The Supercomputing People

Corporate Profile

Session 4:

Servlet Session Tracking

Contents

Session Tracking

- Session tracking is a mechanism that servlets use to maintain state about a series of requests from the same user (that is, requests originating from the same browser) across some period of time.
- Sessions are shared among the servlets accessed by a client.
- This is convenient for applications made up of multiple servlets.
- For example, a **Bookstore** system uses session tracking to keep track of the books being ordered by a user.

Why do we need session tracking?

- HTTP is a stateless protocol:
  - it doesn't provides a way for a server to recognize which client is using what part of the application.
- Great number of web-oriented application requires that application has to keep track of the clients performing actions, and thus can't be supported without an additional API.
- To support the software that needs keep track of the state, **Java Servlet technology provides an API for managing sessions and allows several mechanisms for implementing sessions**.

Three ways for session tracking

- **Hidden Form Fields**
  - fields are added to an HTML form which are not displayed in the client's request.

  e.g., <input type ="hidden" name = "name" value=“***“>
- **Cookies**
  - By default the server creates a cookie and the cookie get stored on the client machine.
  - The cookie is sent back to the server when the user sends a new request. By this cookie, the server is able to identify the user. In this way the session is maintained. Cookie is nothing but a name- value pair, which is stored on the client machine.
- **URL Rewriting**
  - A session object is created when the first request goes to the server.
  - Then server creates a token which will be used to maintain the session.
  - The token is transmitted to the client by the response object and gets stored on the client machine.

What are cookies?

- A cookie is a *message* given to a *Web browser* by a *Web server*.
- The browser stores the message in a text file. The message is then sent back to the server each time the browser requests a page from the server.
- **Purpose**
  - to identify users and possibly prepare customized Web pages for them.
- Example
  - When you enter a Web site using cookies, you may be asked to fill out a form providing such information as your name and interests.
  - This information is packaged into a cookie and sent to your Web browser which stores it for later use.
  - The next time you go to the same Web site, your browser will send the cookie to the Web server.
  - The server can use this information to present you with custom Web pages.
  - So, for example, instead of seeing just a generic welcome page you might see a welcome page with your name on it.

Cookie API

Cookies

- Creating Cookies
- Cookie Attributes
- Reading Cookies
- Session & Persistent Cookies
- Examples
  1. Repeat Visitor
  2. Tracking User Access Counts
  3. Remembering User Preferences

Creating Cookies

- Three steps to creating a new cookie:

1) Create a new Cookie Object
   - Cookie cookie = new Cookie (name, value);

2) Set any cookie attributes
   - Cookie.setMaxAge (60);

3) Add your cookie to the response object:
   - Response.addCookie (cookie)

1. Cookie Constructor

- You create a new cookie by calling the Cookie constructor and specifying:
  - Name
  - Value
- Example:
  - Cookie cookie = new Cookie (“school”, “NYU”);
- Neither the name nor the value should contain *whitespace* or any of the following characters:
  - [ ] ( ) = , “ / ? @ ;

2. Set Cookie Attributes

- Before adding your cookie to the response object, you can set any of its attributes.

- Attributes include:
  - Name/Value
  - Domain
  - Maximum Age
  - Path
  - Version

Cookie Name

```
public String getName();
public void setName (String name);
```

- You rarely call setName() directly, as you specify the name in the cookie constructor.
- getName() is useful for reading in cookies.

Cookie Value

```
public String getValue();
public void setValue (String value);
```

- You rarely call setValue() directly, as you specify the name in the cookie constructor.
- getValue() is useful for reading in cookies.

Domain Attributes

```
public String getDomain ();
public void setDomain(String domain);
```

- Normally, the browser only returns cookies to the exact same host that sent them.
- You can use setDomain() to instruct the browser to send cookies to other hosts within the same domain.

Domain Example

- Example: Cookies sent from a servlet at bali.vacations.com would not be forwarded to mexico.vacations.com.

- If you do want to the cookie to be accessible to both hosts, set the domain to the highest level:
  - cookie.setDomain (“.vacations.com”);

- Note that you are always required to include at least two dots. Hence, you must specify .vacations.com, not just vacations.com

Cookie Age

```
public int getMaxAge ();
public void setMaxAge (int lifetime);
```

- In general there are two types of cookies:
  - <u>Session Cookies</u>: Temporary cookies that expire when the user exits the browser.
  - <u>Persistent Cookies</u>: Cookies that do <u>not</u> expire when the user exits the browser. These cookies stay around until their expiration date, or the user explicitly deletes them.

Cookie Expiration

- The **setMaxAge()** method tells the browser how long (in seconds) until the cookie expires.

- Possible values:
  - **negative value** (default): creates a session cookie that is deleted when the user exits the browser.
  - **0**: instructs the browser to delete the cookie.
  - **positive value**: any number of seconds.

    e.g., to create a cookie that lasts for one hour, setMaxAge (3600);

Path

```
public String getPath();
public void setPath (String path);
```

- By default, the browser will only return a cookie to URLs in or below the directory that created the cookie.

Path Example

- Example: If you create a cookie at http://ecommerce.site.com/toys.html then:
  - The browser will send the cookie back to http://ecommerce.site.com/toys.html
  - The browser will not send the cookie back to http://ecommerce.site.com/cds
- If you want the cookie to be sent to all pages, set the path to /
  - Cookie.setPath ("/");
  - Very common, widely used practice.

Cookie Version

```
public int getVersion ();
public void setVersion (int version);
```

- By default, the Servlet API will create Version 0 cookies.
- Via the **setVersion()** method you can specify version 1. But, since this is not widely implemented, stick with the default.

Security

```
public int getSecure ();
public void setSecure (boolean);
```

- If you set Secure to true, the browser will only return the cookie when connecting over an encrypted connection.
- By default, cookies are set to non-secure.

Comments

```
public int getComment ();
public void Comment (String)
```

- Comments: you can specify a cookie comment via the setComment() method. But, comments are only supported in Version 1 cookies.
- Hence, no one really uses these methods.

3. Add Cookies to Response

- Once you have created your cookie, and set any attributes, you add it to the response object.

- By adding it to the response object, your cookie is transmitted back to the browser.

- Example:

```
Cookie school = new Cookie (“school”, “NYU”);
school.setMaxAge (3600);
response.addCookie (school);
```

Reading Cookies

Reading Cookies

- To create cookies, **add** them ***to*** the **response object**.
- To read incoming cookies, **get** them ***from*** the **request object**.
- HttpServletRequest has a **getCookies()** method.
  - Returns an array of cookie objects. This includes all cookies sent by the browser.
  - Returns a zero-length array if there are no cookies.

Reading Cookies

- Once you have an array of cookies, you can iterate through the array and extract the one(s) you want.
- Our next few examples illustrate how this is done.

Session & Persistent Cookies

- **Session Cookies**
  - these are temporary and are erased when you close your browser at the end of your surfing session.
  - The next time you visit that particular site it will not recognise you and will treat you as a completely new visitor as there is nothing in your browser to let the site know that you have visited before.

- **Persistent Cookies**
  - these remain on your hard drive until you erase them or they expire.
  - How long a cookie remains on your browser depends on how long the visited website has programmed the cookie to last .

Understanding Examples

Example 1: RepeatVisitor.java

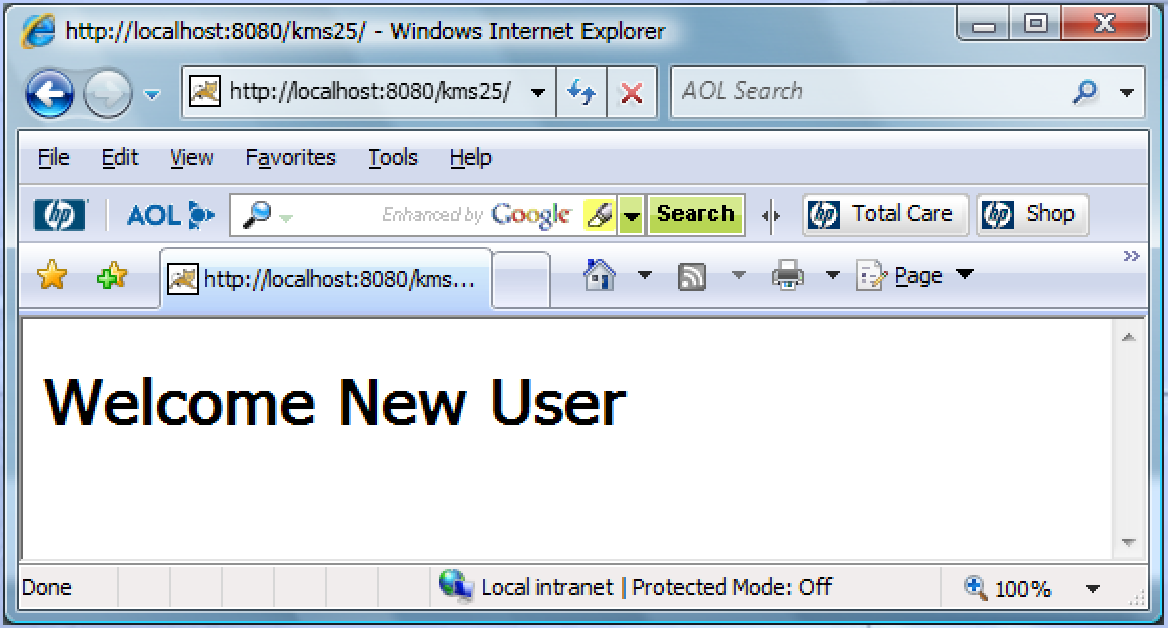
- This servlet checks for a unique cookie, named “repeatVisitor”
  - If the cookie is present, servlet says “Welcome Back”
  - Otherwise, servlet says “Welcome New User”.

```
public class RepeatVisitors extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
            boolean newUser = true;
            Cookie[ ] cookies = request.getCookies();
            if (cookies != null)
            {
                  for(Cookie c: cookies)
                  {
                        if ((c.getName().equals("repeatVisitor")) &&(c.getValue().equals("yes")))
                        {
                                    newUser = false;
                                    break;
                        }
                   }
            }
```

RepeatVisitor (Cont.)

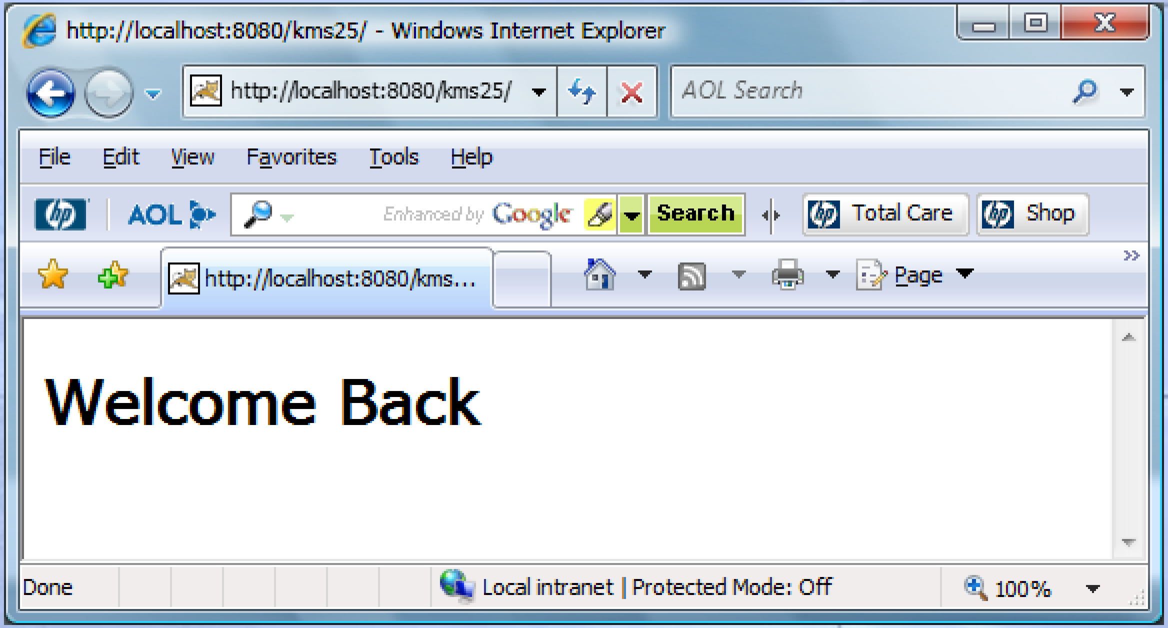
```
        String title=null;
        if (newUser)
        {
                Cookie returnVisitorCookie =new Cookie("repeatVisitor", "yes");
                returnVisitorCookie.setMaxAge(60*60*24*365);
                response.addCookie(returnVisitorCookie);
                title="Welcome New User";
        }
        else
        {
                title = "Welcome Back";
        }

                response.setContentType("text/html");
                PrintWriter out = response.getWriter();
                out.println("<H1>" + title + "<H1>");
    }
    public void doPost(HttpServletRequest request,HttpServletResponse response)
                throws ServletException, IOException
    {
                doGet(request,response);
    }
}
```

First Time Visit


http://localhost:8080/kms25/ - Windows Internet Explorer
http://localhost:8080/kms25/
AOL Search
File Edit View Favorites Tools Help
AOL
Enhanced by Google
Search
Total Care
Shop
http://localhost:8080/kms...
Page
Done
Local intranet | Protected Mode: Off
100%

Welcome New User

Next Time Visits


http://localhost:8080/kms25/ - Windows Internet Explorer
http://localhost:8080/kms25/
AOL Search
File Edit View Favorites Tools Help
AOL
Enhanced by Google
Search
Total Care
Shop
http://localhost:8080/kms...
Page
Done
Local intranet | Protected Mode: Off
100%

Welcome Back

Example 2: ClientAccessCount

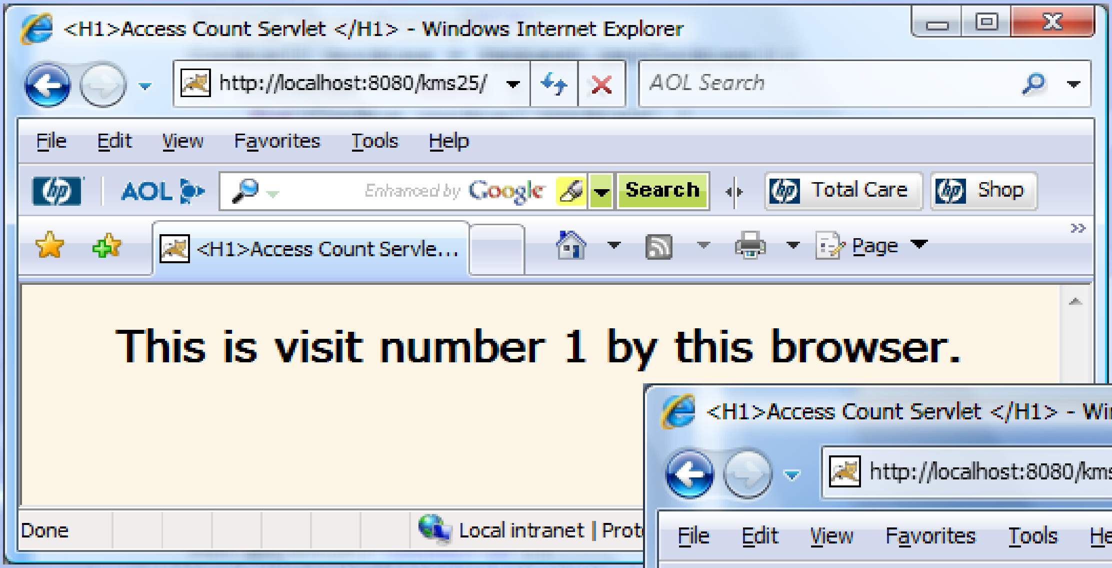
- We created a simple Counter servlet that keeps track of the number of “hits”.
- Now, we want to display the number of hits for *each* user.
- This is relatively simple to do. We just create a counter cookie, and increment the counter cookie at each visit.

```
public class ClientAccessCounts extends HttpServlet {
        public void doGet(HttpServletRequest request,HttpServletResponse response) throws
        ServletException, IOException {
                String countString="1";
                Cookie[ ] cookies = request.getCookies();
                if (cookies != null) {
                    for(Cookie cookie: cookies) {
                                if (cookie.getName().equals("accessCount"))
                                                countString=cookie.getValue();
                                }
                     }
                int count = 1;
                try {
                                count = Integer.parseInt(countString);
                } catch(NumberFormatException nfe) { }
                Cookie c = new Cookie("accessCount",String.valueOf(count+1));
                response.addCookie(c);
```

ClientAccessCount(cont.)

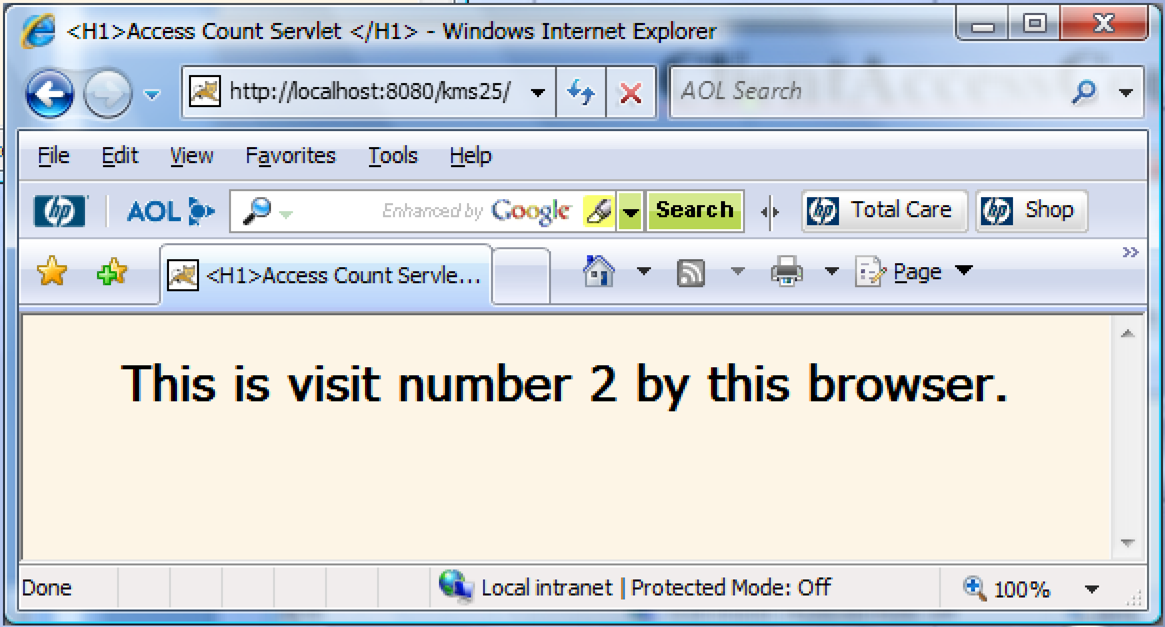
```
		response.setContentType("text/html");
		PrintWriter out = response.getWriter();
		out.println("<HTML>\n");
		out.println("<HEAD><TITLE> <H1>Access Count Servlet </H1> </TITLE>");
		out.println("<BODY BGCOLOR=\"#FDF5E6\">\n<CENTER>\n");
		out.println("<H2>This is visit number " + count + " by this browser.</H2>\n");
		}
}
```

ClientAccessCount(cont.)



<H1>Access Count Servlet </H1> - Windows Internet Explorer
http://localhost:8080/kms25/
AOL Search
File Edit View Favorites Tools Help
AOL
Enhanced by Google
Search
Total Care
Shop
<H1>Access Count Servle...
Page
Done
Local intranet | Prot

This is visit number 1 by this browser.



<H1>Access Count Servlet </H1> - Windows Internet Explorer
http://localhost:8080/kms25/
AOL Search
File Edit View Favorites Tools Help
AOL
Enhanced by Google
Search
Total Care
Shop
<H1>Access Count Servle...
Page
Done
Local intranet | Protected Mode: Off
100%

This is visit number 2 by this browser.

Example 3: Remembering User Preferences

- RegistrationForm servlet
  - Uses cookie values to prepopulate form field values
  - Uses default values if no cookies are found
  - Will also be done in JSP

- ServletRegistration servlet
  - Creates cookies based on request parameters received
  - Displays values if all parameters are present
  - Redirects to form if any parameter is missing

RegistrationForm servlet

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class RegistrationForm extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        String firstName = "“;
        String lastName = "“;
        String emailAddress = "“;
        Cookie[] cookies = request.getCookies();
        if (cookies != null) {
        for(Cookie cookie: cookies) {
                if (cookie.getName().equals("firstName"))
                                firstName=cookie.getValue();
                else if (cookie.getName().equals("lastName"))
                                lastName=cookie.getValue();
                else if (cookie.getName().equals("emailAddress"))
                emailAddress=cookie.getValue();
        }
}
```

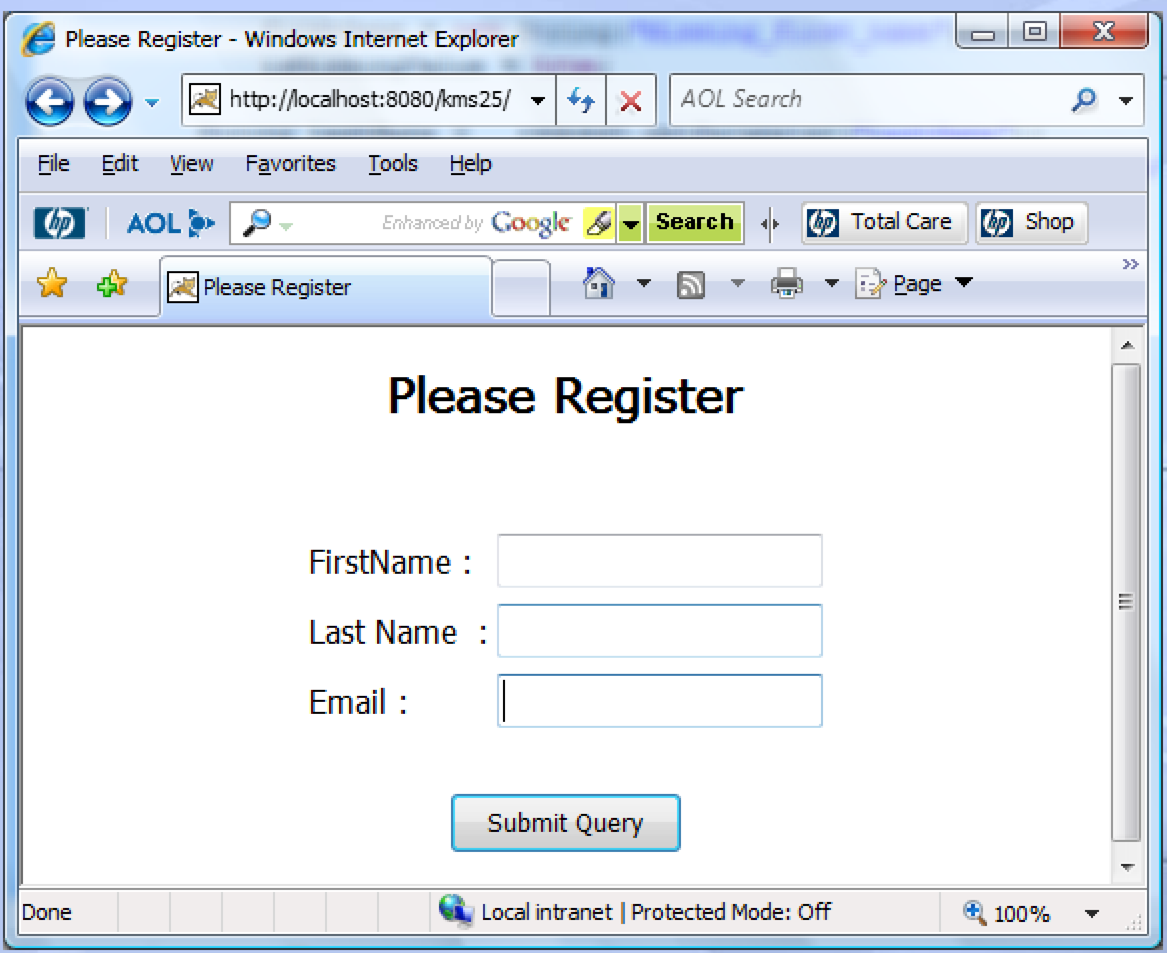
RegistrationForm servlet (Cont.)

```
response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<HTML>");
out.println("<HEAD> <CENTER> <TITLE> Please Register </TITLE>  </CENTER></HEAD>");
out.println("<BODY><CENTER>");
out.println("<H2> Please Register </H2><BR/>");
out.println("<FORM method='GET' ACTION=\"registrationServlet\">");
out.println("<TABLE>" +
        "<TR>" +
                "<TD>FirstName : </TD> " +
                "<TD><INPUT Type=\"text\" name=\"firstName\" value=" + firstName + "> </TD>" +
        "</TR>" +
        "<TR>" +
                "<TD>Last Name&nbsp : </TD>" +
                "<TD><INPUT Type=\"text\" name=\"lastName\" value=" + lastName + "> </TD>" +
        "</TR>" +
        "<TR>" +
                "<TD>Email : </TD>" +
                "<TD> <INPUT Type=\"text\" name=\"emailAddress\" value=" + emailAddress + "> </TD>" +
        "</TR>" +
  "</TABLE>" +
"<BR/><INPUT Type=\"submit\" name=\"submit\"> ");
```

RegistrationForm servlet(Cont.)

```
      out.println("</FORM>");
      out.println("</CENTER></BODY>");
      out.println("</HTML>");
  }
}
```

Remembering User Preferences (cont.)

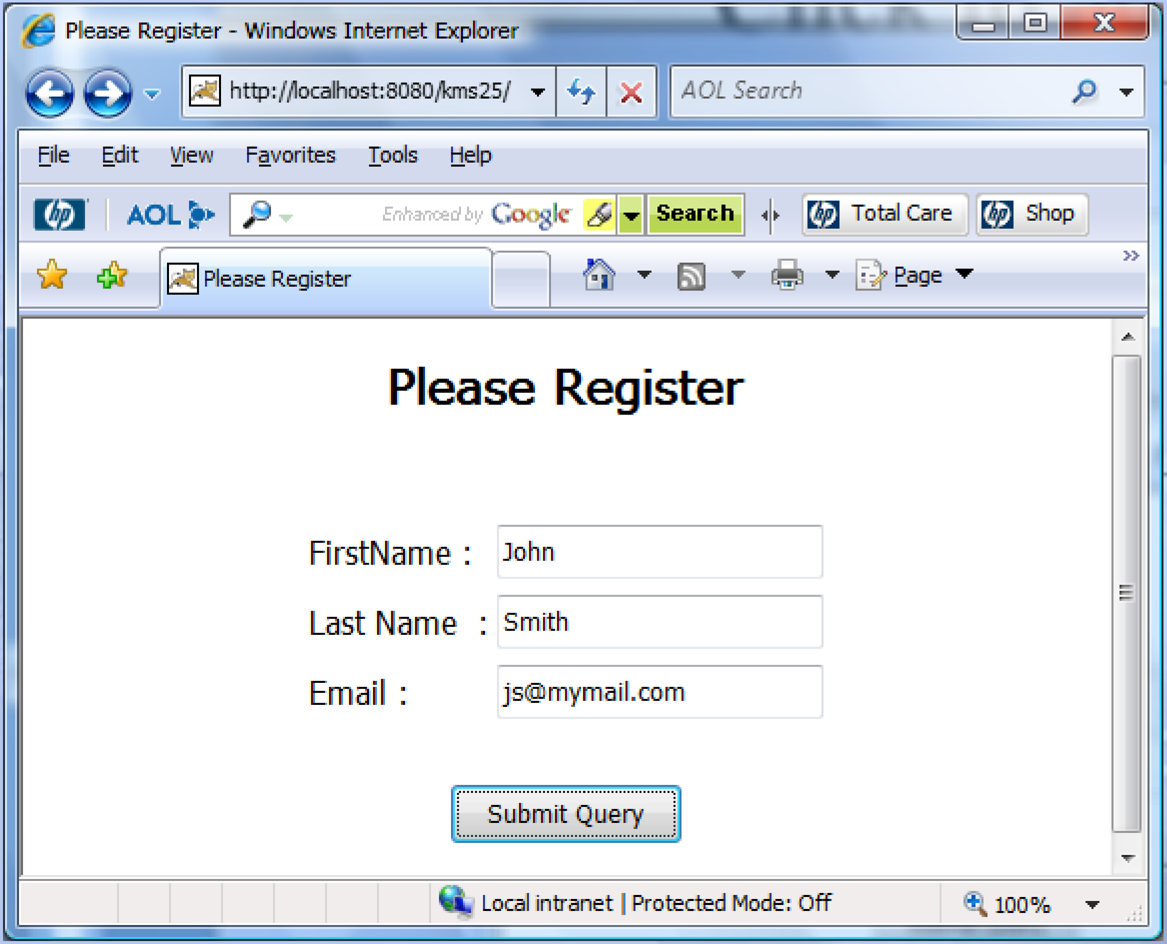

Please Register - Windows Internet Explorer
http://localhost:8080/kms25/
AOL Search
File Edit View Favorites Tools Help
Search
Total Care
Shop
Please Register
Page
Done
Local intranet | Protected Mode: Off
100%

Please Register

FirstName :	
Last Name :	
Email :	

Submit Query

Remembering User Preferences (cont.)


Please Register - Windows Internet Explorer
http://localhost:8080/kms25/
AOL Search
File Edit View Favorites Tools Help
AOL
Enhanced by Google
Search
Total Care
Shop
Please Register
Page
Local intranet | Protected Mode: Off
100%

Please Register

FirstName :	John
Last Name :	Smith
Email :	js@mymail.com

Submit Query

RegistrationServlet.java

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

public class RegistrationServlet extends HttpServlet {
public void doGet(HttpServletRequest request,HttpServletResponse response)
throws ServletException, IOException {
	response.setContentType("text/html");
	boolean isMissingValue = false;
	String firstName =request.getParameter("firstName");
	if (firstName.equals("") || firstName.equals("Missing_first_name"))
	{
		firstName = new String("Missing_first_name");
		isMissingValue = true;
	}
	String lastName =request.getParameter("lastName");
	if (lastName.equals("")|| lastName.equals("Missing_last_name"))
	{
		lastName = new String("Missing_last_name");
		isMissingValue = true;
	}
```

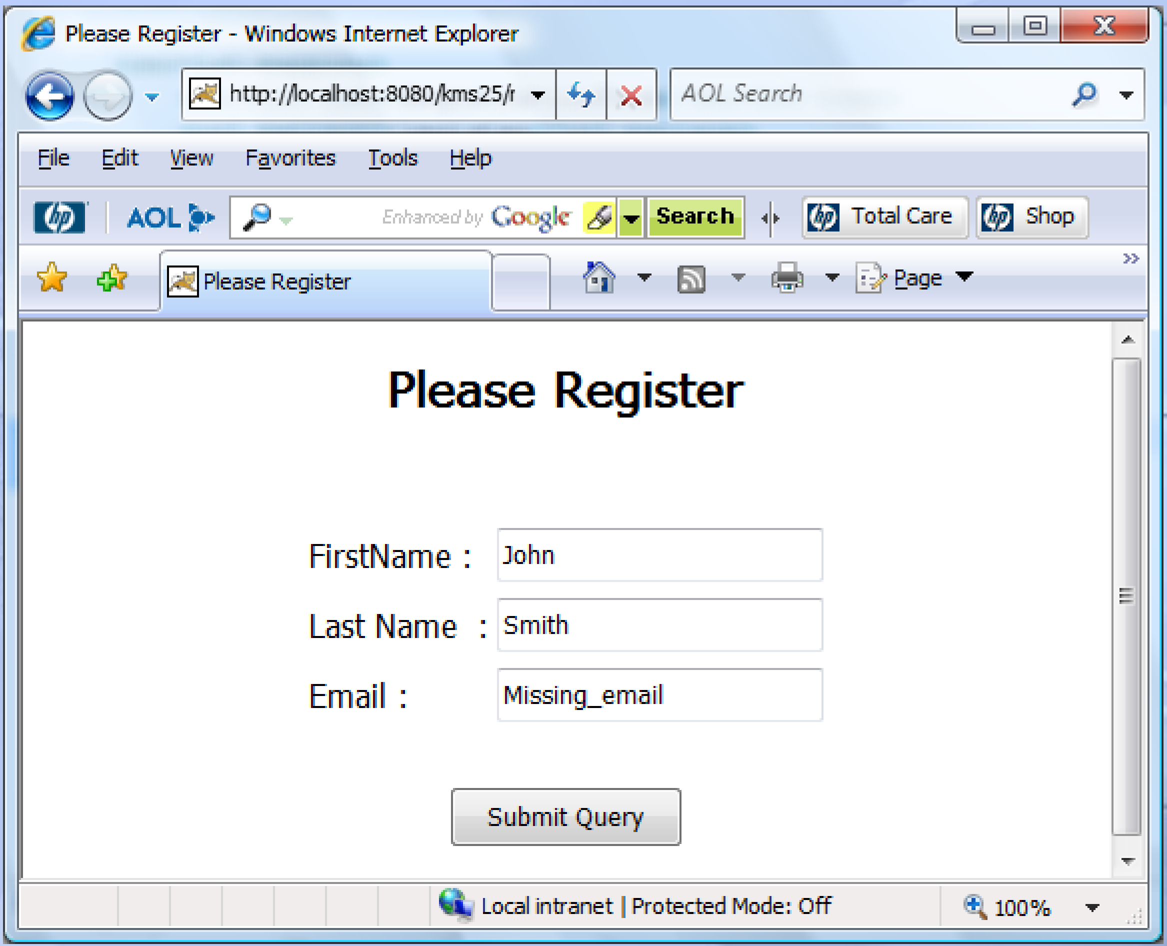
RegistrationServlet.java (cont.)

```
String email = request.getParameter("emailAddress");
if (email.equals("")|| email.equals("Missing_email")) {
    email = new String("Missing_email");
    isMissingValue = true;
}

    Cookie c1 =new Cookie("firstName“,firstName);
    c1.setMaxAge(60);
    response.addCookie(c1);
    Cookie c2 = new Cookie("lastName", lastName);
    response.addCookie(c2);
    c2.setMaxAge(60);
    Cookie c3 = new Cookie("emailAddress",email);
    response.addCookie(c3);
    c3.setMaxAge(60);
    String formAddress ="./register";
```

RegistrationServlet.java (cont.)

```
    if (isMissingValue) {
            response.sendRedirect(formAddress);
    }
    else
    {
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            out.println("<HTML>");
            out.println("<HEAD> <CENTER> <TITLE> thanks </TITLE>  </CENTER></HEAD>");
            out.println("<BODY><CENTER>");
            out.println("<H2> Thanks for registration! </H2><BR/>");
            out.println("<LI> First Name : " + firstName);
            out.println("<LI> Last Name : " + lastName);
            out.println("<LI> Email  : " + email);
            out.println("</CENTER> </BODY>");
            out.println("</HTML>");
    }
  }
}
```

Remembering User Preferences (cont.)

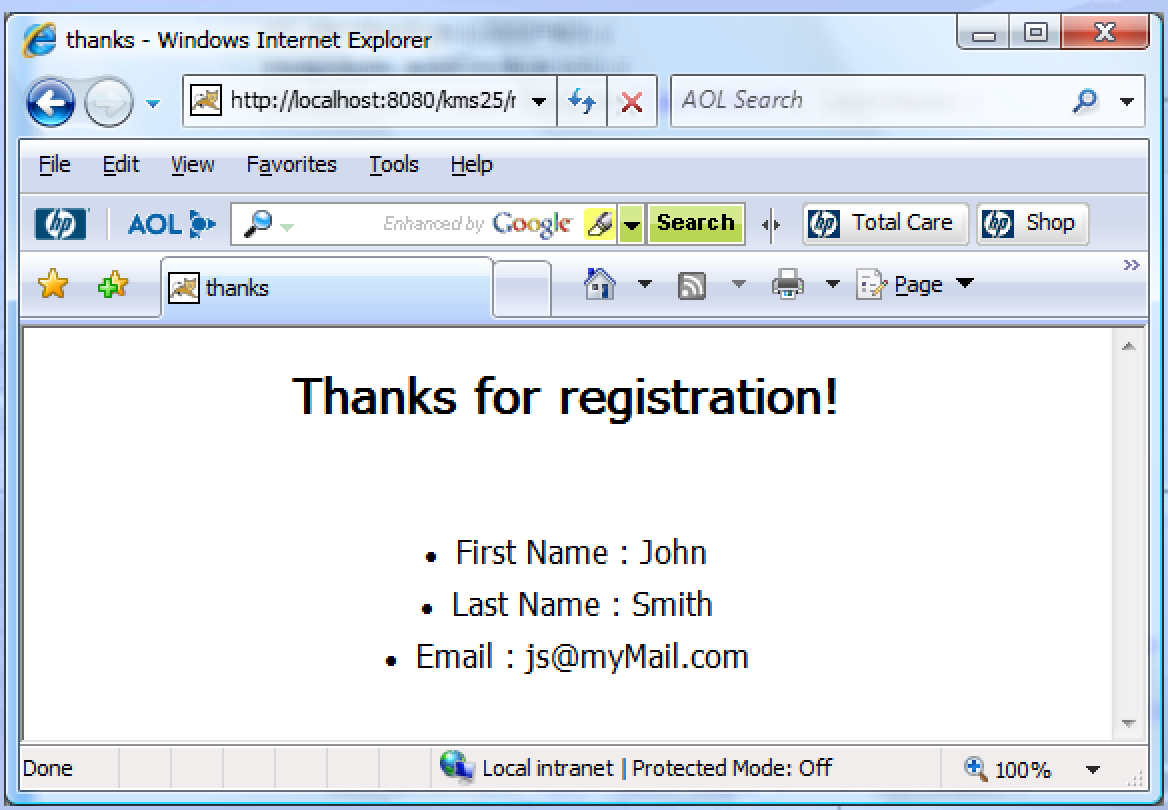

Please Register - Windows Internet Explorer
http://localhost:8080/kms25/r
AOL Search
File Edit View Favorites Tools Help
AOL
Enhanced by Google
Search
Total Care
Shop
Please Register
Page
Local intranet | Protected Mode: Off
100%

Please Register

FirstName :	John
Last Name :	Smith
Email :	Missing_email

Submit Query

Remembering User Preferences (cont.)


thanks - Windows Internet Explorer
http://localhost:8080/kms25/r
AOL Search
File Edit View Favorites Tools Help
AOL
Enhanced by Google
Search
Total Care
Shop
thanks
Page
Done
Local intranet | Protected Mode: Off
100%

Thanks for registration!

- First Name : John
- Last Name : Smith
- Email : js@myMail.com

Session API

Session

- Using the Java Session API
  - Overview of what the Session API provides
  - Extracting Data from the Session
  - Extracting Session Information
  - Adding Data to the Session
- Example:
  - Repeat Visitor with accessCount
  - Shopping Cart System

Overview of Session API Functionality

Overview of Session API

- Servlets include a built-in Session API.
- Enables you to very easily create applications that depend on individual user data.
- For example:
  - Shopping Carts
  - Personalization Services
  - Maintaining state about the user’s preferences.

Using the Session API

- Steps to using the Java Session API

1) Get the Session object from the HttpRequest object.

2) Extract Data from the user’s Session Object

3) Extract information about the session object, e.g. when was the session created?

4) Add data to the user’s Session Object.

Getting a Session Object

- To get the user's session object, call the **getSession()** method of the **HttpServletRequest** class.

- Example:

  HttpSession session = request.getSession();

- If user already has a session, the existing session is returned. If no session exists, a new one is created and returned.

- If you want to know if this is a new session, call the Session **isNew()** method.

Getting a Session Object

- If you want to disable creation of new sessions, pass **false** to the getSession() method.

- For example:

  HttpSession session = request.getSession(false);

- If no current session exists, you will now get back a null object.

Behind the Scenes

- When you call getSession() there is a lot going on behind the scenes.
  - Each user is automatically assigned a unique session ID.

- How does this sessionID get to the user?
  - **<u>Option 1</u>:** If the browser supports cookies, the servlet will automatically create a session cookie, and store the session ID within the cookie. (In Tomcat, the cookie is called: JSESSIONID)
  - **<u>Option 2</u>:** If the browser does not support cookies, the servlet will try to extract the session ID from the URL.

Extracting Data from the Session

Extracting Data From Session

- The Session object works like a Hash Map that enables you to store any type of Java object.
- You can therefore store any number of keys and their associated values.
- To extract an existing object, use the **getAttribute()** method.

Extracting Data from Session

- The getAttribute () method will return an Object type, so you will need to perform a type cast.

- Example:

```
Integer accessCount =
     (Integer)session.getAttribute("accessCount");
```

Extracting Data from Session

- Tip:
  - If you want to get a list of all “keys” associated with a Session, use the **getAttributeNames()** method.
  - This method returns an Enumeration of all Attribute names.

Additional Session Info.

- The Session API includes methods for determining Session specific information.
- **public String getId();**
  - returns the unique session ID associated with this user, e.g. *gj9xswvw9p*
- **public boolean isNew();**
  - indicates if the session was just created.
- **public long getCreationTime();**
  - indicates when the session was first created.
- **public long getLastAccessedTime();**
  - indicates when the session was last sent from the client.

Additional Methods

- **public int getMaxInactiveInterval()**
  - Determine the length of time (in seconds) that a session should go without access before being automatically invalidated.

- **public void setMaxInactiveInterval (int seconds)**
  - Sets the length of time (in seconds) that a session should go without access before being automatically invalidated.
  - A negative value specifies that the session should never time out.

Adding Data to the Session

Adding Data To Session

- To add data to a session, use the **setAttribute()** or **putValue()** methods, and specify the key name and value.

  e.g.,

  - session.setAttribute("accessCount", accessCount);

- To remove a value, you can use the **removeAttribute (String name)** method.

Terminating Sessions

- **public void invalidate()**
  - If the user does not return to a servlet for XX minutes*, the session is automatically invalidated and deleted.
  - If you want to manually invalidate the session, you can call **invalidate().**

* For the exact number of minutes before automatic expiration, check the **getMaxInactiveInterval()** method.

Encoding URLs

- If a browser does not support cookies, you need some other way to maintain the user’s session ID.

- The Servlet API takes care of this for you by automatically appending the session ID to URLs if the browser does not support cookies.

- To automatically append the session ID, use the **encodeURL()** method.

Encoding URLs

- Example:
  - **String url = response.encodeURL (originalURL);**

- Remember that if you do this, every single URL must include the sessionID.

- Since this is hard to ensure, lots of sites (e.g. Yahoo require cookies.)

Understanding Examples

Example 1: RepeatVisitor & accessCount

- Our example tracks the number of visits for each unique visitor.
  - If this is a first time visit, the servlet creates an accessCount Integer variable and assigns it to the Session.
  - If the user has visited before, the servlet extracts the accessCount variable, increments it, and assigns it to the Session.
  - Servlet also displays basic information regarding the session, including: creation time and time of last access.

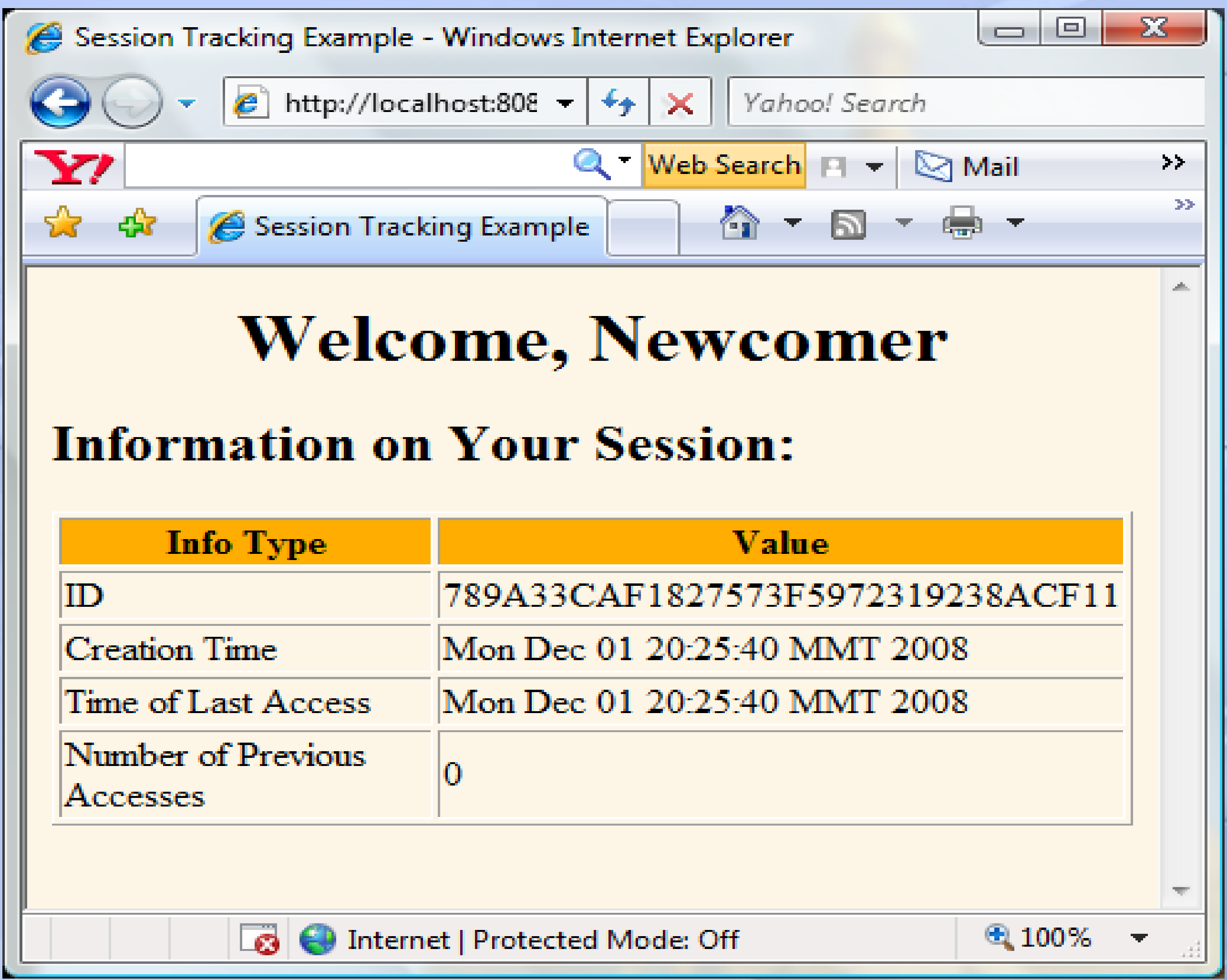
RepeatVisitor (Cont.)

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

public class ShowSession extends HttpServlet
{
  public void doGet(HttpServletRequest request,HttpServletResponse response) throws
  ServletException,IOException
  {
    String title = "Session Tracking Example";
    HttpSession session = request.getSession(true);
    String heading;
    Integer accessCount = (Integer)session.getAttribute("accessCount");
    if (accessCount == null) {
      accessCount = new Integer(0);
      heading = "Welcome, Newcomer";
    } else {
      heading = "Welcome Back";
      accessCount = new Integer(accessCount.intValue() + 1);
    }
    session.setAttribute("accessCount", accessCount);
```

RepeatVisitor (Cont.)

```
response.setContentType("text/html");
   PrintWriter out = response.getWriter();
   out.println("<HTML>");
   out.println("<HEAD> <CENTER> <TITLE>" + title + "</TITLE> </CENTER> </HTML>");
   out.println("<BODY BGCOLOR=\"#FDF5E6\">\n" +
            "<H1 ALIGN=\"CENTER\">" + heading + "</H1>\n" +
            "<H2>Information on Your Session:</H2>\n" +
            "<TABLE BORDER=1 ALIGN=\"CENTER\">\n" +
            "<TR BGCOLOR=\"#FFAD00\">\n" +  "  <TH>Info Type<TH>Value\n" + "<TR>\n" +
            "  <TD>ID\n" + "  <TD>" + session.getId() + "\n" + "<TR>\n" +
            "  <TD>Creation Time\n" +
            "  <TD>" +  new Date(session.getCreationTime()) + "\n" +
            "<TR>\n" + "  <TD>Time of Last Access\n" +
            "  <TD>" + new Date(session.getLastAccessedTime()) + "\n" +
            "<TR>\n" + "  <TD>Number of Previous Accesses\n" +
            "  <TD>" + accessCount + "\n" +
            "</TR>"+
            "</TABLE>\n" +
      "</BODY></HTML>");
   }
}
```

First Time Visit

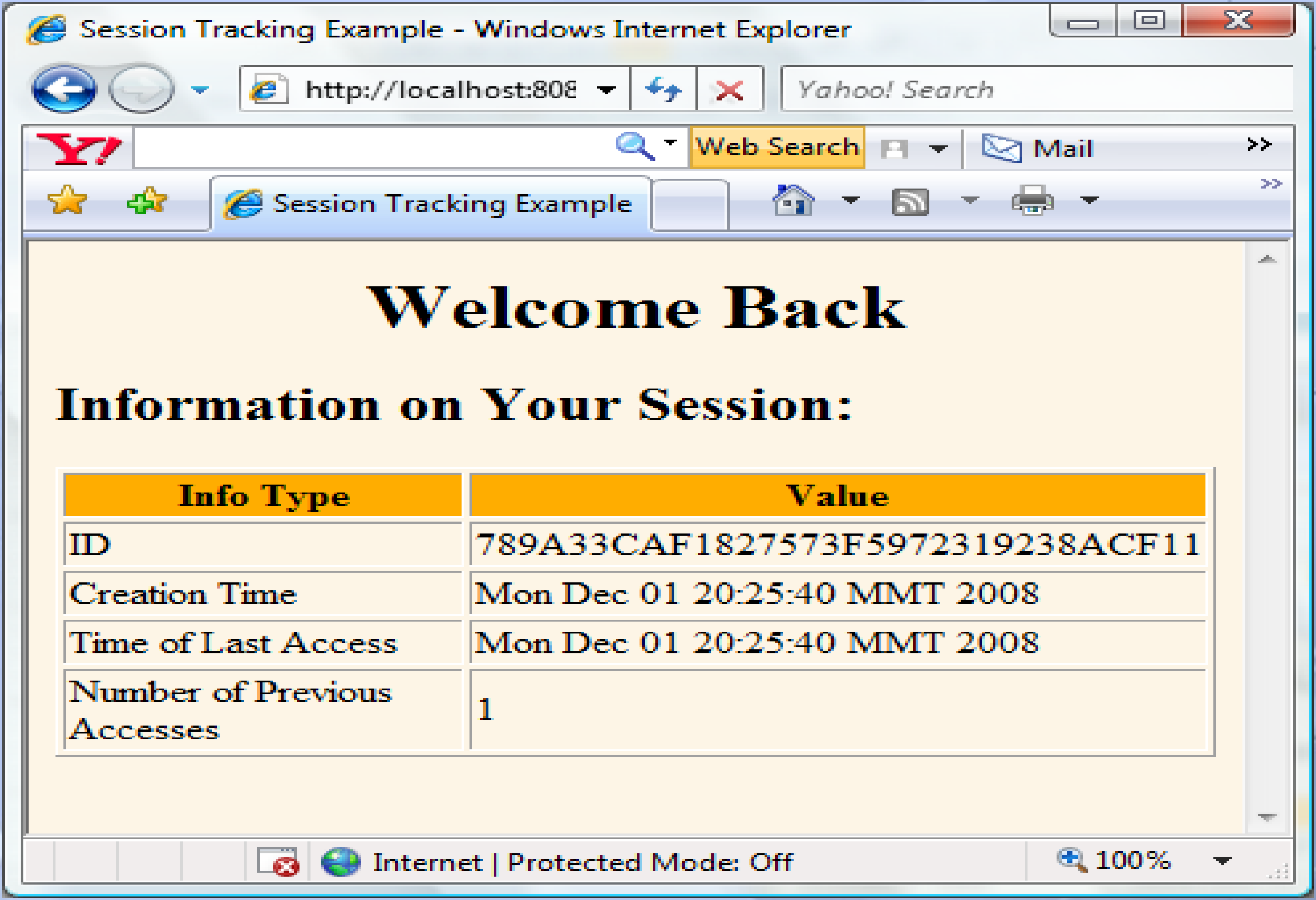

Session Tracking Example - Windows Internet Explorer
http://localhost:808
Yahoo! Search
Web Search
Mail
Session Tracking Example
Internet | Protected Mode: Off
100%

Welcome, Newcomer

Information on Your Session:

Info Type	Value
ID	789A33CAF1827573F5972319238ACF11
Creation Time	Mon Dec 01 20:25:40 MMT 2008
Time of Last Access	Mon Dec 01 20:25:40 MMT 2008
Number of Previous Accesses	0

Next Time Visits


Session Tracking Example - Windows Internet Explorer
http://localhost:808
Yahoo! Search
Web Search
Mail
Session Tracking Example
Internet | Protected Mode: Off
100%

Welcome Back

Information on Your Session:

Info Type	Value
ID	789A33CAF1827573F5972319238ACF11
Creation Time	Mon Dec 01 20:25:40 MMT 2008
Time of Last Access	Mon Dec 01 20:25:40 MMT 2008
Number of Previous Accesses	1

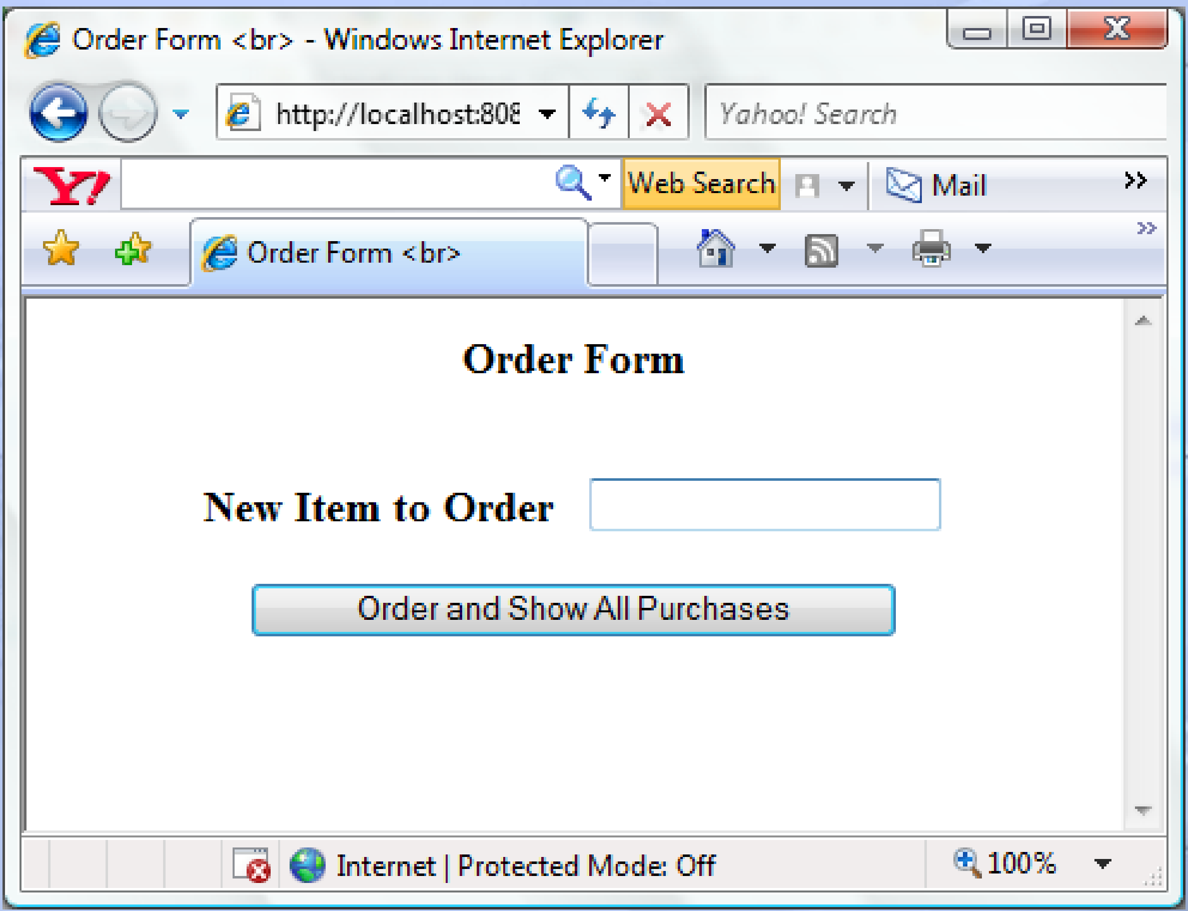
Example 2 : Shopping Cart

- Provides a simple **shopping cart**.
- Stores an ArrayList is the session; session attribute is called, “previousItems”
- Each time you add a new item, the item is added to the ArrayList.

OrderForm.html

```
<html>
        <head><center><title>  Order Form </title></center></head>
        <body>
                <center>
                <h3> Order Form <h3><br>
                <form method="get" action="stest">
                        New Item to Order &nbsp
                        <input type="text" name="newItem"><br><br>
                        <input type="submit" value="Order and Show All
                                                Purchases"><br>
                </form>
                </center>
        </body>
</html>
```

OrderForm.html (Cont.)


Order Form
 - Windows Internet Explorer
http://localhost:808
Yahoo! Search
Web Search
Mail
Order Form
Internet | Protected Mode: Off
100%

Order Form

New Item to Order

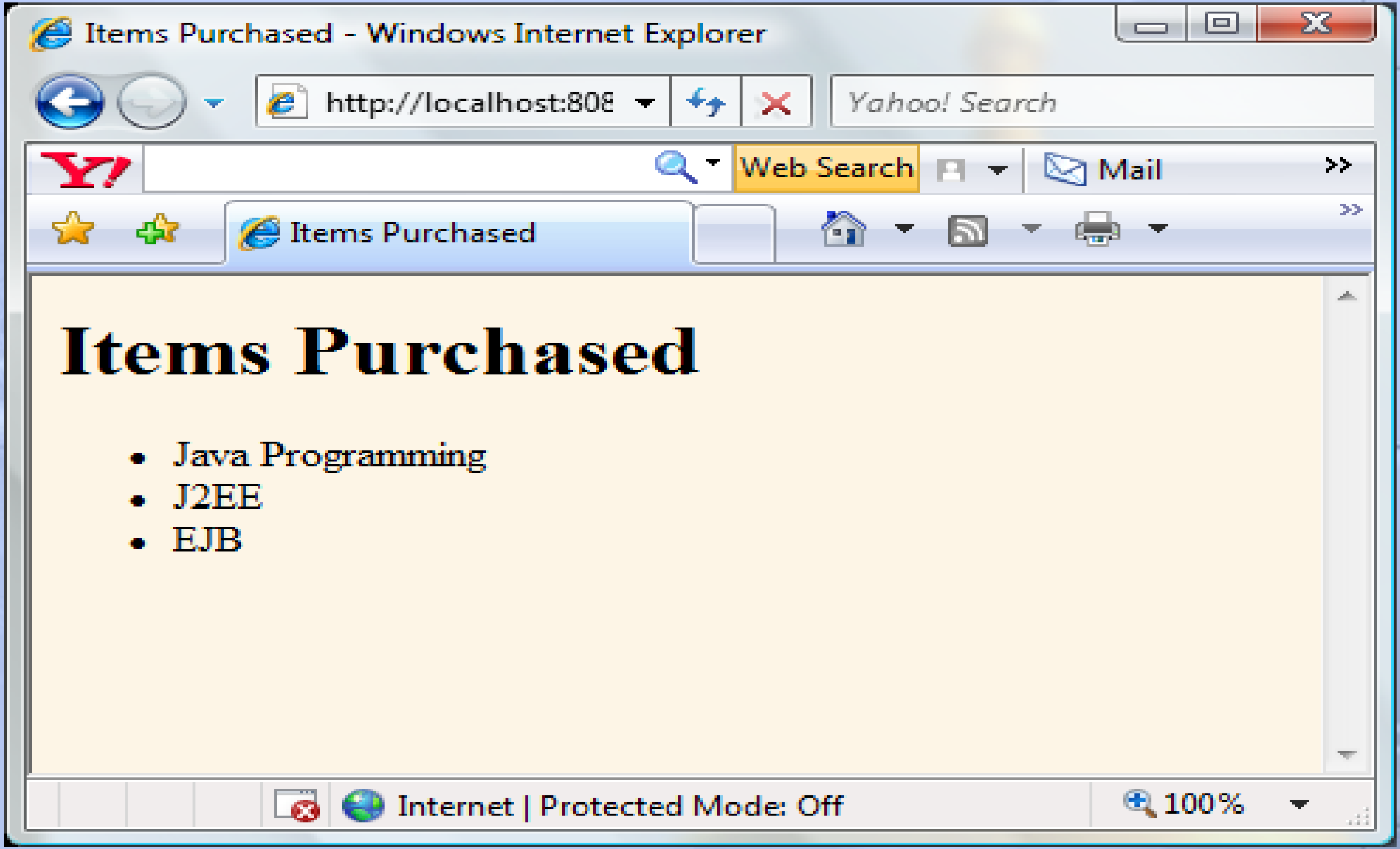
Order and Show All Purchases

ShowItems.java

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

public class ShowItems extends HttpServlet  {
		public void doGet(HttpServletRequest request,
		HttpServletResponse response)throws ServletException,
IOException {
		HttpSession session = request.getSession();
		ArrayList<String> previousItems =(ArrayList<String>)
							session.getAttribute("previousItems");
		if (previousItems == null) {
			previousItems = new ArrayList<String>();
			session.setAttribute("previousItems", previousItems);
		}
		String newItem = request.getParameter("newItem");
		if ((newItem != null) && !newItem.trim().equals(""))
			previousItems.add(newItem);
```

ShowItems.java (Cont.)

```
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String title = "Items Purchased";
        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=\"#FDF5E6\">\n" + "<H1>" +
        title + "</H1>");
        if (previousItems.size() == 0)
                out.println("<I>No items</I>");
        else {
                out.println("<UL>");
                for(String item : previousItems) {
                        out.println(" <LI>" + item);
                }
         out.println("</UL>");
    }
  out.println("</BODY></HTML>");
} }
```


Items Purchased - Windows Internet Explorer
http://localhost:808
Yahoo! Search
Web Search
Mail
Items Purchased
Internet | Protected Mode: Off
100%

Items Purchased

- Java Programming
- J2EE
- EJB

Lets make exercise!

- **Shopping Cart (Book store)**
  - Consists of
    - User Registration
    - Selling Java Programming Books
    - Selling .Net Programming Books
  - Calculates and sends the total cost for all the selected books to user.

❖ write two systems by using

  ❖ Session

  ❖ Cookies

Thank You!