

Servlets 2

Cookie Tricky Timing

Set on response

Read on later request

Timing: must plan ahead, send the cookie now, so you can get it back later

Cookie vs. Param

Cookies come in on the request

So do form parameters (bindings)

Web app must be able to deal with one or the other or both

I check for parameters before cookies, so the parameters take precedence

Cookie Features

(previous handout)

Servlet code

Domain

Lifetime

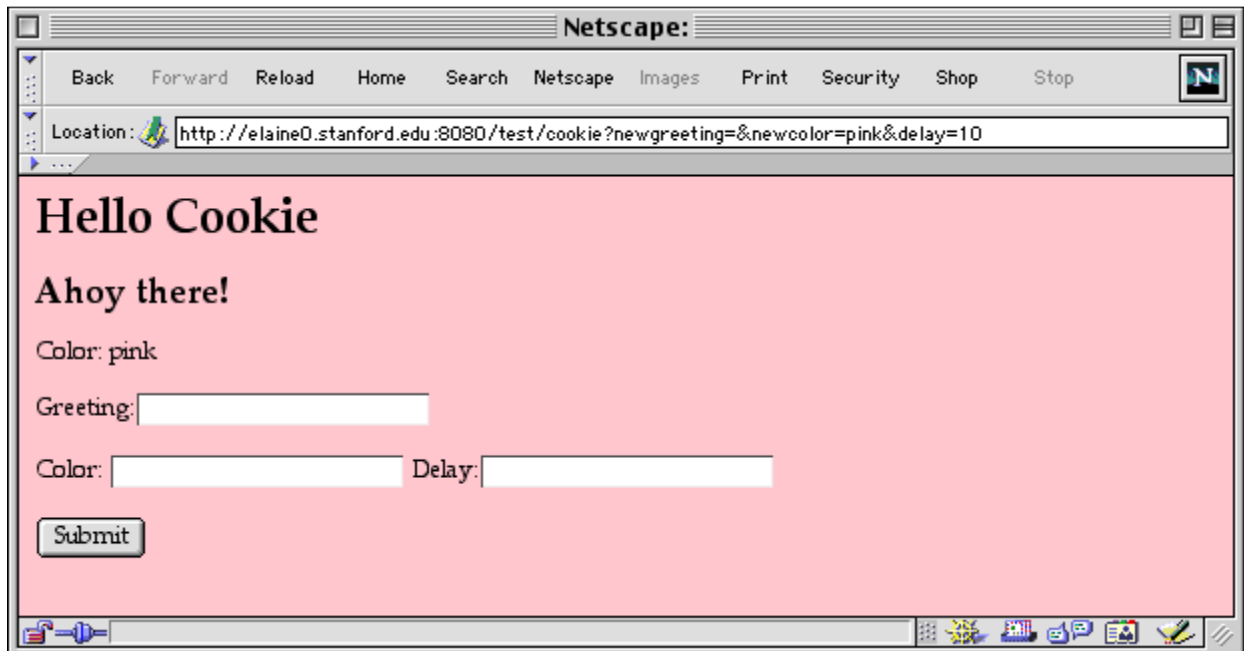
 session -- until browser exits

 # of seconds -- stored in file, works across browser exits

Cookie Example

Stores greeting as persistent cookie -- 1e9 seconds

Stores color with user defined lifetime



```
// HelloCookie.java
/*
 A servlet that demonstrates the basic features of cookies.
*/
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloCookie extends HttpServlet
{
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws IOException, ServletException
    {
        String color = "white"; // default color
        String greeting = "Hello"; // default greeting

        String param = null;
        Cookie cookie = null;

        // 1. Deal with greeting
        // Note: check for param before cookie,
        // addCookie() before response is sent
        param = request.getParameter("newgreeting");
        if (param!=null && !param.equals("")) {
            greeting = param;
            // Store a persistent cookie back to the client on the response
            Cookie c = new Cookie("greetingCookie", greeting);
            c.setMaxAge(1000000000); // 1 billion secs = (30 yrs)
            response.addCookie(c);
        }
        else {
            // Check for the greeting cookie
            cookie = findCookie(request, "greetingCookie");
            if (cookie != null) greeting = cookie.getValue();
        }
    }
}
```

```

// 2. Deal with color preference
param = request.getParameter("newcolor");
if (param!=null && !param.equals("")) {
    color = param;

    // Store back a cookie, using the given max-age
    cookie = new Cookie("colorCookie", color);

    String delay = request.getParameter("delay");
    if (delay!=null && !delay.equals("")) {
        cookie.setMaxAge(Integer.parseInt(delay));
    }

    response.addCookie(cookie);
}
else {
    cookie = findCookie(request, "colorCookie");
    if (cookie != null) color = cookie.getValue();
}

response.setContentType("text/html");
PrintWriter out = response.getWriter();

out.println("<html><body>");
out.println(" bgcolor=\"\" + color + "\"");
out.println(">");
out.println("<h1>Hello Cookie</h1>");

out.println("<h2>" + greeting + "</h2>");
out.println("<p>Color: " + color);

out.println("<form><p>Greeting:<input type=text name=newgreeting>");
out.println("<p>Color: <input type=text name=newcolor>");
out.println("Delay:<input type=text name=delay><p><input type=submit
value=Submit>");
out.println("</form>");

out.println("</body></html>");
}

// Finds the cookie with the given name, or returns null.
// (The cookies!=null test should not be necessary, but I
// I saw one case where it was needed because of a bug in the
// servlet container)
public Cookie findCookie(HttpServletRequest request, String name) {
    Cookie[] cookies = request.getCookies();

    if (cookies!=null) {
        for (int i=0; i<cookies.length; i++) {
            if (cookies[i].getName().equals(name)) return(cookies[i]);
        }
    }

    return(null);
}
}

```

Mozilla Cookie Features

The open source "mozilla" browser (mozilla.org) has good cookie features

Turn on the "warn about cookies" option

Use the "manage cookies" option in the security preferences

Application: Session

Servlet sessions are implemented with a session cookie

Application: Remember login

e.g. Amazon

e.g. Slashdot

The browser knows its me -- uses my prefs to display things

Amazon -- Half Auth

Amazon stores a cookie to remember the user

However, amazon does not really know who is on the browser

Therefore, amazon treats it as a "half auth" -- the user can see the cart, but not complete an order

To complete and order, the user must give the password (full auth)