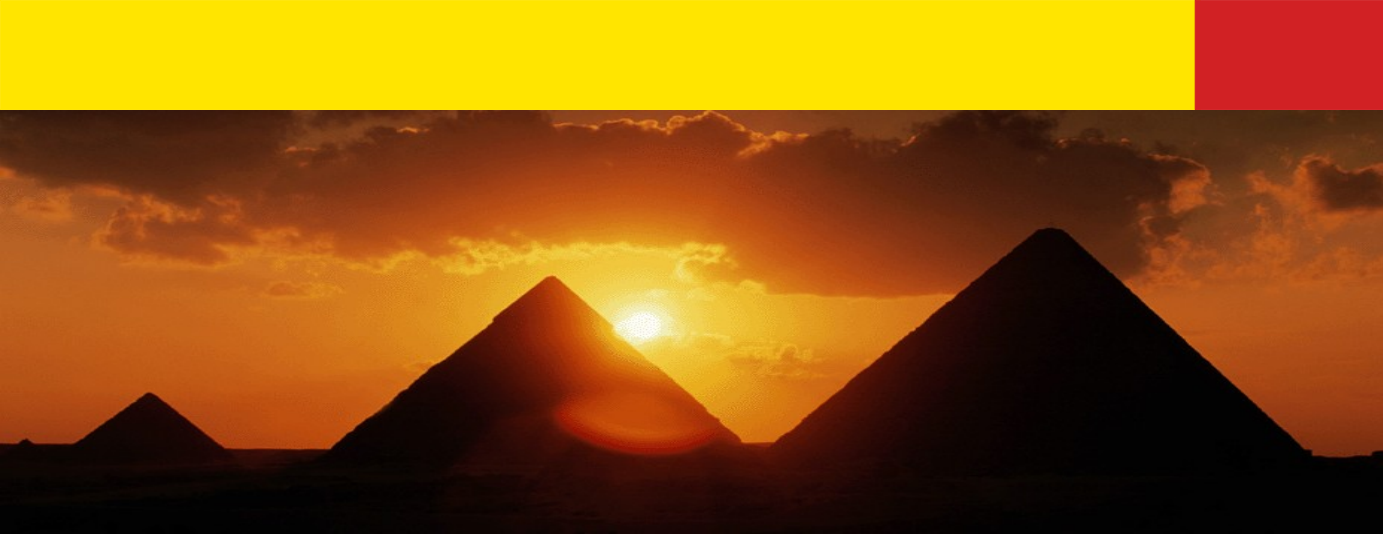




شعبان 1429
الإصدار 1.0

Web Application Architecture (based J2EE 1.4 Tutorial)

بنية تطبيق الشبكة

(معتمد على الكتاب التعليمي الخاص بـ J2EE 1.4)

Disclaimer & Acknowledgments إخلاء مسؤولية و عرفان

- Even though Sang Shin is a full-time employee of Sun Microsystems, the contents here are created as his own personal endeavor and thus does not reflect any official stance of Sun Microsystems.

على الرغم من أن Sang Shin يعمل في شركة SUN فإن محتوى هذه الدروس تم إنجازها بمجهوده الشخصي وبالتالي لا تعبر بأي حال عن الموقف الرسمي لشركة SUN

- Sun Microsystems is not responsible for any inaccuracies in the contents.

لا تتحمل شركة Sun أي مسؤولية عن الأخطاء الموجودة في محتوى هذه الدروس

- Acknowledgments عرفان

- The slides of this presentation are made from “Web Application” section of J2EE 1.4 tutorial written by Sun Microsystems

محتوى هذه الشريحة مقتبس من فصل “Web Application” من الكتاب التعليمي الذي ألفته Sun لتعليم J2EE 1.4

Agenda مفكرة

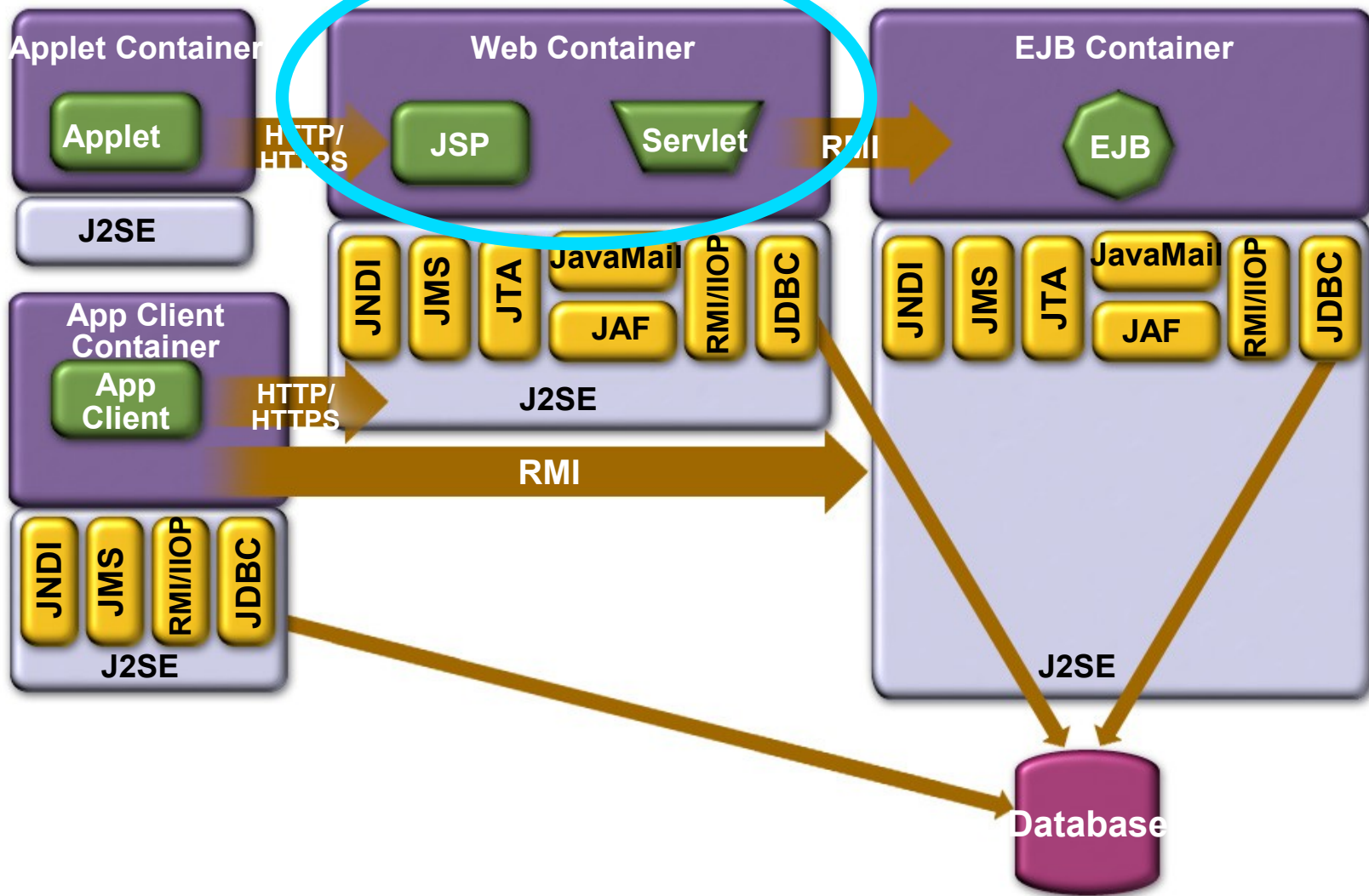
- Web application, components and Web container
تطبيق شبكة و المكونات و حاوية الشبكة
- Technologies used in Web application
التقنيات المستعملة في تطبيق الشبكة
- Web application development and deployment steps
خطوات تطوير و نشر تطبيق الشبكة
- Web Application Archive (***.WAR file**)
 - *.WAR directory structure بنية مجلد WAR
 - WEB-INF subdirectory الملف الفرعي WEB-INF
- Configuring Web application تشكيل تطبيق الشبكة
 - Web application deployment descriptor (**web.xml file**)
واصف نشر تطبيق الشبكة



Web Application & تطبيق شبكة **Web Components** & مكونات شبكة **Web Container** حاوية شبكة

Web Components & Container

حاوية و مكونات الشبكة

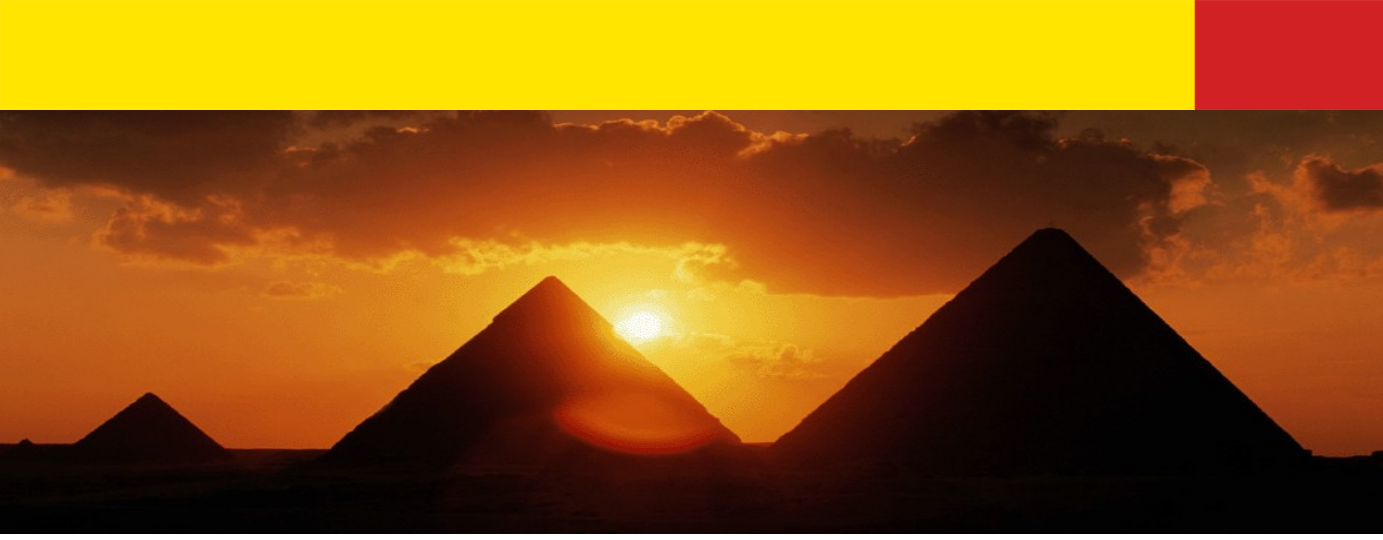


Web Components & Container

- Web components are in the form of either Servlet or JSP (along with JavaBean's and custom tags)
 مكونات الشبكة هي إما على شكل Servlet أو على شكل JSP إلى جانب JavaBean والأوسمة الخاصة
- Web components run in a Web container
 تشغل مكونات الشبكة فوق حاوية الشبكة
 - Tomcat and Resin are popular ^{شعبية} web containers
 - All J2EE compliant app servers (Sun Java System App Server) provide web containers
 جميع إصدارات * خدمات تطبيقات J2EE توفر حاويات شبكة
- Web container provides system services to Web components
 حاويات الشبكة توفر خدمات النظام لمكونات الشبكة
 - Request dispatching , security , and life cycle management
 طلب إفاد* , حماية وإدارة دورة الحياة

Web Components & Container حاوية و مكونات الشبكة

- Web Application is a **deployable package**
تطبيق الشبكة هي حزمة قابلة للنشر
 - Web components مكونات الشبكة (Servlets and JSP's)
 - Static resource files such as images ملفات موارد ساكنة مثل الصور
 - Helper classes *مساعد الطبقات
 - Libraries مكتبات
 - Deployment descriptor (web.xml file) واصف النشر
- Web Application can be represented as
يمكن أن نمثل تطبيق الشبكة عن طريق
 - **A hierarchy of directories and files** ملفات و مجلدات هرمية (unpacked form) **or**
 - ***.WAR file** reflecting the same hierarchy يمثل نفس الهرمية (packed form)

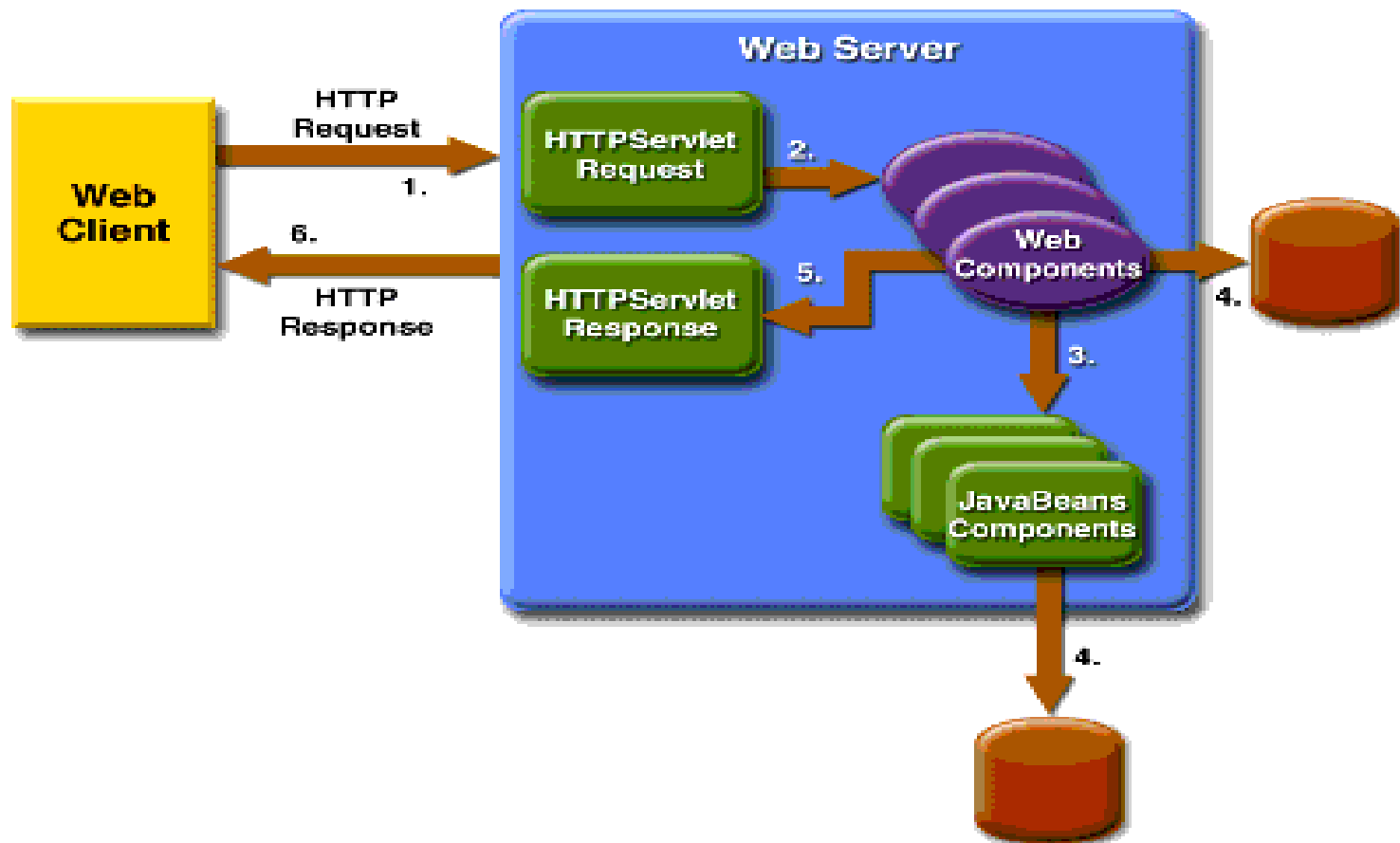


Technologies Used In Web Application

التقنيات المستعملة في تطبيق الشبكة

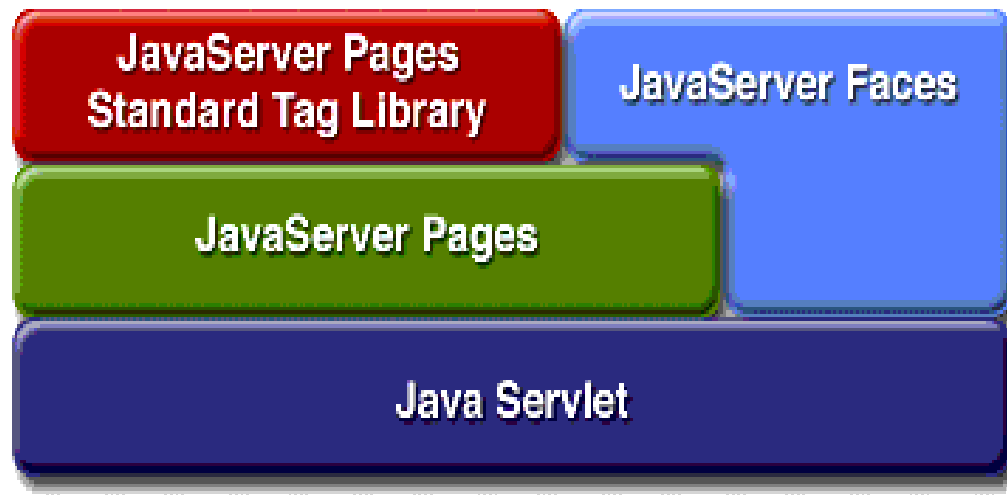
Web Request Handling

معالجة طلبات الشبكة



Java Web Application Technologies

تقنيات جافا في تطبيقات الشبكة





Web Application Development and Deployment Steps (using hello2 example under J2EE 1.4 tutorial)

خطوات تطوير و نشر تطبيقات الشبكة

(باستخدام مثال hello2 المقتبس من الكتاب التعليمي ل J2EE 1.4)

Web Application Development and Deployment Steps

خطوات تطوير و نشر تطبيقات الشبكة

1. Write (and compile) the Web component code (Servlet or JSP) and helper classes referenced by the web component code
قم بكتابة و تجميع مصدر مكون الشبكة (Servlet أو JSP) و مساعد الأصناف المحالة من قبل مكون * الشبكة
2. Create any static resources (for example, images or HTML pages)
إنشاء أي مصادر ثابتة (مثل الصور أو صفحات HTML)
3. Create deployment descriptor (web.xml)
إنشاء واصف النشر
4. Build the Web application (* .war file or deployment-ready directory)
قم ببناء تطبيق الويب
5. Deploy the web application into a Web container
أنشر تطبيق الشبكة من خلال حاوية الشبكة
 - Web clients are now ready to access them via URN
حرفاء الشبكة الآن مستعدون للنفوذ من خلال URN

1. Write and compile the Web component code

كتابة و تجميع مصدر مكون الشبكة

- Create development tree structure

إنشاء شجرة بنية التطوير

- Write either servlet code or JSP pages along with related helper code

كتابة إما مصدر JSP أو مصدر Servlet مع مصدر المساعدة المتصل

- Create **build.xml** for Ant-based build **build.xml** إنشاء (and other application development life-cycle management) process الذي يساعد Ant على بناء المشروع
- IDE (i.e. NetBeans) handles all these chores

بيئة التطوير المتكاملة تقوم بكل هذه الأعمال الروتينية

Development Tree Structure

شجرة بنية التطوير

- Keep Web application source separate from compiled files
تحافظ على فصل مصدر تطبيق الشبكة عن ملفات التجميع
- facilitate iterative development
- Root directory **الدليل الجذري** (example from hello2 sample code from J2EE 1.4 tutorial)
 - **build.xml**: Ant build file
 - **src**: Java source of servlets and JavaBeans components
مصدر جافا لـ **servlets** و مكونات **JavaBeans**
 - **web**: JSP pages and HTML pages, images
صفحات **JSP** و صفحات **HTML** و الصور

Example: hello2 Tree Structure (before “ant build” command)

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

2. Create any static resources

- HTML pages
 - Custom pages
 - Login pages
 - Error pages
- Image files that are used by HTML pages or JSP pages الصور المستخدمة في صفحات HTML أو صفحات JSP
 - Example: [duke.waving.gif](#)

3. Create deployment descriptor (web.xml)

إنشاء واصف النشر

- Deployment descriptor contains deployment time runtime instructions to the Web container

واصف النشر يتضمن تعليمات وقت التشغيل الزمنية للنشر الذي تقوم به حاوية الشبكة

- URL that the client uses to access the web component

URL التي يستعملها الحريف للنفاذ إلى مكون الشبكة

- Every web application has to have it

جميع تطبيقات الشبكة تحتاج إليه

4. Build the Web application بناء تطبيق شبكة

- Either *.WAR file or unpacked form of *.WAR file

إما ملف WAR أو ملف مفتوح الشكل من WAR

- Build process is made of عملية البناء تعتمد على

- create **build** directory (if it is not present) and its subdirectories

إنشاء مجلد **build** مع جميع مجلداته الفرعية (إذا كان غير موجود)

- compile Java code into **build/WEB-INF/classes** directory

تجميع مصدر جافا عن طريق مجلد **build/WEB-INF/classes**

- Java classes reside under **./WEB-INF/classes** directory

أصناف جافا تستقر تحت مجلد **./WEB-INF/classes**

- copy **web.xml** file into **build/WEB-INF** directory

إنسخ ملف **web.xml** تحت مجلد **build/WEB-INF**

- copy image files into **build** directory

إنسخ ملفات الصور تحت مجلد **build**

Example: hello2 Tree Structure (after “asant build” command)

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

5. Deploy Web application

أنشر تطبيق الشبكة

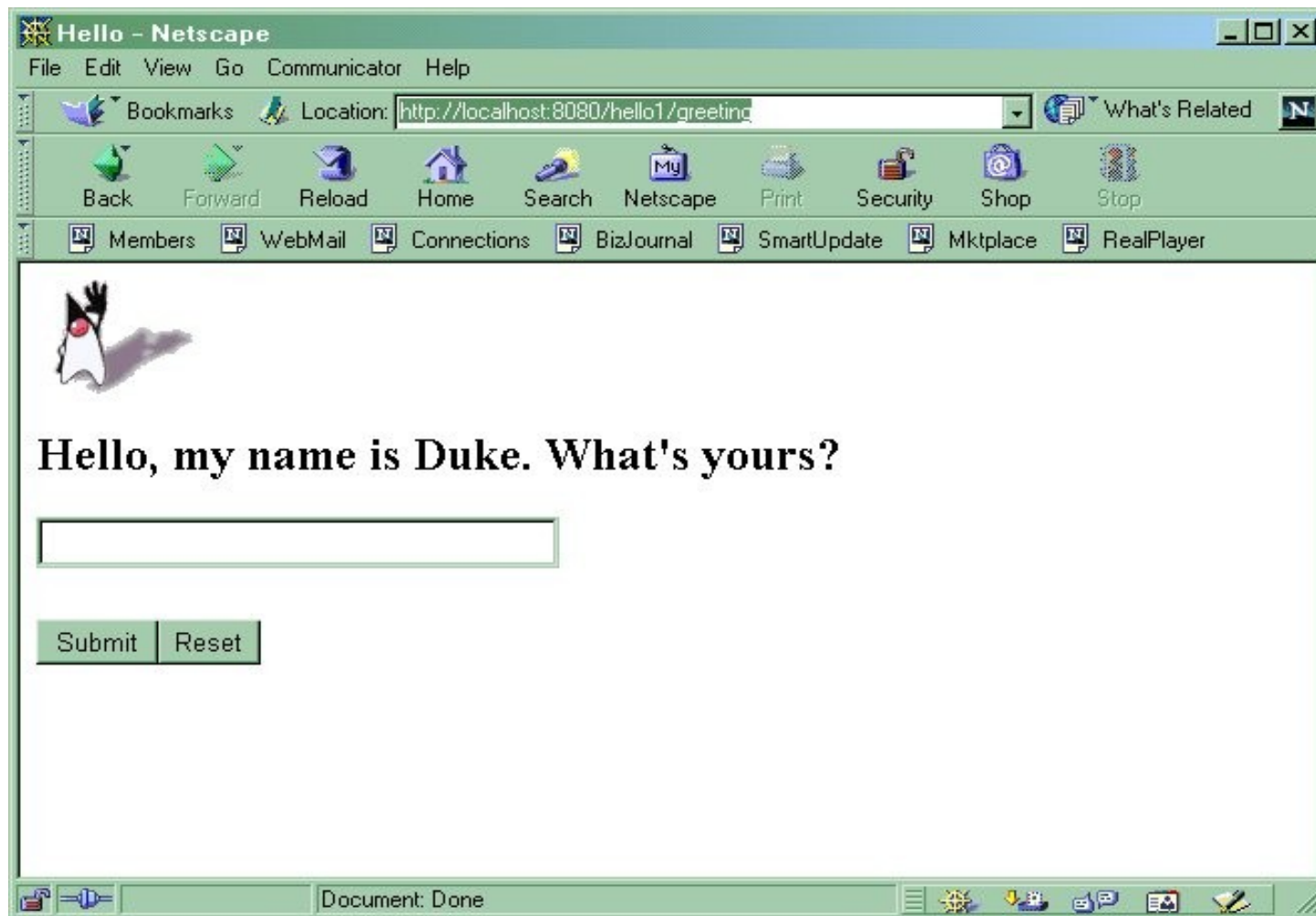
- Deploy the application over deployment platform such as Sun Java System App Server or Tomcat
Sun Java System App Server أنشر التطبيق من خلال منصة النشر مثل
أو Tomcat
- 3 ways to deploy to Sun Java System App server
ثلاث طرق للنشر من خلال Sun Java System App Server
 - asadmin deploy --port 4848 --host localhost --passwordfile "c:\j2eetutorial14\examples\common\admin-password.txt" --user admin hello2.war (asant deploy-war)
 - App server admin console
 - NetBeans

6. Perform Client Access to Web Application

إجعل الحريف ينفذ إلى تطبيق الشبكة

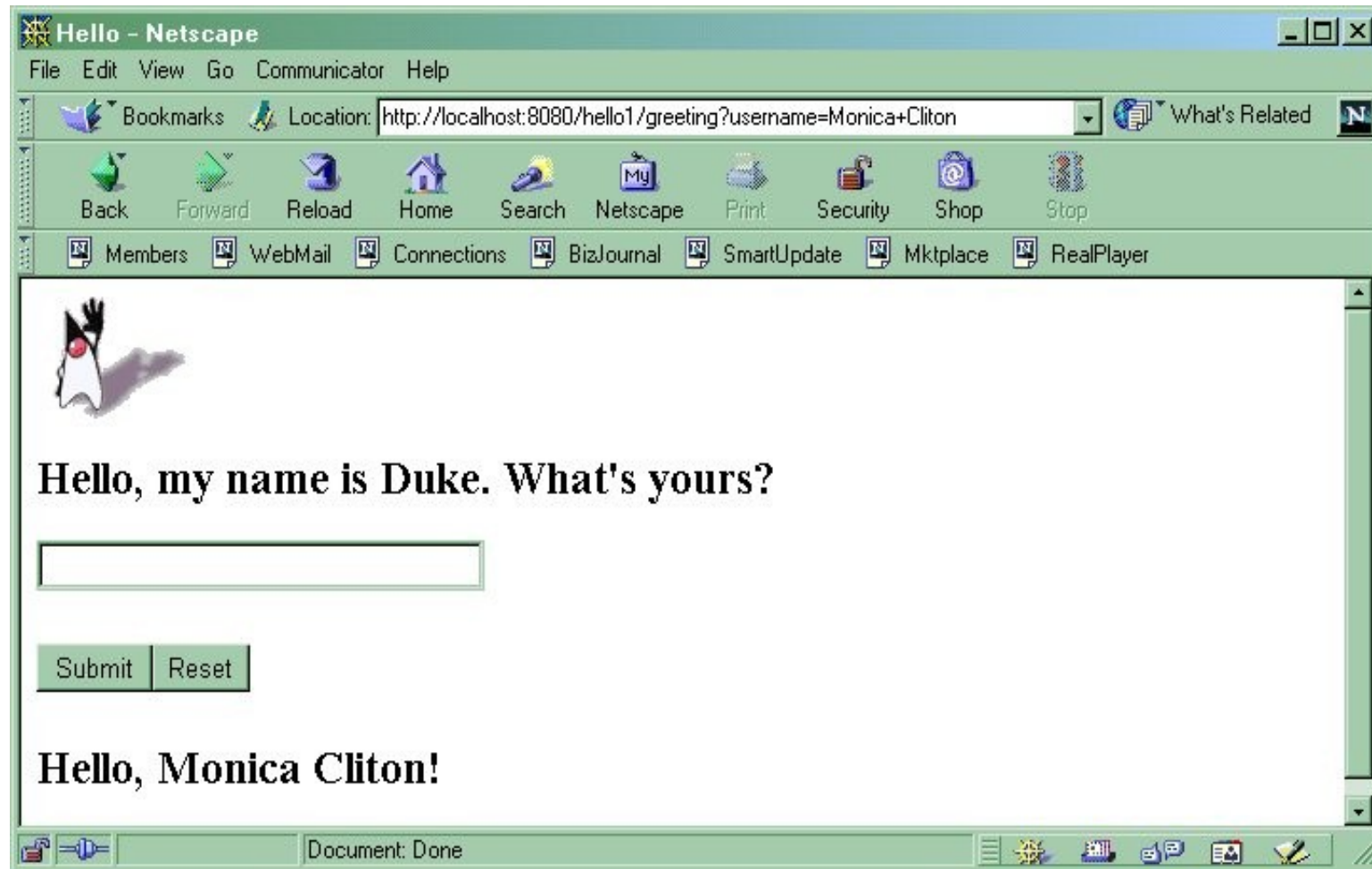
- From a browser, go to URN of the Web application
عن طريق المصفح إذهب إلى URN الخاصة بتطبيق الشبكة

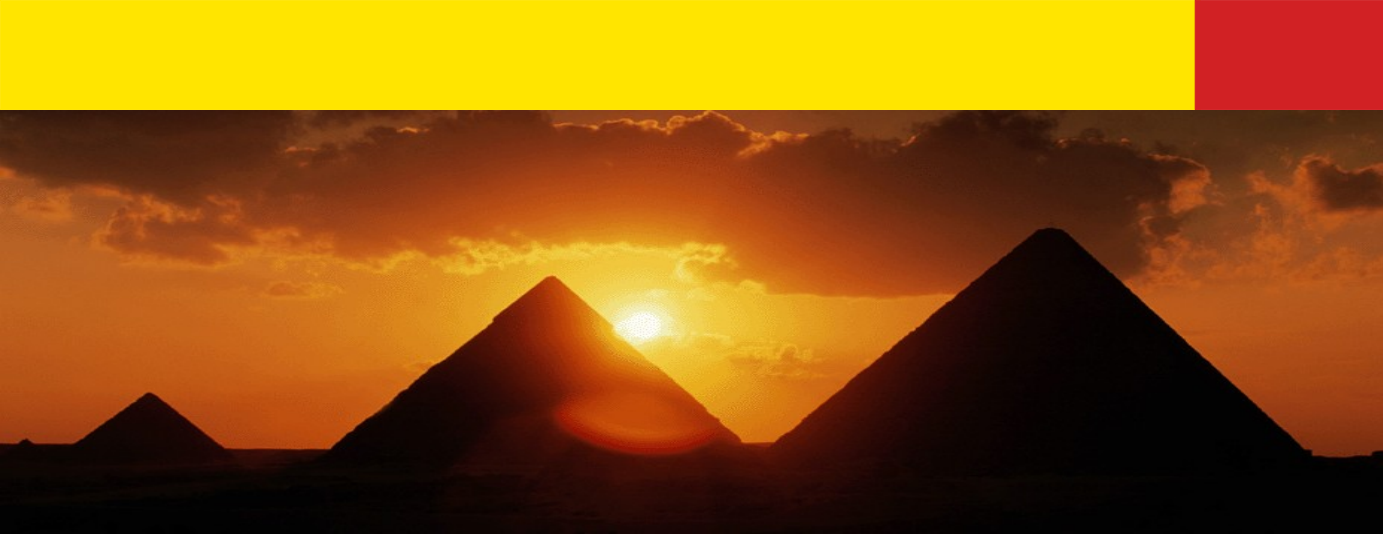
http://localhost:8080/hello1/greeting



Running Web Application

تشغيل تطبيق الشبكة





Web Application Archive (*.WAR)

أرشيف تطبيق الشبكة

Web Application

تطبيق الشبكة

- Web application can be deployed in two different forms
تطبيق الشبكة يمكن نشرها في شكلين مختلفين
 - a *.war file or
 - an unpacked directory laid out in the same format as a *.war file (build directory)
- Use *.war file when you have to deploy on a remote machine
إستعمل ملف war عندما تريد النشر على جهاز آخر
 - `asant deploy-war` command

What is *.WAR file?

ماهو ملف WAR ؟

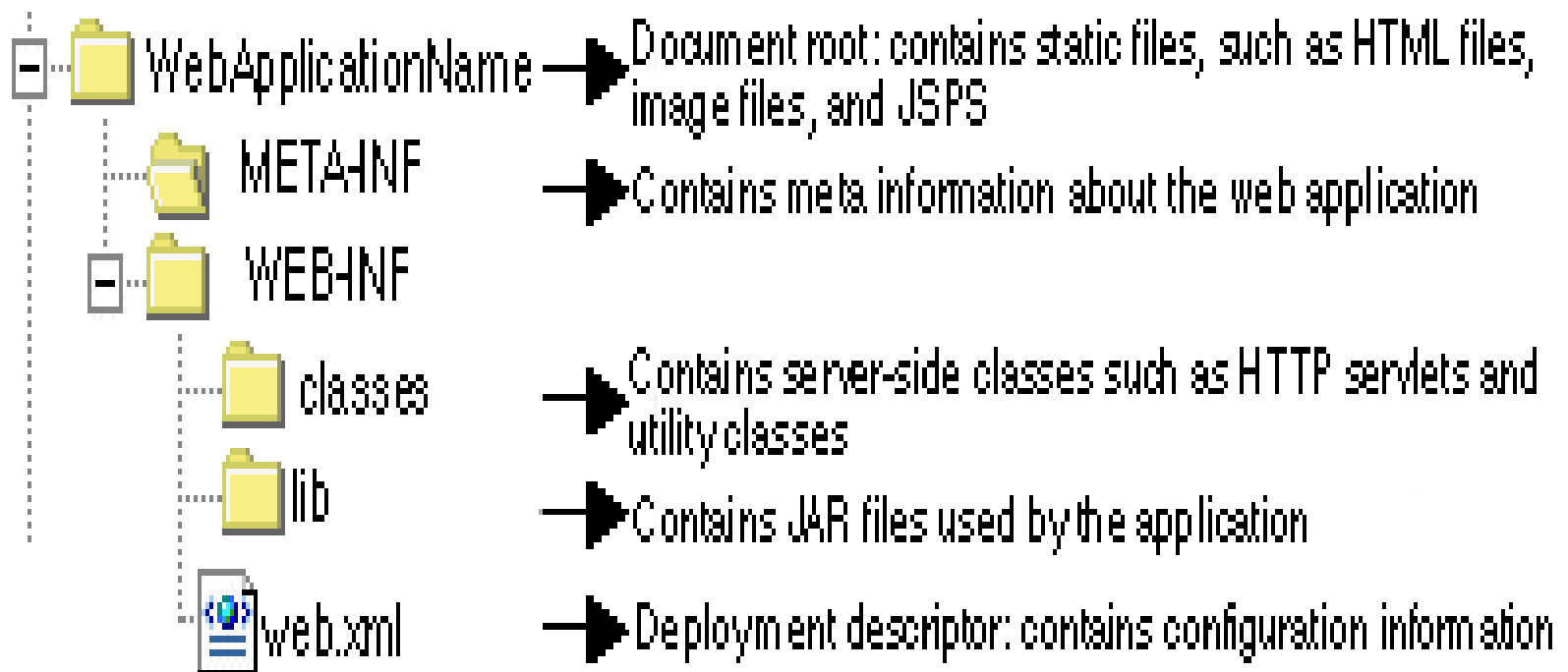
- Ready to deploy'able package over web container
جاهز لنشر الحزم من خلال حاوية الشبكة
- Similar to *.jar file
شبيه بملف jar
- Contains things to be deployed
يتضمن أشياء يتعين نشرها
 - Web components (servlets or JSP's) مكونات الشبكة
 - Server-side utility classes أصناف مفيدة من جهة الخادم
 - Static Web presentation content (HTML, image, etc)
مكونات العرض الثابتة
 - Client-side classes (applets and utility classes)
أصناف من جهة الحريف (أصناف مفيدة و applets)
- Reflects contents in **build** directory

Document Root & Context الوثائق الجذرية و السياق

- **Document Root** الوثائق الجذرية of the Web application
 - Top-level directory of WAR مجلد war الأعلى مستوى
 - Contains JSP pages, client-side classes and archives, and static Web resources are stored يتضمن صفحات JSP و أصناف و موارد الشبكة الثابتة من جهة الحريف كلها مخزنة
 - Also contains **WEB-INF** directory كذلك يتضمن مجلد **WEB-INF**
- A **context** is a name that gets mapped to the document root of a Web application
 - `/hello1` is context for hello1 example
 - Distinguishes a Web application in a single Web container
 - Has to be specified as part of client URN

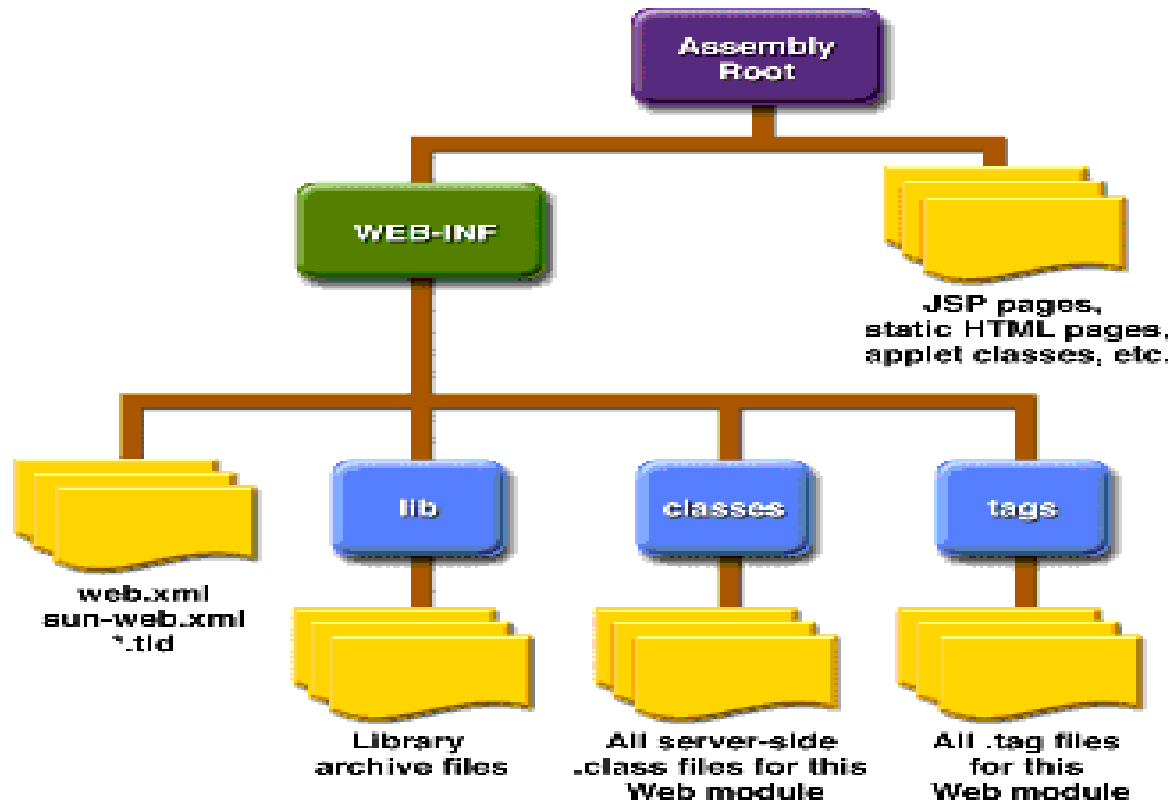
Directory Structure of *.WAR file

بنية مجلد ملفات war



Directory Structure of *.WAR file

بنية مجلد ملفات war



How to Create *.WAR file? كيف يتم إنشاء ملف war

- 3 different ways هناك ثلاث طرق مختلفة
 - Use IDE (NetBeans) باستعمال بيئة تطوير متكاملة
 - Use ant tool after putting proper build instruction in `build.xml` file باستعمال الأداة لبرمجية `ant` بعد إدخال تعليمات البناء اللازمة داخل ملف `build.xml`
 - “asant create-war” (under J2EE 1.4 tutorial)
 - Use “`jar cvf <filename>.war .`” command under `build` directory إستخدم الأمر المذكور سابقا تحت مجلد `build`

Example: Creating hello2.war via “asant create-war” command

```
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
```

Example: Creating hello2.war via jar command

```
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
```

WEB-INF Directory

- Subdirectory of Document root دليل فرعي للمستند الجذرية
- Contains يتضمن
 - **web.xml** : Web application deployment descriptor واصف نشر تطبيق الشبكة
 - JSP tag library descriptor files الملفات الواصفة لمكتبة وسوم JSP
 - **Classes** : A directory that contains server-side classes: servlets, utility classes, and JavaBeans components الدليل الذي يحتوي الأصناف من جهة الخادم : **servlet** والأصناف المفيدة و مكونات JavaBeans
 - **lib** : A directory that contains JAR archives of libraries (tag libraries and any utility libraries called by server-side classes) الدليل الذي يتضمن أرشيفات JAR الخاص بالمكتبات (الوسوم المكتبية و جميع المكتبات المفيدة التي تستدعى من الأصناف من جهة الخادم)

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

- Request URL: User specified access point of a web resource *النقطة المخصصة للحريف حتى يتمكن من النفاذ إلى موارد الشبكة*
 - `http://[host]:[port]/[request path]?[query string]`
 - [request path] is made of context and web component's URN
 - `http://localhost:8080/hello1/greeting?username=Monica`
- Context: Name of the root document of a web application – Identifies a particular application on that server *إسم المستند الجذري لتطبيق الشبكة – تحدد * تطبيق معين على الخادم*
 - `/hello1` is context



Configuring Web Application via web.xml

تشكيل تطبيق الشبكة من خلال web.xml

Configuring Web Application

تشكيل تطبيق الشبكة

- Configuration information is specified in **web.xml** (Web Applications Deployment Descriptor) **web.xml** المعلومات الخاصة بالتشكيل يقع تحديدها في واصف نشر تطبيق الشبكة

Web Applications Deployment Descriptor (web.xml)

واصف نشر تطبيق الشبكة

- Prolog
- **Alias Paths**
- Context and Initialization Parameters
- Event Listeners
- Filter Mappings
- Error Mappings
- Reference to Environment Entries, Resource environment entries, or Resources

Web Applications Deployment Descriptor (web.xml) واصف نشر تطبيق الشبكة

- Case sensitive حساس لحالة الحرف
- Order sensitive حساس * للترتيب (in the following order)
 - 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

Prolog (of web.xml)

- Every XML document needs a prolog

كل مستند XML يحتاج إلى محرك 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">
```

Alias Paths (of web.xml) * كنية الطرقات أو المسارات

- When a request is received by Servlet container, it must determine which Web component in a web application should handle the request. It does so by mapping the **URL path** contained in the request to a Web component

عندما يتم إستقبال طلب من قبل حاوية Servlet يتعين عليها تحديد أي مكون في تطبيق الشبكة الذي سيقوم بمعالجة أو القيام بهذا الطلب, وهي تفعل ذلك بتعيين مسار URL الموجود في الطلب مع مكون الشبكة

- A **URL path** contains the **context root** and **alias path**
 - `http://<host>:8080/context_root/alias_path`

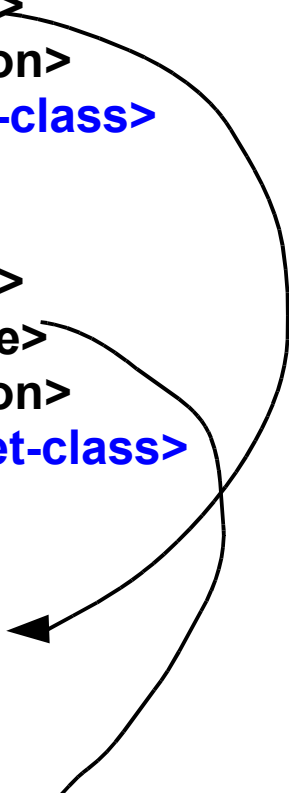
- Alias Path can be in the form of either

كنية المسارات يمكن أن تكون على إحدى الشكلين التاليين

- `/alias-string` (for servlet) or
- `/*.jsp` (for JSP)

Alias Paths (of web.xml) كنية الطرق *

```
<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>
```



The diagram consists of two curved arrows. The first arrow originates from the `<servlet-class>GreetingServlet</servlet-class>` line in the first `<servlet>` block and points to the `<servlet-name>greeting</servlet-name>` line in the first `<servlet-mapping>` block. The second arrow originates from the `<servlet-class>ResponseServlet</servlet-class>` line in the second `<servlet>` block and points to the `<servlet-name>response</servlet-name>` line in the second `<servlet-mapping>` block.

Context and Initialization Parameters (of web.xml)

السياق و تهيئة المتوسطات

- Represents application context
- Can be shared among Web components in a WAR file

تمثل سياق التطبيق

يمكن إشتراكها بين مكونات الشبكة في ملف 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>
```

Event Listeners (of web.xml) مستمع الأحداث

- Receives servlet life-cycle events

يستقبل أحداث دورة حياة ال servlet

```
<listener>
```

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

```
</listener>
```

Filter Mappings (of web.xml) تعيين المرشح

- Specify which filters are applied to a request, and in what order

يقوم بتحديد أي مرشح وقع تطبيقه في الطلب و في أي ترتيب كذلك

```
<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>
```

Error Mappings (of web.xml) تعيين الأخطاء

- Maps status code returned in an HTTP response to a Java programming language exception returned by any Web component and a Web resource

يتم إرجاع خرائط أحوال * المصدر باستعمال إجابة HTTP إلى مسؤول الإستثناءات البرمجي في لغة جافا و التي يتم إرسالها من قبل مكون الشبكة و موارد الشبكة

```
<error-page>
```

```
    <exception-type>exception.OrderException</exception-type>
```

```
    <location>/errorpage.html</location>
```

```
</error-page>
```

References (of web.xml) المراجع

- Need when web components make references to environment entries, resource environment entries, or resources such as databases
نحتاجها إذا قامت مكونات الشبكة بالإشارة إلى مداخل البيئة أو إلى مداخل موارد البيئة أو إلى الموارد مثل قاعدة البيانات
- Example: declare a reference to the data source
إعلان مرجع إلى مصدر البيانات

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

Example web.xml of 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>
```



Passion!