Programmazione WEB

Lezione del 29 Aprile 2019

Docente: Novella Bartolini

JSP e Bean

- Azione Standard **`<jsp:useBean>`**
  - Il codice contenuto negli scriptlet Java può esser inserito in una classe apposita invocata al momento del bisogno
  - Necessaria la conformità agli standard fissati per i JavaBean (architettura a componenti)

```
// classe JellyBean.java

package beans; //NOTARE CHE IL BEAN VIENE MESSO IN UN PACKAGE

public class JellyBean {
  private String color;

  public JellyBean()  {
  }

  public String getColor() {
                return color;
  }

  public void setColor(String newColor) {
                color=newColor;
  }
}
```

JavaBeans

- Ai fini dello sviluppo di servlet e JSP contano i seguenti aspetti della specifica:
  - I JavaBean non devono avere nessuna proprietà pubblica (attributi di classe privati).
  - Tutte le proprietà di un JavaBean che devono essere esposte dovranno avere metodi pubblici **get** e **set** secondo necessità.
    - Tali metodi dovranno avere nome **getVariableName** e **setVariableName**
  - Tutti i JavaBean devono avere un metodo costruttore senza argomenti

Utilizzo di un JavaBean in una JSP

- Affinchè sia visibile alla web application il bean deve essere incluso nella sotto-directory `classes` di `WEB-INF` con tutto il percorso del suo package

- Una volta creato il JavaBean, per poterlo utilizzare in una JSP vengono definiti 3 tag:
  - Per individuare o creare un JavaBean nell'area di visibilità specificata
  - Per definire una proprietà di un JavaBean (set)
  - Per leggere una proprietà di un JavaBean (get)

Caricamento di un JavaBean

- Il tag <jsp:useBean> ha la seguente sintassi:
  - <jsp:useBean id="nome" scope="page|request|session|application" class="nomeClasse" />
    - id è l'etichetta che viene assegnata al bean all'interno dell'applicazione (nome dell'istanza cui si fa riferimento)
    - scope definisce le modalità con cui l'istanza del Bean deve essere ricercata
- Esempio:
  - <jsp:useBean id="jb" scope="application" class="beans.JellyBean" />

Attributi dell'azione <jsp:useBean>

Attribute	Description
id	The name used to manipulate the Java object with actions **<jsp:setProperty>** and **<jsp:getProperty>**. A variable of this name is also declared for use in JSP scripting elements. The name specified here is case sensitive.
scope	The scope in which the Java object is accessible—**page**, **request**, **session** or **application**. The default scope is **page**.
class	The fully qualified class name of the Java object.
type	The type of the JavaBean. This can be the same type as the **class** attribute, a superclass of that type or an interface implemented by that type. The default value is the same as for attribute **class**. A **ClassCastException** occurs if the Java object is not of the type specified with attribute **type**.

Attributes of the **<jsp:useBean>** action.

Impostazione di una proprietà JavaBean

- Una volta creato il Bean è possibile definirne le proprietà con il tag **jsp:setProperty**
  - <jsp:setProperty name="beanName" property="propertyName" value="propertyValue" />

    (assegnazione di un valore)
  - <jsp:setProperty name="beanName" property="propertyName" param="parameterName" />

    (passaggio di un parametro della richiesta)

Si noti che qui "name" è il nome dell'istanza, ovvero l'"id"

Specificato nell'azione standard <jsp:usebean …>

Impostazione di una proprietà JavaBean

- Modo rapido per definire le proprietà del Bean attraverso dati provenienti da un form:
  - <jsp:setProperty name="beanName" property="*">
- Aggiungendo "*" il metodo setProperty fa corrispondere tutti i parametri dell'oggetto richiesta ai metodi set del bean e passa i valori tra di essi.
- Definizione automatica delle proprietà possibile solo se si rispettano le convenzioni sui nomi dei metodi set e get

Attributi dell'azione <jsp:setProperty>

Attribute	Description
name	The ID of the JavaBean for which a property (or properties) will be set.
property	The name of the property to set. Specifying **"*"** for this attribute causes the JSP to match the request parameters to the properties of the bean. For each request parameter that matches (i.e., the name of the request parameter is identical to the bean's property name), the corresponding property in the bean is set to the value of the parameter. If the value of the request parameter is **""**, the property value in the bean remains unchanged.
param	If request parameter names do not match bean property names, this attribute can be used to specify which request parameter should be used to obtain the value for a specific bean property. This attribute is optional. If this attribute is omitted, the request parameter names must match bean property names.
value	The value to assign to a bean property. The value typically is the result of a JSP expression. This attribute is particularly useful for setting bean properties that cannot be set using request parameters. This attribute is optional. If this attribute is omitted, the JavaBean property must be of a data type that can be set using request parameters.

Lettura delle proprietà di un bean

- Una volta creato il Bean è possibile leggerne le proprietà con il tag **jsp:getProperty**
  - <jsp:getProperty name="beanName" property="propertyName" />

Esempio: definizione dell’attributo di un bean per l’accesso da una pagina jsp

```
<!– Pagina di scrittura delle proprietà: scrivi.jsp -->

<html>
<head>
  <title> Impostazione del colore di JellyBean </title>
</head>
<&@ page language=“java” %>
<jsp:useBean id=“jb” scope=“session” class=“beans.JellyBean” />
<body>
<form action = “leggi.jsp" method = "get">


        <strong>Scegli il colore del bean:</strong><br />
        <label>Rosso
          <input name = "newColor" type = "radio"
            value = "red" checked = "checked" />
        </label>
```

Esempio: definizione dell'attributo di un bean per l'accesso da una pagina jsp

```
        <label>Verde
          <input name = "newColor" type = "radio"
            value = "green" /></label>

        <label>giallo
          <input name = "newColor" type = "radio"
            value = "yellow" /></label>



          <input type = "submit" value = "Submit" />
</form>
</body>
```

Come ottenere una proprietà di un JavaBean

```
  <!– leggi proprietà: leggi.jsp →

<html>
  <head>
    <title> Ottenere il colore di JellyBean </title>
  </head>
<%@ page language="java" %>
<jsp:useBean id="jb" scope="session" class="beans.JellyBean" />




<body>
   <jsp:setProperty name="jb" property="color" param="newColor" />
   Il colore del bean è stato impostato al valore:</td>
   <jsp:getProperty name="jb" property="color" /></td>
</body>
</html>
```

Esercizio:

- Scrivere semplici esempi di pagine jsp contenenti:
  - La direttiva “include”
  - Le azioni standard
    - include
    - forward
    - useBean (con diversi contesti di visibilità)
    - setProperty (sia il caso in cui viene passato un valore, sia il caso in cui viene preso il valore di un parametro della richiesta)
    - getProperty
  - Descrivere (non codice) come vengono tradotte dal jsp container (consegnare solo quest’ultima parte)

JSP useBean: contesti di visibilità

- request
  - <jsp:useBean id="..." class="..." scope="request" />
- session
  - <jsp:useBean id="..." class="..." scope="session" />
- application
  - <jsp:useBean id="..." class="..." scope="application" />
- page
  - <jsp:useBean id="..." class="..." scope="page" />
    oppure
    <jsp:useBean id="..." class="..." /> (page è il contesto di default)

Visibilità di un JavaBean

- Si può accedere ad un bean da più pagine JSP, a condizione che
  - Il bean sia stato dichiarato con scope *session* e sia lo stesso utente ad accedere ad entrambe le pagine
    - Se si osserva il codice generato per le JSP si noterà una chiamata al metodo **`pageContext.getSession()`**
  - Oppure che il bean sia stato dichiarato con scope *application*
    - Tutti gli utenti avranno accesso alla stessa istanza del bean

Esercitazione

- Scrivere due servlet A e B che producono ciascuna una pagina html propria in cui si dica se in concomitanza con l'elaborazione della richiesta corrente si è avuta o no la creazione di una nuova sessione.
- Le due servlet citate accedono allo stesso bean con visibilità di applicazione. Il suddetto bean deve contenere informazioni generiche sull'applicazione e in particolare due contatori: un contatore di accessi alle singole risorse e un contatore di sessioni.
- Scrivere infine una pagina JSP riassuntiva che oltre ad aggiornare come le servlet A e B i dati del bean ne legga il contenuto e lo stampi attraverso opportune azioni standard.