Sessions

- **Session:** Set of pages submitted by user to accomplish goal
  - Example: Most on-line shopping

  ![Diagram showing steps of an on-line shopping process](image)

  **Basic steps:**
  - Assign each new client a unique ID at the start of session.
  - Pass ID to client as part of each response
    - Now client knows it as well
    - Stored as cookie by default
  - Client passes ID back to server with subsequent requests
    - Server can associate this request can be associated with initial request.
  - Server stores client data in a table indexed by session ID

Need for Session Handling

- **Problem:**
  - No easy way to associate steps if multiple clients
    - Nothing built into the web allows server to know where a request is coming from.
    - Server may have to simultaneously manage thousands of sessions.

![Diagram illustrating session handling](image)

Who submitted this request?
Session Handling

- Sessions can be accessed from both servlet and JSP
  - Servlet: Construct a new session object from the request
    ```java
    HttpSession session = request.getSession();
    ```
  - JSP: Just use built-in `session` object which Tomcat creates from request (like `request` object)

Creating a New Session

- Done automatically first time session requested by servlet or JSP
  - ```java
    HttpSession session = request.getSession();
    ```
    in servlet
  - Use of `session` object in JSP
  - Tomcat:
    - Knows this because no session ID included in request
    - Generates new ID not used for current session (or recent past session)
    - Creates new session table entry for that ID

Passing Session IDs

- Automatically included in response sent back to client
- Stored in cookie on client machine
  - Cookies only data that persist between pages in browser
  - Associated with server domain name, directory, etc.

Passing Session IDs

- Automatically included in request sent in future to same server
  - All cookie values associated with server sent with request
  - Server now knows who client is!
Associating Session Data

- Servlets/JSPs can store data associated with session ID
- Servlets/JSPs can look up that data in future when passed the session ID in request

Storing Session Data

- Syntax:
  
  ```java
  session.setAttribute("name", object);
  ```

  - Like parameters, session data stored as name/value pairs
  - Like attributes, can store any Java object
  - Often a “shopping cart” object

Retrieving Session Data

- Syntax:
  
  ```java
  type variable = (type)session.getAttribute("name");
  ```

  - Same syntax as retrieving attribute added to request
  - Since value could be any object, must cast back to original type

- Will be null if
  - No session created for this client
  - That value not stored for this client
Retrieving Session Data

```java
int quantity = Integer.parseInt(request.getParameter("quantity"));
double totalCost = quantity * 9.95;
String name = (String)session.getAttribute("name");
String email = (String)session.getAttribute("email");

<p>
  Thank you for your order of <%= quantity %> widgets, <%= name %>.
</p>
<p>
  Your bill will be $<%= totalCost %>.<br>
</p>
<p>
  You will shortly receive an e-mail confirmation at <%= email %>.<br>
</p>
```

Session Example

StoreInfo servlet creates session and stores the information in new session

Session ID = fieh4K39Rdk

Session data
- name: "Mai Anh Tho"
- email: "tho@hcmuaf.edu.vn"

Cookies
- ID= fieh4K39Rdk
- server=www.widgets.com

Sending request to www.widgets.com, so retrieve its cookies

Cookies
- quantity=27
- ID= fieh4K39Rdk

Server at www.widgets.com
Session Example

Receipt JSP retrieves information associated with the session ID and inserts into the response page:

- Quantity: 27
- ID: fieh4K39Rdk

Session ID = fieh4K39Rdk

Session data:
- Name: Mai Anh Tho
- Email: tho@hcmuaf.edu.vn

URL Encoding

- Many users disable cookies!
  - Often default in some browsers
  - Need alternative way of storing session information on server

Solution:
- Pass session ID to the client as part of every response
- Insure that client sends that session ID back to the server as part of every request
- Since you have no way of knowing whether user has cookies, you must do this!

URL Encoding

- Syntax:
  

```html
<form action="<%= response.encodeURL("url") %>">
  method=...>
  
  Page being requested

  • If browser detects cookies not enabled, it appends the session ID to the request
    - Like other form data

```  

Session Expiration

- Can set time until session expiration
  - Property of web.xml file

  • Session inactive
  • Session id and all attributes destroyed
  • Request for session attributes returns null
Sessions for Access Control

• Users can skip pages in a sequence
  – Bookmarked page in middle

  Goal:
  Prevent users from directly going to other pages without first going to initial page

Solution:
• Set session attribute at servlet called from first page
  – Use this in other pages to determine whether initial page requested in this session

```java
protected void processRequest(HttpServletRequest req,
    HttpServletResponse res)
    throws ServletException, IOException {

    HttpSession session = request.getSession();

    // This attribute used for access control
    session.setAttribute("sessionOK", "yes");
```

Sessions for Access Control

• All other JSPs test whether attribute is null
• If so, redirect to another page
  – Initial page in sequence
  – Error page telling session has expired
• Syntax for redirection from JSP:
  `<jsp:forward page="url to forward to"/>`

```jsp
<% if (session.getAttribute("sessionOK") == null) { %>
  <jsp:forward page="SessionExpired.jsp"/>
<% } %>
```

Attempt to start here

Redirected here