

Java Server Pages

Adolfo Bulfoni

12/5/2004

Adolfo Bulfoni

1

Scaletta

- La tecnologia JSP
- Sintassi JSP
- Componenti riutilizzabili: JavaBean
- Collegamento a database: JDBC

Adolfo Bulfoni

2

Scopi e utilizzi tipici delle JSP

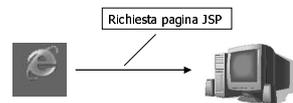
- Scopo della tecnologia: realizzare pagine Web dinamiche e personalizzabili
- Utilizzo: siti web dal contenuto dinamico...
 - siti di e-commerce
 - portali
 - forum
 - ...In genere qualsiasi sito che voglia rendere dinamico l'ambiente (magari anche in funzione dell'utente)
- Lo scopo è lo stesso delle Servlet, le circostanze d'utilizzo cambiano

Adolfo Bulfoni

3

Caratteristiche principali -1-

- Pagina JSP = HTML + codice Java inserito in tag specifici, di fatto "immerso" nel codice HTML
- Come funziona:
 1. Il browser del Client richiede un URL, che corrisponde a una pagina JSP



Adolfo Bulfoni

4

Caratteristiche principali -2-

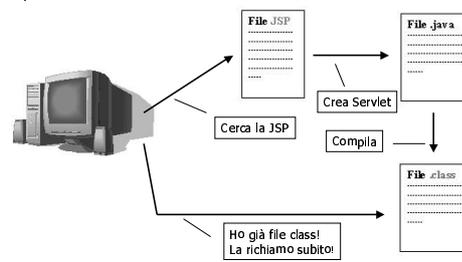
2. Il Server web riceve la richiesta
3. Traduce tale file in una Servlet Java (file .java)
4. Compila la classe Java creando la servlet (file .class)

- Nota importante:
il processo viene svolto 1 volta sola! Se il file .jsp non viene modificato, non serve ri-creare la servlet, ce l'ho già!
(vedi frecce rosse e nere dello schema successivo)

Adolfo Bulfoni

5

Caratteristiche principali -3-

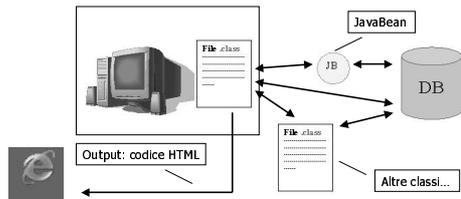


Adolfo Bulfoni

6

Caratteristiche principali -4-

- 5. Il Server esegue la Servlet la quale spesso interagisce con DB ed altre componenti Java



Adolfo Bulfoni

7

Dalle Servlet alle JSP

- Servlet: classi java con codice HTML interno, spedito in output al browser:

```
out.write("<html>\r\n<head>\r\n<title>\r\nJava Server  
Pages e  
JavaBean\r\n</title>\r\n</head>\r\n<body>\r\n\r\n")
```

- JSP: file HTML con segmenti di codice Java interni

Numero indovinato dopo <%=contatore.getNumGuesses()%>
tentativi

Adolfo Bulfoni

8

Scaletta

- La tecnologia JSP
- Sintassi JSP
- Componenti riutilizzabili: JavaBean
- Collegamento a database: JDBC

Adolfo Bulfoni

9

Sintassi JSP

- Direttive
- Elementi di script
 - Dichiarazioni
 - Espressioni
 - Scriptlet
- Azioni

Adolfo Bulfoni

10

Sintassi JSP: direttive

- Controllano la struttura e forniscono informazioni globali di un'intera pagina JSP
- Sintassi:
<%@ direttiva attributo= "valore" %>
dove:
direttiva e attributo sono predefiniti dal linguaggio:
direttiva assume i valori page, include, taglib
attributo assume valori che dipendono dal tipo di direttiva.
- Esempi:
<%@ page import = "miopkg.miaclasse" %>
<%@ include file = "miofile.html" %>

Adolfo Bulfoni

11

Sintassi JSP: elementi di script

- Contengono qualsiasi tipo di comando Java.

1. Dichiarazioni

```
<%=! String name = new String("Bob"); %>  
<%=! public String getName(){return name;} %>
```

- Dichiarazioni di metodi, variabili di classe...
- Vengono richiamati dall'interno di Espressioni e Scriptlet (vediamo ora cosa sono...)

Adolfo Bulfoni

12

Sintassi JSP: elementi di script -2-

2. Espressioni

- ```
<%= getName() %>
```
- Cosa posso inserire? Qualsiasi espressione Java valutabile e convertibile in stringa
  - Esempio:

```
<html><body>
<%= String name = new String("Bob"); %>
<%= public String getName(){return name;} %>
<p>Ciao <%= getName() %> </p>
</body></html>
```

Adolfo Bulfoni

13

## Sintassi JSP: elementi di script -3-

### 3. Scriptlet

- Includo porzioni di codice Java all'interno di pagine HTML
- Sintassi: `<% codice java %>`
- Esempio:

```
<%= int random; %>
...
<% for(int n = 0; n < 10; n++){
 random = Math.abs(new Random().nextInt()*100) + 1;
 %>
 Numero casuale: <%= random %>

<% } %>
```

Adolfo Bulfoni

14

## Sintassi JSP: le azioni

- Sono tag XML che vengono convertiti in comandi Java
- Eseguono operazioni comuni senza scrivere in Java
- Le più importanti..

```
- <jsp:include page="miadir/miapagina.jsp" />
- <jsp:forward page="miadir/paginaTarget.jsp" />
```
- A che servono?
  1. `<jsp:include/>` permette inclusione dinamica di pagine statiche o dinamiche
  2. `<jsp:forward />` inoltra la richiesta a una nuova pagina
- Usate anche nei JavaBean

Adolfo Bulfoni

15

## Esempio: contatore accessi

- Dov'è? `../webapps/ROOT/counter.jsp`
- Codice:

```
<html>
<head>
<title>Contatore</title>
</head>
<body>
<%= int contatore = 0; %>
<h4>Numero di accessi alla pagina: <%= ++contatore %>
</h4>
</body>
</html>
```

Adolfo Bulfoni

16

## Servlet generata: metodo \_jspService -1-

```
public final class counter_jsp extends
org.apache.jasper.runtime.HttpJspBase
implements org.apache.jasper.runtime.JspSourceDependent
{
int contatore = 0; /* variabile contatore */

public void _jspService(HttpServletRequest request,
HttpServletResponse response)
throws java.io.IOException, ServletException {
/* Inizializzazioni varie (non ci interessa) */

try {
/* altre inizializzazioni che non ci interessano ora... */
out.write("<html>\r\n");
out.write("<head>\r\n");
out.write("<title>Contatore</title>\r\n");
```

Adolfo Bulfoni

17

## Servlet generata: metodo \_jspService -2-

```
out.write("</head>\r\n");
out.write("\r\n");
out.write("<body>\r\n");
out.write("\r\n");
out.write("\r\n");
out.write("<h4>Numero di accessi alla pagina: ");
out.print(++contatore);
out.write("</h4>\r\n");
out.write("</body>\r\n");
out.write("</html>\r\n");
} // blocco catch - finally... niente di interessante
} //chiuso _jspService
} // chiusa la Servlet
```

Adolfo Bulfoni

18

## Confronto: Servlet vs JSP

- Servlet: orientate ai contenuti (codice Java)
  - JSP: orientate alla presentazione (codice HTML)
  - Vantaggi di JSP: si hanno se non devo eseguire operazioni logiche complesse
1. Facile scrivere e mantenere codice HTML
  2. Posso usare i tools di sviluppo per pagine web che conosco già
  3. Posso dividere il team di sviluppo in programmatori Java e sviluppatori web
- In progetti normali vengono usate entrambe, proprio a causa delle loro diverse funzionalità

Adolfo Bulfoni

19

## Scaletta

### Visto:

- La tecnologia JSP
- Sintassi JSP

### Approfondimenti:

- Componenti riutilizzabili: JavaBean
- Collegamento a database: JDBC

Adolfo Bulfoni

20

## JavaBean

- Cosa sono: classi java di utilità che vengono richiamate nei file .jsp
- Hanno una struttura particolare:
  1. Costruttore con zero parametri
  2. Non ha variabili pubbliche d'istanza
  3. Possiede metodi pubblici che iniziano con get e set, (chiamati "getters" e "setters") che servono per accedere alle sue proprietà

Adolfo Bulfoni

21

## JavaBean -2-

- Esempio di Bean:

```
public class MioBean{
 private String cognome = new String("");

 public MioBean(){}

 public String getCognome() {
 return this.cognome;
 }

 public void setCognome(String cognome){
 this.cognome = cognome;
 }
}
```

Adolfo Bulfoni

22

## JavaBean -3-

- Come vengono usati:
  - Compilati dall'utente come i normali file Java
  - Inseriti come tutte le classi in ..webapps/ROOT/WEB-INF/classes/
  - Inizializzati nella JSP col comando XML:

```
<jsp:useBean id="mioBean" class="miopkg.MioBean" />
```
  - di fatto viene eseguito il costruttore con zero parametri del Bean

Adolfo Bulfoni

23

## JavaBean -4-

- Le loro proprietà vengono richiamate in 2 modi:
  - tramite sintassi Java:

```
<%= mioBean.getCognome(); %>
<%= mioBean.setCognome(); %>
```

- tramite sintassi XML (cioè usando le azioni viste prima):

```
<jsp:getProperty name="mioBean" property="cognome" />
<jsp:setProperty name="mioBean" property="cognome"
value="Bob" />
```

Adolfo Bulfoni

24

## JavaBeans -5-

- Perché non usare classi java normali?  
Il programmatore web cerca di evitare di scrivere codice Java (lo conosce poco); richiama/imposta le proprietà dei bean con:  

```
<jsp:getProperty name="mioBean" property="cognome" />
<jsp:setProperty name="mioBean" property="cognome"
value = "Rossi" />
```
- Ulteriori vantaggi dei JavaBean
  1. Possono essere utilizzati facilmente da più pagine JSP
  2. Permettono interazione trasparente col database

Adolfo Bulfoni

25

## Scaletta

Visto:

- La tecnologia JSP
  - Sintassi JSP
  - Componenti riutilizzabili: JavaBean
- Ultimo argomento:
- Collegamento a database: JDBC

Adolfo Bulfoni

26

## JDBC -1-

- Java DataBase Connectivity: interfaccia completamente Java utilizzata per la connessione a database.
- Apertura connessione:
  1. Carico il driver Jdbc:  

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```
  2. Connessione: specifico l'URL del DB e nome (più eventualmente username + password)  

```
Connection conn = DriverManager.getConnection
("jdbc:odbc:MioDatabase","admin","miapassword");
```

Adolfo Bulfoni

27

## JDBC -2-

- Operazioni sul database.
  1. Creo oggetto Statement  

```
Statement s = conn.createStatement();
```
  2. Uso uno dei metodi di Statement per
    - a. eseguire una query
    - b. eseguire un'operazione di update/delete/insert
  3. Se ho eseguito una query, scandisco tramite l'oggetto ResultSet il risultato, riga per riga
- Chiusura della connessione

Adolfo Bulfoni

28

## Esempio: connessione e query

```
package jspdb;
import java.sql.*;
public class Database {
 Connection conn;
 private String sql = "SELECT * FROM clienti";
 public Database(){
 try{
 Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
 conn = DriverManager.getConnection("jdbc:odbc:Azienda");
 }catch(Exception e){}
 }
 public ResultSet getResult(){
 try{
 Statement s = conn.createStatement();
 ResultSet r = s.executeQuery(sql);
 return r;
 }catch (Exception e){}
 return null; }
}
```

Adolfo Bulfoni

29

## Esempio: connessione e query -2-

```
... ..
<%@ page import = "java.sql.*, jspdb.Database" %>
... ..
<% Database d = new Database();
 ResultSet r = d.getResult(); %>
... ..
<% while(r.next()){ %>
 <tr>
 <td> <%= r.getString("Nome")%> </td>
 <td> <%= r.getString("Cognome")%> </td>
 <td> <%= r.getString("Città")%> </td>
 </tr>
 <%}%>
```

Adolfo Bulfoni

30

## Riassunto

Di cosa abbiamo parlato:

- JSP:
  1. Come funzionano
  2. La sintassi base + esempio
  3. Differenze con le Servlet
- Approfondimenti:
  1. JavaBean
  2. JDBC + esempio

Adolfo Buffoni

31

## Bibliografia

- JSP – Java Server Pages, Guida di riferimento  
*J. Goodwill* - APOGEO - 2000
- Core Servlets and JavaServer Pages (vol.1)  
*M. Hall, L. Brown* - Prentice Hall - 2003
- Articoli:
  - Sviluppare applicazioni web con Java *L. Balzerani* su n.2 di "<tag/>" - 6/2003
- Web:
  - Corso JSP su <http://www.html.it>

Adolfo Buffoni

32