

Tecnologie di Sviluppo per il Web

Applicazioni Web J2EE Framework per il Modello 2 Struts

versione 3.0

Questo lavoro è concesso in uso secondo i termini di una licenza Creative Commons
(vedi ultima pagina)

G. Mecca – mecca@unibas.it – Università della Basilicata



Sommario

- Introduzione
- Modello 2 con Struts
- Differenza Principali con pinco
- Altre Funzionalità



Introduzione

○ Apache Struts

- ⇒ il framework industriale attualmente leader di mercato per lo sviluppo J2EE con modello 2
- ⇒ un progetto open source della Apache Software Foundation fondato da Craig McClanahan
- ⇒ disponibile gratuitamente sul sito struts.apache.org
- ⇒ attualmente alla versione 1.2



Introduzione

>> indovinam2struts

○ In sintesi

- ⇒ una implementazione del modello 2 con controllore frontale e controllo dichiarativo
- ⇒ basato sulla sequenza comando-azione-esito-schermo successivo
- ⇒ con alcune funzionalità aggiuntive (in particolare: varie forme di supporto alla convalida e alla costruzione delle viste)



Modello 2 con Struts

○ La terminologia di Struts

- ⇒ il controllore frontale si chiama ActionServlet (org.apache.struts.action.ActionServlet)
- ⇒ il servlet viene inizializzato in web.xml con vari parametri, tra cui il riferimento al file di configurazione struts-config.xml
- ⇒ possono esserci vari file di configurazione
- ⇒ i comandi hanno tipicamente estensione .do



Modello 2 con Struts

>> web.xml

○ Nota

- ⇒ Struts non prevede il concetto di azione iniziale e azione finale
- ⇒ per eseguire l'azione iniziale due modi
- ⇒ I: registrare un servlet in modo che sia eseguito il metodo init()
- ⇒ II: registrare un metodo per la gestione dell'evento di creazione del contesto dell'applicazione



Modello 2 con Struts

- Il file di configurazione
 - ⇒ tipicamente struts-config.xml
 - ⇒ contiene una serie di descrittori chiamati action-mapping
- Per ciascun descrittore
 - ⇒ comando (senza .do !)
 - ⇒ azione
 - ⇒ "action form"
 - ⇒ esiti ed inoltri ("forward")



Modello 2 con Struts

>> struts-config.xml

- I formbean
 - ⇒ si chiamano "action form"
- Una prima differenza significativa
 - ⇒ Struts offre supporto al processo di convalida
 - ⇒ gli action form vengono dichiarati nel file di configurazione e vengono manipolati automaticamente dal framework (>>)
 - ⇒ di conseguenza non è necessario crearli esplicitamente nelle azioni



Modello 2 con Struts

○ Le azioni di Struts

⇒ devono estendere la classe
`org.apache.struts.action.Action`

```
package org.apache.struts.action;
public class Action {
    public abstract ActionForward execute(ActionMapping mapping,
                                         ActionForm form,
                                         HttpServletRequest request,
                                         HttpServletResponse response)
        throws Exception;
}
```



Modello 2 con Struts

○ Il riferimento ActionMapping mapping

⇒ necessario perchè i metodi di esecuzione
delle azioni NON restituiscono stringhe

⇒ ma riferimenti di tipo ActionForward
utilizzando l'oggetto mapping

○ Esempio: in pinco: return "indovinato";

⇒ ActionForward indovinato =
 mapping.findForward("indovinato");
return indovinato;



Modello 2 con Struts

>> AzioneIniziaPartita

- Una tipica azione di struts
 - ⇒ preleva i parametri dal formBean (dopo aver fatto un cast)
 - ⇒ effettua le operazioni di controllo necessarie
 - ⇒ produce l'oggetto ActionForward corrispondente all'esito necessario e lo restituisce
- NOTA
 - ⇒ l'azione non si occupa della convalida



Modello 2 con Struts

- Nota
 - ⇒ il framework fornisce una serie di azioni predefinite per operazioni ricorrenti
 - ⇒ esempio: ForwardAction, che consente di spostarsi ad uno schermo senza eseguire logica applicativa

```
<action path="/esci"
        type="org.apache.struts.actions.ForwardAction"
        parameter="/schermoUscita.jsp">
</action>
```



Differenze Principali con pinco

- Le principali differenze
 - ⇒ la gestione del processo di convalida
 - ⇒ la produzione dei messaggi negli schermi
- In sintesi
 - ⇒ Struts automatizza gran parte del processo di convalida lato server
 - ⇒ Struts obbliga ad internazionalizzare i messaggi (es: messaggi di errore)



Differenze Principali con pinco

- La gestione dei formbean
 - ⇒ il contenitore istanzia il formbean
 - ⇒ popola le proprietà prelevando i parametri dalla query string
 - ⇒ chiama l'eventuale metodo di convalida
`public ActionErrors validate(ActionMapping mapping, HttpServletRequest request);`
 - ⇒ in caso di errori rimanda allo schermo di immissione, altrimenti esegue l'azione



Differenze Principali con pinco

- La produzione dei messaggi di errore
 - ⇒ è basata sull'utilizzo di oggetti di tipo ResourceBundle
 - ⇒ in modo che l'applicazione sia predisposta per l'internazionalizzazione
- La classe `java.util.ResourceBundle`
 - ⇒ una collezione di risorse (es: stringhe) sensibile all'internazionalizzazione



Differenze Principali con pinco

- Utilizzo dei ResourceBundle
 - ⇒ gli oggetti di tipo ResourceBundle sono normalmente costruiti a partire da file di properties
 - ⇒ possono esserci vari file in lingue diverse
 - ⇒ a seconda dall'impostazione di lingua del sistema (l'impostazione "locale"), la classe è in grado di prelevare la versione linguisticamente adatta



Differenze Principali con pinco

○ In concreto, con Struts

- ⇒ per la produzione dei messaggi di errore viene predisposto uno file di properties contenente coppie chiave=messaggio
es: risorse/messaggiErrore.properties
- ⇒ le chiavi sono identificatori dei messaggi neutri rispetto alla lingua
- ⇒ nei metodi di convalida dei bean i messaggi di errore vengono specificati utilizzando le chiavi, e non il testo



Differenze Principali con pinco

○ In concreto, con Struts (cont.)

- ⇒ negli schermi, i tag della libreria personalizzata di Struts sono in grado di risalire dalle chiavi dei messaggi alle corrispondenti stringhe
- ⇒ oltre al file di properties con la lingua "predefinita", il programmatore può produrre altre versioni localizzate
- ⇒ es: risorse/messaggiErrore_fr.properties, risorse/messaggiErrore_en.properties



Differenze Principali con pinco

- In concreto, con Struts (cont.)
 - ⇒ nel caso in cui l'impostazione locale (del server o del browser) è francese o inglese, verrà selezionato automaticamente il file di messaggi corrispondente
 - ⇒ altrimenti viene selezionato il file di messaggi standard
- Nota
 - ⇒ i bundle vanno dichiarati nel file struts-config



Differenze Principali con pinco

- Un tipico formBean di Struts
 - >> formTentativo
 - >> messaggi.properties
 - >> struts-config.xml
 - ⇒ estende org.apache.struts.action.ActionForm
 - ⇒ contiene i metodi set e get per i vari parametri della query string
 - ⇒ contiene il metodo validate() nel quale viene prodotto un oggetto di tipo ActionErrors, una collezione di ActionError
 - ⇒ ciascun ActionError contiene la chiave di un messaggio di errore da prelevare dai bundle



Differenze Principali con pinco

>> schermoTentativo.jsp

- Per l'utilizzo del framework di convalida
 - ⇒ nelle pagine JSP è necessario utilizzare i tag di azione della libreria di Struts
 - ⇒ `<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html" %>`
 - ⇒ in particolare: `<html:form>`, `<html:text>`, `<html:submit>`, invece che le ordinarie versioni html
 - ⇒ e `<html:errors />`



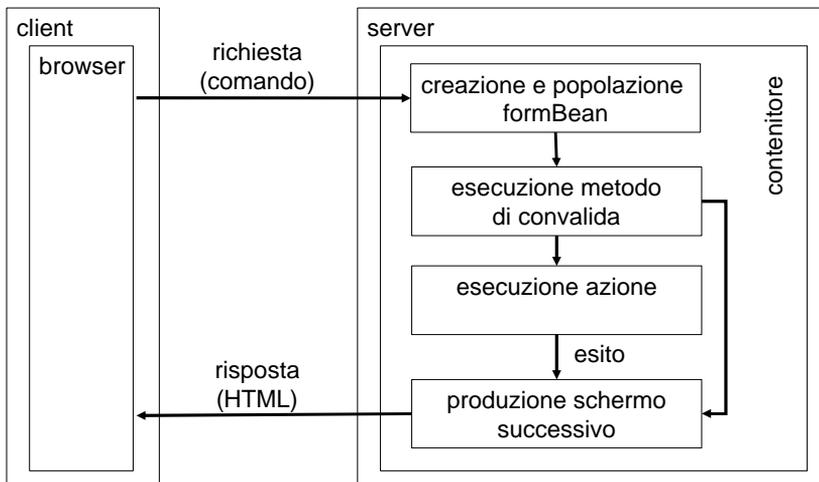
Differenze Principali con pinco

- Riassumendo
 - ⇒ il ciclo di vita della richiesta HTTP con Struts è più articolato che con pinco
- Le fasi principali
 - ⇒ creazione e popolazione del formBean
 - ⇒ esecuzione del metodo di convalida
 - ⇒ esecuzione dell'azione
 - ⇒ produzione dello schermo successivo



>> discutere indovinam2struts

Differenze Principali con pinco



Altre Funzionalità

- Client-Side Validation Framework
 - ⇒ Struts fornisce un framework per la convalida lato client in aggiunta a quello per la convalida lato server
 - ⇒ in sintesi, è possibile descrivere i vincoli di convalida sugli attributi dei formbean in un file di configurazione
 - ⇒ il framework genera automaticamente il codice JavaScript negli schermi per la verifica lato client



Altre Funzionalità

○ Tiles

- ⇒ Tiles è un framework per la costruzione modulare degli schermi
- ⇒ consente di assemblare gli schermi a partire da porzioni più piccole (es: intestazione, barra laterale, corpo principale, piè di pagina) invece che specificare staticamente il codice HTML nelle pagine JSP



Riassumendo

- Introduzione
- Modello 2 con Struts
- Differenza Principali con pinco
- Altre Funzionalità



Termini della Licenza

- This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
- Questo lavoro viene concesso in uso secondo i termini della licenza "Attribution-ShareAlike" di Creative Commons. Per ottenere una copia della licenza, è possibile visitare <http://creativecommons.org/licenses/by-sa/1.0/> oppure inviare una lettera all'indirizzo Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.