Lecture 7
Handling Cookies

Lecture Agenda

Applied

1. Understanding the benefits and drawbacks of cookies.
2. Sending outgoing cookies.
3. Receiving incoming cookies.
4. Track repeat visitors.
5. Session cookies vs Persistent cookies
6. Simplifying cookie usage with utility classes.
7. Modifying cookie values
8. Remembering user preferences.

What is a Cookie?

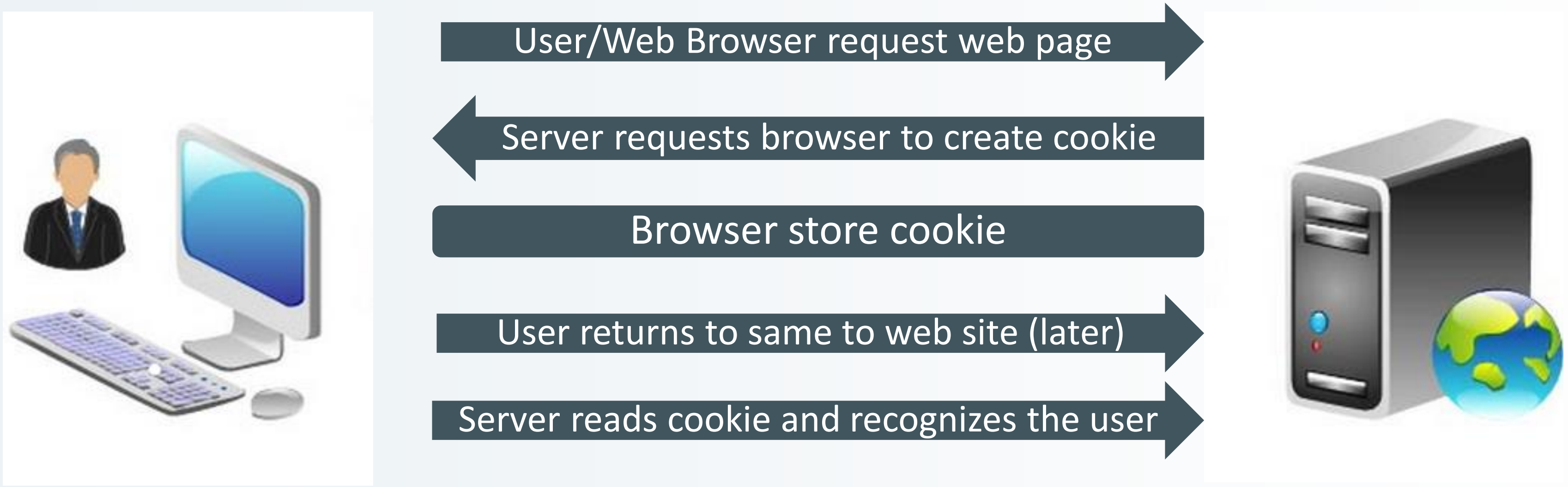
What is a Cookie?

Cookie background

- Cookie:
  - A small amount of information sent by a server to a browser, and then back by the browser on future page requests.
- Cookies have many uses:
  - authentication
  - user tracking
  - maintaining user preferences, shopping carts, etc.
- Cookie composition
  - A cookie’s data consists of a single name/value pair, sent in the header of the clients HTTP GET or HTTP POST request.

What is a Cookie?

High-Level Process



User/Web Browser request web page

Server requests browser to create cookie

Browser store cookie

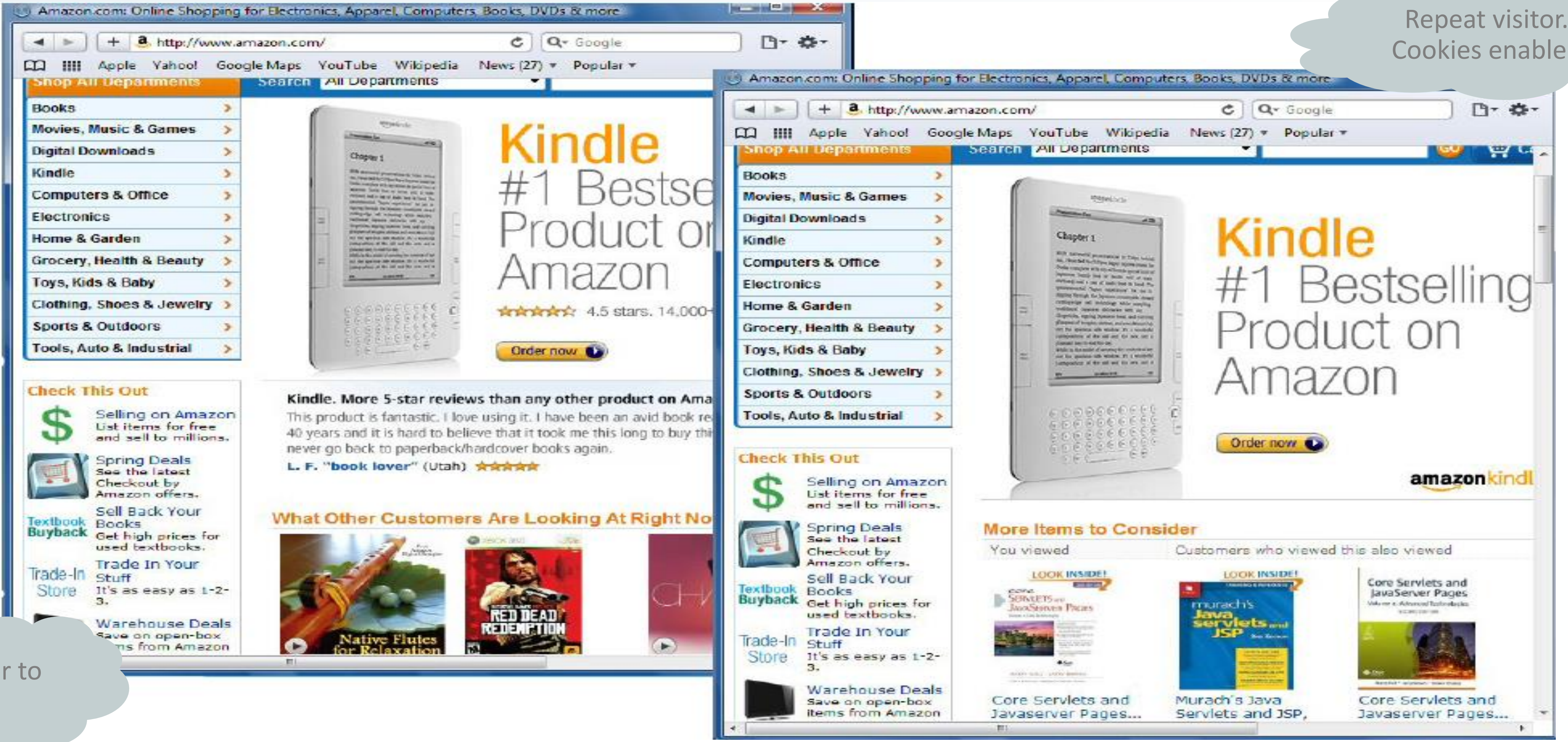
User returns to same to web site (later)

Server reads cookie and recognizes the user

What is a Cookie?

Cookie background

Repeat visitor. Cookies enabled



Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & more

http://www.amazon.com/

Kindle. More 5-star reviews than any other product on Ama

This product is fantastic. I love using it. I have been an avid book re 40 years and it is hard to believe that it took me this long to buy thi never go back to paperback/hardcover books again.

L. F. "book lover" (Utah) ★★★★★

New visitor to site

Some Problems with Cookies

Cookie background

- Privacy:
  - Servers can remember your previous action.
  - If you provide your information to a cookie, servers can link that information to your previous actions.
  - Servers can share cookie information
  - Poorly designed site store sensitive information like credit-card numbers directly in cookie.

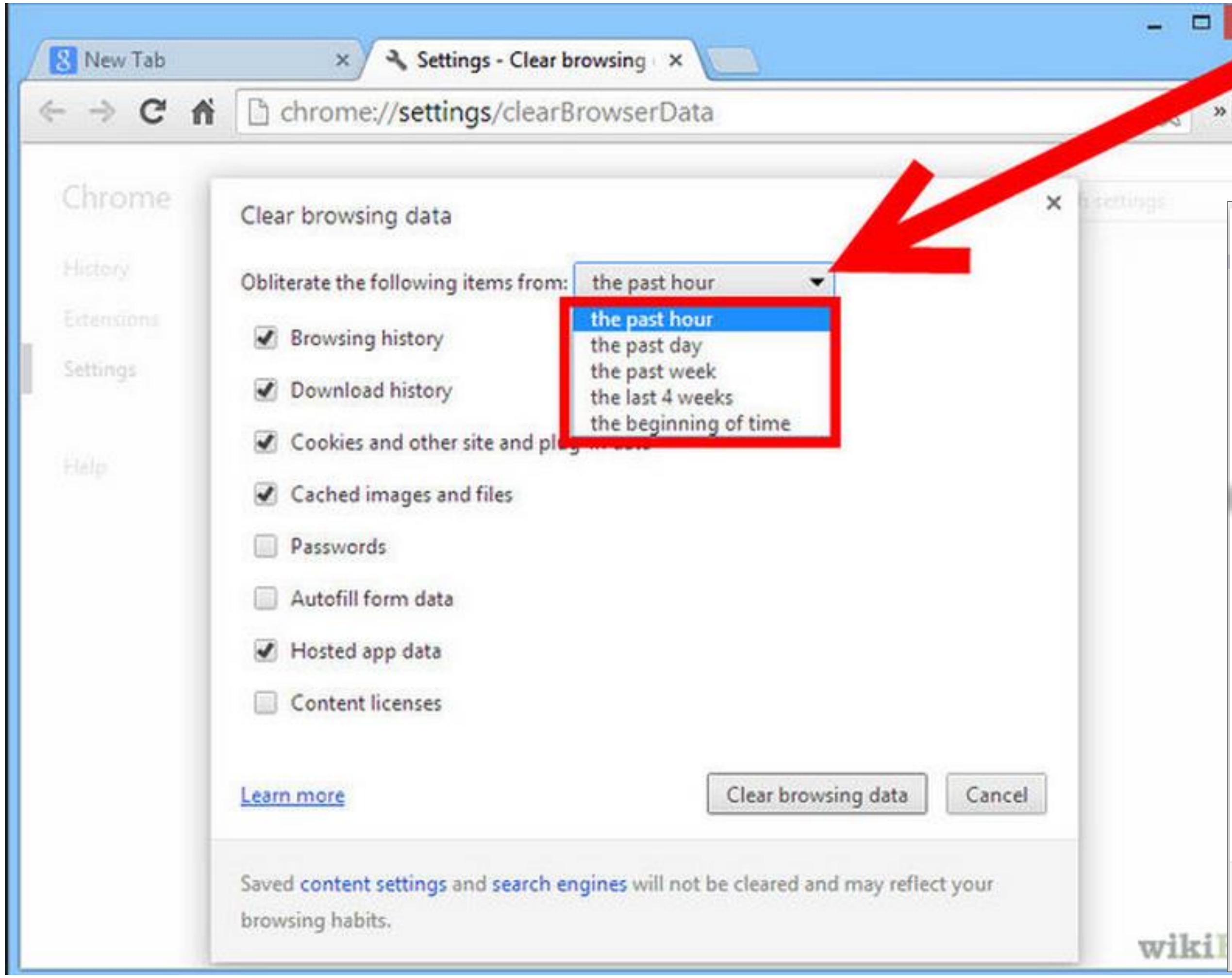
Servlet Authors – Cookie Moral

Cookie Develop Guide

- Moral for Servlet Authors:
  - If cookies are not critical to your task, avoid servlets that totally fail when cookies are disabled.
  - Don’t put sensitive information in cookies.

Manually Deleting Cookies

Deleting cookies for testing etc …



Clear browsing data

Obliterate the following items from: the past hour

- the past hour
- the past day
- the past week
- the last 4 weeks
- the beginning of time

☑ Browsing history

☑ Download history

☑ Cookies and other site and pl

☑ Cached images and files

□ Passwords

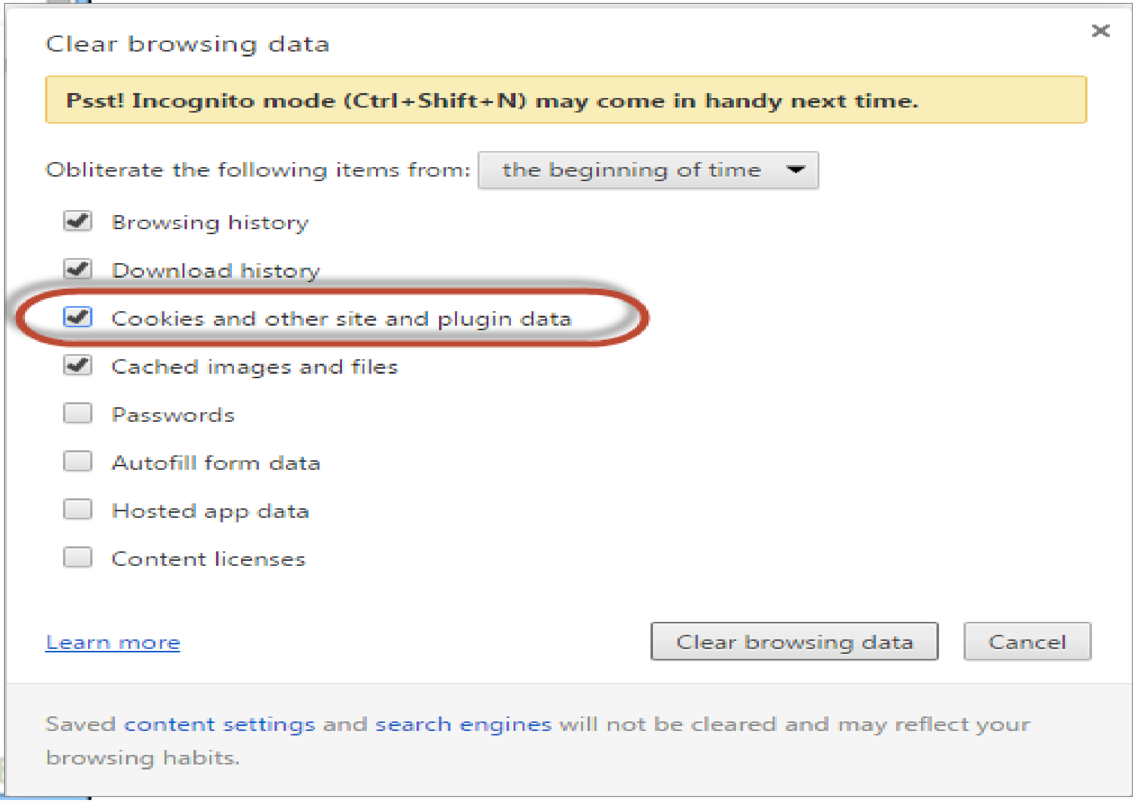
□ Autofill form data

☑ Hosted app data

□ Content licenses

Learn more · Clear browsing data · Cancel

Saved content settings and search engines will not be cleared and may reflect your browsing habits.



Clear browsing data

Psst! Incognito mode (Ctrl+Shift+N) may come in handy next time.

Obliterate the following items from: the beginning of time

☑ Browsing history

☑ Download history

☑ Cookies and other site and plugin data

☑ Cached images and files

□ Passwords

□ Autofill form data

□ Hosted app data

□ Content licenses

Learn more · Clear browsing data · Cancel

Saved content settings and search engines will not be cleared and may reflect your browsing habits.

Sending Cookies to the Client

Writing Cookies

- Create a Cookie object
  - Call the cookie constructor with a cookie name and a cookie value (both Strings).

```
Cookie c = new Cookie(“userId”, “a1234”);
```

- Set maximum age
  - To tell the browser to store a cookie on disk instead of just in memory.
  - Argument in seconds

```
c.setMaxAge( 60 * 60 * 24 * 7 );
```

- Place Cookie in response
  - Call the cookie constructor with a cookie name and a cookie value (both Strings).

```
Cookie c = new Cookie(“userId”, “a1234”);
```

Reading Cookies from the Client

Reading Cookies

- Call request.getCookies()
  - This returns an array of Cookie objects.
- Example: Looping through the Cookie object array:

```
String cookieName  = ”userId”;
Cookie[] cookies = request.getCookies();
if(cookies != null) {
for(Cookie cookie: cookies){
    if(cookieName.equals(cookie.getName()){
      doSomething(cookie.getValue())
    }
 }
}
```

Get all cookie objects

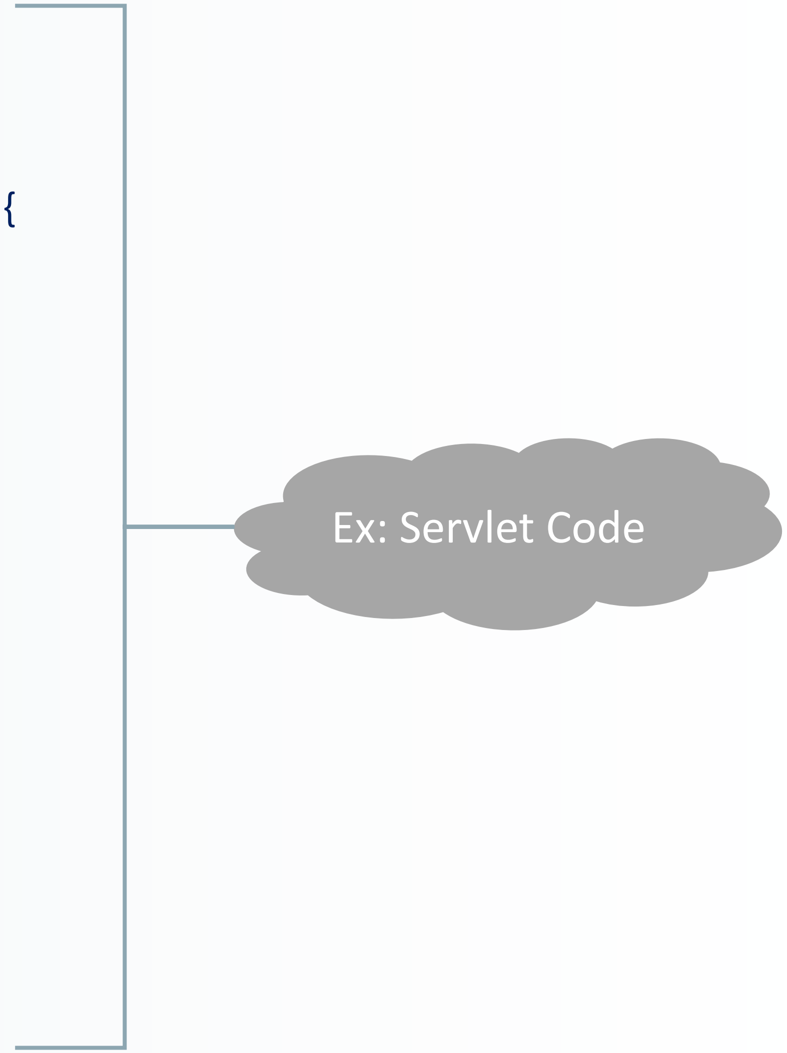
Loop through cookie objects

Using Cookies to Determine First-Time Web Site Visitor

First-Time Visitor

Extract Sample Code: Using cookies to determine first-time visitor

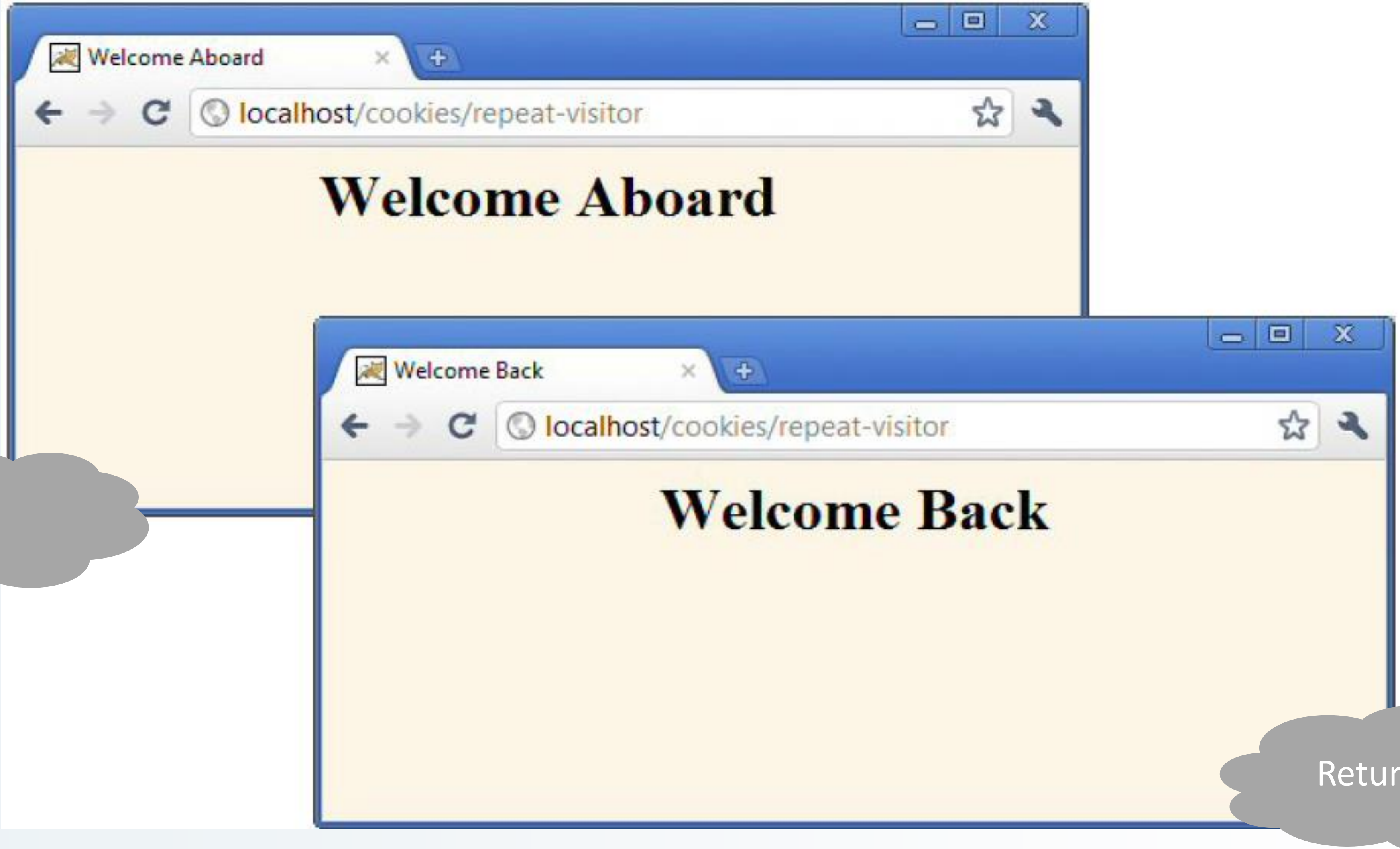
```
boolean newbie = true;
Cookie[] cookies = request.getCookies();
If( cookies != null ){
            for( Cookie c: cookies ){
             if( (c.getName).equals(“repeatVisitor”)) &&  (c.getValue().equals(“yes”))){
                  newbie = false;
                  break;
             }
         }
}
String title;
If(newbie) {
   Cookie returnVisitorCookie = new Cookie(“repeatVisitor”, “yes”);
   returnVisitorCookie.setMaxAge(60 * 60 * 24 * 365);  // 1 year
   response.addCookie(returnVisitorCookie);
   title = “Welcome Aboard”;
else {  title = “Welcome Back”; }
    // CREATE HTML OUTPUT
```



Ex: Servlet Code

First-Time Visitor

Result



Welcome Aboard

localhost/cookies/repeat-visitor

Welcome Aboard

Welcome Back

localhost/cookies/repeat-visitor

Welcome Back

First Time Visitor

Return Visitor

Setting/Using Cookie Attributes

Response Method	Description
getDomain() / setDomain()	- Lets you specify domain to which cookie applies. cookie.setDomain("yourdomain.com");
getMaxAge() / setMaxAge()	- Gets/Sets the cookie expiration time (in seconds). - If this is not set, the cookie applies only to current session.
getName()	- Gets the cookie name. - These is no setName() method you must supply your desired name to the constructor.
getPath() / setPath()	- Gets/Sets the path to which the cookie applies. - If unspecified cookie applies to URLs that are within or below directory containing the current page. cookie.setPath("www.domain.com/home/index.jsp"); *cookie available to all pages in home folder and its subfolders
getValue()/setValue()	- Gets/Sets value associated with cookie. - For new Cookie, value is supplied to the constructor (new Cookie())

Session Cookies
versus
Persistent Cookies

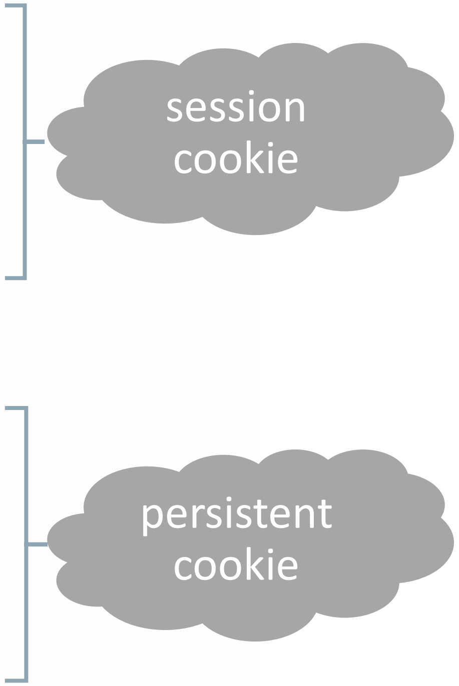
Session Cookie vs. Persistent Cookie

Session Cookie	Persistent Cookie
• These are temporary cookie files	• These files will remain on the client hard-drive
• The file (cookie) is erased once the browser is closed	• The file (cookie) is erased either manually or once max age expires.
• Short-Lived cookie	• Long-Lived cookie.

Session Cookies vs. Persistent Cookies

Differentiating between Session Cookies and Persistent Cookies

```
for (int =0; i<3; i++) {
  Cookie cookie = new Cookie(“Session-Cookie-” + i, “Cookie-Value-S” + i);
   //no maxAge (ie default = -1)
  response.addCookie(cookie);

                                  for
  cookie = new Cookie(“Persistent-Cookie-” + i, “Cookie-Value-P” + i);
  cookie.setMaxAge(3600);  // 1 hour
  response.addCookie(cookie);
}
```



session cookie

persistent cookie

Building Cookie Helper Utilities

Cookie Utility

Example: Finding Cookies with Specified Names

```
 public static String getCookieValue(HttpServletRequest request,
                                        String cookieName, String defaultValue) {
Cookie[] cookies = request.getCookies();
if( cookies != null ) {
    for( Cookie cookie : cookies ){
          if(cookieName.equals(cookie.getName())){
              return (cookie.getValue());
          }
    }
}
 return {defaultValue);
}
```

for
search for
cookie by name

Cookie Utility

Example: Creating Long-Lived Cookies

```
public class LongLivedCookie extends Cookie {

   public static final int SECONDS_PER_YEAR = 60 * 60 * 24 * 365;  // 1 year

   public LongLivedCookie(String name, String value){
         super(name, value);
        setMaxAge(SECONDS_PER_YEAR )
    }
 }
```

Constructor sets cookie life of 1 year

Applying Cookie Helper Utility

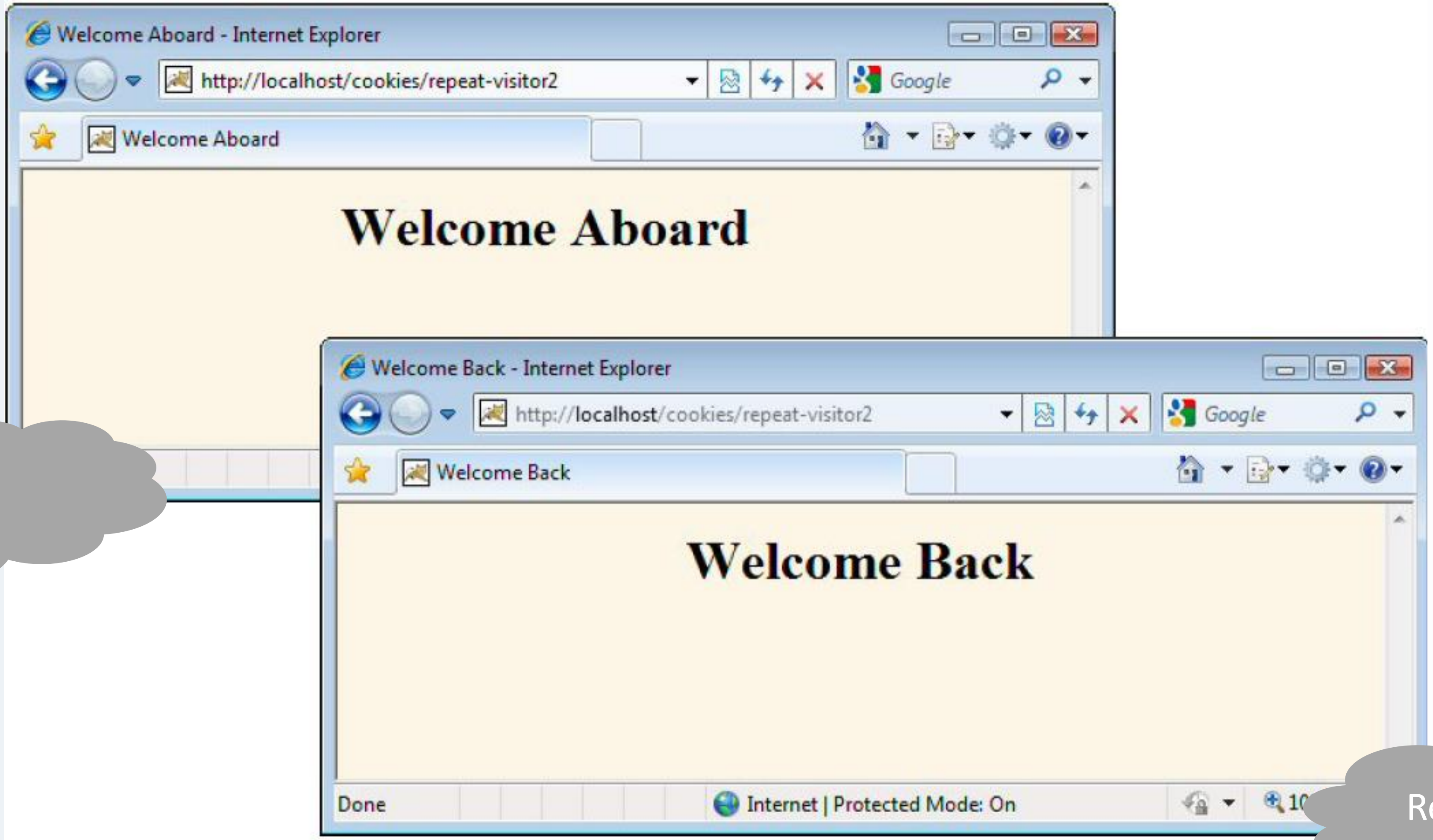
Applying Cookie Utility

Applying Cookie Utility Helper

```
boolean newbie = true;
String value = CookieUtilities.getCookieValue(request, “repeatVisitor2”, “no”);
If(value.equals(“yes”)) {
   newbie = false;
}
String title;
If( newbie ){
   LongLivedCookie returnVisitorCookie = new LongLivedCookie(“repeatVisitor2”, “yes”);
   response.addCookie(returnVisitorCookie);
   title = “Welcome Aboard”;
} else{  title = “Welcome Back”}
```

Applying Utilities

Result



Welcome Aboard - Internet Explorer

http://localhost/cookies/repeat-visitor2

Welcome Aboard

Welcome Aboard

Welcome Back - Internet Explorer

http://localhost/cookies/repeat-visitor2

Welcome Back

Welcome Back

Done

Internet | Protected Mode: On

First Time Visitor

Return Visitor

Modifying Cookie Value

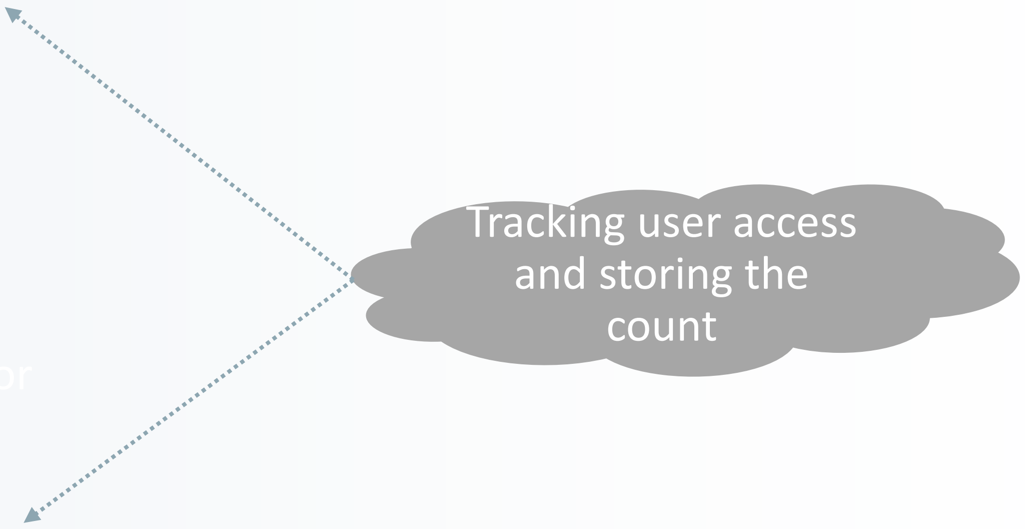
Modifying Cookie Values

Goal	Instructions
Replacing a cookie	1. Send the same cookie name with a different cookie value. 2. Call response.addCookie() not merely setValue() 3. Must reapply setMaxAge() etc … Usually not worth the bother, better to create a new Cookie();
Instruct the browser to delete a cookie	• Call setMaxAge(0).

Tracking User Access Counts

Example: Tracking User Access

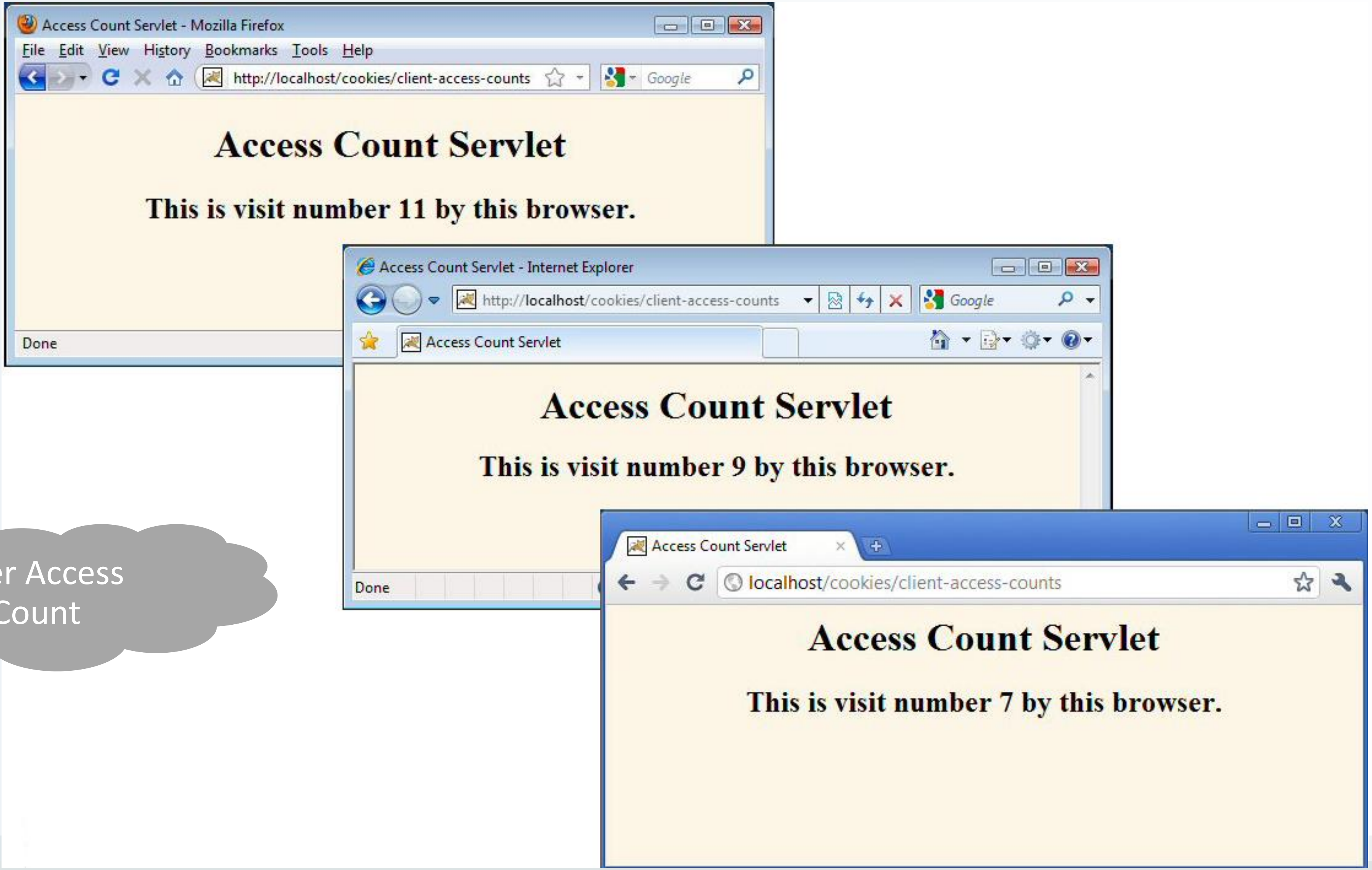
```
String countString = CookieUtilities.getCookieValue(request, “accessCount”, ‘’0”);
int count = 0;
try {
   count = Integer.parseInt(countString);

} catch(NumberFormatException nfe){
    nfe.getMessage();
}
LongLivedCookie  c = new LongLivedCookie(“accessCount”,  String.valueOf(count+1));
Response.addCookie(c);

//OUTPUT HTML
```



Tracking user access and storing the count

Tracking User Access Counts

Result


Access Count Servlet - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/cookies/client-access-counts
Google
Done
Access Count Servlet - Internet Explorer
http://localhost/cookies/client-access-counts
Google
Access Count Servlet
Done
Access Count Servlet
localhost/cookies/client-access-counts

Access Count Servlet

This is visit number 11 by this browser.

Access Count Servlet

This is visit number 9 by this browser.

Access Count Servlet

This is visit number 7 by this browser.

User Access Count

Using Cookies to Remember User Preferences

User Preferences

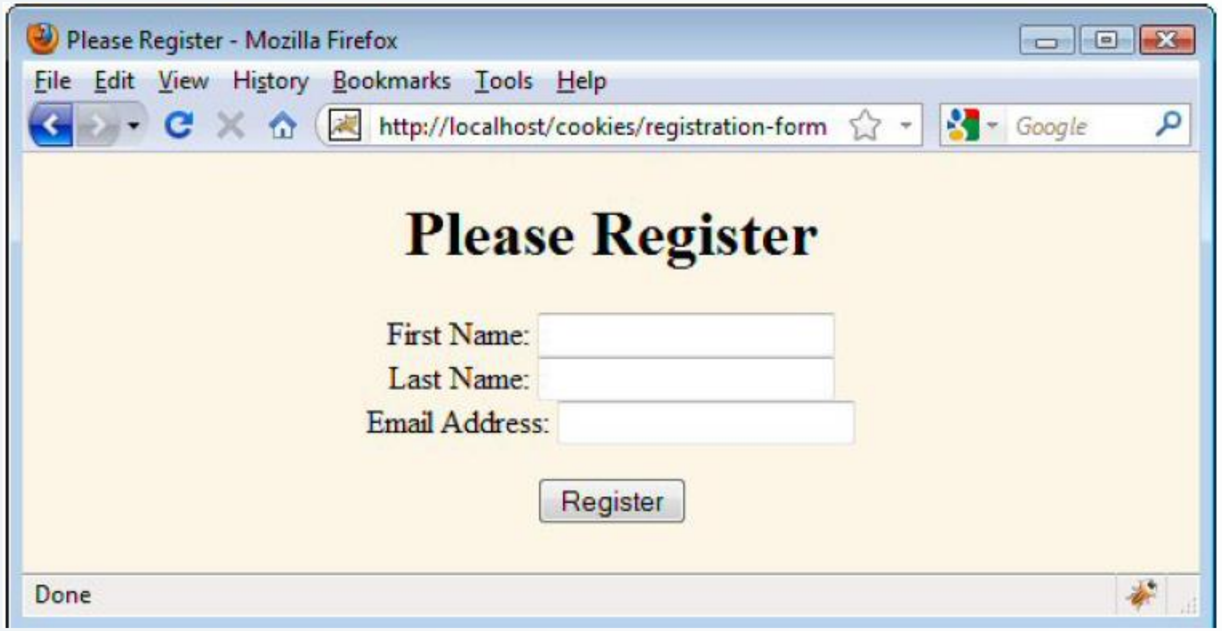
Design: Using Cookies to Remember User Preferences

1. Registration Form Servlet
   - Use cookie value to prepopulate form field values
   - Uses default values if no cookies are found.

2. Registration Servlet
   - Creates cookies based on request parameters received
   - Displays values if all parameters are present
   - Redirects to form if any parameter is missing

Registration Form

Initial Form



Please Register - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/cookies/registration-form
Google

Please Register

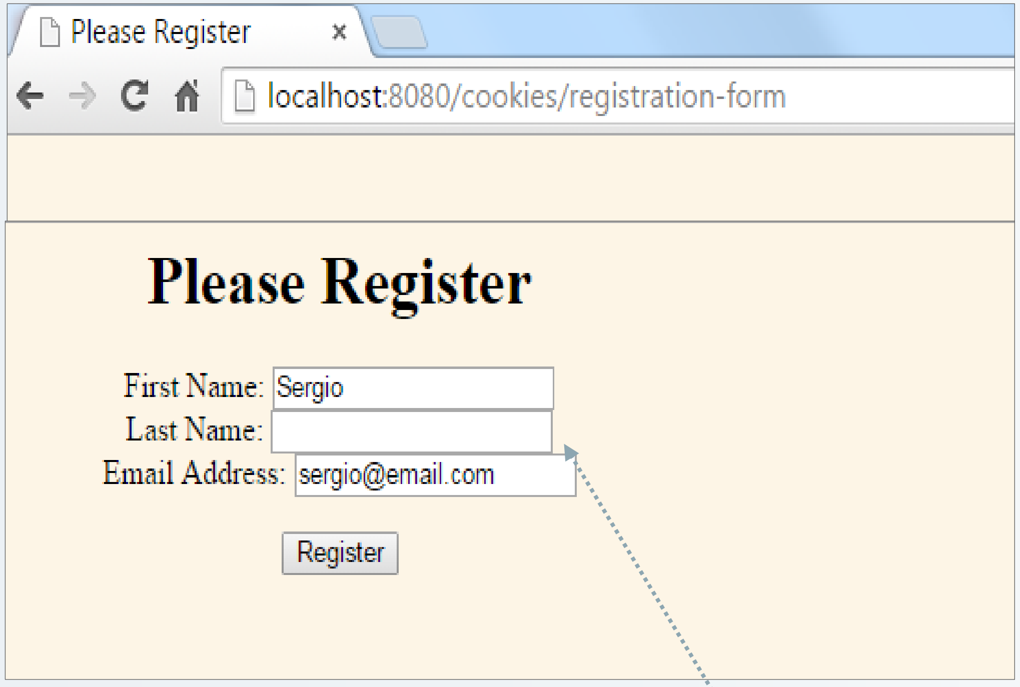
First Name:

Last Name:

Email Address:

Registration Form

Submitting Incomplete Form



Please Register

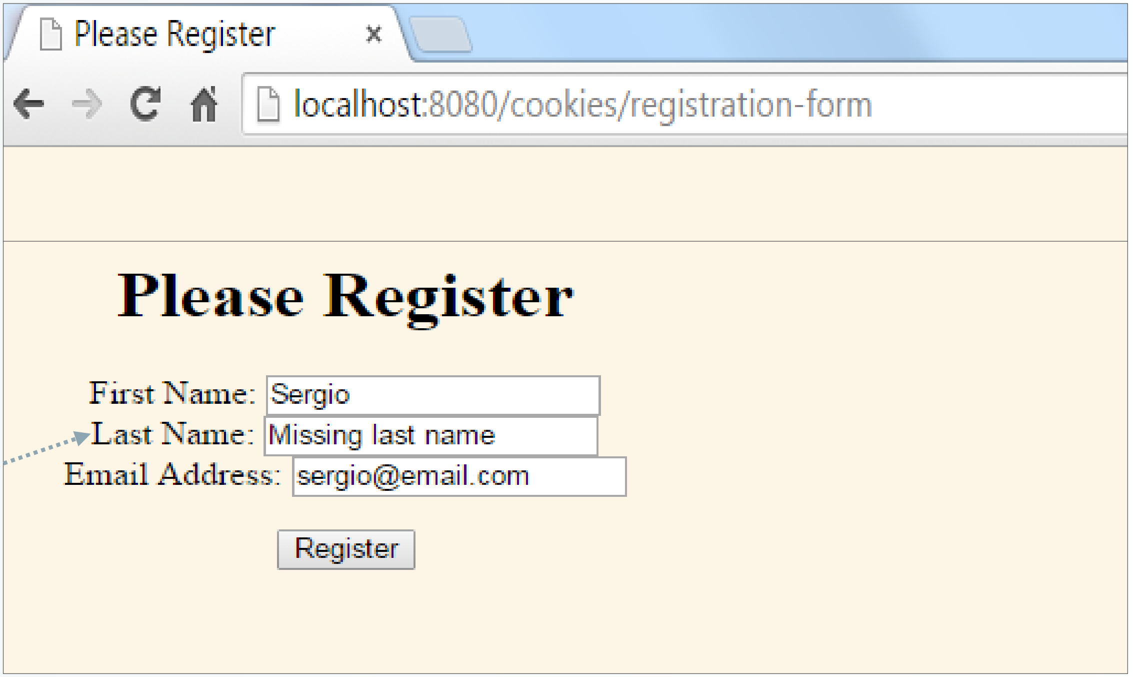
localhost:8080/cookies/registration-form

Please Register

First Name: Sergio

Last Name:

Email Address: sergio@email.com



Please Register

localhost:8080/cookies/registration-form

Please Register

First Name: Sergio

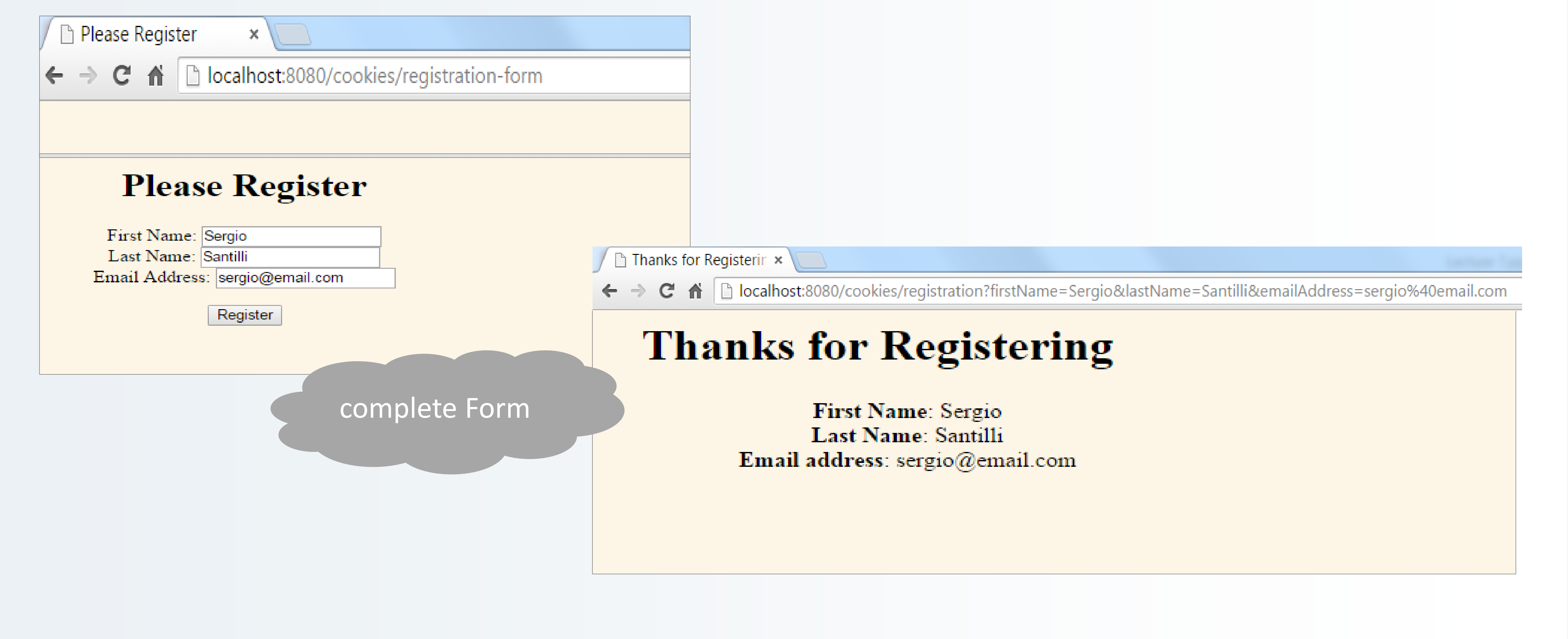
Last Name: Missing last name

Email Address: sergio@email.com

Incomplete Form

Registration Form

Submitting a Complete Form



localhost:8080/cookies/registration-form

Please Register

First Name: Sergio
Last Name: Santilli
Email Address: sergio@email.com

complete Form

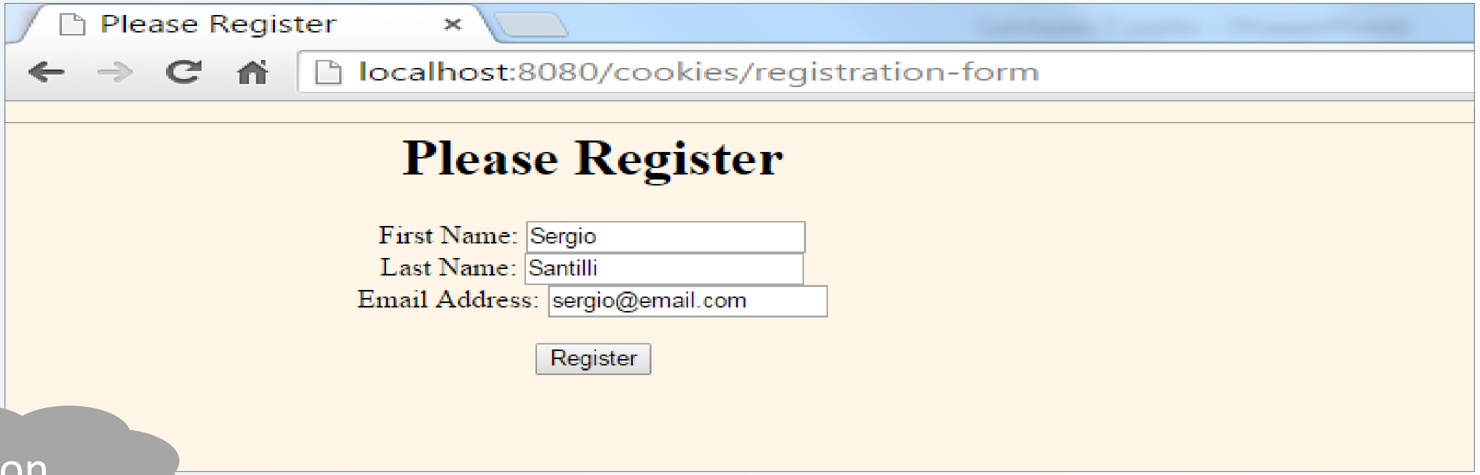
localhost:8080/cookies/registration?firstName=Sergio&lastName=Santilli&emailAddress=sergio%40email.com

Thanks for Registering

First Name: Sergio
Last Name: Santilli
Email address: sergio@email.com

Registration Form

Initial Result on Later Visit



Please Register

First Name: Sergio
Last Name: Santilli
Email Address: sergio@email.com

Initial Result on Later Visit

RegistrationForm Servlet

Servlet: registration-form

```
@WebServlet(“/registration-form”)
public class RegistrationForm extends HttpServlet {

   public void doGet(HttpServletRequest request, HttpServletResponse response){
        response.setContentType(“text/html”);
        PrintWriter out = response.getWriter();

        String firstName     =  CookieUtilities.getCookieValue(request, “firstName”, “”);
        String lastName      =  CookieUtilities.getCookieValue(request, “lastName”, “”);
        String emailAddress =  CookieUtilities.getCookieValue(request, “emailAddress”, “”);

        //CONSTRUCT HTML OUTPUT HERE
   }
}
```

read cookie values

Registration Servlet

Servlet: registration

```
@WebServlet("/registration")
public class RegistrationServlet extends HttpServlet {


  public void doGet(doGet(HttpServletRequest request, HttpServletResponse response){
      response.setContentType("text/html");
      boolean isMissingValue = false;

      String firstName = request.getParameter("firstName");
      if( isMissing(firstName) ){
          firstName = "Missing first name";
          isMissingValue = true;
      }

      String lastName = request.getParameter("lastName");
      if( isMissing(lastName) ){
          lastName = "Missing last name";
          isMissingValue = true;
      }
```

test cookie values
if exist

Registration Servlet Continued …

Servlet: registration

```
    Cookie c1 = new LongLivedCookie(“firstname”, firstName ) ;
    response.addCookie(c1);

    Cookie c2 = new LongLivedCookie(“lastname”, lastName ) ;
    response.addCookie(c2);

    if( isMissingValue ){
     response.sendRedirect(“registration-form”);
    } else {
        // Do Something …
    }


  }
}
```

set persistent cookie(s)

If any data missing, redirect back to form

for

Questions?