

BÀI GIẢNG MÔN HỌC LẬP TRÌNH MẠNG



NGÔN NGỮ LẬP TRÌNH JSP-SERVLET

Java Server Pages

Nội Dung

- * Giới thiệu
- * Cơ chế hoạt động của JSP
- * Cách xây dựng Web bằng JSP
- * Các cú pháp cơ bản của JSP
- * Nhúng và chuyển hướng giữa các trang
- * Các đối tượng mặc định của JSP
- * Phương thức truy xuất cơ sở dữ liệu

Java Server Pages

I. Giới Thiệu

JSP là cạnh tranh của Sun đối với công nghệ ASP của Microsoft.

ASP cũng là các trang Web thực hiện cơ chế diễn dịch thay các lệnh Java là các lệnh Visual Basic. ASP không có cơ chế diễn dịch ra mã nhị phân như JSP.

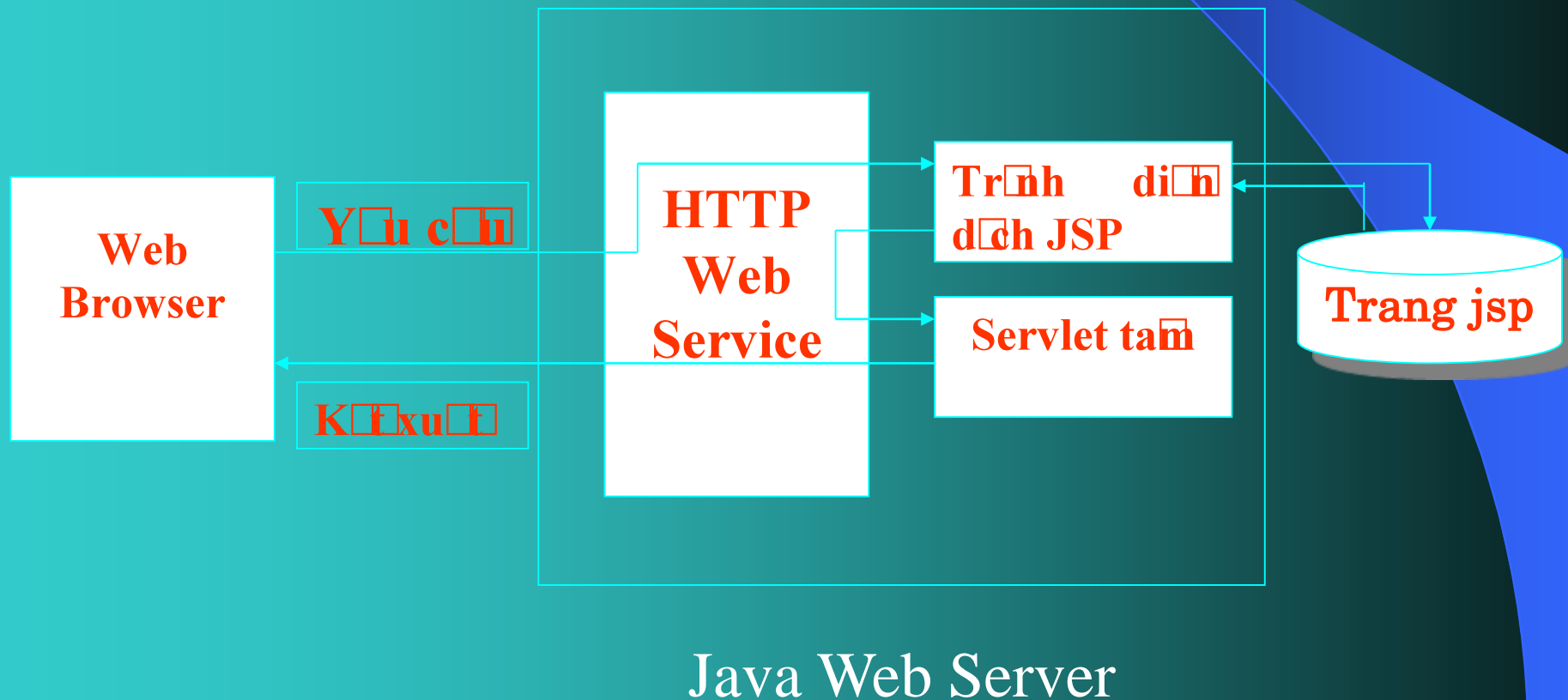
JSP (Java Server Page) là cách đơn giản để người dùng tiếp cận đọc hớng lập trình Web phía máy chủ hiệu quả và nhanh hơn.

Nếu như đối với Servlet bạn phải viết mã Java và biên dịch bằng tay trước khi đưa vào sử dụng với trình chủ Web Server thì JSP không cần điều này

Java Server Pages

II. Cơ Chế Hoạt Động Của JSP

Cơ chế hoạt động của trang JSP được minh họa như hình sau :



Java Server Pages

II. Cơ Chế Hoạt Động Của JSP :

- + Khi bạn yêu cầu một trang JSP, trình chủ sẽ đọc trang JSP từ đĩa cứng,
- + Bộ diễn dịch JSP sẽ diễn dịch mã lệnh Java chứa trong trang JSP thành một servlet.
- + Sau đó trình chủ Java Web Server sẽ triệu gọi servlet trả kết xuất thuần HTML về cho trình khách

Java Server Pages

III. Cách Xây Dựng Web Bằng JSP

Một trang JSP với phần mở rộng là .jsp như là một trang HTML, có thể chèn thêm các câu lệnh Java bằng cặp

`<% . . . %>`

hoặc: `<jsp:scriptlet>... </jsp:scriptlet>.`

Giữa cặp dấu này là các câu lệnh của ngôn ngữ Java nằm trong phần thân (`<Body>. . . </Body>`) của tài liệu, hoặc phần đầu tiên trước cả tag `<HTML>` của tài liệu.

Java Server Pages

III. Cách Xây Dựng Web Bằng JSP

Mô tả cấu trúc của một trang JSP

<%

----- Các câu lệnh Java ở đây -----

%>

< HTML>

----- Các thành phần định nghĩa khác -----

<BODY>

<%

----- Các câu lệnh Java ở đây -----

%>

----- Các định dạng khác của HTML -----

< /BODY>

</HTML>

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

Tương tự thẻ HTML, thẻ lệnh JSP cũng bao gồm thẻ mở và thẻ đóng. Mỗi thẻ có các thuộc tính qui định cách sử dụng thẻ.

1. Thẻ `<jsp:scriptlet>` hay `<% %>`

Với thẻ này bạn có thể đặt các đoạn mã lệnh Java ở giữa cặp thẻ tương tự một chương trình Java thông thường. Bên trong mã Java nếu muốn kết xuất dữ liệu HTML bạn gọi phương thức `println()` của đối tượng `out`.

```
<jsp:scriptlet>
```

```
-----
```

```
mã lệnh Java
```

```
-----
```

```
</jsp:scriptlet>
```

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

1. Thẻ `<jsp:scriptlet>` hay `<% %>`

Khi gặp thẻ này, bộ diễn dịch JSP sẽ biết được lệnh cần thực hiện tiếp theo là mã lệnh Java. Bạn hoàn toàn sử dụng mọi cú pháp của Java ở đây, kể cả lệnh chú thích thông thường. Nếu muốn viết tắt thì có thể sử dụng cặp thẻ `<% %>` thay thế.

`<%`

mã lệnh Java

`%>`

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

2. Hiển thị kết xuất bằng cú pháp `<%= %>`

Thay vì sử dụng cú pháp `<% %>` để diễn đạt một khối gồm nhiều lệnh, ta có thể sử dụng cú pháp `<%= %>` chỉ để hiển thị kết xuất của một giá trị biến hay hàm nào đó.

Riêng biểu thức nằm trong thẻ `<%= %>` không được tận cùng bằng dấu chấm phẩy (;) như các lệnh hay khai báo biến của ngôn ngữ Java.

Ví dụ:

```
Welcome <%=username%>
```

username : là tên biến do người dùng đăng nhập.

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

3. Chèn chú thích vào trang JSP :

Cũng như Java, JSP cho phép bạn dùng cú pháp `//` để chú thích một dòng mã lệnh trong khi cú pháp `/* */` áp dụng cho nhiều dòng. Các dòng chú thích sẽ bỏ qua khi trình chủ diễn dịch trang JSP.

```
<%
```

```
// Lấy biến dữ liệu mang tên username từ đối tượng session
```

```
String username=session.getAttribute("username");
```

```
%>
```

JSP còn cung cấp cho bạn cú pháp chú thích `<%-- --%>`. Tất cả các khối lệnh Java và HTML nằm giữa hai dấu chú thích này sẽ được trình biên dịch trang bỏ qua không quan tâm đến.

Ví dụ : `<%--`

```
out.println("You will never see this line");
```

```
--%>
```

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

4. Thẻ khai báo phương thức và biến bằng `<%! %>`

Ta có thể dùng thẻ `<%! %>` để khai báo một biến hoặc một phương thức dùng cho toàn trang jsp. Biến khai báo phải theo cú pháp của ngôn ngữ Java. Phương thức và biến sau đó có thể được gọi bất kỳ nơi đâu trong trang JSP.

5. Thẻ chỉ dẫn biên dịch trang `<%@ page%>`

Thẻ `<%@ page%>` chỉ dẫn một số tính chất biên dịch áp dụng cho toàn trang jsp. Ta có thể sử dụng thẻ này để khai báo các thư viện import của Java, chỉ định tùy chọn trang jsp có cần giữ trên cache bộ nhớ của trình chủ để tăng tốc hay không.

Ví dụ:

```
<%@ page import="java.sql.*" %>
```

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

6. Nhúng Và Chuyển Hướng Giữa Các Trang

Trong quá trình xử lý và chuyển dữ liệu cho form bạn có thể nhúng và triệu gọi những trang JSP khác với trang hiện hành.

Có thể đưa một trang HTML hoặc JSP vào trang JSP hiện tại để hỗ trợ thêm thư viện hay tính năng bổ sung nào đó. Hay ghi nhận đọc dữ liệu submit từ trình khách, trang JSP nhận được dữ liệu có thể chuyển hướng hoặc triệu gọi đến trang JSP khác.

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

6.1. Nhúng file vào trang với chỉ thị include file :

Ta có thể nhúng một trang jsp hay trang html vào trang jsp hiện hành bằng thẻ `<%@ include file %>`.

Thẻ này tương tự như chỉ dẫn `#include` trong ngôn ngữ C. Cú pháp đầy đủ của thẻ này là :

`<%@ include file="URL or Filepath"%>`

6.2. Sử dụng thẻ `<jsp:include>`

Chỉ thị `<@ include %>` chỉ dùng để nhúng các mã nguồn tĩnh. Nếu muốn nhúng kết quả kết xuất từ các trang jsp, servlet hay html khác vào trang hiện hành bạn sử dụng thẻ `<jsp:include>` với cú pháp như sau :

`<jsp:include page="filename" flush="true" />`

Lu ý : `<jsp:include>` đưa kết quả kết xuất của trang được nhúng vào trang hiện hành, trong khi `<%@ include %>` dùng để đưa mã nguồn của trang được nhúng vào trang hiện hành.

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

6.3 Chuyển tiếp đến trang khác với thẻ `<jsp:forward>`

Ta sử dụng thẻ `<jsp:forward>` để triệu gọi và chuyển dữ liệu cho trang jsp khác xử lý.

Ví dụ : Trang Page1.jsp chỉ hiển thị thông tin để người dùng nhập vào username và password

```
<html>
```

```
  <form method=post  action=page1.jsp>
```

```
    username : <input type=text  name="username" >
```

```
    password : <input type=password  name="password" >
```

```
      <input type=submit  value="Login" >
```

```
  </form>
```

```
</html>
```


Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

6.3 Chuyển tiếp đến trang khác với thẻ `<jsp:forward>`

Trang Page2.jsp kiểm tra nếu username khác rỗng sẽ chuyển tiếp quyền xử lý file Page3.jsp, ngược lại nếu username bằng rỗng sẽ đưa ra thông báo yêu cầu người dùng nhập vào.

```
<% if (!request.getParameter("username").equals(" ")) {  
    %>  
<jsp:forward page="Page3.jsp" />  
<%} else { out.println(" Please input username and password ");}  
%>  
<html>  
    Your username : <%=request.getParametter("username")%>  
    <br>  
    Your password : <%=request.getParameter("password")%>;  
    <br>  
</html>
```

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

6.4. Chuyển hướng sang trang mới với sendRedirect()

Ngoài việc chuyển tiếp bạn có thể sử dụng phương thức sendRedirect() để chuyển hướng và triệu gọi trang JSP khác xử lý.

Sự khác nhau giữa chuyển tiếp (forward) và chuyển hướng (redirect) đó là lệnh senRedirect() chỉ thực hiện triệu gọi trang mà không chuyển tham số của trang hiện hành cho trang chuyển hướng.

Ví dụ : Từ Page2.jsp bạn có thể chuyển hướng đến trang Page3.jsp như sau :

```
<%
```

```
-----
```

```
response.sendRedirect("Page3.jsp");
```

```
%>
```

Java Server Pages

V. Các Đối Tượng Mặc Định Của Trang JSP

Trình diễn dịch JSP cho phép ta sử dụng một số đối tượng đã khai báo trước. Điều này sẽ giúp ta viết mã lệnh trong trang jsp nhanh hơn servlet.

1. Đối tượng out : Xuất phát từ lớp `java.io.PrintWriter`. Bạn có thể sử dụng đối tượng này để định dạng kết xuất gửi về máy khách.

Ví dụ: `<% out.println("Result "+7*3); %>`

Java Server Pages

V. Các Đối Tượng Mặc Định Của Trang JSP

2. Đối tượng Request :

Xuất phát từ lớp `HttpServletRequest`, đối tượng này giúp bạn lấy về các tham số hay dữ liệu do trình khách chuyển lên.

Một trong những ứng dụng thông thường nhất của đối tượng request là nắm giữ các tham số. Bạn có thể thấy điều này bằng cách gọi phương thức `getParameter()` của request.

Phương thức này kế thừa từ lớp cha là `javax.servlet.ServletRequest`. Phương thức này nhận tên tham số và trả về giá trị chuỗi tương ứng với tên của tham số đó.

Ví dụ: Lấy về các thông tin đăng nhập từ file `Bc_td1.jsp`

```
String USERNAME=request.getParameter("username");  
String PASSWORD=request.getParameter("password");
```

Java Server Pages

V. Các Đối Tượng Mặc Định Của Trang JSP

3. Đối tượng Session :

Xuất phát từ lớp HttpSession. Bạn sử dụng đối tượng session để theo dõi kết nối và lưu vết một phiên làm việc giữa trình khách và trình chủ.

Đối tượng session tham chiếu đến đối tượng javax.servlet.http.HttpSession. Đối tượng session được khởi tạo bằng cách gọi phương thức pageContext.getSession() để tạo ra trang servlet.

```
session = pageContext.getSession();
```

Khi cần một biến nào đó có giá trị toàn cục từ khi mở cho đến khi kết thúc trình duyệt đó, bạn nên nghĩ đến session. Đối tượng session sẽ thực hiện việc đó giúp bạn. Đối tượng session tạo biến cục bộ cho phép lưu một giá trị nào đó từ trang JSP này sang trang JSP khác.

Java Server Pages

V. Các Đối Tượng Mặc Định Của Trang JSP

3. Đối tượng Session :

Ví dụ : Để lưu thông tin về phiên làm việc vào session ta thực hiện như sau:

```
session.putValue("NGAY", NGÀY);  
session.putValue("THANG", THANG);  
session.putValue("NAM", NAM);
```

Sau khi ghi thông tin vào session, khi cần lấy về giá trị trong session ta thực hiện nh sau :

```
String NGÀY=(String) session.getValue("NGAY");  
String THANG=(String) session.getValue("THANG");  
String NAM=(String) session.getValue("NAM");
```

Java Server Pages

V. Các Đối Tượng Mặc Định Của Trang JSP

4. Đối tượng Response :

Đối tượng response cho phép phản hồi thông tin xử lý từ trình chủ trở về trình duyệt. Thực ra đối tượng response thường gọi phương thức `response.getWriter()` để thực hiện kết xuất. Các phương thức mà đối tượng response thường được liệt kê như sau :

Liên kết trực tiếp đến trang Bc_td2.jsp

```
response.sendRedirect("Bc_td2.jsp") ;
```

Định kết xuất ra HTML hay text

```
response.setContentType("text/html") ;
```

Lấy dữ liệu trong vùng đệm nhờ `OutputStream()`;

```
response.getOutputStream();
```

Java Server Pages

V. Các Đối Tượng Mặc Định Của Trang JSP

5. Đối tượng Exception :

Đối tượng exception chỉ tồn tại trong trang xử lý lỗi (errorPage). Nó dùng để tham chiếu đến nguyên nhân gây ra lỗi mà trang error có liên quan.

Java Server Pages

VI. Truy Xuất CSDL Bằng JSP :

1. JDBC và trình điều khiển truy xuất cơ sở dữ liệu

Công việc lập trình mà JSP thường thực hiện nhất đó là lưu trữ thông tin và truy xuất cơ sở dữ liệu (database) cung cấp thông tin cho trình khách.

Việc truy xuất cơ sở dữ liệu trong JSP dựa vào trình truy xuất JDBC mà Java hỗ trợ. JDBC là các trình đạo diễn cho phép bạn truy cập vào cơ sở dữ liệu của các hệ như Access, MS SQL Server, Oracle,... bằng ngôn ngữ truy vấn SQL. Có 4 kiểu JDBC nh sau:

- + JDBC sử dụng cầu nối ODBC*
- + JDBC kết nối trực tiếp với các trình điều khiển cơ sở dữ liệu.*
- + JDBC kết nối thông qua các ứng dụng mạng trung gian.*
- + JDBC kết nối thông qua các trình điều khiển đặc thù ở xa*

Java Server Pages

VI. Truy Xuất CSDL Bằng JSP :

1. JDBC và trình điều khiển truy xuất cơ sở dữ liệu

+ Kiểu 1: JDBC sử dụng cầu nối ODBC

Nếu bạn đã từng làm việc với các hệ cơ sở dữ liệu trên windows bạn sẽ biết đến cách kết nối ODBC (Open Database Connectivity).

ODBC là cách kết nối tổng quát nhất đến mọi hệ cơ sở dữ liệu thông qua trình quản lý ODBC của Windows. Mỗi hệ cơ sở dữ liệu cung cấp một trình điều khiển (driver) có khả năng đọc hiểu đọc cơ sở dữ liệu của chúng.

Trình điều khiển này sẽ được đăng ký với bộ quản lý ODBC. Java cung cấp cho bạn khả năng sử dụng JDBC để gọi ODBC và bằng cách này bạn có thể kiểm soát rất nhiều hệ quản trị cơ sở dữ liệu khác nhau.

Java Server Pages

VI. Truy Xuất CSDL Bằng JSP :

1. JDBC và trình điều khiển truy xuất cơ sở dữ liệu

+ Kiểu 2: JDBC kết nối trực tiếp với các trình điều khiển cơ sở dữ liệu.

Khi Java phát triển, các nhà cung cấp hỗ trợ các trình điều khiển JDBC kết nối trực tiếp với các trình điều khiển đặc thù của mỗi hệ cơ sở dữ liệu (native database driver) mà không cần phải qua trung gian ODBC của windows nữa.

Các kết nối này nhanh và hiệu quả hơn kiểu 1 JDBC-ODBC

Java Server Pages

VI. Truy Xuất CSDL Bằng JSP :

1. JDBC và trình điều khiển truy xuất cơ sở dữ liệu

+ Kiểu 3: JDBC kết nối thông qua các ứng dụng mạng trung gian.

Ngày nay, các ứng dụng đều gắn liền với Internet. Theo mô hình lập trình phân tán, ứng dụng trung gian (middle-ware) trên máy chủ sẽ chịu trách nhiệm kết nối với cơ sở dữ liệu.

Nhiệm vụ của trình khách là sử dụng những trình điều khiển JDBC gọn nhẹ có khả năng giao tiếp với ứng dụng trung gian qua Internet để truy vấn dữ liệu.

Java Server Pages

VI. Truy Xuất CSDL Bằng JSP :

1. JDBC và trình điều khiển truy xuất cơ sở dữ liệu

+ Kiểu 4: JDBC kết nối thông qua các trình điều khiển đặc thù ở xa.

Mô hình kết nối theo kiểu 3 do phải thông qua ứng dụng trung gian (middle-ware) nên có thể ảnh hưởng đến tốc độ truy xuất cơ sở dữ liệu phía máy khách.

Java đã ra kết nối JDBC kiểu 4 cho phép trình khách sử dụng các trình điều khiển gọn nhẹ kết nối trực tiếp vào trình điều khiển cơ sở dữ liệu đặc thù (thuần java) trên máy chủ ở xa thông qua mạng Internet.

Java Server Pages

VI. Phương Thức Truy Xuất CSDL Bằng JSP :

2. Phương Thức Truy Xuất CSDL Bằng Jsp

2.1 Thực hiện một kết nối với CSDL :

Để kết nối với CSDL, JDBC đòi hỏi hai yếu tố là **trình điều khiển** (tương ứng với các kiểu kết nối) và **thông tin để kết nối** (địa chỉ máy chủ, tên tài khoản, mật khẩu,...).

+ Nạp trình điều khiển: Ta sử dụng trình điều khiển miễn phí do Sun cung cấp mang tên JdbcOdbcDriver. Ta khai báo tên lớp này như sau :

```
String drivename="sun.jdbc.odbc.JdbcOdbcDriver";
```

Tiếp đến ta gọi phương thức tĩnh `forName` của lớp Class để nạp và khởi tạo trình điều khiển :

```
Class.forName("drivename").newInstance();
```

Java Server Pages

VI. Phương Thức Truy Xuất CSDL Bằng JSP :

2. Phương Thức Truy Xuất CSDL Bằng Jsp

2.1 Thực hiện một kết nối với CSDL :

+ *Tạo thông tin kết nối* : Lúc này ta đã hoàn toàn có khả năng sử dụng trình điều khiển JDBC-ODBC để truy xuất dữ liệu. Tuy nhiên trình điều khiển cần biết thêm những thông tin cụ thể như tài khoản đăng nhập, thông tin kết nối,...Ta đặt những thông tin này trong chuỗi kết nối như sau :

```
String url="jdbc:odbc:tên CSDL";  
String username="";  
String password="";
```

Ở đây chuỗi jdbc:odbc: là bắt buộc với trình điều khiển JDBC-ODBC. Tiếp sau chuỗi này là thông tin kết nối cụ thể đến CSDL do ODBC qui định.

Java Server Pages

VI. Phương Thức Truy Xuất CSDL Bằng JSP :

2. Phương Thức Truy Xuất CSDL Bằng Jsp

2.1 Thực hiện một kết nối với CSDL :

+ *Tạo thông tin kết nối :*

Tiếp tục ta tạo một đối tượng Connection bằng cách gọi phương thức getConnection() của lớp DriverManager để yêu cầu trình điều khiển nạp bởi Class.forName() trước đó tiếp nhận thông tin và thực thi kết nối như sau:

```
Connection conn = null;
```

```
conn = DriverManager.getConnection(url, "username", "password ");
```

url : chuỗi nêu lên đặc điểm của CSDL có dạng

jdbc:subprotocol:subname

subprotocol : giao thức con tương ứng với loại CSDL

subname : tên của CSDL

username : tên người dùng khi đăng nhập vào CSDL

password : mật khẩu người dùng khi đăng nhập CSDL

Java Server Pages

VI. Phương Thức Truy Xuất CSDL Bằng JSP :

2. Phương Thức Truy Xuất CSDL Bằng Jsp

2.2 Thực hiện các lệnh :

Các chương trình JSP chỉ thực hiện được các lệnh trên CSDL thông qua đối tượng Statement. Đối tượng này được dùng để thực thi mọi câu lệnh sql.

Ta tạo đối tượng Statement từ kết nối *conn*

```
Statement stmt =null;  
stmt = conn.createStatement();
```

Các câu lệnh sql có thể được thực hiện tức thì thông qua đối tượng Statement, có thể là một câu lệnh biên dịch trước (đối tượng PreparedStatement) hay có thể là một lệnh gọi các thủ tục cài sẵn (store procedure) trong CSDL (đối tượng CallableStatement).

Java Server Pages

VI. Phương Thức Truy Xuất CSDL Bằng JSP :

2. Phương Thức Truy Xuất CSDL Bằng Jsp

2.2 Thực hiện các lệnh :

```
rs=stmt.executeQuery(sql);
```

hoặc :

```
rs=stmt.executeUpdate(sql);
```

Các câu lệnh sql có thể thực hiện thông qua :

- + phương thức `executeQuery()` -nếu kết quả là một đối tượng `ResultSet`

- + phương thức `executeUpdate()`-nếu kết quả là một số nguyên cho biết tổng số các dòng dữ liệu chịu tác dụng của câu lệnh vừa thực hiện.

Java Server Pages

VI. Phương Thức Truy Xuất CSDL Bằng JSP :

3. Đoạn lệnh dùng JSP truy cập dữ liệu trên Server

// Nạp trình điều khiển JDBC

- `String drivename="sun.jdbc.odbc.JdbcOdbcDriver";`
- `Class.forName(drivename).newInstance();`

// Thông tin kết nối JDBC-ODBC

- `String url="jdbc:odbc:Database_Name";`
- `String username="";`
- `String password="";`

Java Server Pages

VI. Phương Thức Truy Xuất CSDL Bằng JSP :

3. Đoạn lệnh dùng JSP truy cập dữ liệu trên server

- `Connection con=null;`
// Thực hiện kết nối
- `con=DriverManager.getConnection(url, username, password);`
- `Statement stmt=null;`
- `stmt=con.createStatement();`
// Thực thi câu lệnh truy vấn và nhận kết quả trả về
- `ResultSet rs=null;`
// Tạo câu lệnh truy vấn
- `String sql="select * from TABLE_NAME";`
- `rs=stmt.executeQuery(sql);`

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

1. Giới thiệu về Jrun

+ Để nghiên cứu về JSP trước tiên phải có trình chủ Web Server "hiểu Java". Trình chủ đó phải có khả năng thông dịch và gọi thực thi được các đơn thể Java theo đặc tả của Sun.

+ Mặc dù hiện nay có rất nhiều trình chủ Web dành cho Java như Java Web Server, TomCat, IPLanet, Jrun,...Nhưng do Jrun viết bằng Java nên bạn có thể sử dụng và cài đặt Jrun cả trên môi trường Linux lẫn Unix.

+ Trình Jrun cung cấp khả năng triệu gọi mã Java thông qua Servlet. Jrun cung cấp cho ta hai dịch vụ Web hoạt động trên hai cổng:

- Cổng 8100 là dịch vụ Web thông thường và cổng 8000 là dịch vụ Web phục vụ cho cấu hình trình chủ.

Cũng như những Web server khác, tìm hiểu Jrun bao gồm các bước sau:

- Cài đặt trình chủ Jrun.
- Cấu hình kết nối với các Web server khác.
- Quản trị Jrun.
- Tìm hiểu các file quan trọng của Jrun.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

2. Cài đặt Jrun

2.1. Cài đặt JDK

+ Trước khi cài đặt trình chủ Jrun Web Server ta phải cài đặt JDK. Mặc định JDK của Java sẽ được cài vào thư mục C:\Jdk. Khi Jrun yêu cầu chỉ định thư mục cài đặt JDK ta hãy chỉ đồng dẫn đến thư mục đó.

2.2. Cài đặt Jrun Web Server

- + Yêu cầu về phần cứng và phần mềm :
 - 32 MB Ram
 - 20 MB dung lượng đĩa cứng
 - Hệ điều hành : Windows hay NT/2000, Unix và Linux.
 - Sử dụng trình duyệt Netscape hay Internet Explorer
 - Môi trường máy ảo JRE (Java Runtime Environment)
 - JDK

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

2. Cài đặt Jrun

2.3. Cấu trúc thư mục của Jrun

Thư mục

Nội dung

/bin	Chứa các tập tin chương trình của Jrun
/connectors	Chứa các tập tin Web server kết nối
/docs	Chứa các trang tài liệu html hướng dẫn sử dụng
/lib	Chứa các tập tin thư viện .jar
/lib/ext	Chứa các tập tin .jar gồm servlet.jar & ejb.jar
/logs	Chứa các tập tin log của Jrun
/samples	Chứa các trang jsp và servlet mẫu
/servers	Chứa các ứng dụng Web của Jrun
/servers/lib	Chứa các tập tin .jar và các lớp đối tượng .class
/servlets	Chứa các tập tin .class của servlet truy cập
/uninst	Thông tin về tháo bỏ cài đặt của Jrun

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3. Cửa sổ điều khiển (Jrun Management Console)

Jrun Management Console là một ứng dụng Web nhằm giúp cho thao tác trên giao diện Web để cấu hình Jrun.

Để dùng JMC nh một người quản trị Web Server, ta có thể thực hiện theo các bước sau:

Nhập địa chỉ URL : <http://local:8000>

+ Chọn Start -> Program -> Jrun3.1 -> Jrun Admin Server

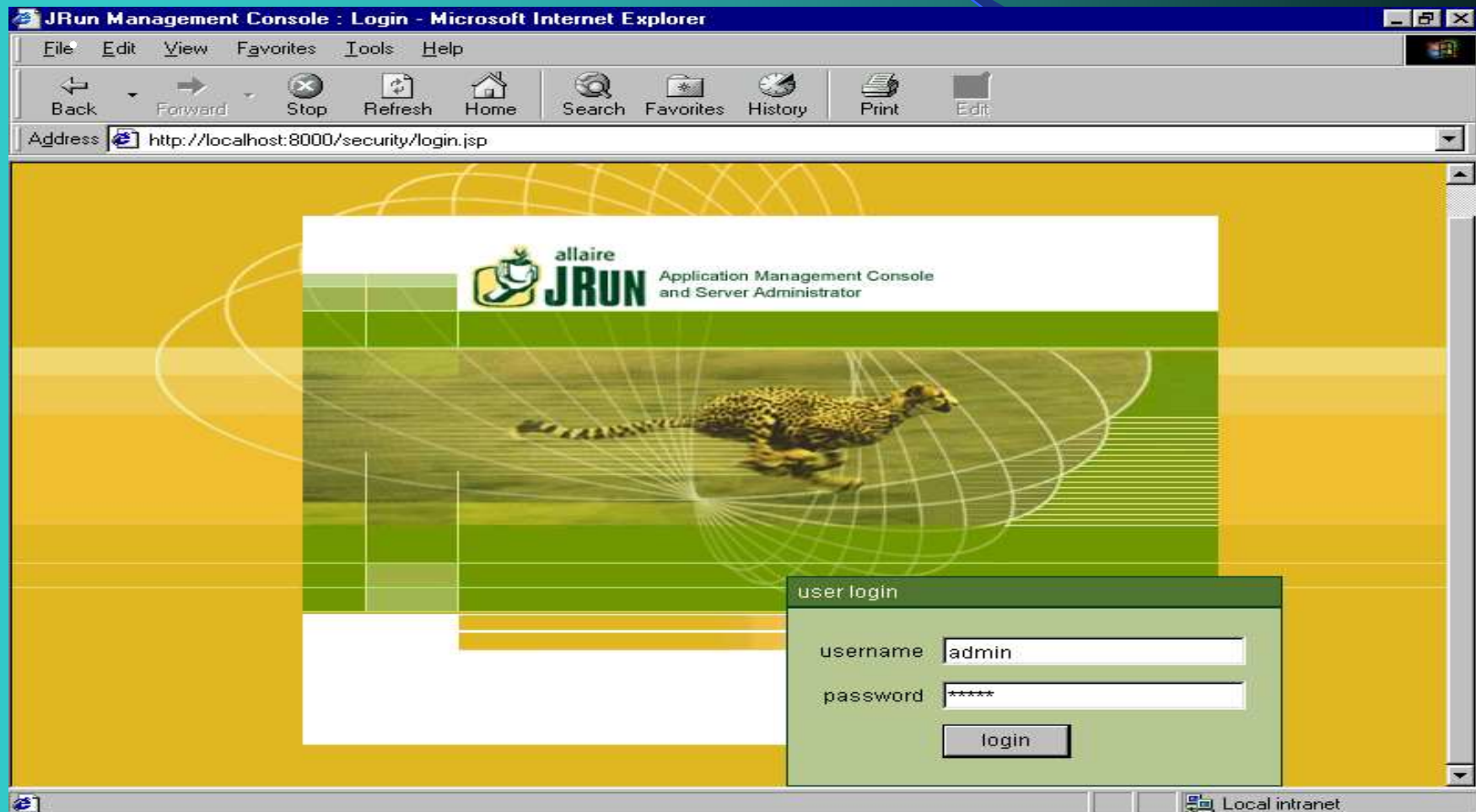
+ Chọn Start -> Program -> Jrun3.1 -> Jrun Management Console

Nếu thành công Jrun sẽ hiển thị màn hình đăng nhập (login) yêu cầu bạn cung cấp tài khoản người dùng (username) và mật khẩu (password) để đăng nhập vào phần quản trị (Administrator) của trình chủ.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3. Cửa sổ điều khiển (Jrun Management Console)

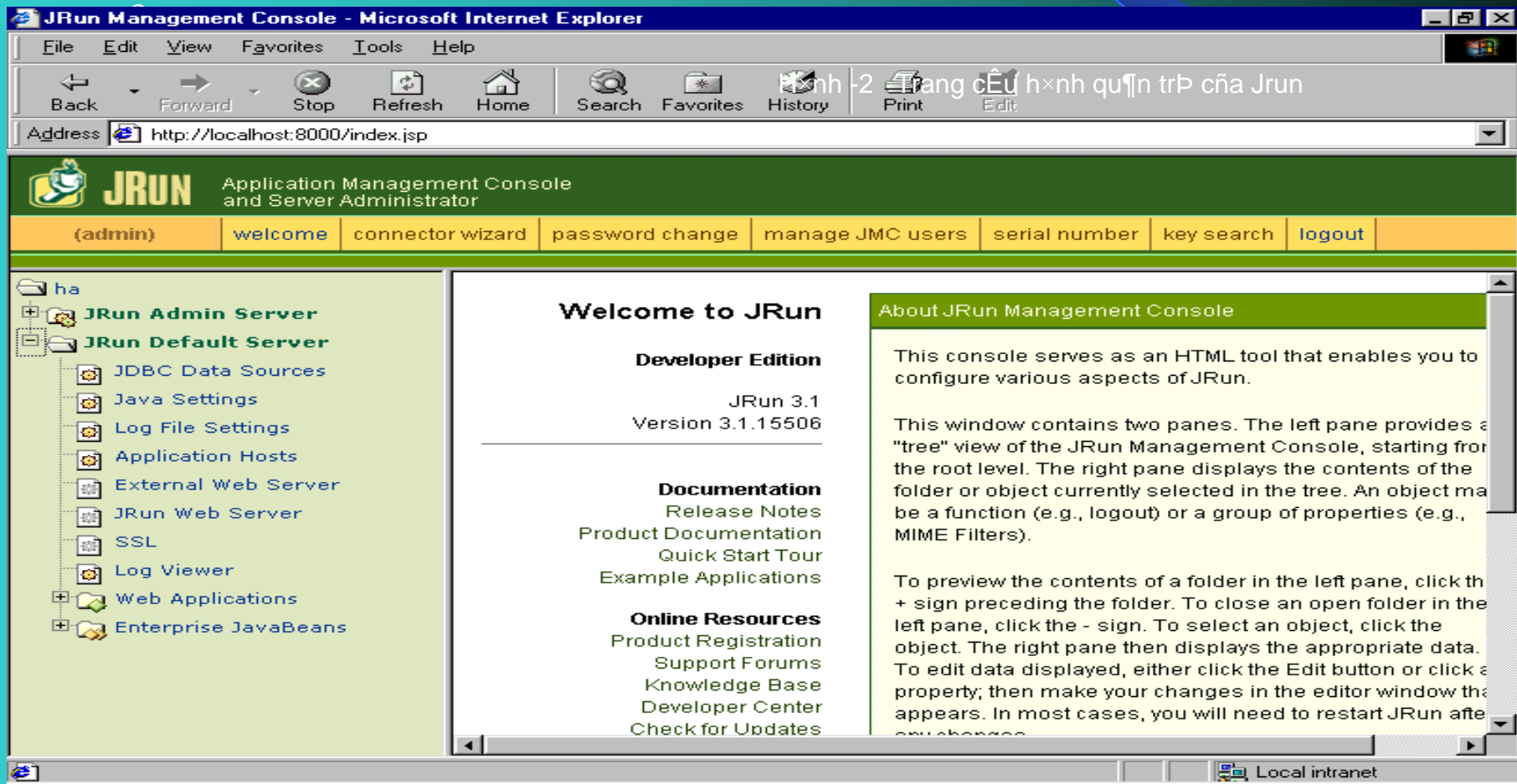


CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3. Cửa sổ điều khiển (Jrun Management Console)

Sau khi đăng nhập, các thông tin quản trị sẽ



CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3. Cửa sổ điều khiển (Jrun Management Console)

Sau khi đăng nhập, các thông tin quản trị sẽ



CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.1 Cấu hình quản trị Web Server

Thường bạn sử dụng trang quản trị để quản lý, khởi động, thay đổi tham số cho trình chủ Default Web Server phục vụ ứng dụng Web.

3.1.1 Cấu hình Jrun Default Web Server

Ta chỉ cần cấu hình theo các thông tin sau :

+ Web Server Address	*
+ Client IP filter	*
+ Web Server Port	8100
+ Idle thread timeout	300
+ Minimum thread count	1
+ Maximum active request	100
+ Maximum concurrent request	100
+ Jrun Web Server	on

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.1 Cấu hình quản trị Web Server

3.1.2 Tự tạo một Web Server

+ Ngoài Jrun Default Server, bạn có thể tự tạo một Web Server khác cho riêng mình

Lưu ý : trên một máy có thể có nhiều Web Server chạy cùng một lúc miễn sao chúng được cấu hình với số hiệu cổng khác nhau.

+ Để tạo Web Server khác, bạn chọn nhánh gốc bên khung trang trái. Nhấn nút *Add a server* bên khung trang phải và đặt tên cho Web Server cũng như các thông tin cấu hình khác mà Jrun yêu cầu

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.1 Cấu hình quản trị Web Server

3.1.2 Tự tạo một Web Server của riêng bạn

JRun Management Console - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Edit

Address <http://localhost:8000/index.jsp>

JRUN Application Management Console and Server Administrator

(admin) welcome connector wizard password change manage JMC users serial number key search logout

JRun > Managing JRun Servers

The following table displays information about your JRun servers. Each server is a separate Java Virtual Machine with different property files, ports, and applications. You may want to add additional JRun servers to keep the test and development environments separate, or if you develop applications for deployment on multiple Web Servers.

	Display Name	Root Directory	Status	Start/Restart
	JRun Admin Server	C:/Program Files/Allaire/JRun/servers/admin	running	<input type="button" value="restart"/>
	JRun Default Server	C:/Program Files/Allaire/JRun/servers/default	running	<input type="button" value="restart"/>

Done, but with errors on page. Local intranet

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.2 Cấu hình và tạo ứng dụng Web (Web Application)

3.2.1 Tạo mới ứng dụng Web

Thông thường khi tạo một ứng dụng Web, ta cần phải cấu hình ứng dụng này trong Jrun Web Server. Để tạo một ứng dụng trên Web Server của Jrun, ta chỉ cần theo các bước sau:

- + Trước khi cấu hình Jrun, ta cần chú ý địa chỉ IP của máy chủ nơi Jrun đang chạy. Nếu không kết nối mạng, ta có thể sử dụng localhost hay 127.0.0.1 là địa chỉ IP cục bộ của mỗi máy.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.2 Cấu hình và tạo ứng dụng Web (Web Application)

3.2.1 Tạo mới ứng dụng Web

- + Chọn Jrun Web Server (Admin Server hay Default Server)
- + Bấm vào link Create Application
- + Chọn Jrun Admin Server hay Jrun Default Server.
- + Nhập tên của ứng dụng. (Ví dụ : DATN)
- + Nhập vào địa chỉ gốc URL cho ứng dụng.
- + Chọn thư mục mà ta đặt các trang JSP của ứng dụng.

Ví dụ: C:/DATN/Jsp là thư mục chứa các trang JSP của ứng dụng.

- + Sau khi điền thông tin ta nhấn nút Create.
- + Tiếp đến ta khởi động lại Web Server tương ứng với tên mà ta dùng để tạo ứng dụng(Jrun Default Server).
- + Cuối cùng ta thử trên màn hình Browser của trình duyệt bằng cách gõ vào địa chỉ URL nh sau : `http://localhost:8100/DATN/Jsp/index.jsp`

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.2 Cấu hình và tạo ứng dụng Web (Web Application)

3.2.1 Tạo mới ứng dụng Web

JRun Management Console - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Edit

Address <http://localhost:8000/index.jsp>

JRUN Application Management Console and Server Administrator

(admin) welcome connector wizard password change manage JMC users serial number key search logout

ha

- JRun Admin Server
- JRun Default Server
 - JDBC Data Sources
 - Java Settings
 - Log File Settings
 - Application Hosts
 - External Web Server
 - JRun Web Server
 - SSL
 - Log Viewer
 - Web Applications
 - Default User Application
 - JRun Demo
 - Allaire Invoice Sample A
 - Enterprise JavaBeans

Create a Web Application

Return to Main Page
Edit an Application
Create an Application
Deploy an Application
Remove an Application

Please Note:
You must restart the JRun server before the new Web application will become available.

Web Application Information

JRun Server Name: JRun Admin Server
JRun Default Server

Application Name: DATN

Application Host: All Hosts

Application URL: /DATN

Application Root Dir: C:\DATN

create

Application Root Directory

Done Local intranet

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.2 Cấu hình và tạo ứng dụng Web (Web Application)

3.2.2 Xóa một ứng dụng Web

Do nhu cầu sử dụng đôi khi bạn tạo ra một ứng dụng web, nhưng có thể sau một thời gian sử dụng bạn không cần dùng đến nó nữa, ta có thể xóa nó đi. Để xóa một ứng dụng web:

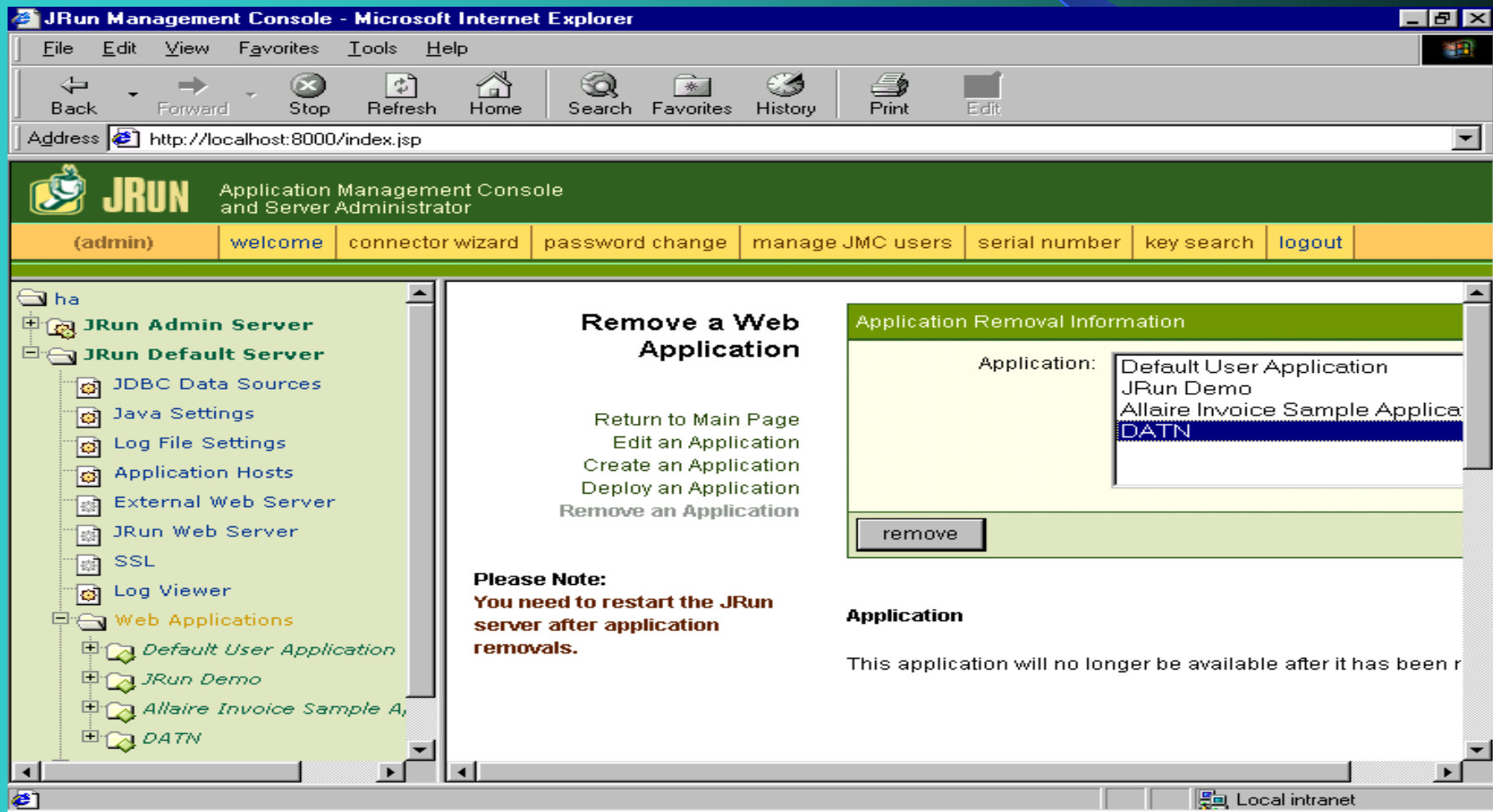
- + Chọn Web Application từ màn hình quản trị của Jrun
- + Tiếp đến chọn Web Application. Jrun sẽ hiển thị danh sách các ứng dụng ta đã đăng ký hoặc tạo ra trước đó.
- + Để loại bỏ ứng dụng Web, ta chọn mục Remove Application.
- + Khi Jrun hiển thị danh sách các đề mục cần xóa, bạn chọn tên của ứng dụng rồi chọn Remove.
- + Để Jrun nhận ra ứng dụng đã bị xóa hoàn toàn, cần khởi động lại trình chủ Jrun Web Server.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.2 Cấu hình và tạo ứng dụng Web (Web Application)

3.2.2 Xóa một ứng dụng Web



CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.2 Cấu hình và tạo ứng dụng Web (Web Application)

3.2.3 Cập nhật một ứng dụng Web

Sau khi tạo ra một ứng dụng Web nếu cần bổ sung hoặc thay đổi, ta có thể cập nhật lại nội dung ứng dụng theo các bước sau:

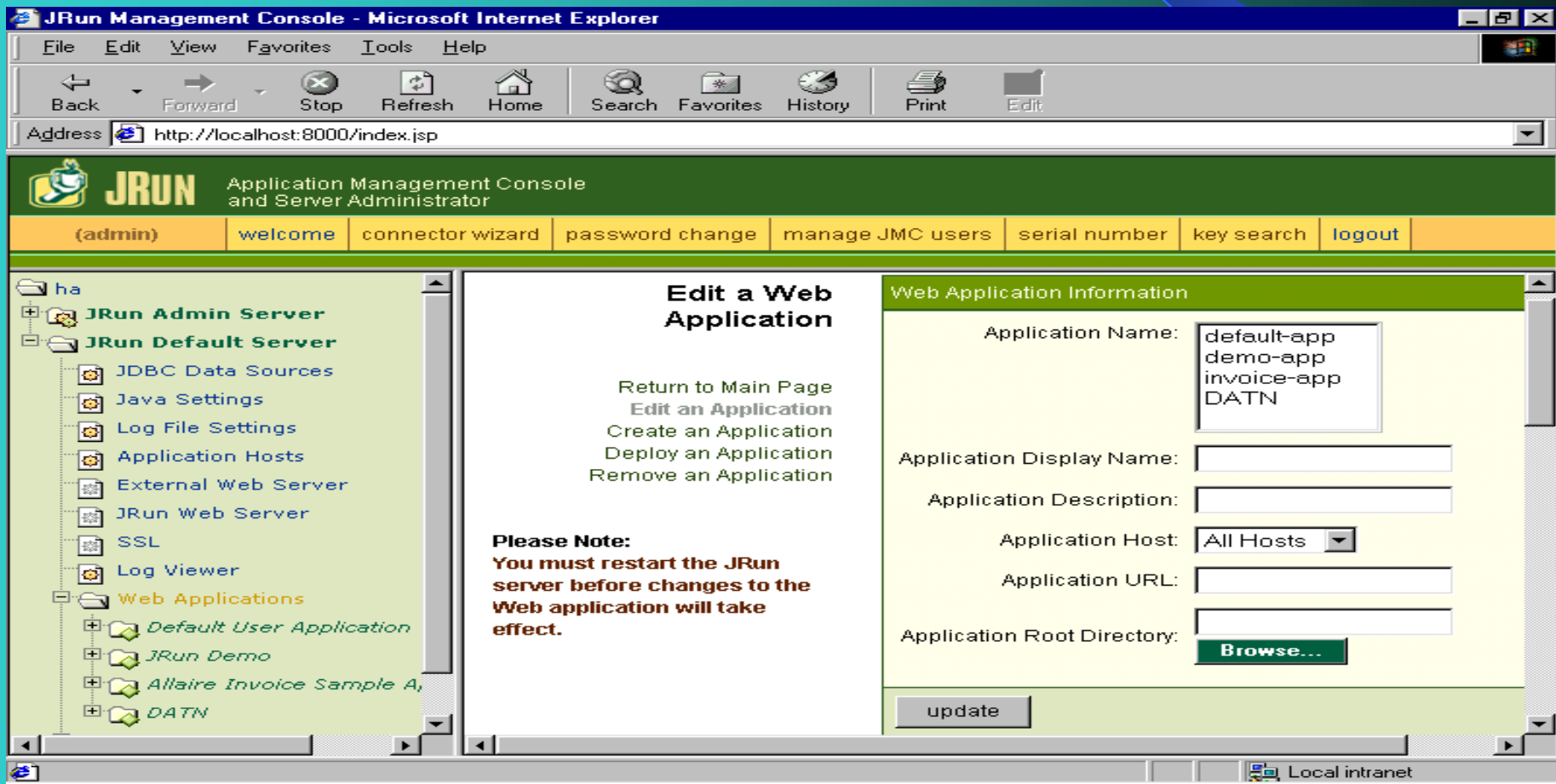
- + Chọn Web Application từ màn hình quản trị của Jrun Web Server.
- + Jrun sẽ hiển thị danh sách các ứng dụng đã đăng ký trước đó.
- + Chọn mục Edit Application.
- + Khi Jrun hiển thị danh sách ứng dụng, ta cần chọn lấy ứng dụng muốn cập nhật.
- + Thông tin ứng dụng sẽ hiện ra để ta bổ sung hoặc sửa đổi.
- + Cần phải khởi động lại trình chủ Jrun Web Server.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.2 Cấu hình và tạo ứng dụng Web (Web Application)

3.2.3 Cập nhật một ứng dụng Web



CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

4. Tạo và quản lý user trong Jrun Web Server

Đối với Jrun Web Server ta có thể tạo ra một số thành viên giới hạn cho phép truy cập trình chủ Web Server. Ta có thể thực hiện theo các bước sau :

- + Chọn mục manage JMC users từ trang quản trị của Jrun.
- + Nhập tên user và password của thành viên đó. Mỗi lần ta chỉ cần đăng ký cho hai thành viên.
- + Ta chọn nút Update JMC Users để cập nhật user vào cơ sở dữ liệu của Jrun.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

4. Tạo và quản lý user trong Jrun Web Server

JRun Management Console - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Edit

Address <http://localhost:8000/index.jsp>

JRUN Application Management Console and Server Administrator

(admin) welcome connector wizard password change **manage JMC users** serial number key search logout

ha

- JRun Admin Server
- JRun Default Server
 - JDBC Data Sources
 - Java Settings
 - Log File Settings
 - Application Hosts
 - External Web Server
 - JRun Web Server
 - SSL
 - Log Viewer
 - Web Applications
 - Default User Application
 - JRun Demo
 - Allaire Invoice Sample A
 - DATN

Manage JMC Users

Edit the password and/or the permission level for the appropriate user.

Please Note:
Passwords - Leave the password as ***** if no changes to the password are needed. Otherwise, the value entered will be saved.

** is not allowed in the password. Leading and trailing spaces will be trimmed.

User Names - Leading and trailing spaces will be trimmed.

Currently Available JMC Users

	User Name	Password	Permissions
<input type="checkbox"/>			All servers JRun Admin Server JRun Default Server
<input type="checkbox"/>			All servers JRun Admin Server JRun Default Server

Update JMC Users

Local intranet

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

5. Kiểm tra lỗi thông qua Log Files

Thông thường bất kỳ lỗi nào xảy ra đối với Jrun đều được lưu vào file logs. Những file logs này nằm trong thư mục [Jrun_Home]\logs.

Mỗi file log lưu lại những sự kiện và lỗi phát sinh khác nhau. Thông thường có 3 file log sau :

- + default-err.log : file này lưu lại những lỗi của Jrun xảy ra trong quá trình dịch

- + default-even.log : lưu lại những lỗi trong quá trình thực hiện truy xuất tài nguyên của Java.

- + default-out.log : lưu lại những lỗi liên quan đến môi trường bên ngoài. Ví dụ : Khi bạn gửi thông tin đến một địa chỉ không có thì lỗi sẽ được ghi vào default-out.log.

- Tóm lại : Để theo dõi quá trình làm việc của Jrun, ta có thể dựa vào nội dung của file log. File log là nơi giúp ta phát hiện và khắc phục những lỗi ngoại lệ của hệ thống

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Poll

1. Giới thiệu :

- + Nhu cầu về kết nối cơ sở dữ liệu trung tâm để truy vấn dữ liệu trong các ứng dụng của Web ngày càng tăng.
- + Các hệ cơ sở dữ liệu phải tiêu tốn rất nhiều tài nguyên để duy trì liên tục kết nối trong quá trình truy vấn.
- + Ta chỉ nên mở kết nối truy vấn đến cơ sở dữ liệu chỉ khi nào thật sự cần thiết. Một khi đã hoàn tất truy xuất, bạn phải trả lại kết nối cho người dùng khác truy cập. Mặc dù vậy, mỗi một kết nối đến cơ sở dữ liệu đều phải đòi hỏi có thời gian thiết lập, phân bổ tài nguyên,...
- + Các trình chủ Web server đã ra cơ chế POLL cho phép các yêu cầu về kết nối đến cơ sở dữ liệu chỉ đọc mở một lần và dùng chung bởi các trang JSP với nhau.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Poll

1. Giới thiệu :

- + Vì vậy, các yêu cầu về truy vấn dữ liệu thường chỉ diễn ra trong thời gian ngắn

- + Những trang JSP trước khi truy xuất dữ liệu sẽ yêu cầu hệ thống cung cấp một kết nối, hệ thống sẽ kiểm tra xem kết nối đã được thiết lập hay chưa. Nếu chưa có hệ thống sẽ tạo một kết nối cung cấp cho trang JSP sử dụng.

- + Trang JSP sau khi sử dụng xong không đóng kết nối mà trả lại kết nối cho hệ thống. Hệ thống sẽ lưu kết nối trở lại ngăn xếp hoặc hàng đợi.

- + Trong lời triệu gọi trang JSP ở phiên làm việc khác, nếu cần đến truy xuất cơ sở dữ liệu, hệ thống sẽ cung cấp kết nối đã có trước đó mà không cần sử dụng tài nguyên của hệ thống tạo lại kết nối.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Pool

1. Giới thiệu :

Ta sẽ cài đặt một hệ thống như vậy cho cơ chế Pool (cơ chế chia sẻ kết nối). Trước hết hệ thống của ta cần hỗ trợ hai phương thức chính cho yêu cầu kết nối đó là cung cấp kết nối (`getConnection()`) và nhận lại kết nối khi trang JSP đã sử dụng (`releaseConnection()`). Khai báo giao tiếp interface mang tên `IconnectionPool` sẽ cung cấp thông tin này.

// Interface định nghĩa các phương thức cơ bản cung cấp và chia sẻ kết nối

```
public interface IconnectionPool
```

```
    { public Connection  getConnection() throws SQLException;
      public void  releaseConnection(Connectionconn)  throws
      SQLException;
    }
```

Ta hình dung Pool tương tự cài đặt của ngăn xếp hoặc hàng đợi dùng để lưu trữ các phần tử cho mục đích sử dụng lại. Pool không đòi hỏi thứ tự lấy ra của các phần tử. Các phần tử lưu trong Pool thường là đối tượng `Connection` có thể lấy ra ngẫu nhiên hoặc tuần tự.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Pool

2. Cơ Chế Pool Hỗ Trợ Bởi Web Server

- + Hiện nay hầu hết các trình chủ Web server đều cung cấp sẵn cơ chế Pool gọi là Data Source.
- + Việc quản lý và điều hành Pool sẽ do trình chủ Web server đảm nhiệm và bạn chỉ cần cung cấp cho trình chủ những thông tin tạo Pool là đủ.
- + Chúng ta sẽ cấu hình và sử dụng cơ chế Pool hỗ trợ bởi trình chủ Jrun. Trước hết ta mở trang <http://localhost:8100/> để vào trang quản trị của Jrun. Tiếp đến ta chọn Default Jrun Server\JDBC Data Source. Nhấn nút Add trang cung cấp thông tin để cấu hình cho DataSource hiển thị nh sau :

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Pool

2. Cơ Chế Pool Hỗ Trợ Bởi Web Server

The screenshot shows the JRun Management Console in a Microsoft Internet Explorer browser window. The address bar displays `http://localhost:8000/index.jsp?`. The console interface includes a navigation menu on the left with items like **JRun Admin Server**, **JRun Default Server**, **JDBC Data Sources**, **Java Settings**, **Log File Settings**, **Application Hosts**, **External Web Server**, **JRun Web Server**, **SSL**, **Log Viewer**, **Web Applications**, and **Enterprise JavaBeans**. The main content area is titled **JRun Default Server > JDBC Data Sources** and contains the following text:

JDBC data sources currently available to Web applications.
Ensure that the JDBC driver is in the classpath of the **default** server. If not, you can use the Java Set
In addition, testing the data source setting requires the JDBC driver to be in the classpath of the adm

	Name	Driver	URL
	TaglibDataSource	com.pointbase.jdbc.jdbcUniversalDriver	jdbc:pointbase://embedded/sample
	Sample9bDataSource	com.pointbase.jdbc.jdbcUniversalDriver	jdbc:pointbase://embedded/sample
	KDPHUYENC	allaire.jrun.jdbc.JRunDriver	jdbc:jrun:sqlserver://127.0.0.1:1433;date
	forum	allaire.jrun.jdbc.JRunDriver	jdbc:jrun:sqlserver://127.0.0.1:1433;date
	Baomat	sun.jdbc.odbc.JdbcOdbcDriver	jdbc:odbc:Baomat

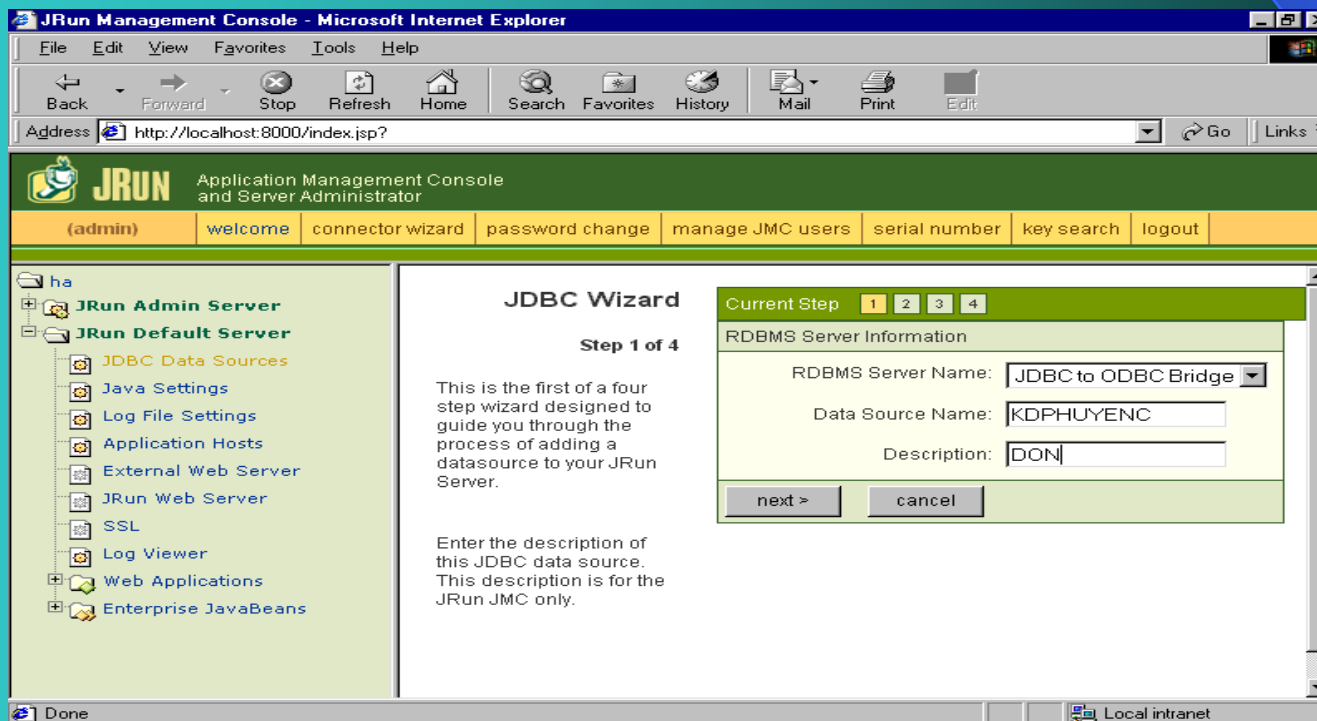
Below the table is an **add** button. The status bar at the bottom indicates **Done** and **Local intranet**.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Pool

2. Cơ Chế Pool Hỗ Trợ Bởi Web Server

+ Jrun cho phép sử dụng nhiều loại kết nối JDBC. Trong trường hợp này ta chọn mục danh sách JDBC to ODBC Bridge. Mục Data Source Name ta gõ vào tên *CSDL nguồn*. Đây là sẽ chuỗi đọc dùng để tham chiếu đến Pool sau này.



CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Pool

2. Cơ Chế Pool Hỗ Trợ Bởi Web Server

Tiếp tục ta nhập vào các thông tin liên quan đến chuỗi kết nối cơ sở dữ liệu.

The screenshot displays the JRun Management Console in a Microsoft Internet Explorer browser window. The address bar shows `http://localhost:8000/index.jsp?`. The console interface includes a top navigation bar with links like (admin), welcome, connector wizard, password change, manage JMC users, serial number, key search, and logout. A left-hand tree view shows the system hierarchy, including JRun Admin Server, JRun Default Server, and various settings like JDBC Data Sources, Java Settings, and Application Hosts. The main content area is titled 'JDBC Wizard' and shows 'Step 2 of 4'. The instructions for this step are: 'Complete the fields and click Next to proceed to Step 3.' and 'Enter the database name that corresponds to the data source. For example, in ODBC it would be the data source name. In Oracle it would be the SID.' To the right, the 'JDBC to ODBC Bridge Server Connection Settings' form is shown with the following fields: 'Server User' (sa), 'Server Password' (empty), and 'Database Name' (KDPHUYENC). Navigation buttons '< back', 'next >', and 'cancel' are at the bottom of the form. The status bar at the bottom indicates 'Done' and 'Local intranet'.

JRun Management Console - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Edit

Address `http://localhost:8000/index.jsp?` Go Links >>

JRUN Application Management Console and Server Administrator

(admin) welcome connector wizard password change manage JMC users serial number key search logout

ha

- JRun Admin Server
- JRun Default Server
 - JDBC Data Sources
 - Java Settings
 - Log File Settings
 - Application Hosts
 - External Web Server
 - JRun Web Server
 - SSL
 - Log Viewer
 - Web Applications
 - Enterprise JavaBeans

JDBC Wizard

Step 2 of 4

Complete the fields and click Next to proceed to Step 3.

Enter the database name that corresponds to the data source. For example, in ODBC it would be the data source name. In Oracle it would be the SID.

Current Step 1 2 3 4

JDBC to ODBC Bridge Server Connection Settings:

Server User:

Server Password:

Database Name:

< back next > cancel

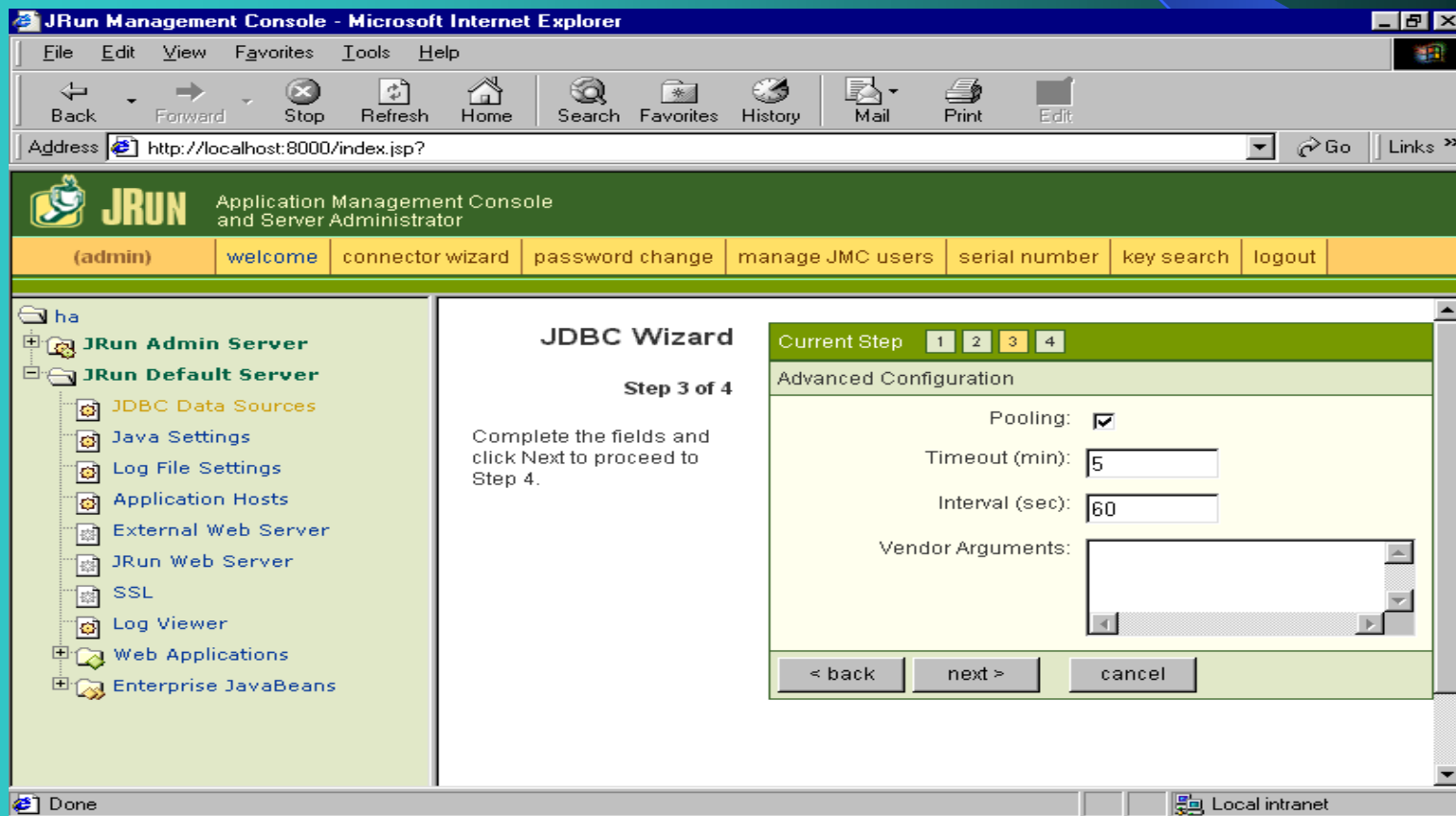
Done Local intranet

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Pool

2. Cơ Chế Pool Hỗ Trợ Bởi Web Server

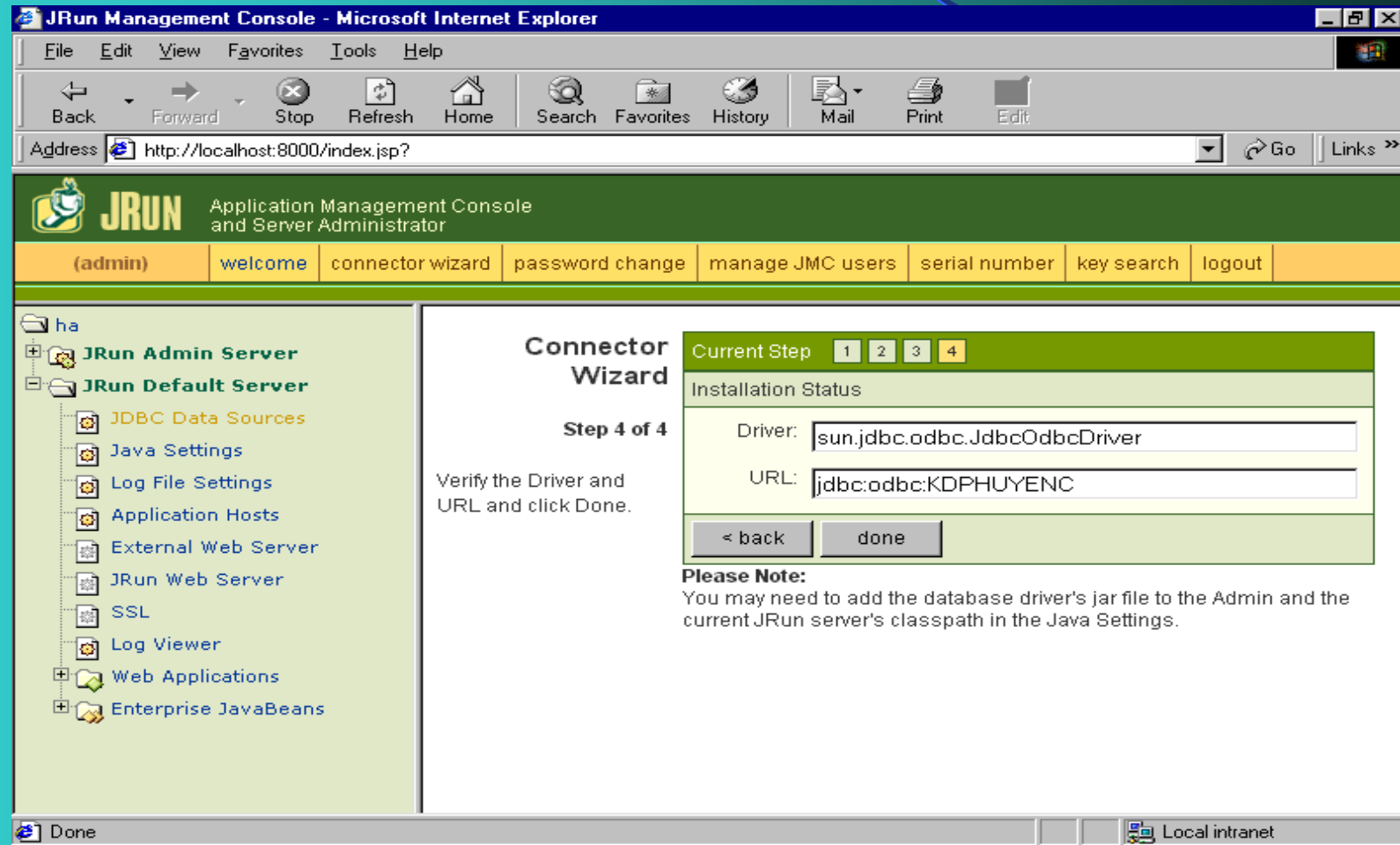
Tiếp tục nhấn Next Jrun sẽ hoàn tất những thao tác còn lại.



CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Pool

2. Cơ Chế Pool Hỗ Trợ Bởi Web Server



Ngôn Ngữ Truy Vấn SQL

Ngôn ngữ Truy vấn dữ liệu (SQL)

I. Giới thiệu

Ngôn ngữ SQL (Structured Query Language) được hãng máy tính IBM đưa ra vào năm 1970.

Là một ngôn ngữ có cấu trúc, có nghĩa là bằng tập hợp các dòng lệnh sẽ chỉ cho hệ quản trị CSDL biết là người sử dụng sẽ làm gì chứ không chỉ ra làm như thế nào.

Ngôn ngữ SQL cung cấp một công cụ truy vấn (query) nhằm cho phép người sử dụng đặt câu hỏi với dữ liệu đang chứa trong các bảng, dữ liệu được rút ra từ những truy vấn có thể có nguồn từ một hay nhiều bảng khác nhau.

Sau đó tham khảo kết quả này, xem xét, phân tích nó kỹ lưỡng để có những xử lý thông tin thiết thực hơn.

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

Trong mục này trình bày một số lệnh cơ bản dùng để thao tác trên cơ sở dữ liệu như: chọn dữ liệu, thêm vào dữ liệu mới, xoá dữ liệu cũ, sửa đổi dữ liệu.

1. Lệnh Select : Lệnh này dùng để lấy dữ liệu trong một bảng thoả mãn điều kiện nào đó.

Cú pháp:

```
Select [ *fields ] From tablename  
[Where expression]  
[Group by fields1]  
[Having expression1]  
[Order by {colname } [ASC | DESC]]
```

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

- * : tất cả các trường đều được chọn.
- *colname*: tên cột.
- *tablename*: là tên của một Table hoặc nhiều Table mà các Record được chọn.
- *expression*: là biểu thức điều kiện được đặt ra mà các bản ghi được chọn từ *tablename* phải thoả mãn biểu thức này.
- *expression1*: một biểu thức điều kiện đi sau mệnh đề Having. Mệnh đề Having thông thường đi kèm với mệnh đề Group by, *expression1* là biểu thức không tác động vào toàn bảng được chỉ ra bởi mệnh đề From mà chỉ tác động lần lượt từng nhóm các bản ghi đã được chỉ ra bởi mệnh đề Group by.
- Order by: dùng để sắp các bản ghi tăng hay giảm

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

1. Lệnh Select

- Ví dụ: Giả sử có một Table có tên NHANSU như sau:

Maso	Hoten	Ngaysinh	Nghenghiep
1	Lê Văn Tuấn	10/8/1990	Học sinh
2	Nguyễn Văn Hà	7/8/1970	Kỹ s
3	Nguyễn Thị Hoa	30/07/1980	Sinh viên

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

1. Lệnh Select

- Nếu sử dụng lệnh: **Select Hoten, Ngaysinh From nhansu**
- Sẽ chọn tất cả các bản ghi nhưng chỉ giới hạn bởi hai trường Hoten và Ngaysinh.
- Cụ thể gồm các bản ghi sau:

Lê Văn Tuấn	10/8/1990
Nguyễn văn Hà	7/8/1970
Nguyễn Thị Hoa	30/07/1980

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

2. Lệnh Insert Into : Lệnh này dùng để chèn thêm một Record dữ liệu mới vào cho bảng.

- Cú pháp:

**Insert Into *table*[(*field1*[,*field2*[,...]])]
Values(*value1*[, *value2*[,...])**

- Trong đó:

Table: tên của Table cần bổ sung bản ghi

field1, field2: tên của các trường cần bổ sung dữ liệu

value1, value2: giá trị cần bổ sung ứng với các trường *field1, field2...*

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

2. Lệnh Insert Into

- Ví dụ:
- Giả sử ta có được Table Nhansu như trên, bây giờ bổ sung vào bản ghi mới:

```
Insert Into Nhansu(Hoten, Ngaysinh, Nghenghiep)  
Values("Nguyễn văn An","10/7/1995","Học sinh")
```
- Sau khi thực hiện lệnh này bảng Nhansu bây giờ như sau:

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

2. Lệnh Insert Into

Maso	Hoten	Ngaysinh	Nghenghiep
1	Lê Văn Tuấn	10/08/1990	Học sinh
2	Nguyễn văn Hà	07/08/1970	Kỹ s
3	Nguyễn Thị Hoa	30/07/1980	Sinh viên
4	Nguyễn văn An	10/07/1995	Học sinh

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

3. Lệnh Update : Lệnh này dùng để thay đổi dữ liệu cho các trường trong một bảng thỏa mãn điều kiện nào đó

- Cú pháp:

Update table Set Field1=value1[,...] Where expression

- Trong đó:

table: là tên của Table chứa bản ghi cần sửa đổi.

Field1: trường cần thay đổi dữ liệu

Value1: giá trị dữ liệu mới

expression : là biểu thức, mà các bản ghi được cập nhật phải thỏa mãn biểu thức này

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

3. Lệnh Update

- Ví dụ:
- Sửa đổi trường Nghenghiep có giá trị mới thành "CNVC" cho tất cả các record thoả mãn điều kiện là trường Ngaysinh="07/08/1970":

Update Nhansu Set Nghenghiep="CNVC" Where Ngaysinh="07/08/1970"

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

3. Lệnh Update

Table Nhansu bây giờ như sau:

Maso	Hoten	Ngaysinh	Nghenghiep
1	Lê Văn Tuấn	10/08/1990	Học sinh
2	Nguyễn văn Hà	07/08/1970	CNVC
3	Nguyễn Thị Hoa	30/07/1980	Sinh viên
4	Nguyễn văn An	10/07/1995	Học sinh

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

4. Lệnh Delete

Lệnh này dùng để xoá một hoặc nhiều record trong một bảng thoả mãn điều kiện nào đó.

- Cú pháp:

Delete * From *table* Where *expression*

- Trong đó:

table: là tên Table chứa các bản ghi cần xoá.

*: tất cả các trường của bản ghi đều bị xoá.

expression : là tất cả các bản ghi được xoá phải thoả mãn biểu thức này.

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

4. Lệnh Delete

Ví dụ:

- Xoá tất cả các bản ghi trong Table Nhansu có trường Nghenghiep="Học sinh".
 - Delete * From nhansu Where Nghenghiep="Học sinh"
- Bây giờ table Nhansu trở thành:

Maso	Hoten	Ngaysinh	Nghenghiep
2	Nguyễn văn Hà	07/08/1970	CNVC
3	Nguyễn Thị Hoa	30/07/1980	Sinh viên

Ngôn ngữ Truy vấn dữ liệu (SQL)

Chú ý: Trong mệnh đề WHERE, các phép toán được sử dụng là:

Phép toán	Mô tả
=	So sánh bằng
<>	So sánh khác
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng
BETWEEN	Nằm giữa một khoảng
LIKE	So sánh mẫu chuỗi

Lưu ý: Trong một số phiên bản của SQL, phép toán <> có thể được viết dưới dạng !=

Ngôn ngữ Truy vấn dữ liệu (SQL)

* Sử dụng dấu nháy

Lưu ý rằng ở ví dụ trên ta đã sử dụng hai dấu nháy đơn (') bao quanh giá trị điều kiện 'Sandnes'.

SQL sử dụng dấu nháy đơn bao quanh các giá trị ở dạng chuỗi văn bản (text).

Nhiều hệ CSDL còn cho phép sử dụng dấu nháy kép ("). Các giá trị ở dạng số không dùng dấu nháy để bao quanh.

+ Với dữ liệu dạng chuỗi văn bản:

Câu lệnh đúng:

```
SELECT * FROM Persons WHERE FirstName = 'Tove'
```

Câu lệnh sai:

```
SELECT * FROM Persons WHERE FirstName = Tove
```

+ Với dữ liệu dạng số:

Câu lệnh đúng:

```
SELECT * FROM Persons WHERE Year > 1965
```

Câu lệnh sai:

```
SELECT * FROM Persons WHERE Year > '1965'
```

* **Phép toán điều kiện LIKE**

Phép toán LIKE được dùng để tìm kiếm một chuỗi mẫu văn bản trên một cột.

Cú pháp :

Cú pháp của phép toán LIKE như sau:

```
SELECT tên_cột FROM tên_bảng  
WHERE tên_cột LIKE mẫu
```

Một ký hiệu % có thể được sử dụng để định nghĩa các ký tự đại diện. % có thể được đặt trước và/hoặc sau mẫu.

+ Sử dụng LIKE

Câu lệnh SQL sau sẽ trả về danh sách những người có tên bắt đầu bằng chữ O:

```
SELECT * FROM SINHVIEN  
WHERE ten LIKE 'O%'
```

Câu lệnh SQL sau sẽ trả về danh sách những người có tên kết thúc bằng chữ a:

```
SELECT * FROM sinhvien  
WHERE ten LIKE '%a'
```

Câu lệnh SQL sau sẽ trả về danh sách những người có tên kết chứa chuỗi la:

```
SELECT * FROM sinhvien  
WHERE ten LIKE '%la%'
```

Hết !!!