

Un contatore di accessi

- Scriviamo una servlet che conta il numero di accessi e lo visualizza.
- Sfruttiamo il fatto che Tomcat, quando riceve la richiesta di una servlet, crea un'istanza della classe corrispondente.
- Utilizziamo una variabile d'istanza privata **i** nel modo seguente:
 - la azzeriamo all'inizio dell'esecuzione della servlet;
 - la incrementiamo ad ogni accesso.

Il sorgente Java (I)

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Contatore extends HttpServlet {
    private int i;

    // init viene eseguito solo alla prima richiesta
    // della servlet.
    public void init() {
        i=0;
    }

    // continua nel lucido successivo ...
}
```

Il sorgente Java (II)

```
// ... continua dal lucido precedente
public void doGet(HttpServletRequest req,
                  HttpServletResponse res)
    throws ServletException , IOException {
    res.setContentType("text/html");
    PrintWriter out = res.getWriter();
    out.println("<HTML>");
    out.println("<HEAD><TITLE>Contatore di "+
                "accessi</TITLE></HEAD>");
    out.println("<BODY>");
    out.println("<STRONG>Accessi registrati "+
                "finora: "+(++i)+"</STRONG>");
    out.println("</BODY></HTML>");
}
```

```
}
```

Il metodo `init()`

- Il metodo `init()` di una servlet viene eseguito alla prima richiesta di quest'ultima.
- E' utile quindi per inserire il codice che inizializza la servlet.
- Nel nostro caso il metodo `init()` contiene soltanto l'assegnamento `i = 0` (che inizializza a zero il contatore di accessi).

Modifiche al file `index.html`

```
<HTML>
<HEAD>
<TITLE>Servlet di prova</TITLE>
</HEAD>

<BODY>
  <TABLE>
    ...
    <TR>
      <TD>
        <A HREF="servlet/Contatore">Un contatore di
          accessi</A>
      </TD>
    </TR>
  </TABLE>
</BODY>

</HTML>
```

Modifiche al file web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE web-app
```

```
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"  
    "http://java.sun.com/dtd/web-app_2_3.dtd">
```

```
<web-app>
```

```
    ...
```

```
    <servlet>
```

```
        <servlet-name>Contatore</servlet-name>
```

```
        <servlet-class>Contatore</servlet-class>
```

```
    </servlet>
```

```
    <servlet-mapping>
```

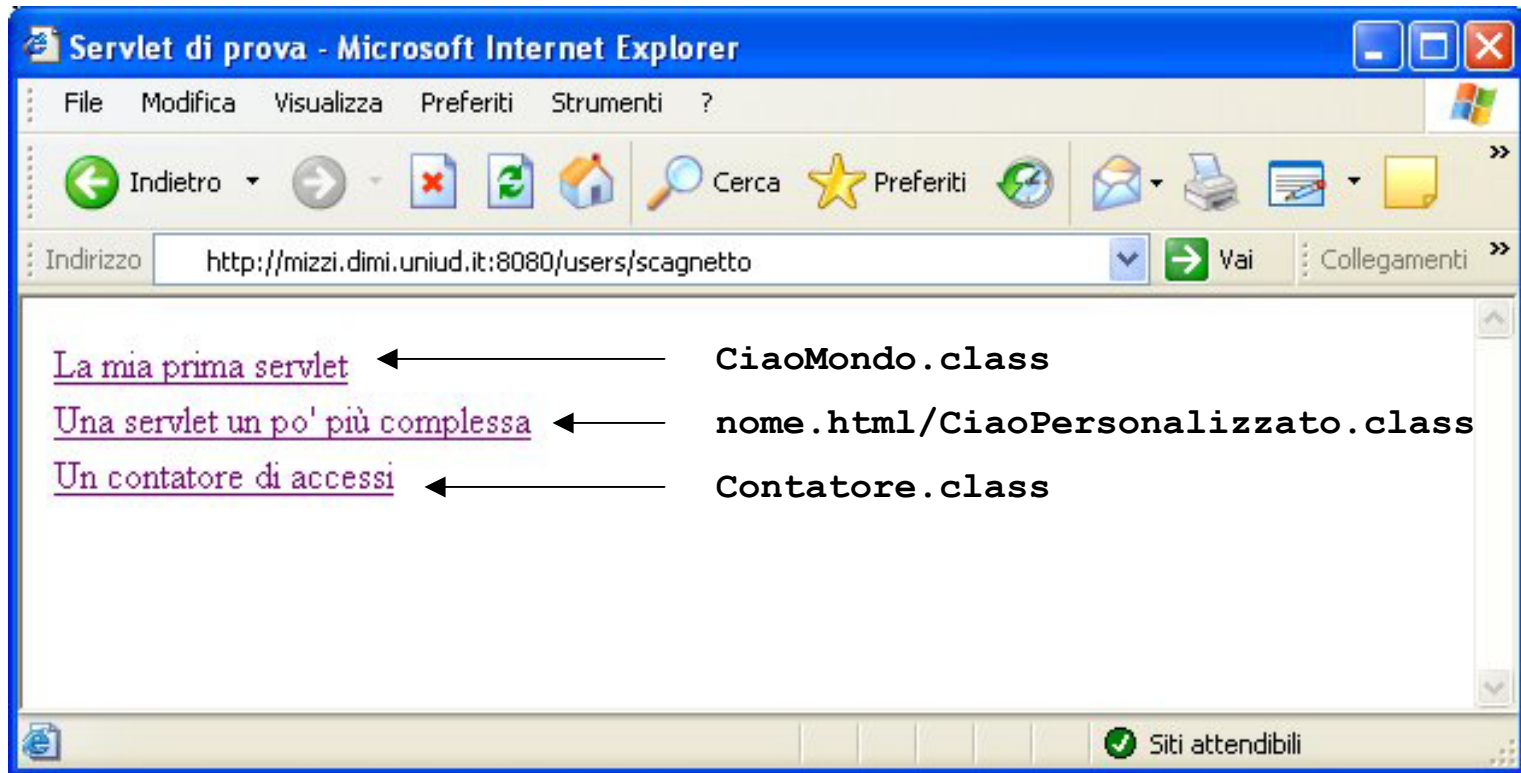
```
        <servlet-name>Contatore</servlet-name>
```

```
        <url-pattern>/servlet/Contatore</url-pattern>
```

```
    </servlet-mapping>
```

```
</web-app>
```

Aspetto della pagina `index.html`



Generare una tabella HTML a partire da un file di testo

- In questo esempio faremo generare alla servlet **StampaTabella** una tabella HTML 10x2 a partire dal contenuto di un file di testo.
- Il formato dei dati nel file di testo è il cosiddetto CSV (i.e., Comma Separated Value: valori separati da virgole).
- I dati della tabella saranno contenuti in un file di testo chiamato **tabella.txt** nella cartella **servlets** della propria home directory.

Il codice Java (I)

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class StampaTabella extends HttpServlet {

    public void doGet(HttpServletRequest req,
                      HttpServletResponse res)
                      throws ServletException , IOException,
FileNotFoundException {
        String filePath=getServletContext().getRealPath("")+"/tabella.txt";
        BufferedReader input = new BufferedReader(new FileReader(filePath));
        String linea = null;
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        out.println("<HTML>");
        out.println("<HEAD><TITLE>Tabella letta da file</TITLE></HEAD>");
    }
}
```

Il codice Java (II)

```
out.println("<BODY>");
    out.println("<TABLE BORDER=\"1\">");

    while((linea = input.readLine()) != null) {
        out.println("<TR>");
        String[] celle = linea.split(",");
        out.println("<TD>" + celle[0] + "</TD>");
        out.println("<TD>" + celle[1] + "</TD>");
        out.println("</TR>");
    }

    input.close();
    out.println("</TABLE>");
out.println("</BODY></HTML>");
}
```

Modifiche al file `index.html`

```
<HTML>
<HEAD>
<TITLE>Servlet di prova</TITLE>
</HEAD>

<BODY>
  <TABLE>
    ...
    <TR>
      <TD>
        <A HREF="servlet/Tabella">Stampa di una tabella
          letta da file</A>
      </TD>
    </TR>
  </TABLE>
</BODY>

</HTML>
```

Modifiche al file `web.xml`

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE web-app
```

```
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"  
  "http://java.sun.com/dtd/web-app_2_3.dtd">
```

```
<web-app>
```

```
  ...
```

```
  <servlet>
```

```
    <servlet-name>Tabella</servlet-name>
```

```
    <servlet-class>StampaTabella</servlet-class>
```

```
  </servlet>
```

```
  <servlet-mapping>
```

```
    <servlet-name>Tabella</servlet-name>
```

```
    <url-pattern>/servlet/Tabella</url-pattern>
```

```
  </servlet-mapping>
```

```
</web-app>
```

Note sul dispiegamento

- Ricordarsi di copiare nella cartella **servlets** della propria home directory su **mizzi.dimi.uniud.it** anche il file **tabella.txt**.
- Esempio di contenuto del file **tabella.txt**:

Riga 1 Colonna 1,Riga 1 Colonna 2

Riga 2 Colonna 1,Riga 2 Colonna 2

Riga 3 Colonna 1,Riga 3 Colonna 2

Riga 4 Colonna 1,Riga 4 Colonna 2

Riga 5 Colonna 1,Riga 5 Colonna 2

Riga 6 Colonna 1,Riga 6 Colonna 2

Riga 7 Colonna 1,Riga 7 Colonna 2

Riga 8 Colonna 1,Riga 8 Colonna 2

Riga 9 Colonna 1,Riga 9 Colonna 2

Riga 10 Colonna 1,Riga 10 Colonna 2

Esempio di esecuzione di StampaTabella

