crud.htm>