

JAVA Server Side: Web & Other

Ing Cesare Monti 5 maggio 2004

1

Cosa vedremo

- JSP e JAVA
- Servlet
- Beans
- Legacy
 - Socket
 - JDBC
- JSP & Application Container

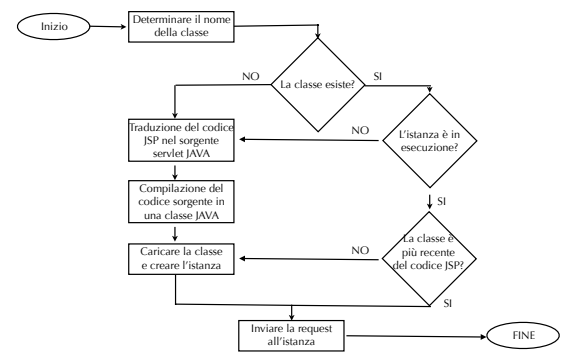
2

JSP e JAVA

- come già detto, JSP presenta tag che eseguono codice,
- come è possibile ?

3

JSP e JAVA



4

JSP e JAVA

- quindi? io scrivo jsp o java?
 - in realtà io scrivo jsp, che viene tradotto in java dal container in cui le piazzo
 - Tomcat non fa altro che creare servlet senza che l'utente se ne accorga
 - SONO APPLICAZIONI SERVER SIDE!
 - nello specifico di Tomcat esiste una servlet principale (detta ROOT con molta fantasia) che viene eseguita di default ... ma si può benissimo dire a Tomcat di eseguire e pubblicare altre servlet oltre a questa

5

JSP e JAVA

- ... quindi scrivo servlet?
 - sì ... ma in maniera trasparente
 - ... me ne astraggo!
 - no! scrivo java che viene trasformato in servlet
 - entrambe le risposte sono accettabili
 - posso scrivere servlet e farle interagire con classi e jsp
 - così riutilizzo codice
 - ... ma prima devo capire bene cosa fa jsp e cosa fa una servlet

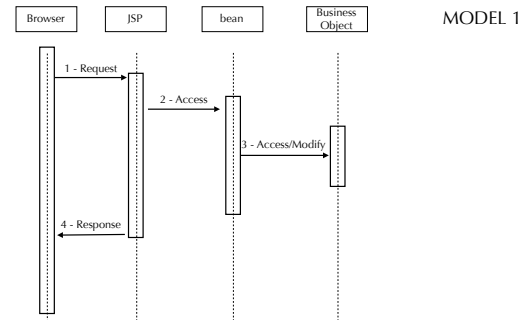
6

JSP e JAVA

- JSP mi serve principalmente per:
 - presentare il risultato di un elaborazione
 - interagire con classi
 - interagire con servlet
 - interagire con i beans
- dove sta il vantaggio
 - Model 1
 - JSP, Beans
 - Model 2 (MVC)
 - JSP, Servlet, Beans

7

JSP e JAVA



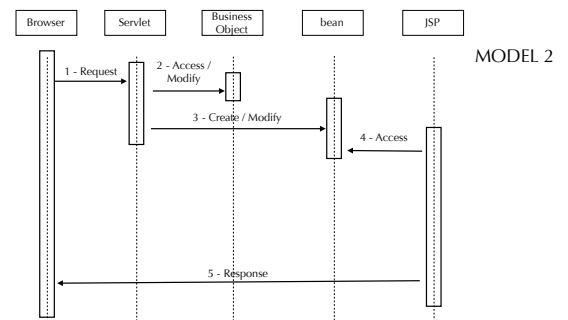
8

JSP e JAVA

- PRO:
 - si riducono le dipendenze tra le pagine jsp e gli oggetti business
 - si possono sviluppare pagine in parallelo
- CONTRO:
 - non è sempre separata la parte View dal Model
 - gli "sviluppatori jsp" non sono svincolati da quelli java

9

JSP e JAVA



10

JSP e JAVA

- PRO:
 - giusto livello di astrazione
 - ... che permette un largo riutilizzo del codice
 - possibilità di "spezzare" la realizzazione
- CONTRO:
 - ... bisogna progettare bene il sistema prima di partire

11

JSP e SERVLET

- e allora le servlet come sono fatte?
 - anzitutto fanno parte di J2EE
 - e quindi siamo più in alto di java Standard Edition
 - ... sono "componenti Web" ... (definizione della SUN)
 - ... manipolati da contenitori (tipo Tomcat), ... che generano contenuti dinamici

12

JSP e SERVLET

- le Servlet JAVA trattano come oggetti:
 - sia la Request che la Response
- e non solo...
 - nascono con il concetto di stato legato ad HTTP superando il limite delle CGI
 - restano flessibilmente aperte per essere modificate per produrre qualsiasi altra cosa che possa diventare Request e Response
 - un filtro su una Request potrebbe cambiare il protocollo e diventare un firewall
 - un filtro su una Response potrebbe cambiare il tipo di oggetto in risposta

13

JSP e SERVLET

- Request
 - si trattano di qualsiasi method siano astruendo da questo
 - si estrae agilmente qualsiasi dato
 - ... si può lavorare in streaming
 - pensate ad un upload di un file grosso
 - si trattano i cookies
 - si può lavorare sui parametri ssl
 - si lavora su line, header, body

14

JSP e SERVLET

- Response
 - si lavora su Header Line, Header Field, Entity Body
 - si lavora in streaming
 - pensate al download di un file grosso
 - si reindirizza la risposta
 - si aggiunge SSL
 - ...

15

JSP e SERVLET

- quindi potrei fare a meno di scrivere codice JSP?
 - ... spesso dimezzo i tempi se scrivo JSP invece che Servlet
 - ... ed ottengo la stessa cosa
- ma in alcuni casi mi conviene utilizzarle
 - ovunque ci sia bisogno di un MODEL 2
 - ovunque io voglia fare MVC

16

JSP e BEANS

- *"Any Java class that adheres to certain property and event interface conventions can be a Bean."* (SUN - 1997 - JavaBeans Technology Survey)
- Dato il nome i beans
 - (la tecnologia si chiama Enterprise Java Beans)
- fanno parte di J2EE
 - e sono quindi applicazioni server side

17

JSP e BEANS

- posso utilizzarli
 - sempre sulla macchina server
 - per sincronia
 - per disciplinare attività
 - per comunicazione
 - ...

18

JSP e BEANS

- o distribuirli tramite container
 - a livello di modello non sono diversi dagli altri
 - basta pensare alle funzionalità che hanno i beans e paragonarle nel contesto web
- la differenza la fa il tipo di bean che utilizziamo
 - session
 - entity
 - message
 - ...

19

JSP e LEGACY

- in questa sezione spendiamo due parole su alcune delle applicazioni java che hanno molto “senso” nel contesto web
 - socket
 - JDBC

20

JSP e LEGACY

- abbiamo già detto che si può riutilizzare qualsiasi classe dentro ad una pagina jsp
 - basta dichiararla globale
 - `<%! ..... %>`

21

JSP e LEGACY

```
<%!  
public class quadrato{  
    double lato;  
    public quadrato (double L){ lato = L;}  
    public double area() {return lato*lato;}  
}  
%>  
.....HTML.....  
<% quadrato quad1 = new quadrato(request.getParameter("LatoQuadrato")) ;  
double AreaQuad1 = quad1.area();  
%>
```

22

JSP e LEGACY

- ... quindi posso utilizzare qualsiasi cosa
 - primo fra tutti le socket

23

JSP e SOCKET

- Le socket (con tutti i pro e contro) rappresentano una possibilità di aumentare la granularità di distribuzione del nostro sistema
 - nel mondo jsp possiamo utilizzare le socket per comunicare con processi remoti e non.
 - nessuno mi vieta di aprire dinamicamente
 - socket
 - serverSocket
 - attenzione all'uso, leggere attentamente le istruzioni del caso !!!!
 - e di recuperare informazioni da queste

24

JSP e SOCKET

- Socket...
- import java.net.*;
import java.io.*;

```

public class Client {

    public static void main (String arg[]) {
        /*vuole ip porta arg?*/
        try {Socket s = new Socket (arg[0] , Integer.parseInt(arg[1]));
            System.out.println("connesso");

            /*%i costruiamo l'acquisizione s f la Socket*/
            InputStreamReader is = new InputStreamReader(s.getInputStream());
            BufferedReader q = new BufferedReader(is);

            OutputStreamWriter os = new OutputStreamWriter(s.getOutputStream());
            BufferedWriter wq = new BufferedWriter(os);

            /*%i write vuole argomento e lunghezza*/
            eq.write(arg[2],0,arg[2].length());
            eq.newLine();
            eq.flush();
            System.out.println(q.readLine());
            s.close();
            System.out.println("disconnesso");
        } catch (IOException e) {
            System.out.println("e");
        }
    }
}

```

25

JSP e SOCKET

- ServerSocket
- import java.net.*;
import java.io.*;

```

public class Server {

    public static void main (String arg[]) {
        ServerSocket ss = null;
        try { ss = new ServerSocket(8080);
            catch (IOException e) {
                System.out.println("e");
            }
        } while (true) {
            try {Socket s = ss.accept();

                InputStreamReader is = new InputStreamReader(s.getInputStream());
                BufferedReader q = new BufferedReader(is);

                OutputStreamWriter os = new OutputStreamWriter(s.getOutputStream());
                BufferedWriter wq = new BufferedWriter(os);

                eq.write(q.readLine());
                eq.newLine();
                eq.flush();
                s.close();

            } catch (IOException e) {
                System.out.println("e");
            }
        }
    }
}

```

26

JSP e JDBC

- JDBC (Java DataBase Connectivity) rappresenta un metodo semplice ed efficace per accedere ai DB
- fisicamente sono un'insieme di interfacce a cui i DB devono attenersi
 - se vogliono lavorare in java
- sono dentro a J2SE

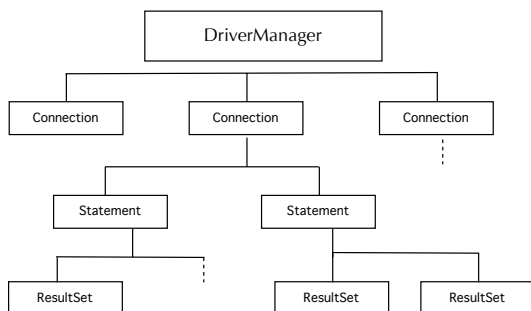
27

JSP e JDBC

- le API JDBC permettono la creazione di 4 tipi di "driver"
 - Tipo 1: JDBC + ODBC Bridge + Driver ODBC + proprietary DBMS access protocol
 - Tipo 2: JDBC + proprietary DBMS access protocol
 - Tipo 3: Driver Client + Driver Server + DBMS access protocol
 - Tipo 4: JDBC only
- in blu quelli "stand alone oriented"
- in rosso quelli "Network Oriented"
- solitamente ...nel distribuito ... si usano quelli di tipo 4

28

JSP e JDBC



29

JSP e JDBC

- solitamente ogni casa produttrice di DB fornisce la rispettiva implementazione delle classi conformi a JDBC
 - è vero che sono conformi a Java
 - ... non è detto che lo siano nei confronti di SQL
- sono una sorta di "driver" per l'utilizzo del db

30

JSP e JDBC

- JSP permette l'utilizzo di JDBC
 - e lo incentiva con i tag personalizzati

```
<database:query id="NomeTabella" scope="session">
SELECT * FROM NomeTabella WHERE id="1"
</database:query>
```
 - ovviamente ... occorre avere un'idea di come ragiona un DB!
 - connessioni DNSless or JNDI
 - connessioni persistenti e transazioni
 - meccanismi di roll-back
 - Tutto trova un suo tag ... ma bisogna sapere cosa si cerca

31

JSP e JDBC

- Negli ultimi periodi per tentare di risolvere i problemi di standardizzazione di SQL, qualcuno ha pensato di creare SQLite
 - un applicativo Server che:
 - ottimizza le performance dei vari DB
 - standardizza l'SQL per le query
 - ... è scritto in C/C++ ma ci si può interfacciare anche da JAVA via JNI
 - <http://www.sqlite.org/>

32

JSP e Application Container

- tiriamo un po' di somme:
 - dalla rete ci si è sempre aspettato grandi applicativi scalabili
 - ... questa è storia
 - grandi quantità di software "legacy" già pronto
 - tutta la parte JAVA fatta "stand alone"
 - dalle ditte vengono poche energie per ri-ingegnerizzare i servizi
 - finché la barca va ... lasciamola andare

33

JSP e Application Container

- grazie al Model 2 di JSP possiamo riuscire a recuperare tutto ma ...
 - chi scrive le Servlet?
 - Ognuno la sua?
 - o Standardizziamo anche quelle?

34

JSP e Application Container

sulla stessa scia di Tomcat e degli altri Servlet Engine sono nati diversi "Application Container" uno per tutti: JBoss

- JBoss ha standardizzato il Model 2 dando:
 - una piattaforma server-side per il legacy JAVA
 - uno standard per riutilizzare la "classi già fatte"
 - uno standard per presentare l'Output sul web

35

JSP e Application Container

- Attorno a JBoss ora ruotano progetti consistenti:
 - Content Management System
 - Nukes
 - ... lo stesso Tomcat ... leggermente esteso
 - JBossIDE on Eclipse
 - un tool di sviluppo per JBoss stesso
 - JBossCache
 - piattaforma per il GRID Computing

36

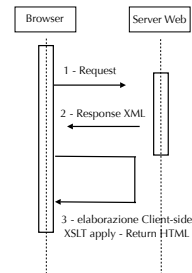
JSP e Application Container

- e xml?
 - ... oramai è ovvio che esistano tag per manipolare XML
 - il problema emerge quando cerchiamo di trasformarlo
 - magari con XSLT

37

JSP e Application Container

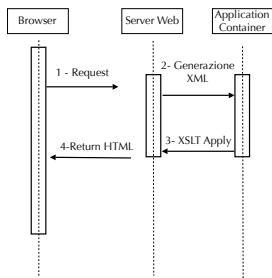
- al momento XSLT viene interpretato dal browser



38

JSP e Application Container

- Nulla vieta di fare così:



39

JSP

- that's all folks!
- any question ?

40