

Introduzione a JSP (Java Server Pages)



© MC 2005

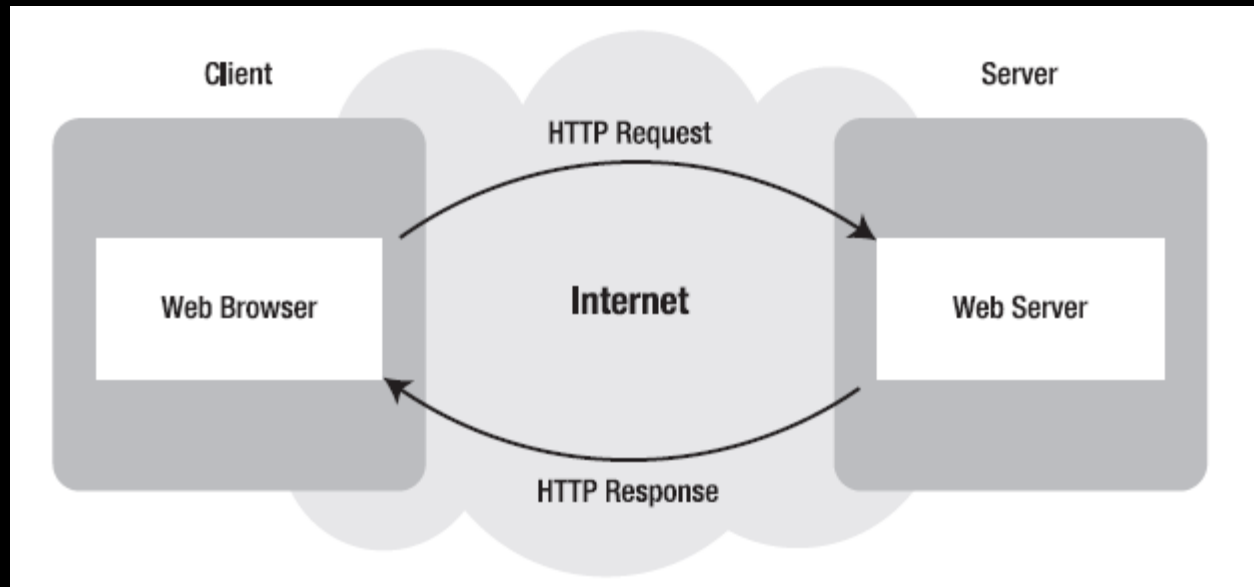
Cosa è JSP

- JSP è una tecnologia per lo sviluppo di pagine web con *contenuti dinamici*
- Miglioramento rispetto alle *CGI* (modello *multithread*)
- Indipendenza dalla piattaforma e dal produttore (*write ones, run everywhere*)
- Integrazione con JDBC, JNDI, RMI ... (tecnologie J2EE)
- Efficienza
- Scalabilità
- Robustezza e sicurezza (non è consentita l'*SQL injection*)

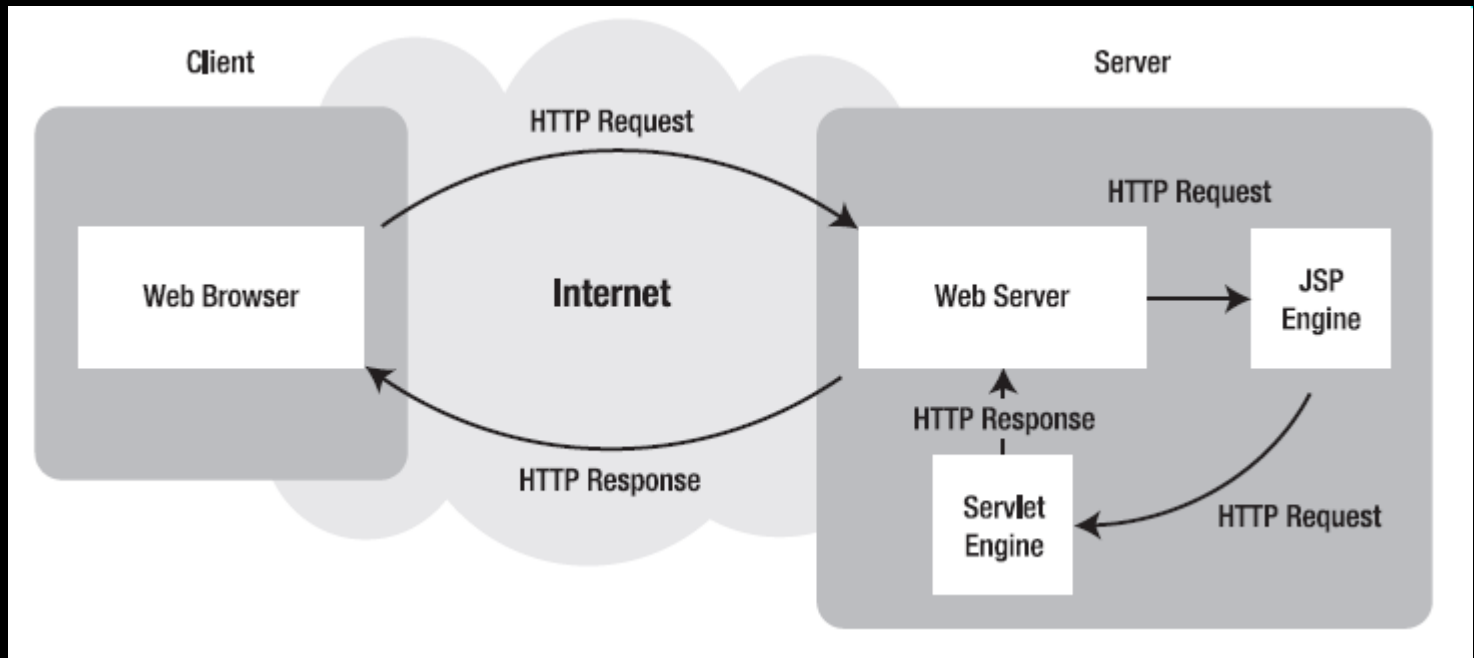
CGI

- Common Gateway Interface
- Programmi scritti in Perl, C o usando la shell (Linux)
- Modello multiprocess (CGI)
- Modello multithread (JSP)

Web server [1]



Web server Java [2]



Servlet



Una Servlet è un programma in esecuzione sul server e in grado di colloquiare col client per mezzo del protocollo HTTP

Ciclo di vita di una Servlet

- Init()
- Service()
 - doGet()
 - doPost()
- Destroy()

Apri

Init

- Il metodo Init serve per inizializzare la Servlet
- Viene eseguito una sola volta
- In genere si usa per effettuare operazioni computazionalmente costose
- Es: connessione ad un database

Service

- Il metodo Service viene mandato in esecuzione al momento di una richiesta di tipo GET o POST
- Manda in esecuzione il metodo doGet() o doPost()

Destroy

- Il Servlet viene terminato dal Servlet Container
- Si usa per liberare la risorsa usata dal servlet
- Es: un file aperto o una connessione di database aperta

Esempio di Servlet



Progettazione di una Servlet con NetBeans



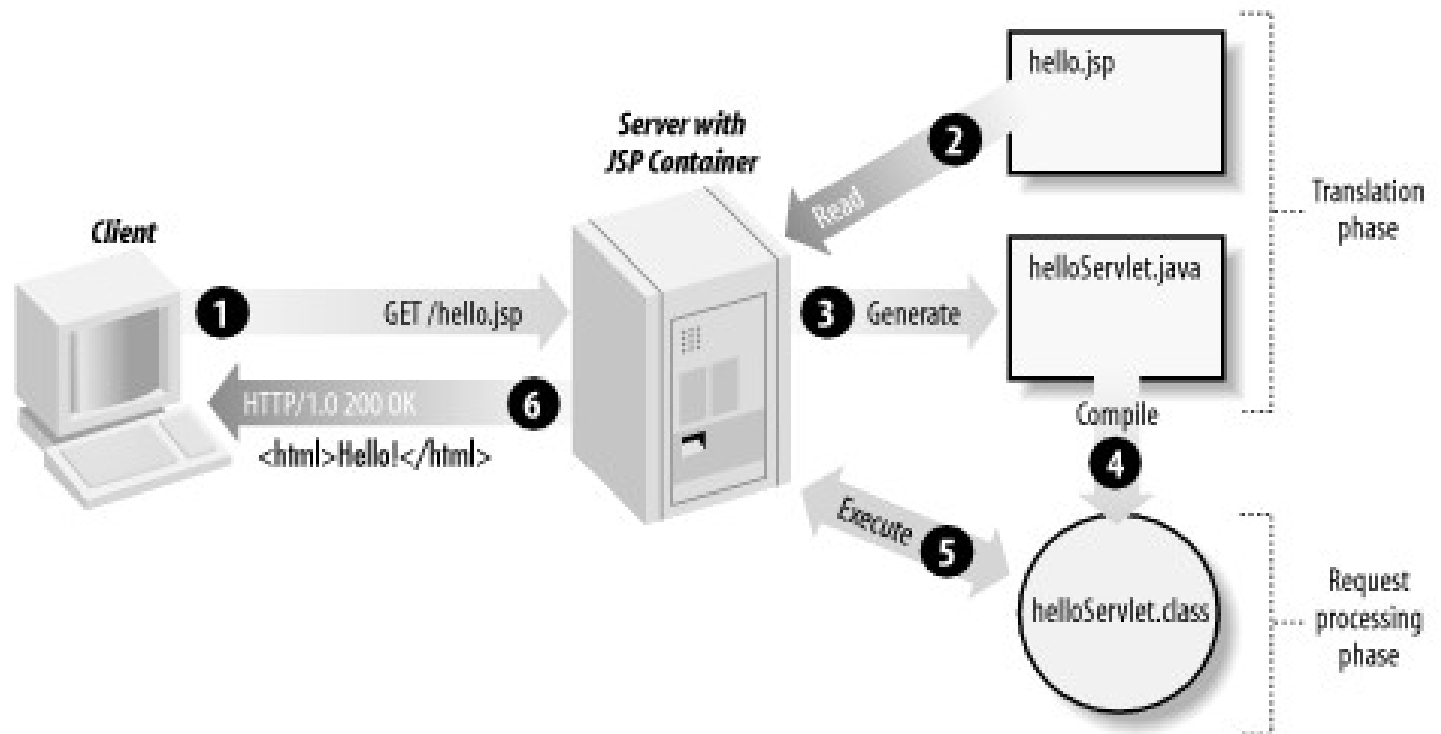
Apri

Le servlet sono “roba” da
programmatori!



Soluzione: Java Server Pages

Fasi di traduzione ed elaborazione di una pagina JSP



JSP Servlet Engine/Application Server

<i>Servlet container</i>	<i>URL</i>
Borland Enterprise Server	http://www.inprise.com
Sun One Application Server	http://www.sun.com
Jetty	http://jetty.mortbay.org/jetty/index.html
JRun	http://www.macromedia.com/software/jrun
Orion Application Server	http://www.orionserver.com
Resin	http://www.caucho.com
Apache Tomcat	http://jakarta.apache.org/tomcat
Weblogic Application Server	http://www.bea.com
WebSphere	http://www-4.ibm.com/software/webservers/appserv
EAServer	http://www.sybase.com

Web Application

- Nelle specifiche J2EE è chiamata *Web Module*
- Consiste di una *logica di Presentazione*
 - Include *HTML, JSP, file di testo, immagini, risorse generiche ecc*
- *Logica di Business*
 - Include *JavaBean, Servlet, classi di utilità generale ecc*
- *Informazioni* su come logica di presentazione e logica di business si legano (*XML*)

Struttura delle cartelle di una Web Application



Apri

Root folder

- Ogni web application ha una struttura di cartelle ben definita. Esiste una cartella principale detta *root folder* o *document base* che contiene una sottocartella *WEB-INF*.
- All'interno troviamo una sottocartella *classes*, una cartella *lib* e un file chiamato *web.xml* (deployment descriptor).

File War



Una applicazione web di solito è impacchettata e distribuita (*deployment*) come file WAR (*web archive*), un formato compatibile con il formato Winzip

Un'applicazione web può contenere

- *Pagine JSP e segmenti JSP*
- *Applets*
- *Classi java per le servlet, filtri, listeners, javabeans, documenti statici (per es file HTML), immagini, suoni ecc*
- *Tag file e tag segment*
- *Custom tag library con file TLD*

Una pagina JSP contiene

- Direttive
- Azioni JSP
- Commenti
- Testo template
- Elementi di scripting (Scriptlet)

Esempio di pagina JSP (Es. 1)

Servlet generata

```
<%@ page contentType="text/html" %>
<%@ taglib prefix="c"
uri="http://java.sun.com/jstl/core" %>
<html>
  <head><title>JSP è facile</title></head>
  <body>
    <h1>JSP è facile come ...</h1>
    <%-- Calcola la somma di 1 + 2 + 3 dinamicamente --
%>
    1 + 2 + 3 = <c:out value="{1 + 2 + 3}" />
  </body>
</html>
```

direttive

Testo template

commento

Azione JSTL

Esecuz

Esempio di pagina JSP (Es. 2)

Servlet generata

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<html>
  <head><title>JSP Page</title></head>
  <body>
    <%
      int a=1,b=2,c=3,d=a+b+c;
      out.println("<h1>Ciao</h1>");
      out.println("<h2>1+2+3="+d+"</h2>");
    %>
  </body>
</html>
```

Scriptlet

Esecuz

Directive e azioni JSP

- Le *directive* hanno effetto soltanto durante la fase di *traduzione* della pagina in Servlet Java e sono utilizzate per specificare *informazioni* che riguardano la pagina stessa
- Le *azioni* rappresentano *operazioni dinamiche*, come inviare e-mail, scrivere un file sul server, leggere da un database ... che devono aver luogo durante l'*esecuzione* della pagina

Direttive

- La sintassi generica è

`<%@ direttiva attr1="val1" attr2="val2" ...%>`

- Ci sono 3 tipi di direttive:
 - Page
 - Include
 - Taglib

Direttiva Page

```
<%@ page contentType="text/html" %>
```

- La direttiva *page* definisce gli attributi dipendenti dalla pagina, come il linguaggio di scripting, la pagina d'errore e i requisiti di buffer
- In questo caso viene utilizzato soltanto l'attributo *contentType* che specifica il tipo *MIME* della pagina da generare

Direttiva Include

```
<%@ include file="nomeFile.htmlf" %>
```

- La direttiva *include* esegue l'inserimento del contenuto di un file statico all'interno di una pagina
- *nomeFile.htmlf* è un file statico e può *non* rappresentare una pagina html completa (*f=fragment*)

Direttiva Taglib

```
<%@ taglib prefix="c"  
    uri="http://java.sun.com/jsp/jstl/core" %>
```

- La direttiva *taglib* dichiara una *tag library*, contenente *azioni personalizzate* utilizzate nella pagina
- L'attributo *prefix* definisce il *prefisso* dello spazio dei nomi XML (*namespace*) che si desidera utilizzare
- L'attributo *uri* è un identificatore unico della tag library

Azioni [1]

- Un'azione è rappresentata da un elemento simile a un tag HTML ordinario

- **Formato1:**

<prefix:action attr1="val1" attr2="val2" ...>

ActionBody

</prefix:action>

- **Formato2:**

<prefix:action attr1="val1" attr2="val2" .../>

Azioni [2] - Tag Library

- Gli elementi o *tag* di azione vengono spesso raggruppati in *librerie* note come *Tag Library*
- Una libreria molto utile è la *JSTL* (*Java Standard Tag Library*)

Azioni [3]

- Azioni standard
- Azioni personalizzate
- Azioni JSTL

Javabean [1]

- è una *componente* software riutilizzabile con la capacità di interagire con altre componenti
- Offre *servizi*
- Può essere utilizzato in ambienti di sviluppo differenti
- *Classi java* manipolabili mediante codice o mediante ambienti di sviluppo grafici

Javabean [2]

- Gli *attributi* di un javabean in genere sono attributi *privati* accessibili attraverso opportuni metodi *getter* e *setter*
- I *metodi* sono *pubblici* e rappresentano la via di accesso ai *servizi* offerti da un javabean
- Consentono la cosiddetta *introspezione* e *personalizzazione*
- La *persistenza* consente ad un javabean di *salvare* il proprio *stato* e di *ripristinarlo* in un secondo tempo (*Object Serialization*)
- I javabean adottano un meccanismo di comunicazione basato sugli *eventi* simile a quello utilizzato da AWT e Swing
- I javabean possono essere distribuiti attraverso *file jar [deployment]*

Azioni standard

Elemento di azione	Descrizione	Esempi
<code><jsp:useBean></code>	Rende disponibile un javaBean	Esempio1
<code><jsp:getProperty></code>	Ricava il valore di una proprietà di un javaBean	Esempio2
<code><jsp:setProperty></code>	Imposta il valore di una proprietà di un javaBean	Esempio3
<code><jsp:include></code>	Include la risposta di una servlet o di una pagina Jsp	Esempio4
<code><jsp:forward></code>	Inoltra l'elaborazione di una richiesta a una servlet o una pagina jsp (reindirizzamento)	Esempio5
<code><jsp:param></code>	Inserisce il valore di un parametro in una richiesta utilizzando <code><jsp:forward></code> o <code><jsp:include></code>	Esempio6
<code><jsp:plugin></code>	Esecuzione di un'applet con un plugin java	Esempio7

JSTL [1]

- Nel giugno 2002, un gruppo di programmatori esperti si riunì nell'ambito del programma *Java Community Process* col nome di JSR052 Expert Group
- Il risultato di quel lavoro fu la release 1.0 della *JSTL (Java Standard Tag Library)*
- In realtà si tratta di un *insieme di 4 librerie* che raggruppano azioni correlate

JSTL [2]

- CORE
 - Elaborazione condizionale e iterativa
- XML
 - Elaborazione di dati XML, come la trasformazione e l'accesso ai singoli elementi
- I18N
 - Formattazione e parsing di informazioni *localizzate*, inserimento di dati *localizzati* in una pagina
- SQL
 - Lettura e scrittura di dati in database relazionali

JSTL [3]

Libreria	URI	Prefisso
CORE	http://java.sun.com/jsp/jstl/core	c
XML	http://java.sun.com/jsp/jstl/xml	x
I18N	http://java.sun.com/jsp/jstl/fmt	fmt
SQL	http://java.sun.com/jsp/jstl/sql	sql

Usare JSTL con NetBeans



Apri demo

Gestione di un database MySql con NetBeans



Apri demo

Gestione di un database MySql con NetBeans



Apri pdf

Linguaggio EL della JSTL [1]

Codice

- EL (Expression Language) è un semplice linguaggio (simile a *Javascript* o *ECMAScript* e *XPath*) per definire i valori degli attributi delle azioni JSTL
- Esempio:
 - `<c:out value="{1 + 2 + 3}" />`
 - `<c:out value="{param.parametroForm}" />` dove *param* è una variabile implicita EL che rappresenta la collezione di tutti i parametri contenuti in un form (*parametroForm* è uno di questi)

Esecuz

- Altri dati relativi alla richiesta sono disponibili sotto forma di proprietà della variabile EL `pageContext`
- Esempio:
 - `<c:out value="{pageContext.request.serverName}" />`
(restituisce il nome del server al quale è stata inoltrata la richiesta)
 - `<c:forEach items="{pageContext.request.cookies}" var="c">`
 `<c:out value="{c.name}" />=<c:out value="{c.value}" />`
 `</c:forEach>`
(restituisce la collection dei cookie memorizzati sul client, almeno JSESSIONID)

Azioni JSTL [Core]

- `<c:out>`
- `<c:if>`
- `<c:when>`
- `<c:choose>`
- `<c:set>`
- `<c:forEach>`

Azione <c:out>

Codice

- Inserisce il valore di un'espressione nel buffer corrente della risposta (*JspWriter*) o un valore di default se l'espressione principale è null
- Sintassi:
 - `<c:out value="espressioneEL" [escapeXml=[true|false]] [default="espressioneDiDefault"] />`

Esecuz

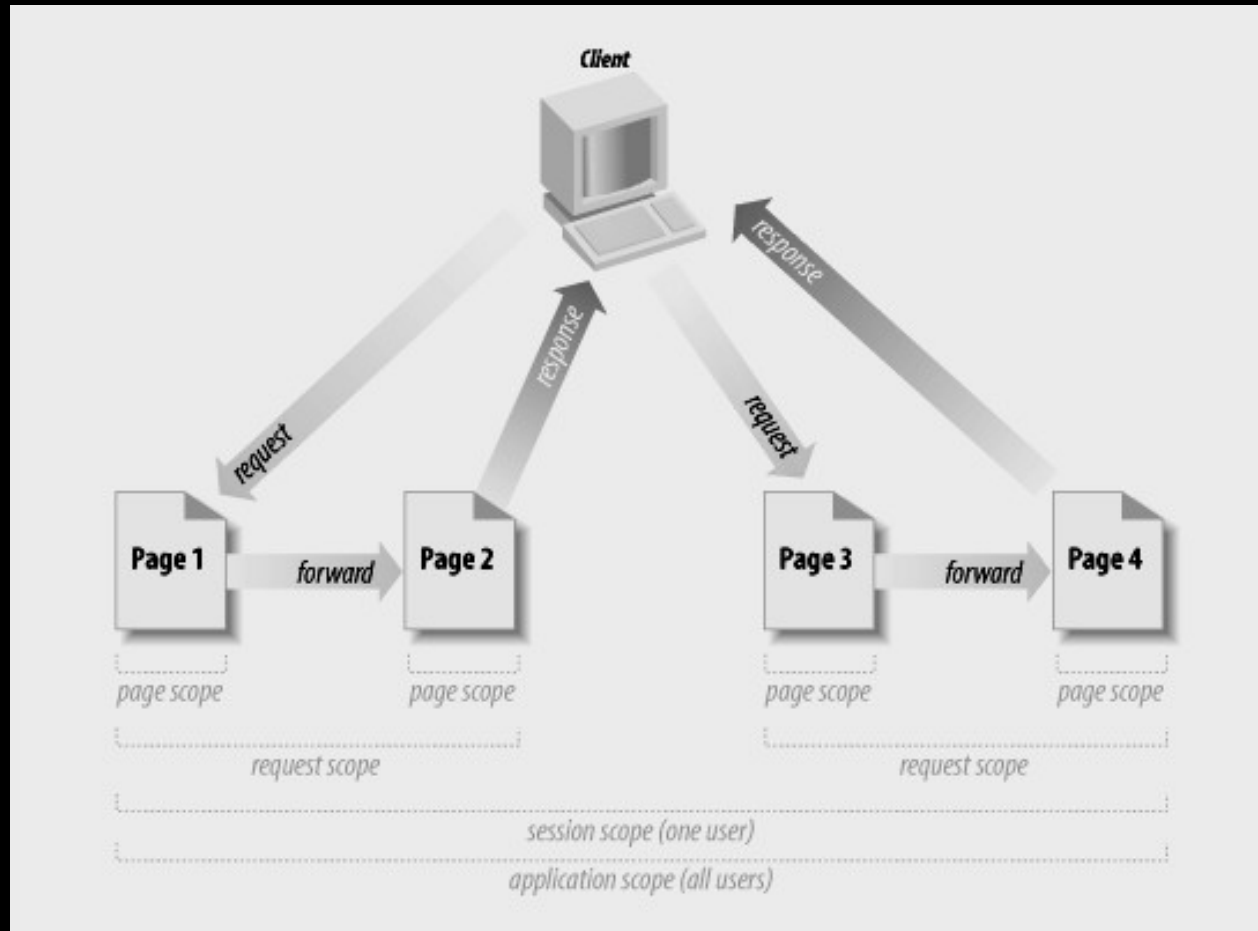
Azione <c:if>

- Valuta il corpo soltanto se l'espressione specificata è true. In alternativa il risultato può essere salvato come variabile Boolean di uno scope
- **Sintassi1:**
 - `<c:if test="espressioneBooleana" var="var" [scope="page|request|session|application"]/>`
- **Sintassi2:**
 - `<c:if test="espressioneBooleana">`
ElementiJSP
 - `</c:if>`

Scope [ambito di visibilità] (1)

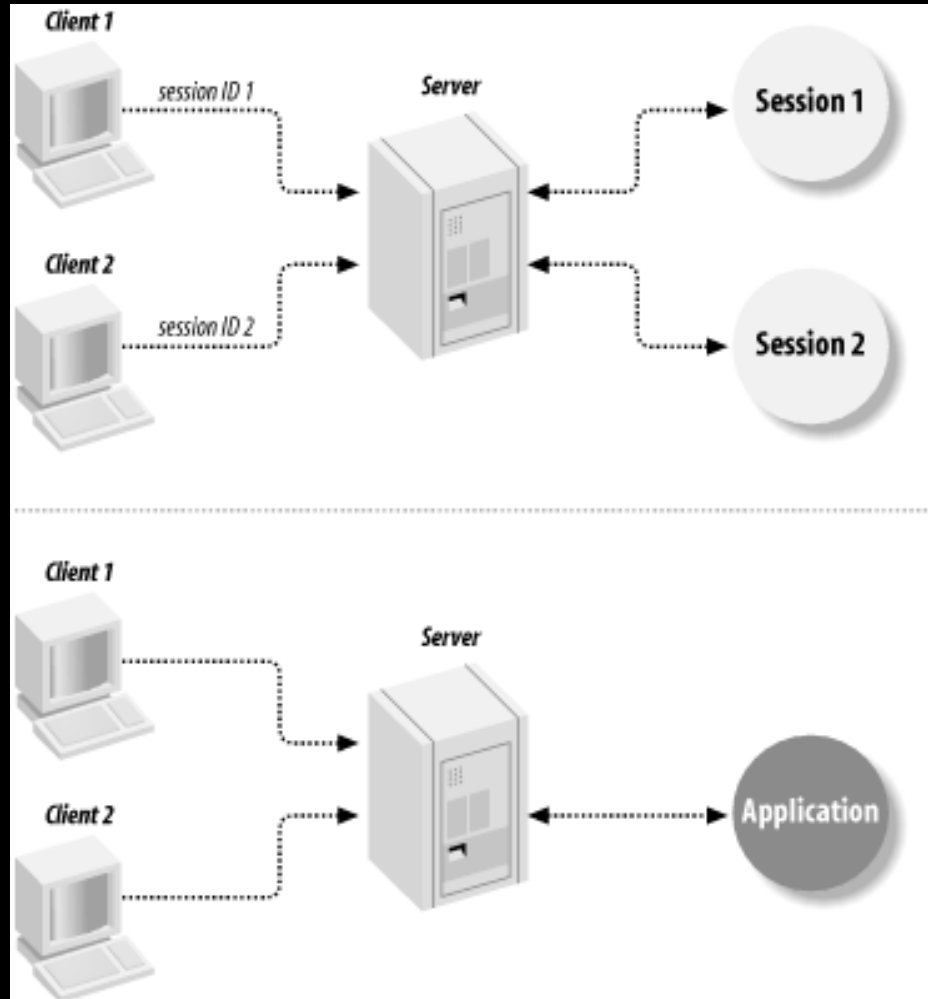
- Lo *scope* definisce la durata di un oggetto per la condivisione dei dati tra le pagine, indicando se è disponibile solo a un utente oppure a tutti gli utenti.
- Sono definiti i seguenti scope:
 - **Pagina** (*page*)
 - **Richiesta** (*request*)
 - **Sessione** (*session*)
 - **Applicazione** (*application*)

Scope [2]



Esempio 1

Codice



Esecuz

Azione <c:when>

- Rappresenta una delle alternative mutuamente esclusive di un blocco <c:choose>
- Sintassi:
 - <c:when test="espressioneBooleana">
ElementiJSP
 - </c:when>

Azione <c:choose>

Codice

- Consente l'elaborazione della prima azione <c:when> con il test che vale true; se non ce n'è nessuna, l'elaborazione passa a <c:otherwise>
- Sintassi:

<c:choose>

<c:when ...> <%-- prima when --%>

-

</c:when>

-

<c:when ...> <%-- seconda when --%>

-

</c:when>

-

<c:otherwise> <%-- altrimenti --%>

-

</c:otherwise>

- </c:choose>

Esecuz

Azione <c:set>

- Imposta una variabile di scope o una proprietà di un oggetto sul valore di un'espressione
- **Sintassi:**
 - `<c:set value="espressione" var="var" [scope="page|request|session|application"] />`

Azione <c:forEach>

- Valuta il corpo un numero di volte prefissato oppure per ciascun elemento di una collezione
- **Sintassi 1**(*collection*):
 - `<c:forEach items="collection" var="var">`
ElementiJSP
`</c:forEach>`
- **Sintassi 2**(*numero prefissato*):
 - `<c:forEach begin="startIndex" end="endIndex"`
`[step="increment"] var="var" >`
ElementiJSP
`</c:forEach>`

Azioni JSTL [XML]



XML è un formato di interscambio dei dati strutturati tra piattaforme differenti

Azioni JSTL [Sql]

- `<sql:setDataSource>`
- `<sql:query>`
- `<sql:param>`
- `<sql:update>`

Azione <sql:setDataSource>

- Imposta la variabile di configurazione dell'origine dati o salva le informazioni relative in una variabile di scope. Usata per prototipi di applicazioni piccole e semplici
- **Sintassi:**
 - `<sql:setDataSource [dataSource=origineDati | url=url [driver=nomeClasseDriver] [user=nomeUtente] [password=password]] [var=var] [scope=page|request|session|application] />`

Azione <sql:query>

- Esegue un'istruzione SQL SELECT e salva il risultato in una variabile di scope. L'istruzione può contenere un punto interrogativo (?) come segnaposto di valori assegnati da <sql:param> nidificate
- Sintassi:
 - `<sql:query sql="istruzioneSQLSelect" [dataSource="origineDati"] [maxRows="massimoNumeroRighe"] [startrow="index"] var="var" [scope="page|request|session|application"] />`

Azione <sql:param>

Codice

- Si utilizza come azione nidificata di <sql:query> e <sql:update> per assegnare un valore a un segnaposto
- **Sintassi:**
 - <sql:param value="valoreParametro" />

Esecuz

Azione <sql:update>

- Esegue un'istruzione SQL INSERT, UPDATE, DELETE. Esegue l'istruzione CREATE (DDL)
- Sintassi:
 - <sql:update sql=*istruzioneSQL* [dataSource=*origineDati*] [var=*var*] [scope=*page|request|session|application*]] />

Schema di autenticazione con JSP



Apri demo

Schema di autenticazione con JSP



Apri pdf

Database example (MySQL)

The screenshot displays the MySQL Control Center 0.9.2-beta interface. On the left, a tree view shows the database structure under 'MySQL Servers' > 'MySQL' > 'Databases' > 'example' > 'Tables', with 'employee' selected. The main area shows the 'employee' table schema:

Field	Type	Null	Key	Default	Extra
UserName	varchar(50)		PRI		
Password	varchar(50)	YES			
FirstName	varchar(50)	YES			
LastName	varchar(50)	YES			
Dept	varchar(50)	YES			
EmpDate	date	YES			
EmailAddr	varchar(50)	YES			
ModDate	datetime	YES			

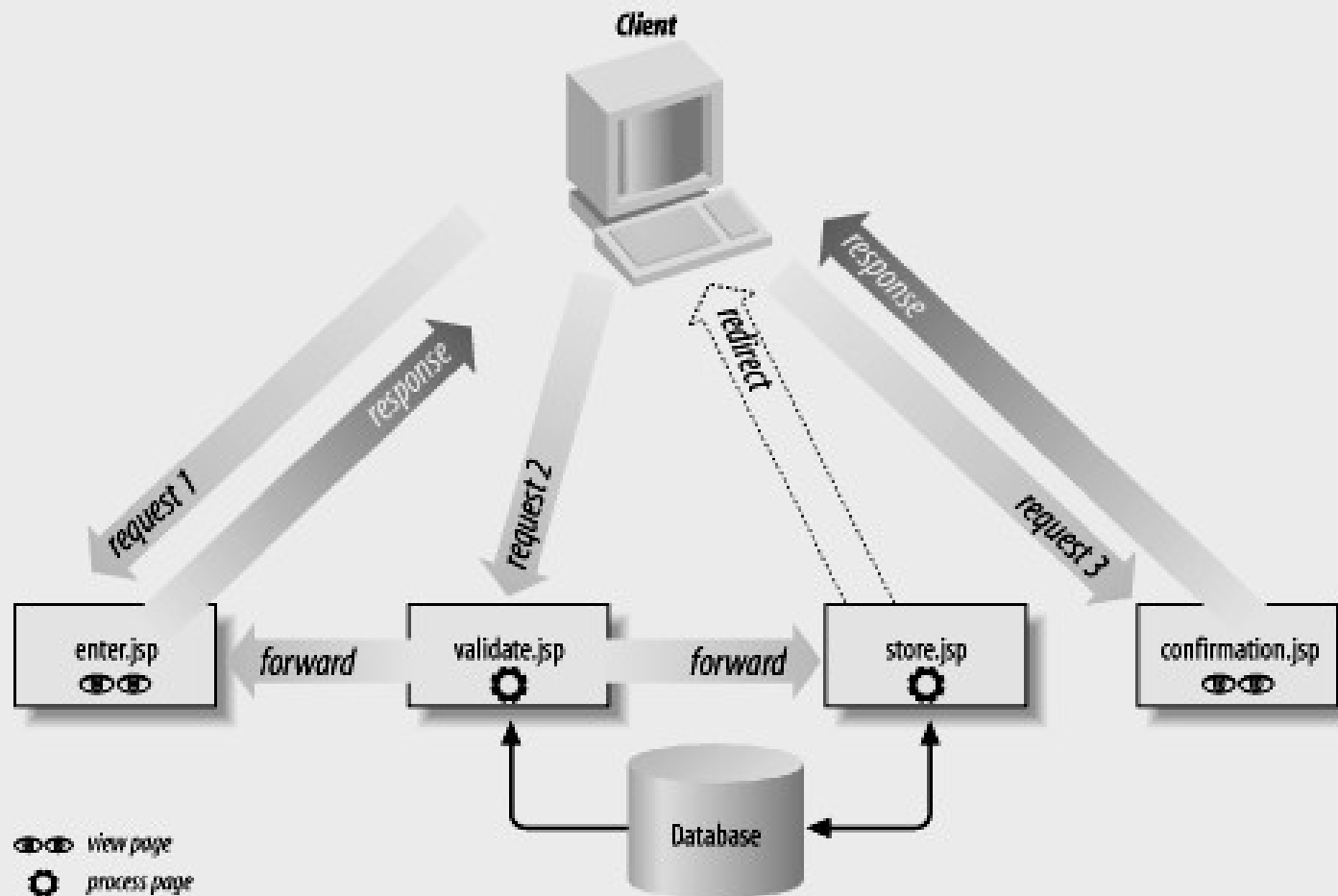
Below the schema, a query window shows the results of a query on the 'employee' table:

UserName	Password	FirstName	LastName	Dept	EmpDate	EmailAddr
ciccio	ciccio	ciccio	formagio	inf	[NULL]	[NULL]
mcozzetto	ciccio	maurizio	cozzetto	inf	0000-00-00	ciccio@ciccio.
dcurtis	dcurtis	antonio	de curtis	INF	1998-02-28	dcurtis@toto.it
mauro4	dfgdfg	dfgf	dfgdfg	INF	2004-09-03	mauro6@maur
ffff	ffff	fff	ffff	ffff	2003-02-01	fff@ffff.it
fgfgfg	fgfg	gffg	reffd	dfdfdd	2004-09-08	dfdf@fdfdfdd.r
ghghgh	ghgh	ghgh	ghg	fgh	2004-09-01	ffkfk@fjff.it
ggggf	hhh	hhh	jjj	yuyu	2004-10-01	ddd@hgddd.g

The query window also shows a status bar indicating '8 rows in set (0.00) sec' and a 'Messages' tab with 'Executing Query Read Only'.

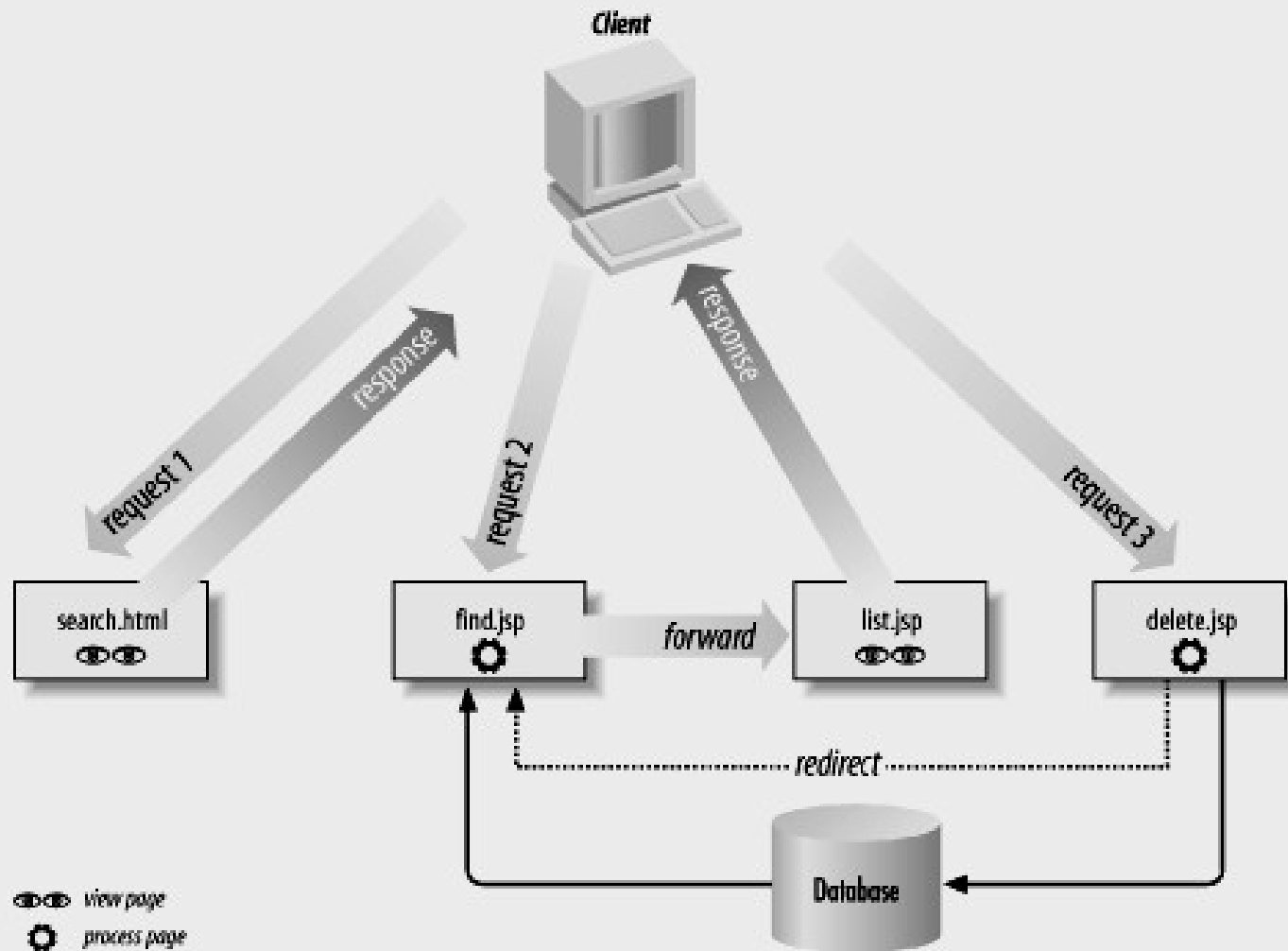
Registrazione dipendenti

Esecuz



Ricerca

Esecuz



Approfondimenti

- Java ServerPages – 3rd edition
 - *Hans Bergsten, O'Really-HopsLibri*
- Web Application in Java
 - *M. Bigatti, Ed. Master (IoProgrammo)*
- JSP-Guida alla programmazione
 - *C. Pelliccia, Ed. Master (IoProgrammo)*
- Java, Manuale pratico
 - *Mokabyte, www.mokabyte.it*
- VisualBuilder.com
 - *www.visualbuilder.com*