

Servlet



ALBERTO BELUSSI
ANNO ACCADEMICO 2009/2010

Servlet



Documentazione:

<http://java.sun.com/products/servlet/>

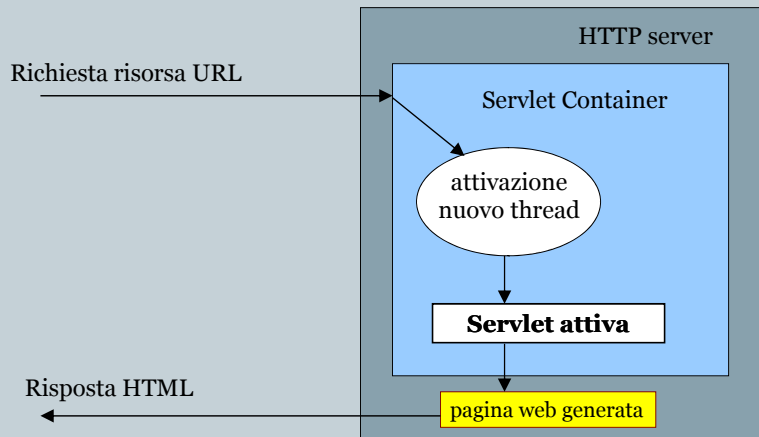
L'approccio **SERVLET** richiede l'installazione di un **SERVLET ENGINE** da integrare nel server HTTP.

Ogni servlet engine predispone un albero di direttori dove vanno inseriti i file contenenti le servlet, le classi java di supporto, i file html statici, ecc..

Servlet engine usato in laboratorio: TOMCAT.

Servlet engine: struttura

3



Servlet container

4

E' un processo sempre attivo che implementa una

JAVA VIRTUAL MACHINE

Servlet

5

Ogni servlet è una classe JAVA ottenuta estendendo la classe `HttpServlet`.

Esempio di servlet semplice:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorld extends HttpServlet {
    public void doGet (
        HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
```

Servlet

6

```
{
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String docType = "<!DOCTYPE HTML ...";
    out.println(docType +
        "<html>\n"+
        "<head><title>Hello World</title>" +
        "</head>\n"+
        "<body>\n"+
        "<h1>Hello World</h1>\n"+
        "</body></html>");
} // end doGet
} // end servlet
```

Servlet

7

Note sui parametri di `doGet` (`doPost`)

`request`: consente di accedere a tutte le informazioni relative alla richiesta HTTP che ha invocato la servlet.

Alcuni metodi di **`request`**

- `request.getParameter(NOME_PARAMETRO)`
restituisce: una stringa (`String`) che rappresenta il valore della prima occorrenza del parametro `NOME_PARAMETRO`; la stringa vuota se il parametro esiste ma non ha valore; `NULL` se il parametro non esiste.
- `request.getParameterValue(NOME_PARAMETRO)`
restituisce un array di stringhe contenenti i valori di tutte le occorrenze del parametro `NOME_PARAMETRO`; un array con una stringa vuota se il parametro esiste ma non ha valore; `NULL` se il parametro non esiste.
- `request.getParameterNames()`
restituisce un array di stringhe contenenti i nomi dei parametri della richiesta HTTP.

Servlet

8

`response`: gestisce l'invio dell'output (HTML) al server HTTP.

Alcuni metodi di `response`

- `response.getWriter()`
restituisce un oggetto della classe `PrintWriter` al quale è possibile inviare stringhe di caratteri con il metodo `println`.
- `response.getBufferSize()`
restituisce la dimensione del buffer degli oggetti `PrintWriter`.
- `request.setBufferSize()`
consente di ridefinire la dimensione del buffer degli oggetti `PrintWriter`.

Servlet

9

Come passare i parametri ad una servlet:

- **Direttamente nell'URL**

Ad esempio `http://server/risorsa?par1="val1"&par1="val2"&par2="val3"`

- **Attraverso una FORM di HTML** (come vedremo in laboratorio)
in questo caso il browser presenta la FORM all'utente come maschera con campi in cui inserire i valori dei parametri; alla pressione del bottone "submit" il browser invia una richiesta HTTP di tipo GET (o POST) come precisato nella specifica della FORM, completando la richiesta con i valori dei parametri inseriti dall'utente.

Questi parametri sono resi accessibili alla servlet attraverso l'oggetto **request**.