

Programmazione per il Web

Note per la lezione del 04/05/2016

Igor Melatti

I Tag (o Azioni) Personalizzati

- Slide 421–438: tag che modificano il corpo
 - questo tipo di tag serve quando il contenuto del corpo può contenere delle azioni personalizzate (esempio del libro) o standard (uso di bean, tipicamente, esempio delle slide e nel progetto)
- Slide 422: vedere pagine di documentazione per TagSupport e BodyTagSupport
- Slide 425: `setBodyContent` è chiamato dal JSP container, praticamente impossibile che lo sviluppatore di tag personalizzati lo debba chiamare direttamente o scriverne il codice; anche `doInitBody` è chiamato dal JSP container, ma tipicamente il suo corpo viene riscritto dallo sviluppatore di tag personalizzati
- Slide 428: “trascrivere” il corpo significa valutarlo, *non* scriverlo nell’out
- `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 `doInitBody`), mentre le chiamate `pageContext.setAttribute("item", o)`; dentro `doAfterBody` non hanno effetto

- Esempi 14.16 – 14.21 dal libro