

# Servlet: utilizzo dei Cookies

---

Web Programming

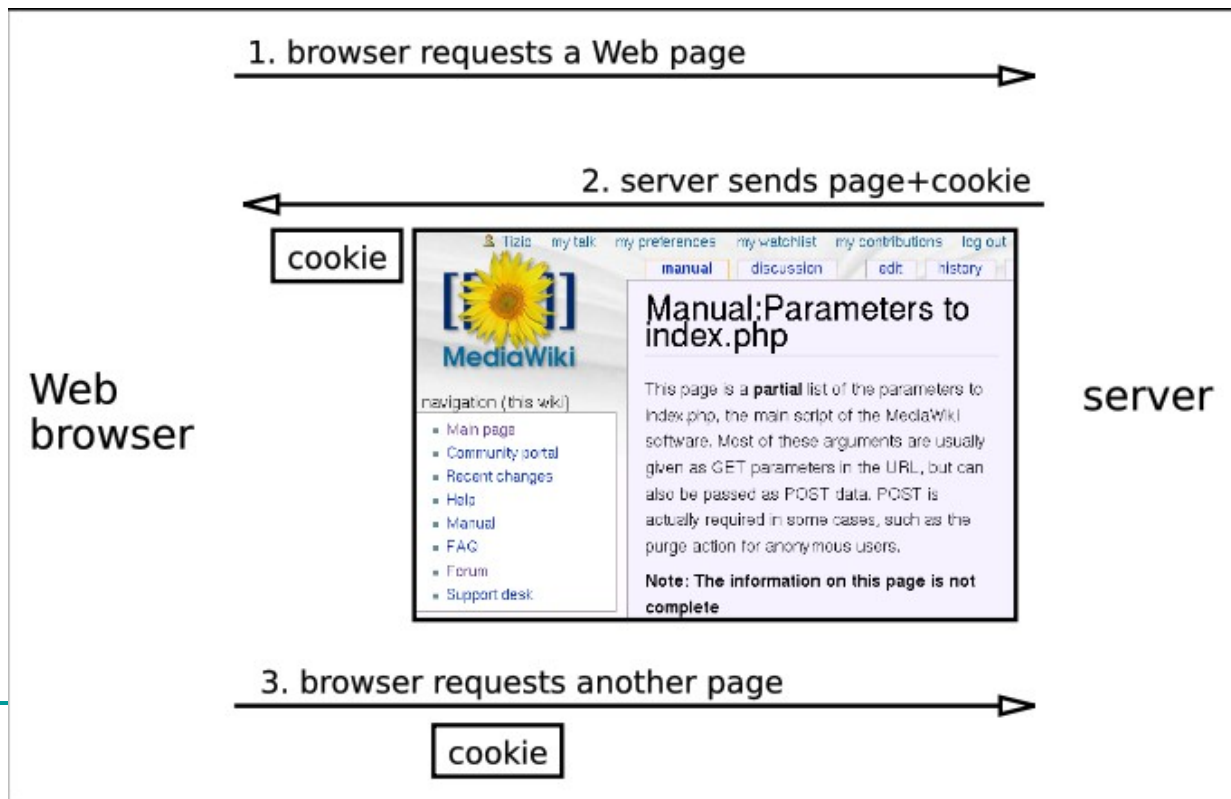
AA 2013-2014 Gino Perna

# Cookies

- La navigazione delle pagine web e' state-less, ovvero il server non puo' ricordare in che stato era il client alla richiesta precedente.
- Questo significa che alcune informazione devono transitare dal client al server (e/o viceversa) per tenere traccia dello stato del client
- I cookies sono uno dei metodi di mantenimento dello stato:

# Cookies:caratteristiche

- I cookies sono un meccanismo generale che le connessioni server-side (come, ad esempio, gli script CGI) possono utilizzare per memorizzare o reperire informazioni sulla parte client del collegamento.



# Cookies:caratteristiche

- Il principio di funzionamento di tale meccanismo è semplice:
  - Il server, restituendo una risposta HTTP al client, invia anche una porzione di informazione, denominata cookie, contenente, fra l'altro una descrizione del range degli URL per i quali è valida (come vedremo, tale descrizione può essere esplicita, mediante il settaggio di opportuni attributi o implicita).
  - Ad ogni successiva richiesta HTTP effettuata dal client che ricade nel range specificato, verrà accodato il cookie, che renderà così al server il contenuto informativo salvato con la connessione preedente.

# Cookies

- I cookie sono delle coppie (attributo, valore) memorizzate come stringhe sul client
  - I cookie sono ritornati nell'header di ogni richiesta fatta al server web (e solo al server) che li ha creati
  - Un client normalmente memorizza un massimo di 20 cookie per ogni server e un massimo di 300 cookie
  - Ogni cookie ha un tempo di scadenza (*max age*) in secondi con i seguenti casi particolari:
    - Se *max age* = 0 il cookie viene cancellato
    - Se *max age* = -1 il cookie viene memorizzato solo finchè il processo del browser è attivo
-

# Cookies - cont

- I cookie sono poi tornati dal client al server se il cookie corrisponde al dominio richiesto
- Ogni client (quindi IE, Firefox, Safari, Konqueror etc..) ne tiene una copia in directory diverse, questo significa che se uso IE e poi uso Firefox i Cookies devono essere reinviati, o risulteranno mancanti.

# Cookies: in pratica perché servono?

- Identificazione di un utente in una sessione di e-commerce.
- Customizzazione di un sito (in funzione delle preferenze utente)
- Pubblicità mirata
- Eliminazione di username e password
- etc...

# Cookies in Java Servlets

- La classe `javax.servlet.http.Cookie` crea un cookie con un nome e un valore  
`Cookie cookie = new Cookie("mycookie", "prova");`
- I metodi principali:
  - `cookie.setValue("nuovo valore");`
  - `cookie.setAge(40);`
- Per spedire un cookie esso va aggiunto alla risposta prima di chiamare il metodo `getWriter()`
  - `response.addCookie(cookie)`
  - Per leggere i cookie da una servlet si usa il metodo `getCookies()` della request



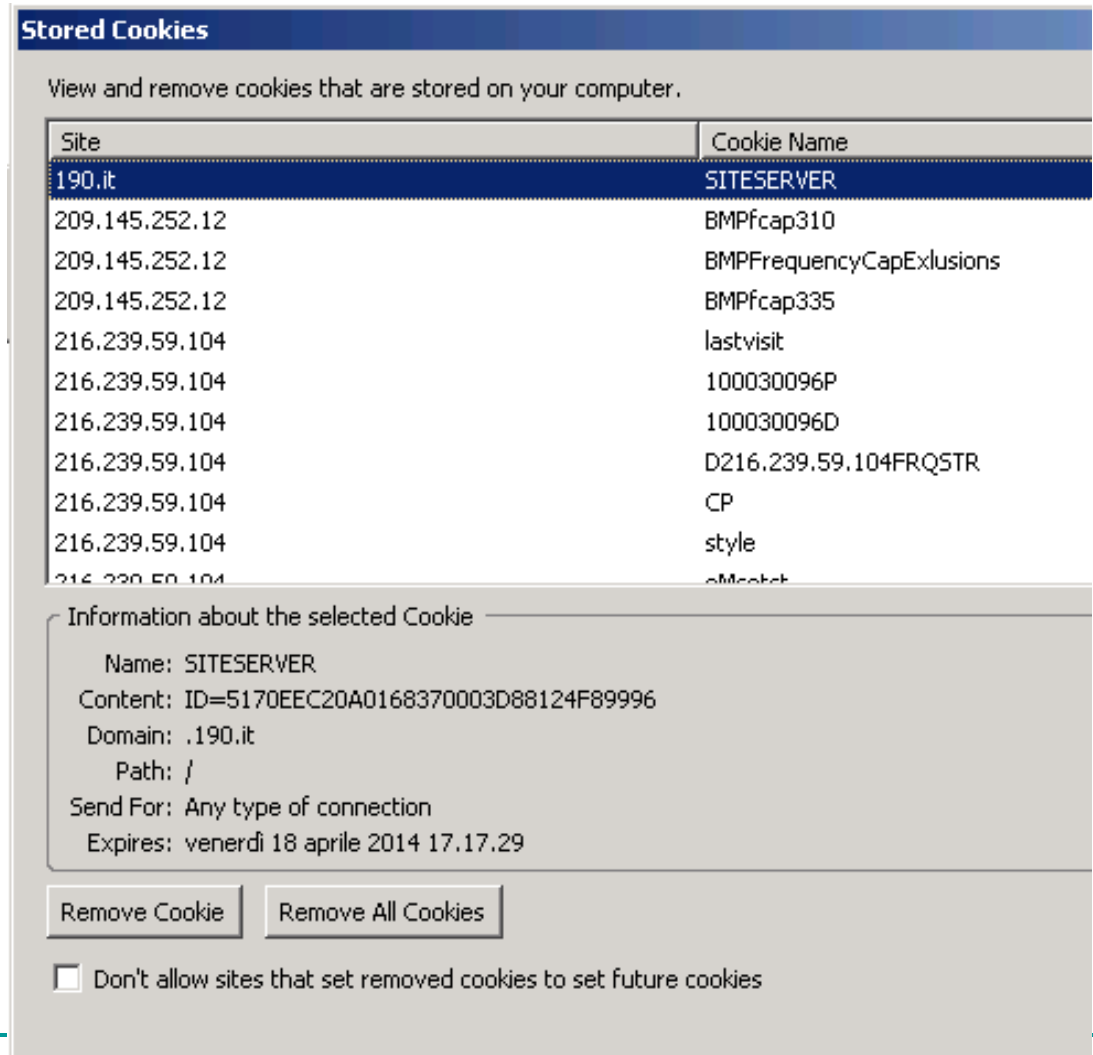
# Cookies:lettura

**Richiedere i cookie prima di qualsiasi scrittura della pagina (prima dell'apertura del writer della pagina) se servisse modificarli**

```
Cookie[] cookies = request.getCookies();  
    Cookie cookie;  
    for(int i=0; i<cookies.length; i++) {  
        cookie = cookies[i];  
        out.println("<TR>\n" +  
            "    <TD>" + cookie.getName() + "\n" +  
            "    <TD>" + cookie.getValue());  
    }
```

# Visualizzazione lato client

- Mozilla Firefox:  
tools>Options>  
privacy>Cookie

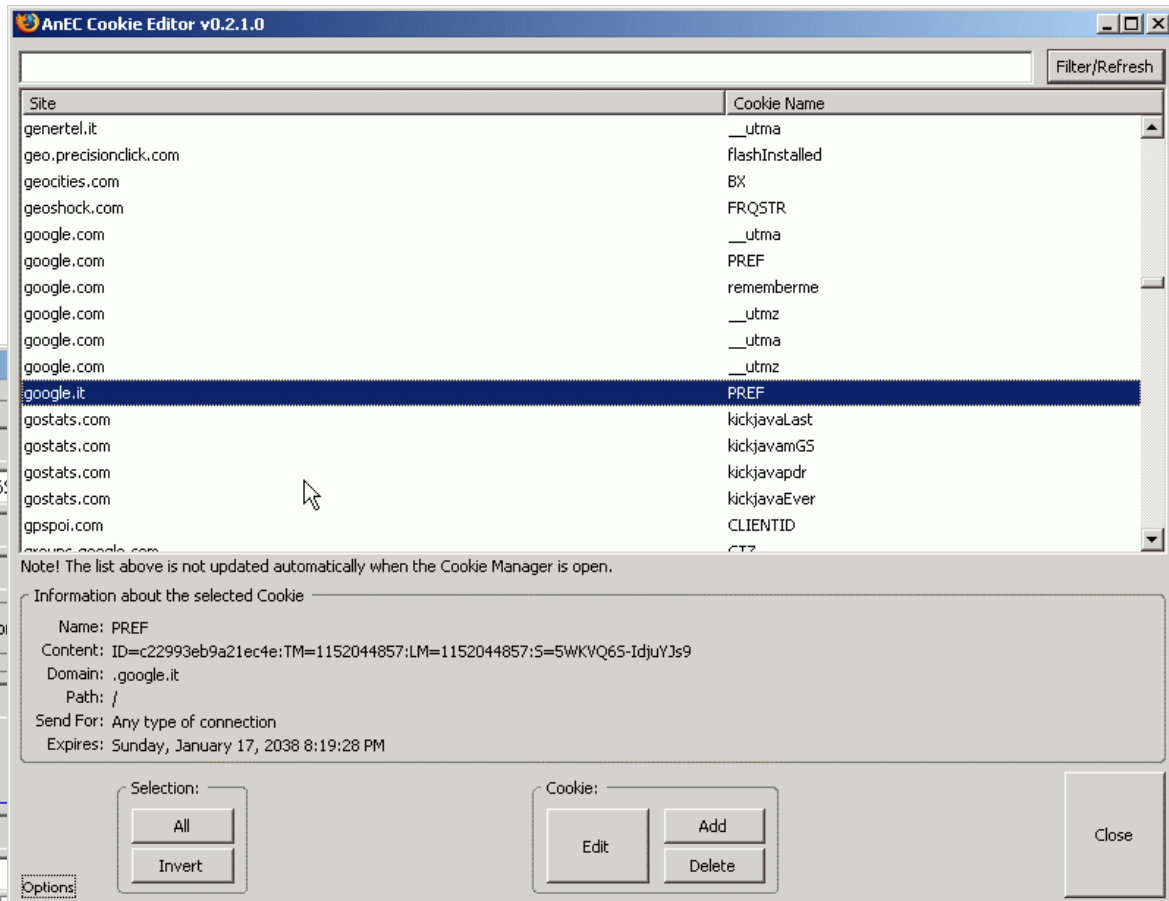


# Visualizzazione/Editing

- Installare l'estensione web developer di firefox per poter agevolmente operare con essi
- In Chrome: menu>strumenti>strumenti sviluppatori>resources>cookies dopo aver acceduto alla pagina interessata

# Correzione (Add N edit Cokie ext.)

- Mozilla Firefox:  
tools>  
Edit Cookie



# Cookies: I metodi in dettaglio

- `public void setComment(String c)`
- `public String getComment()`
  
- `public void setVersion(int c)`
- `public int getVersion ()`
  - Version 0: Netscape standard
  - Version 1: RFC 2109

# Cookies: i metodi

- `public void setMaxAge(int c)`
- `public int getMaxAge()`
  - Positive value: secondi di vita
  - 0: delete cookie (quando arriverà sul client)
  - Negative value: finché dura la sessione del browser

# Cookies: i metodi

- `public void setDomain(String c)`

- `public String getDomain()`

- Specifies a domain restriction pattern. A domain pattern specifies the servers that should see a cookie. By default, cookies are returned only to the host that saved them. Specifying a domain name pattern overrides this. The pattern must begin with a dot and must contain at least two dots. A pattern matches only one entry beyond the initial dot. For example, ".foo.com" is valid and matches `www.foo.com` and `upload.foo.com` but not `www.upload.foo.com`. For details on domain patterns, see Netscape's Cookie Specification and RFC 2109.

- `public void setPath(int c)`

- `public int getPath()`

- Specifies a path for the cookie, which is the subset of URIs to which a cookie should be sent. By default, cookies are sent to the page that set the cookie and to all the pages in that directory or under that directory. For example, if `/servlet/CookieMonster` sets a cookie, the default path is `/servlet`. That path indicates the cookie should be sent to `/servlet/Elmo` and to `/servlet/subdir/BigBird`--but not to the `/Oscar.html` servlet alias or to any CGI programs under `/cgi-bin`. A path set to `/` causes a cookie to be sent to all the pages on a server. A cookie's path must be such that it includes the servlet that set the cookie.

# Cookies: La privacy!

- Tenete presente che attraverso i cookies potete invadere la privacy dell'utente, tenendo memorizzati (lato client teoricamente, ma poi finiscono sul server...) tutte le eventuali sessioni, preferenze etc...
  - Google e' l'esempio per antonomasia di utilizzo di qualsiasi informazione l'utente possa chiedere per memorizzarla
    - <http://www.google.com/privacypolicy.html>
    - <http://epic.org/privacy/gmail/faq.html>
-



# Cookies: Bibliografia

- [http://en.wikipedia.org/wiki/HTTP\\_cookie](http://en.wikipedia.org/wiki/HTTP_cookie)
  - **Attacchi: tipologie e metodi**
  - **Settaggi**
- **Privacy:**
  - <http://cpe.njit.edu/dlnotes/CIS/CIS350/TakingTheByteOutOfCookies.pdf>
- **Quick reference**
  - <http://www.tutorialspoint.com/servlets/servlets-cookies-handling.htm>

# Esempio setCookie:

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

/** Sets six cookies: three that apply only to the current session
 * (regardless of how long that session lasts) and three that persist for an hour
 * (regardless of whether the browser is restarted).
 */

public class SetCookies extends HttpServlet {

    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        for(int i=0; i<3; i++) {

            // Default maxAge is -1, indicating cookie applies only to current browsing session.

            Cookie cookie = new Cookie("Session-Cookie-" + i, "Cookie-Value-S" + i);
            cookie.setMaxAge(-1);
            response.addCookie(cookie);
```

---

# Esempio setCookie (cont):

```
cookie = new Cookie("Persistent-Cookie-" + i,"Cookie-Value-P" + i);
    // Cookie is valid for an hour, regardless of whether
    // user quits browser, reboots computer, or whatever.
    cookie.setMaxAge(3600);
    response.addCookie(cookie);
}
response.setContentType("text/html");
PrintWriter out = response.getWriter();
String title = "Setting Cookies";
out.println(("<HTML><HEAD><TITLE>" +title+ "</TITLE></HEAD>" +
    "<BODY BGCOLOR=\"#FDF5E6\">\n" + "<H1 ALIGN=\"CENTER\">"
    + title + "</H1>\n" +
    "There are six cookies associated with this page.\n" + "</BODY></HTML>");
}
```

---

# Esempio: showCookie

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

/** Creates a table of the cookies associated with the current page. */

public class ShowCookies extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        String title = "Active Cookies";

        out.println("<HTML><HEAD><TITLE>" + title + "</TITLE></HEAD>" +
            "<BODY BGCOLOR=\"#FDF5E6\">\n" +
            "<H1 ALIGN=\"CENTER\">" + title + "</H1>\n" +
            "<TABLE BORDER=1 ALIGN=\"CENTER\">\n" +
            "<TR BGCOLOR=\"#FFAD00\">\n" +
            "    <TH>Cookie Name\n" + "    <TH>Cookie Value");
```

# Esempio: showCookie

```
Cookie[] cookies = request.getCookies();  
    Cookie cookie;  
    for(int i=0; i<cookies.length; i++) {  
        cookie = cookies[i];  
        out.println("<TR>\n" +  
                    "    <TD>" + cookie.getName() + "\n" +  
                    "    <TD>" + cookie.getValue());  
    }  
    out.println("</TABLE></BODY></HTML>");  
}  
}
```

# Esempio: cancellazione cookies

- To delete cookies is very simple. If you want to delete a cookie then you simply need to follow up following three steps:
  - ❑ Read an already existing cookie and store it in Cookie object.
  - ❑ Set cookie age as zero using `setMaxAge()` method to delete an existing cookie.
  - ❑ Add this cookie back into response header.

```
Cookie cookie = null;  
Cookie[] cookies = null;  
    // Get an array of Cookies associated with this domain  
cookies = request.getCookies();  
    // Set response content type  
response.setContentType("text/html");
```

# Esempio: cancellazione cookies

```
PrintWriter out = response.getWriter();
String title = "Delete Cookies Example";
String docType = "<!doctype html public \"-//w3c//dtd html 4.0 \" +
\"transitional//en\">\n";
out.println(docType + "<html>\n" +
    "<head><title>" + title + "</title></head>\n" +
    "<body bgcolor=\"#f0f0f0\">\n" );
if( cookies != null ){
    out.println("<h2> Cookies Name and Value</h2>");
    for (int i = 0; i < cookies.length; i++){
        cookie = cookies[i];
        if((cookie.getName( )).compareTo("first_name") == 0 ){
            cookie.setMaxAge(0);
            response.addCookie(cookie);
            out.print("Deleted cookie : " +
                cookie.getName( ) + "<br/>");
        }
    }
}
```

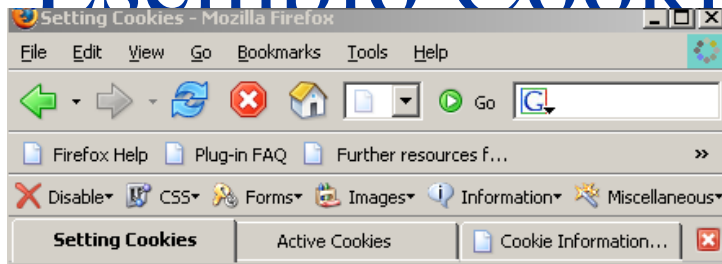
# Esempio: cancellazione cookies

```
        out.print("Name : " + cookie.getName( ) + ",  ");  
        out.print("Value: " + cookie.getValue( )+" <br/>");  
    }  
    }else{  
        out.println(  
            "<h2>No cookies founds</h2>");  
    }
```

- **Attenzione: la risposta e' mescolata al settaggio degli header, l'effettiva cancellazione dipende da come rieseguo la servlet**



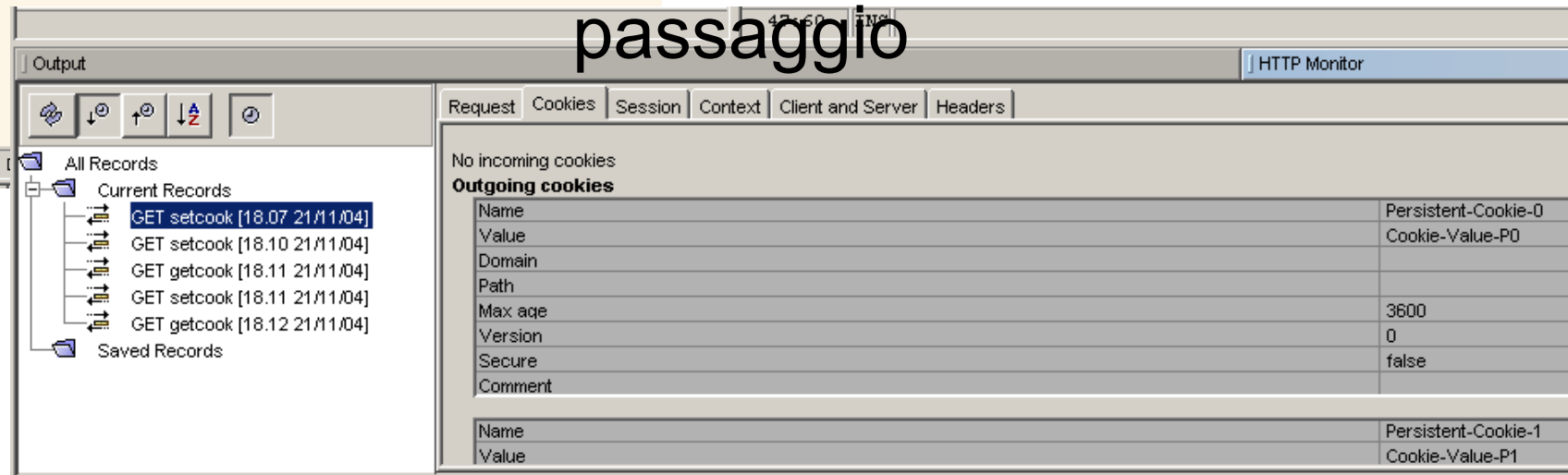
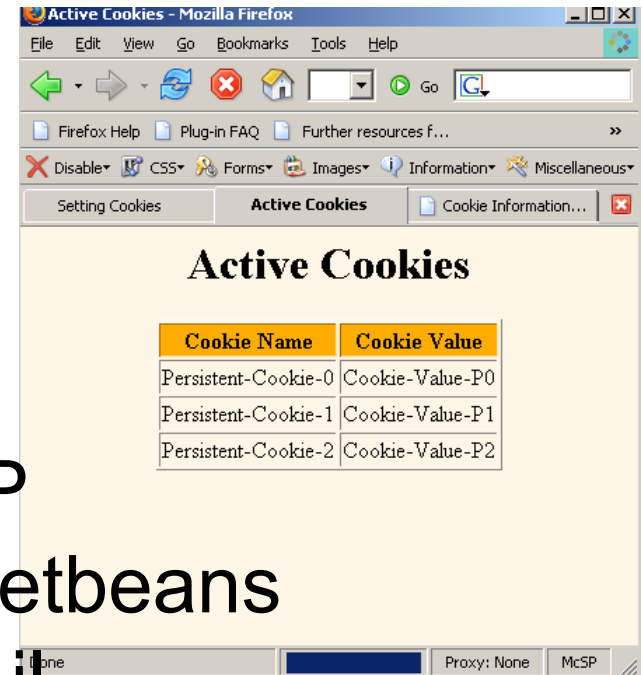
# Esempio Cookies



## Setting Cookies

There are six cookies associated with this page.

■ Utilizzare HTTP MONITOR di netbeans per monitorare il passaggio



Starting browser...

# Esercizio: login con usr/pass salvati

- Creare una servlet che visualizzi un form di login e chieda utente e password se l'utente non e' registrato
- Se l'utente e' registrato visualizzare i dati immessi

# Esercizio: due diversi approcci

- L'esercizio puo' essere svolto in due diversi modi:
  - Una sola servlet che svolge in pratica 3 funzioni:
    - Visualizza il form se ci arriva la prima volta
    - Riceve il form se l'utente manda l'autenticazione
    - Se gia' registrato visualizza i dati
  - Due servlet:
    - La prima e' il login che si preoccupa di visualizzare il form
    - La seconda:
      - Controlla l'autenticazione ed eventualmente spedisce i dati

# Esercizio: due diversi approcci

- La seconda modalita' e' piu' facile
- Verranno di seguito proposti i frammenti di codice che servono a sviluppare l'applicazione con la prima modalita'

# Esercizio: punti attenzione

- La servlet svolge tre funzioni (stesso URI!)
  - Richiedo i cookies
  - Controllo se ho un cookie, se non c'e' devo
    - Madare il form per chiedere i dati oppure
    - Processare il form dove ho dato utente e password
  - Altrimenti visualizzo i dati

# Esercizio: punti attenzione

- **USARE LA TECNICA PROPOSTA IN SEGUITO: MEMORIZZARE LA RISPOSTA IN UNA STRINGA E MANDARLA ALLA FINE**
- **Il problema da evitare e' di mandare qualsiasi risposta al client (anche implicita) senza aver completato le operazioni sui cookies**

# Esercizio: i componenti del form di login generato dalla servlet

- Tipici componenti e nomi (parameter nella servlet)

The image shows a login form with several components highlighted by colored boxes and overlaid with HTML code snippets. A red rectangle outlines the main form area, while a green box highlights the search bar at the top. The form contains a title, a message, two input fields, a submit button, and two links at the bottom.

Cerca

Non sei loggato

```
<form action="/index.php/openit/user/login" method="post">
```

Nome utente

```
<input id="id1" name="Login" size="10">
```

Password

```
<input id="id2" name="Password" size="10">
```

```
<input name="LoginButton">
```

Login

[Hai dimenticato la password?](#)

[Non registrato?](#)

# Punti attenzione

```
protected void processRequest(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {

    Cookie[] cookies = request.getCookies();
    Cookie cookie, cookie1, cookie2;
    int autenticato = 0;

    if( cookies != null) { // ho cookies devo visualizzare la pagina
        for(int i=0; i<cookies.length; i++) {
            cookie = cookies[i];
            if(cookie.getName().equals("UTENTE")){ autenticato = 1;
            }
        }
    }
}
```

---



# Punti attenzione

```
if(authenticato == 0 && request.getParameter("name") == null){  
    // non e' autenticato e non ha compilato il form  
    uscita = ( " <!DOCTYPE html PUBLIC \"-.....  
}  
else  
    if(authenticato == 0 && request.getParameter("name") != null){  
        // ha compilato il form ma devo processare i campi e dare  
        //pagina di conferma  
  
        String name = request.getParameter("name");  
        String passw = request.getParameter("passw");  
        //settare i cookie  
        .....  
    }  
    uscita =  " <!DOCTYPE html PUBLIC \"-.....  
  
}
```

---

# Punti attenzione

```
if(authenticato == 1){ // ero autenticato scrivo i parametri
```

```
    String title = "Active Cookies";
```

```
    uscita = ("<HTML><HEAD><TITLE>" +title+  
"</TITLE></HEAD>" .....  
  
}
```

```
response.setContentType("text/html");
```

```
PrintWriter out = response.getWriter();
```

```
    System.out.println("autenticato="+autenticato);
```

```
    out.println(uscita);
```

```
out.close();
```

```
}
```