Programmazione per il Web
Note per la lezione del 23/03/2016

Igor Melatti

Pagine JSP

- slide 219–220: diversi tipi di commenti, possibilità di usare direttamente il valore delle variabili, valgono gli if delle scriptlet, come se l'HTML fosse tutto in out.println; alcune variabili (come request e response; non ridenominabili come succedeva con le servlet) sono implicitamente definite, se ne possono definire altre a piacimento
  - quindi non è detto che tutto quello che c'è in una pagina JSP finirà poi nell'HTML mandato al client...
- slide 221: stessa pagina JSP, ma dinamicamente il contenuto (ciò che viene mandato al client) è diverso a seconda che l'informazione sullo username sia stata mandata oppure no. Se il metodo fosse stato POST, non ci sarebbe stata differenza nella barra degli indirizzi...
- slide 222: estensione "ovvia": se la pagina HTML risultante è lunga, scriverla con gli out.println è ovviamente error-prone
- JSP container: include il servlet container. In pratica, è il servlet container "wrappato" in modo da gestire le pagine JSP come descritto qui
  - da notare che la prima esecuzione assoluta ci mette quindi di più, proprio a causa della compilazione; di solito, i programmatori compiono anche la prima visualizzazione...
- slide 224: avendo request, ovviamente si hanno le informazioni della richiesta; meno ovvio che si possano avere le informazioni sul contesto
- slide 225: oltre a ciò, ovviamente, vanno considerate le parti di HTML "puro" (tecnicamente: *template text*); nell'esempio di slide 219-220 c'erano solo gli scriptlet (e un'espressione scriptlet). Una partizione leggermente diversa (e più precisa) è presentata nel libro di testo (pag. 233–234), dove i costrutti JSP sono strutturati come segue:
  - Scripting elements, a loro volta divisi in

    - scriptlet (tra `<%` e `%>`)
    - espressioni (tra `<%=` e `%>`)
    - dichiarazioni (tra `<%!` e `%>`)
  - Azioni predefinite (o standard)
    - non sono molte: `jsp:include`, `jsp:forward`, `jsp:setProperty`, `jsp:getProperty`, `jsp:useBean` sono le più importanti
  - Direttive
    - sono solo tre: `page`, `include`, `taglib` (quest'ultima serve per le azioni personalizzate)

- slide 227: le graffe sono ad indicare la ripetizione (nel senso che si possono specificare più attributi)
- slide 231: azioni come funzioni; alcune predefinite, altre da definire (personalizzate)
- slide 232: niente `doGet` o `doPost`; ma c'è il metodo `getMethod` di `HttpServletRequest`...
- slide 234: errori nella traduzione si manifestano al primo uso e non se ne vanno più, a meno di non correggere la pagina JSP
- le traduzioni delle pagine JSP sono dentro la directory `work`, sorella di `webapps` (notare che per le servlet non c'è nessun codice...)

Esempi di Pagine JSP

- Dove può essere messa una servlet: ovunque, purché dentro un context root e non in una cartella `WEB-INF` (con qualunque case)
- `count.jsp`: come si dichiara un membro della classe; essenzialmente, con le pagine JSP si può ricostruire quasi ogni cosa delle servlet
  - "quasi" perché comunque ci sono delle cose fissate, come le righe di `_jspService` prima della `out.write`...
  - poco male, tanto sono righe standard
  - si consideri la classe generata (che dovrebbe trovarsi, almeno sotto Linux, in un path come questo: `work/Catalina/localhost/20160323/org/apache/jsp/count_jsp.java`)
  - `accessCount` è sia un membro (privato) della classe, in quanto dichiarato con `<%!`, che una variabile locale del metodo `_jspService` (che fa le veci di `doGet` e `doPost`), in quanto dichiarata in una scriptlet

– trattandosi di una variabile “globale”, ovvero, comune a tutti i thread che vengono invocati ogni volta che qualche client richiede la pagina JSP, gli incrementi sono una sezione critica e vanno protetti con un blocco `synchronized`

– in questo modo è possibile contare quanti utenti hanno fatto accesso ad una pagina (contatore degli accessi)

– tuttavia, facendo refresh sulla pagina su un browser, si nota che il contatore rimane sempre 1: come occorre correggere le ultime 2 righe di `count.jsp` per risolvere il problema?