

Cơ bản về JSP

Học JSP cơ bản đến nâng cao

Môi trường lập trình là nơi bạn phát triển các chương trình JSP của bạn, test chúng và sau đó là run.

Chương này hướng dẫn bạn chi tiết cách cài đặt môi trường JSP theo các bước sau:

Cài đặt Java Development Kit

Bước này bao gồm tải Java Software Development Kit (SDK) và thiết lập biến môi trường PATH.

Bạn có thể tải SDK từ: [Java SE Downloads](#).

Sau khi tải về, bạn theo các chỉ dẫn đã cho để cài đặt và định cấu hình cho cài đặt. Cuối cùng, thiết lập các biến môi trường PATH và JAVA_HOME để tham chiếu tới thư mục mà chứa java và javac, soạn *java_install_dir/bin* và *java_install_dir*.

Nếu bạn đang chạy trên Windows và cài đặt SDK trong C:\jdk1.5.0_20, bạn đặt dòng sau trong C:\autoexec.bat file:

```
set PATH=C:\jdk1.5.0_20\bin;%PATH%
```

```
set JAVA_HOME=C:\jdk1.5.0_20
```

Trên Windows NT/2000/XP, bạn cũng có thể nhấp chuột phải trên **My Computer**, chọn Properties, và chọn Advanced, sau đó là Environment Variables. Cuối cùng, bạn update giá trị PATH và nhấn nút OK.

Trên Unix (Solaris, Linux, ...), nếu cài đặt SDK trong /usr/local/jdk1.5.0_20 và bạn sử dụng C Shell, bạn đặt dòng sau vào trong .cshrc file của bạn:

```
setenv PATH /usr/local/jdk1.5.0_20/bin:$PATH
```

```
setenv JAVA_HOME /usr/local/jdk1.5.0_20
```

Nếu bạn sử dụng một Development Environment (IDE) ví dụ như Borland JBuilder, Eclipse, IntelliJ IDEA, hoặc Sun ONE Studio, bạn biên dịch và chạy một chương trình đơn giản để xác nhận rằng IDE biết nơi bạn cài đặt Java.

Cài đặt Web Server: Tomcat

Một số Web Server, hỗ trợ JavaServer Pages và Servlets, là có sẵn trên thị trường. Một số chúng là miễn phí và Tomcat là một trong số đó.

Apache Tomcat là một sự triển khai phần mềm mã nguồn mở của JavaServer Pages (JSP) và các công nghệ Servlet, và có thể hoạt động như là một Standalone Server để kiểm tra JSP và Servlet. Apache Tomcat có thể được tích hợp với Apache Web Server. Sau đây là các bước để cài đặt Tomcat:

- Tải phiên bản mới nhất của Tomcat từ: <http://tomcat.apache.org/>.
- Sau khi tải, mở và đặt trong vị trí thích hợp. Ví dụ, trong C:\apache-tomcat-5.5.29 trên windows, hoặc /usr/local/apache-tomcat-5.5.29 trên Linux/Unix và tạo biến môi trường CATALINA_HOME trỏ tới các vị trí này.

Tomcat có thể được bắt đầu bởi thực thi các lệnh sau trên thiết bị Windows:

```
%CATALINA_HOME%\bin\startup.bat
```

hoặc

```
C:\apache-tomcat-5.5.29\bin\startup.bat
```

Tomcat có thể được bắt đầu bởi thực thi các lệnh sau trên thiết bị Unix (Solaris, Linux, ...):

```
$CATALINA_HOME/bin/startup.sh
```

hoặc

```
/usr/local/apache-tomcat-5.5.29/bin/startup.sh
```

Sau khi khởi động thành công, các ứng dụng web mặc định được include với Tomcat sẽ là có sẵn bởi việc chạy **http://localhost:8080/**. Nếu mọi thứ là ổn, thì nó hiển thị kết quả sau:



Các thông tin bổ sung về định cấu hình và chạy Tomcat có thể được tìm thấy trong Document được bao tại đó, cũng như trên: <http://tomcat.apache.org>

Tomcat có thể bị dừng bởi thực thi các lệnh sau trên thiết bị Windows:

```
%CATALINA_HOME%\bin\shutdown
```

hoặc

```
C:\apache-tomcat-5.5.29\bin\shutdown
```

Tomcat có thể bị dừng bởi thực thi các lệnh sau trên thiết bị Unix (Solaris, Linux, ...):

```
$CATALINA_HOME/bin/shutdown.sh
```

hoặc

```
/usr/local/apache-tomcat-5.5.29/bin/shutdown.sh
```

Thiết lập CLASSPATH

Khi Servlet không là một phần của Java Platform, Standard Edition, bạn phải thông báo các lớp Servlet tới bộ biên dịch (compiler).

Nếu bạn đang chạy trên Windows, bạn cần đặt các lệnh sau trong C:\autoexec.bat file:

```
set CATALINA=C:\apache-tomcat-5.5.29
```

```
set CLASSPATH=%CATALINA%\common\lib\jsp-api.jar;%CLASSPATH%
```

Trên Windows NT/2000/XP, bạn cũng có thể nhấp chuột phải vào **My Computer**, chọn Properties, và chọn Advanced, sau đó là Environment Variables. Cuối cùng, bạn update giá trị CLASSPATH và nhấn nút OK.

Trên Unix (Solaris, Linux, ...), nếu bạn đang sử dụng C Shell, bạn đặt các dòng sau vào trong .cshrc file.

```
setenv CATALINA=/usr/local/apache-tomcat-5.5.29
```

```
setenv CLASSPATH $CATALINA/common/lib/jsp-api.jar:$CLASSPATH
```

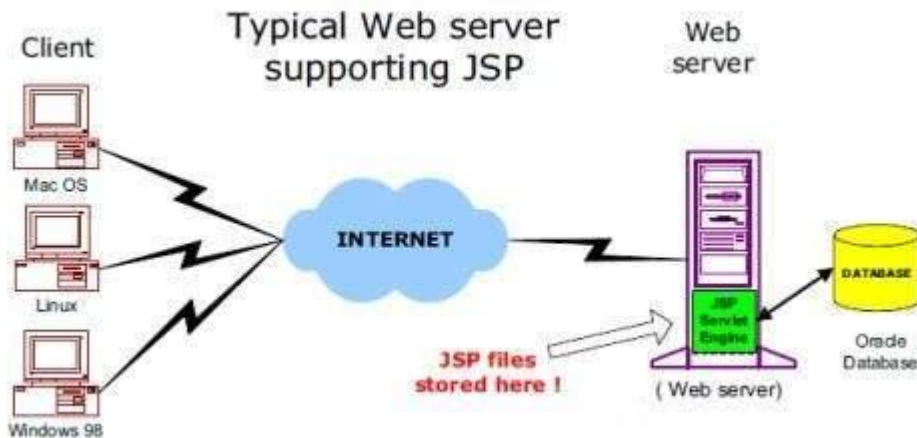
Ghi chú: Giả sử thư mục lập trình của bạn là C:\JSPDev (với Windows) hoặc /usr/JSPDev (với Unix) thì bạn cũng cần thêm các thư mục này vào trong CLASSPATH theo cách tương tự như bạn đã thêm ở trên.

Cấu trúc JSP

Web Server cần một JSP engine (ví dụ: Container) để xử lý các JSP page. JSP container có nhiệm vụ chặn các Request cho các JSP page. Bài hướng dẫn này sử dụng Apache, mà có JSP container có sẵn, để hỗ trợ phát triển các JSP page.

Một JSP container làm việc với Web Server để cung cấp môi trường runtime và các dịch vụ khác mà một JSP cần đến. Nó biết cách để hiểu các phân tử đặc biệt mà là một phần của JSP.

Sơ đồ sau minh họa vị trí của JSP container và JSP file trong một Web Application.



Tiến trình JSP

Dưới đây là các bước giải thích cách Web Server tạo các Webpage bởi sử dụng JSP:

- Với một trang thông thường, trình duyệt gửi một HTTP request tới Web server.
- Web server nhận ra rằng HTTP request là cho một JSP page và chuyển nó tới một JSP engine. Điều này được thực hiện bởi sử dụng một URL hoặc một JSP page mà kết thúc với **.jsp** thay vì **.html**.
- JSP engine tải JSP page từ đĩa và biến đổi nó thành một nội dung Servlet. Sự biến đổi này là rất đơn giản, trong đó tất cả Template Text được biến đổi thành các lệnh `println()` và tất cả phần tử JSP được biến đổi thành Java code mà triển khai các hành vi động tương ứng của trang.
- JSP engine biên dịch Servlet thành một lớp có thể thực thi và chuyển Request ban đầu tới một Servlet engine.
- Một phần của Web server triệu hồi Servlet engine tải lớp Servlet và thực thi nó. Trong quá trình thực thi, Servlet này tạo một output trong định dạng HTML, mà Servlet engine truyền tới Web server bên trong một HTTP response.
- Web server chuyển HTTP Response tới trình duyệt của bạn nội dung HTML tĩnh.
- Cuối cùng, trình duyệt web xử lý trang HTML đã được tạo một cách năng động bên trong HTTP response một cách chính xác như kiểu nó là một trang tĩnh.

Tất cả các bước trên có thể được minh họa trong sơ đồ sau: