**BIBBIA JSP**

Nel contesto della piattaforma Java, la tecnologia JSP è correlata con quella delle servlet: all'atto della prima invocazione, le pagine JSP vengono infatti tradotte automaticamente da un compilatore JSP in servlet.

. Una pagina JSP può quindi essere vista come una rappresentazione ad alto livello di un servlet. Per via di questa dipendenza concettuale, anche l'uso della tecnologia JSP richiede la presenza, sul Web server, di un servlet container, oltre che di un server specifico JSP detto motore JSP (che include il compilatore JSP); in genere, servlet container e motore JSP sono integrati in un unico prodotto (per esempio, Tomcat svolge entrambe le funzioni).

**DIRETTIVE**

Le direttive JSP si possono interpretare come comandi rivolti al motore JSP. Questi comandi vengono eseguiti in una fase di preprocessing, prima che siano elaborate le porzioni della pagina contenenti script. Le due direttive principali sono include e page.

* **include**: ordina al motore JSP di includere un altro file in quello corrente. È un'operazione equivalente a quella di copiare ed incollare il contenuto del file indicato in quello corrente. Questa direttiva può essere usata, per esempio, per realizzare contenuti ricorrenti in diverse pagine di un sito Web.

**<%@ include file="filedaincludere.ext" %>**

* **page**: esistono numerose varianti della direttiva **page**, con scopi piuttosto diversi fra loro:

**page import** corrisponde all'import Java, e indica quali classi saranno utilizzate nel resto del documento;

**<%@ page import="java.util.\*" %> //importa tutte le classi del package java.util**

page contentType indica in quale formato sia scritto il documento (per esempio HTML o XML)

**<%@ page contentType="text/html" %>**

**page errorPage** specifica la "pagina di errore" che deve essere visualizzata se l'elaborazione della pagina dinamica fallisse in seguito al verificarsi di qualche eccezione

**page isErrorPage** indica se quella corrente è una pagina di errore

**<%@ page isErrorPage=false %> //specifica che non è una pagina di errore**

**page isThreadSafe** indica se il servlet generato dalla pagina JSP è thread safe

**page taglib** indica che la pagina JSP fa uso di una libreria di tag. La libreria deve essere identificata tramite una URI (Uniform Resource Identifier)

**<%@ taglib prefix="myprefix" uri="taglib/miataglib.tld" %>**

La pagina JSP può includere dichiarazioni di attributi e metodi, introdotte dalla sequenza <%!. Tali attributi e metodi andranno a far parte della classe "servlet" generata dal compilatore JSP (la loro posizione all'interno del testo della pagina JSP è irrilevante). Lo sviluppatore che ha familiarità con la tecnologia servlet può sfruttare questa possibilità, per esempio, per fare overriding dei "metodi di life-cycle" dei servlet, come jspInit.

**<%! int serverInstanceVariable = 1; %>**

Le espressioni sono introdotte dalla sequenza <%=, e contengono una qualsiasi espressione Java. Durante l'elaborazione della pagina JSP, l'espressione viene valutata dal motore JSP, il risultato viene convertito in stringa, e la stringa inserita nel codice HTML/XML nel punto corrispondente a quello dove compare l'espressione stessa.

**<body>**

**Sei l'utente registrato nr. <%= registroUtenti.conta() %>**

**</body>**

Gli elementi di codice in una pagina JSP possono fare riferimento a un insieme di variabili predefinite:

**out**: rappresenta il flusso di output su cui viene prodotta la pagina Web. Si tratta di un oggetto di classe JSPWriter. Il suo uso è spesso implicito (per esempio nelle espressioni), ma all'occorrenza si può fare riferimento a questa variabile

**page**: rappresenta il servlet generato dalla pagina JSP

**pageContext**: un oggetto di classe PageContext, che contiene dati associati all'intera pagina. L'oggetto può essere trasferito da una pagina JSP a un'altra

**request**: l'oggetto di classe HttpServletRequest che rappresenta la richiesta HTTP che ha portato all'attivazione della pagina JSP/servlet

**response**: l'oggetto di classe HttpServletResponse che rappresenta la risposta HTTP da inviare

**session**: l'oggetto di classe HttpSession che rappresenta la sessione HTTP all'interno della quale è stata invocata la pagina JSP

**application**: permette di accedere e di memorizzare gli oggetti per renderli accessibili da qualsiasi utente e modificabili da ogni pagina

**config**: viene spesso lasciato inutilizzato, visto che le informazioni messe a disposizione da questo oggetto implicito sono generalmente settate e rilevate in automatico.

Le azioni JSP sono tag XML che consentono di controllare alcuni aspetti del comportamento del Web server. Alcuni esempi sono i seguenti:

**jsp:include**: cede il controllo a un'altra pagina JSP, con ritorno del controllo a elaborazione terminata; simile a una chiamata di procedura.

**jsp:forward**: cede il controllo, definitivamente, a un'altra pagina JSP

**jsp:param**: introduce un parametro aggiuntivo nel contesto di azioni come include e forward.