

Programmazione per il Web

Note per la lezione del 13/05/2015

Igor Melatti

Pagine JSP: Chiarimenti sui tag che modificano il proprio corpo

- `test.jsp` e `IteratorTag.java`: mostra come “evolve” l’oggetto `BodyContent` nel corso delle chiamate
 - essenzialmente, ha una “memoria” interna, immaginabile come una stringa: *dopo* l’esecuzione di `doInitBody`, e ogni volta che `doAfterBody` *ritorna* `EVAL_BODY_AGAIN`, questa stringa conterrà la *valutazione* del codice JSP che costituisce il corpo del tag, *concatenata* al precedente valore della stringa stessa (“Trascrizione del corpo” in slide 428)
 - è per questo che si fa tutto questo marchingegno delle chiamate “iterative” di `doAfterBody`
 - infatti, in alternativa, si sarebbe potuto fare come in `test.simple.jsp` e `IteratorTagSimple.java`, dove `doAfterBody` viene chiamata una volta sola (ha un solo `return SKIP_BODY`)
 - qui, la valutazione del corpo avviene una volta sola, ma solo perché viene chiamato il metodo `getString` (riga 20 di `IteratorTagSimple`): non c’è nessuna concatenazione con valori precedenti (e infatti, la concatenazione è gestita esplicitamente con `writeThis`)
 - fatto così, non sarebbe possibile passare informazioni ogni volta diverse tramite beans, come succede invece in `test.jsp`
 - in questo esempio semplice, in cui l’informazione è solo una stringa, si può fare come in `test.simple.jsp`; ma nel caso in cui le informazioni fossero molte e organizzate gerarchicamente, questa via non sarebbe praticabile
 - inoltre: il tag dovrebbe essere per uso generico, e se un programmatore volesse scrivere `REPL_ITEM` senza farlo sovrascrivere?
 - infine, `test.simple.bis.jsp` e `IteratorTagSimpleBis.java` mostrano che la comunicazione tramite beans sarebbe ancora

possibile, ma una volta sola (sfruttando `do InitBody`), mentre le chiamate `pageContext.setAttribute("item", o);` dentro `doAfterBody` non hanno effetto

Autenticazione (fino a slide 483)

- Esempio 1: da pag. 403 in poi, è un esempio di autenticazione programmata (il programmatore scrive del codice per gestire l'autenticazione)
 - il programmatore scrive del codice per gestire l'autenticazione
 - qui l'autenticazione è fatta realizzando un tag personalizzato, che è probabilmente il modo più comodo (direttamente usabile da altri programmatori) ma si può fare in altri modi
 - da esempio, il codice per l'autenticazione può essere messo dentro un file di cui si fa l'`include`
- slide 439: problema generale dell'autenticazione
- slide 440: anziché programmare l'autenticazione come sopra, si possono sfruttare delle funzionalità offerte dai servlet/JSP container (*autenticazione dichiarativa*)