



SoICT
HANOI UNIVERSITY OF
SCIENCE AND TECHNOLOGY



B môn Công nghệ Phần mềm
Viện CNTT & TT
Trường Đại học Bách Khoa Hà Nội

LẬP TRÌNH WEB HƯỚNG JAVA

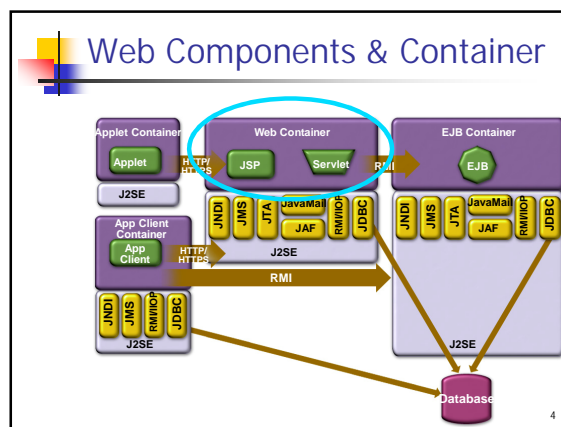
Bài 06: Kiến trúc và cấu trúc ứng dụng WEB

Giảng viên: ThS. Trần Tuấn
B môn CNPM
Email: trinhthuandat.bk@gmail.com/dattt@soict.hut.edu.vn

Nội dung

- 1. Web application, components và Web container
- 2. Các công nghệ sử dụng trong ứng dụng Web
- 3. Các bước phát triển và triển khai ứng dụng Web
- 4. Web Application Archive (*.WAR file)
 - Cấu trúc thông tin trong file *.WAR
 - Thông tin con WEB-INF
- 5. Cấu hình ứng dụng Web
 - **Web application deployment descriptor** (file web.xml)

1. Web Application, Web Components và Web Container



Web Components & Container

- Web components: Servlet hoặc JSP (cùng với **JavaBean** và **custom tags**)
- Các web components chạy trên 1 Web container
 - Các web containers phổ biến: Tomcat và Resin
 - Tất cả các J2EE app servers đều cung cấp các web containers
- Web container cung cấp các dịch vụ hệ thống (**system services**) cho các Web components
 - **Request dispatching**, security, và quản lý vòng đời (life cycle)

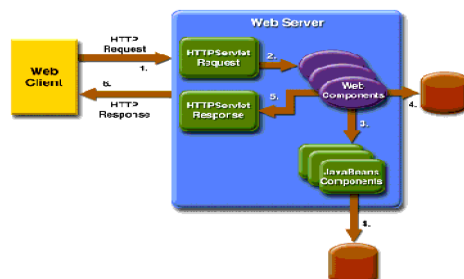
Web Application & Components

- Web Application là 1 gói triển khai, gồm:
 - Web components (Servlets and JSP's)
 - Tài nguyên tĩnh như images
 - Helper classes (sử dụng bởi web components)
 - Thư viện Libraries
 - Deployment descriptor (web.xml file)
- Web Application có thể chia thành
 - Phân chia các thông tin và files (dạng chia gói - **unpacked form**) hoặc
 - *.WAR file: có phân phối trên (dạng gói - **packed form**)

2. Các công nghệ sử dụng trong ứng dụng Web

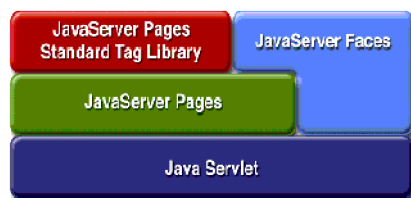
7

Web Request Handling



8

Các công nghệ trong ứng dụng Web Java



9

3. Các bước phát triển và triển khai ứng dụng Web

(Sử dụng ví dụ hello2 trong J2EE 1.4 tutorial)

http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/WCC7.html

10

3. Các bước phát triển và triển khai ứng dụng Web

- Viết code (và biên dịch) cho các Web component (Servlet or JSP), các **helper classes** sử dụng trong web component
- Tổ chức tài nguyên tĩnh (Images, các trang HTML)
- Viết file deployment descriptor (web.xml)
- Build ứng dụng Web (Tạo file *.war hoặc thim c để đóng gói nhúng triển khai)
- Triển khai ứng dụng Web trên 1 Web container
 - Web clients có thể truy cập ứng dụng qua URL

11

3.1. Viết code và biên dịch cho các Web component

- Tổ chức cấu trúc thim c (**Development Tree Structure**)
- Viết code Servlet code hoặc các trang JSP cùng với các lớp phụ trợ
- Tạo file build.xml để build ứng dụng
- IDE (ví dụ NetBeans) hỗ trợ thim c các công việc trên

12

Development Tree Structure

- Cần tách biệt mã nguồn với các file biên dịch
- Thư mục gốc bao gồm:
 - build.xml: Ant build file
 - src: Mã nguồn Java cho các servlets và các JavaBeans
 - web: các trang JSP, HTML, images

13

Ví dụ: cấu trúc cây hello2 (trước khi build với lệnh "asant build")

- Hello2
 - src/servlets
 - GreetingServlet.java
 - ResponseServlet.java
 - web
 - WEB-INF
 - web.xml
 - duke.waving.gif
 - build.xml

14

3.2. Tạo các tài nguyên tĩnh

- Các trang HTML
 - Trang Login
 - Trang Error
- Các file Image cần sử dụng trong trang HTML, JSP
 - Ví dụ: duke.waving.gif



15

3.3. Ví dụ deployment descriptor (web.xml)

- **Deployment descriptor** chứa các chi tiết khi triển khai, khi chạy cho Web container
 - Ví dụ: cấu hình URL
- Bắt buộc tất cả các ứng dụng web phải có

16

3.4. Build ứng dụng Web

- Họ có thể đóng gói (*.WAR file) hoặc không
- Quá trình build bao gồm:
 - Tạo thư mục **build** (nếu chưa có) và các thư mục con
 - Biên dịch mã nguồn Java vào thư mục **build/WEB-INF/classes**
 - Java classes nằm trong thư mục **./WEB-INF/classes**
 - copy **web.xml** file vào thư mục **build/WEB-INF**
 - copy các file image vào thư mục **build**

17

Ví dụ: cấu trúc thư mục hello2 sau khi chạy lệnh "ant build"

- Hello1
 - src
 - web
 - build.xml
 - **build**
 - **WEB-INF**
 - **classes**
 - GreetingServlet.class
 - ResponseServlet.class
 - web.xml
 - duke.waving.gif

18

3.5. Tri n khai ng d ng Web

- Tri n khai ng d ng trên **Sun Java System App Server** hoặc **Tomcat**
- 3 cách tri n khai trên Sun Java System App server
 - asadmin deploy --port 4848 --host localhost --passwordfile "c:\j2eetutorial14\examples\common\admin-password.txt" --user admin hello2.war (asant deploy-war)
 - App server admin console
 - NetBeans**

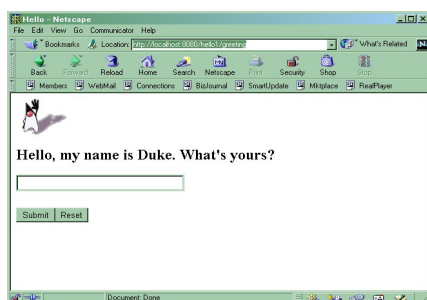
19

3.6. Client th truy c p vào ng d ng Web

- M trình duy t, nh p vào URL c a ng d ng Web

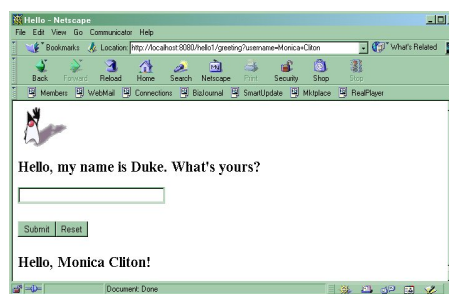
20

<http://localhost:8080/hello1/greeting>



21

Ch y th ch c n ng c a ng d ng Web



22

4. Web Application Archive (*.WAR)

ng d ng Web

- ng d ng Web có th c tri n khai 2 d ng
 - M t file *.war
 - M t th m c, c u trúc gi ng file *.war file (th m c build)
- S d ng d ng file *.war khi mu n tri n khai trên 1 **remote machine**
 - L nh **asant deploy-war**

23

24

File *.WAR là gì?

- Là gói triển khai trên web container
- Thường là file *.jar
- Chứa tất cả thành phần cần thiết:
 - Web components (servlets hoặc JSP)
 - Các Class tĩnh
 - Nội dung tĩnh (HTML, image, ...)
 - Các lớp phía Client (applets và các lớp tiện ích)
- Có nội dung gì không thì mình build

25

Document Root & Context

- **Document Root** của ứng dụng Web
 - Thường nằm trong file WAR
 - Chứa các trang JSP pages, tĩnh, lớp phía client, tài nguyên Web tĩnh
 - Chứa các tệp WEB-INF
- Một **context** là 1 nhãn danh (name) để map với **document root** của ứng dụng Web
 - /hello1 là context cho ví dụ hello1
 - Phân biệt các ứng dụng Web trong **M.T** Web container
 - Chỉ cần nhớ 1 phần của URL

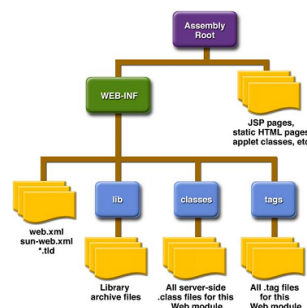
26

Cấu trúc thông tin của file *.WAR



27

Cấu trúc thông tin của file *.WAR



28

Tạo file *.WAR như thế nào?

- 3 cách khác nhau:
 - Sử dụng IDE (NetBeans)
 - Sử dụng công cụ ant
 - "asant create-war" (J2EE 1.4 tutorial)
 - Sử dụng lệnh "jar cvf <filename>.war ." khi trong thư mục build

29

Ví dụ: Tạo file hello2.war với lệnh "asant create-war"

```

C:\j2eetutorial14\examples\web\hello2>asant create-war
Buildfile: build.xml
...
create-war:
[echo] Creating the WAR...
[delete] Deleting:
C:\j2eetutorial14\examples\web\hello2\assemble\war\hello2.war
[delete] Deleting directory
C:\j2eetutorial14\examples\web\hello2\assemble\war\WEB-INF
[copy] Copying 1 file to
C:\j2eetutorial14\examples\web\hello2\assemble\war\WEB-INF
[copy] Copying 2 files to
C:\j2eetutorial14\examples\web\hello2\assemble\war\WEB-INF\classes
[war] Building war:
C:\j2eetutorial14\examples\web\hello2\assemble\war\hello2.war
[copy] Copying 1 file to C:\j2eetutorial14\examples\web\hello2
  
```

30

Ví dụ : Tạo file hello2.war sử dụng lệnh jar

```
C:\j2eetutorial14\examples\web\hello2\build>jar cvf hello2.war .
added manifest
adding: duke.waving.gif(in = 1305) (out= 1295)(deflated 0%)
adding: servlets/(in = 0) (out= 0)(stored 0%)
adding: servlets/GreetingServlet.class(in = 1680) (out= 887)(deflated 47%)
adding: servlets/ResponseServlet.class(in = 1090) (out= 572)(deflated 47%)

C:\j2eetutorial14\examples\web\hello2\build>jar xvf hello2.war
created: META-INF/
extracted: META-INF/MANIFEST.MF
extracted: duke.waving.gif
created: servlets/
extracted: servlets/GreetingServlet.class
extracted: servlets/ResponseServlet.class
```

31

Thư mục WEB-INF

- Là thư mục con của **Document root**
- Chứa:
 - web.xml : **Web application deployment descriptor**
 - Các file **JSP tag library descriptor**
 - Classes : là thư mục chứa các lớp phía server: servlets, lớp tiện ích, các JavaBeans
 - lib : là thư mục chứa các file th viện JAR (tag libraries, các th viện tiện ích) cung cấp các lớp phía server)

32

HTTP request URL & Web component URL (alias) & Context

- Request URL: yêu cầu truy cập tài nguyên Web
 - http://[host]:[port]/[request path]?[query string]
 - [request path] bao gồm **context** + URL của component
 - http://localhost:8080/**hello1/greeting**?username=Monica
- Context: nh danh (Name) của **root document** của 1 ứng dụng Web- nh danh 1 ứng dụng cụ thể trên server đó
 - ./hello1 là **context**

33

5. Cấu hình ứng dụng Web qua file web.xml

34

Cấu hình ứng dụng Web

- Thông số cấu hình được đặt trong file **web.xml** (Web Applications Deployment Descriptor)

35

Web Applications Deployment Descriptor (web.xml)

- Prolog
- **Alias Paths**
- Context & Initialization Parameters
- Event Listeners
- Filter Mappings
- Error Mappings
- Các tham chiếu tới Environment Entries, Resource environment entries, hoặc các Resources

36

Web Applications Deployment Descriptor (web.xml)

- Phân bố tổ chức hóa-thành
- Phức tạp ứng dụng như sau:
 - icon, display-name, description, distributable
 - context-param, filter, filter-mapping
 - listener, servlet, servlet-mapping, session-config
 - mime-mapping, welcome-file-list
 - error-page, taglib, resource-env-ref, resource-ref
 - security-constraint, login-config, security-role
 - env-entry, ejb-ref, ejb-local-ref

37

Prolog (web.xml)

- Tất cả tài liệu XML cần có prolog

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems,
Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
```

38

Alias Paths (web.xml)

- Khi 1 Servlet container nhận 1 request, nó cần biết Web component nào trong ứng dụng Web xử lý request này.
 - Thông tin map URL path trong request tới 1 Web component
- Một URL path chứa 1 context root và 1 alias path
 - http://<host>:8080/context_root/alias_path
- Alias Path có thể có 2 dạng
 - /alias-string (cho servlet) hoặc
 - /*.jsp (cho JSP)

39

Alias Paths (web.xml)

```
<servlet>
  <servlet-name>greeting</servlet-name>
  <display-name>greeting</display-name>
  <description>no description</description>
  <servlet-class>GreetingServlet</servlet-class>
</servlet>
<servlet>
  <servlet-name>response</servlet-name>
  <display-name>response</display-name>
  <description>no description</description>
  <servlet-class>ResponseServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>greeting</servlet-name>
  <url-pattern>/greeting</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>response</servlet-name>
  <url-pattern>/response</url-pattern>
</servlet-mapping>
```

40

Tham số khởi tạo và ng c nh (Context and Initialization Parameters)

- Biện pháp định nghĩa ng c nh trong ứng dụng
- Có thể dùng chung biến các Web components trong cùng 1 file WAR

```
<web-app>
  ...
  <context-param>
    <param-name>
      javax.servlet.jsp.jstl.fmt.localizationContext
    </param-name>
    <param-value>messages.BookstoreMessages</param-value>
  </context-param>
  ...
</web-app>
```

41

Event Listeners (web.xml)

- Nhận các events trong vòng đời của servlet

```
<listener>
  <listener-class>
    listeners.ContextListener
  </listener-class>
</listener>
```

42

Filter Mappings (of web.xml)

- Chỉ ra filters nào sẽ áp dụng cho request nào, và theo thứ tự nào

```
<filter>
  <filter-name>OrderFilter</filter-name>
  <filter-class>filters.OrderFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>OrderFilter</filter-name>
  <url-pattern>/receipt</url-pattern>
</filter-mapping>
```

43

Error Mappings (of web.xml)

- Ánh xạ (Map)
 - mã trạng thái (status code) trả về trong 1 HTTP response do có 1 ngoại lệ Java
 - vị trí 1 Web resource (ví dụ các trang error page)

```
<error-page>
  <exception-type>exception.OrderException</exception-type>
  <location>/errorpage.html</location>
</error-page>
```

44

References (of web.xml)

- Sử dụng khi có web components muốn tham chiếu tới các tài nguyên (database), môi trường.
- Ví dụ: khai báo 1 tham chiếu tới data source

```
<resource-ref>
  <res-ref-name>jdbc/BookDB</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

45

Ví dụ web.xml của hello2

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" version="2.4"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2.4.xsd">
  <display-name>hello2</display-name>
  <servlet>
    <display-name>GreetingServlet</display-name>
    <servlet-name>GreetingServlet</servlet-name>
    <servlet-class>servlets.GreetingServlet</servlet-class>
  </servlet>
  <servlet>
    <display-name>ResponseServlet</display-name>
    <servlet-name>ResponseServlet</servlet-name>
    <servlet-class>servlets.ResponseServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>GreetingServlet</servlet-name>
    <url-pattern>/greeting</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>ResponseServlet</servlet-name>
    <url-pattern>/response</url-pattern>
  </servlet-mapping>
</web-app>
```

46