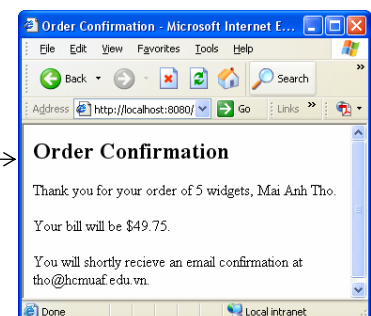
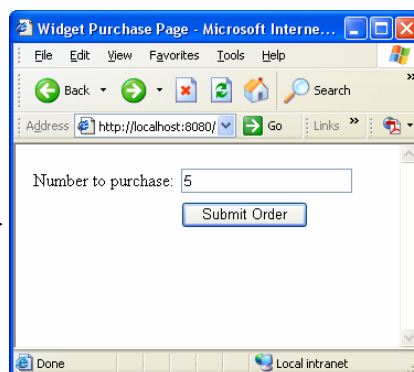
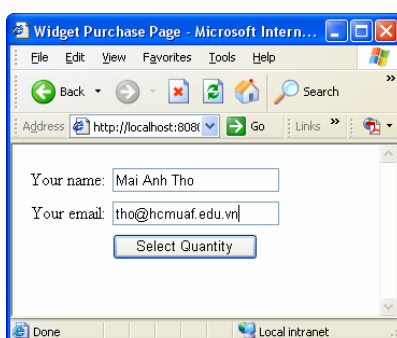
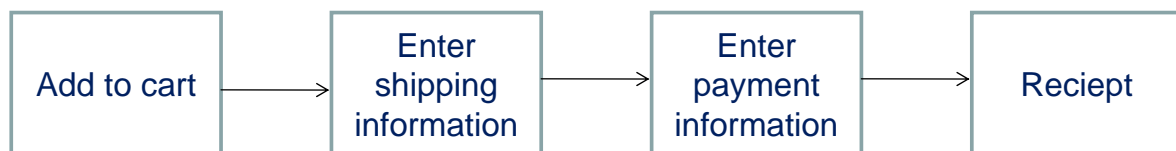


Server-side Web Programming

Lecture 8: Introduction to Sessions

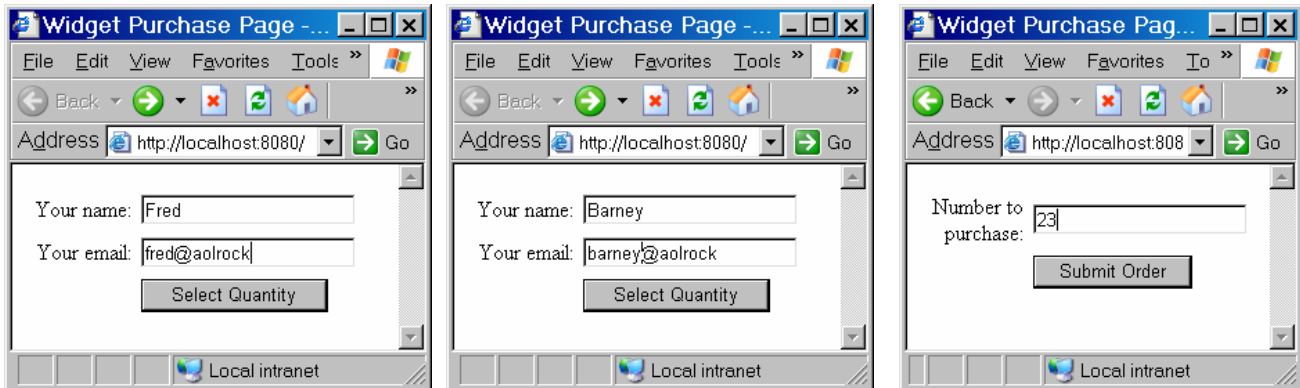
Sessions

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



Need for Session Handling

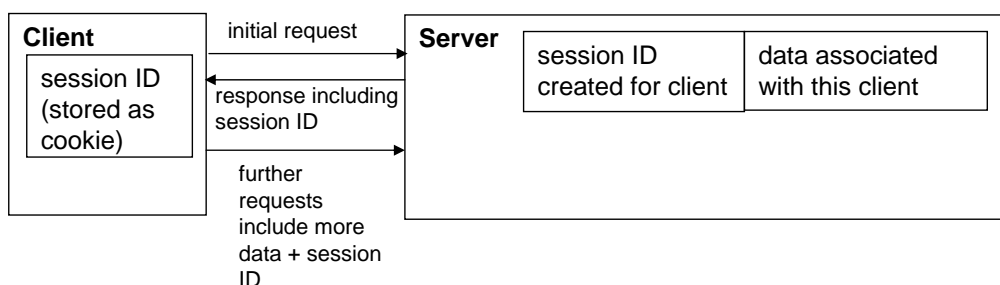
- Problem:
 - No easy way to associate steps if multiple clients
 - Nothing built into the web allows server to know where a request is coming from.
 - Server may have to simultaneously manage thousands of sessions.



Who submitted this request? 3

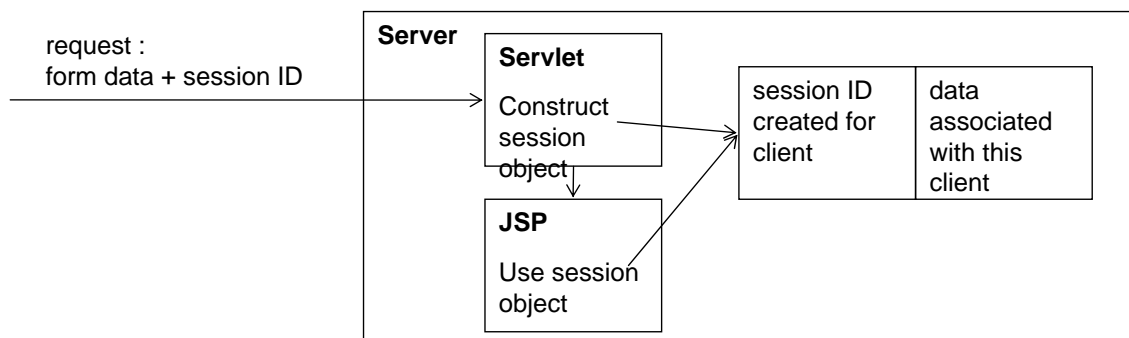
Session Handling

- Basic steps:
 - Assign each new client unique ID at 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 table indexed by session ID



Session Handling

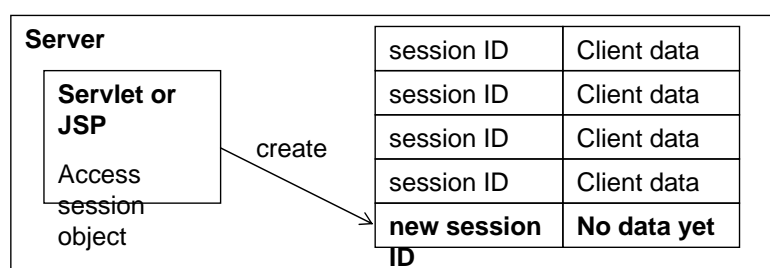
- Sessions can be accessed from both Servlet and JSP
 - Servlet: Construct a new session object from the request
`HttpSession session = request.getSession();`
 - JSP: Just use built-in `session` object which Tomcat creates from request (like `request` object)



5

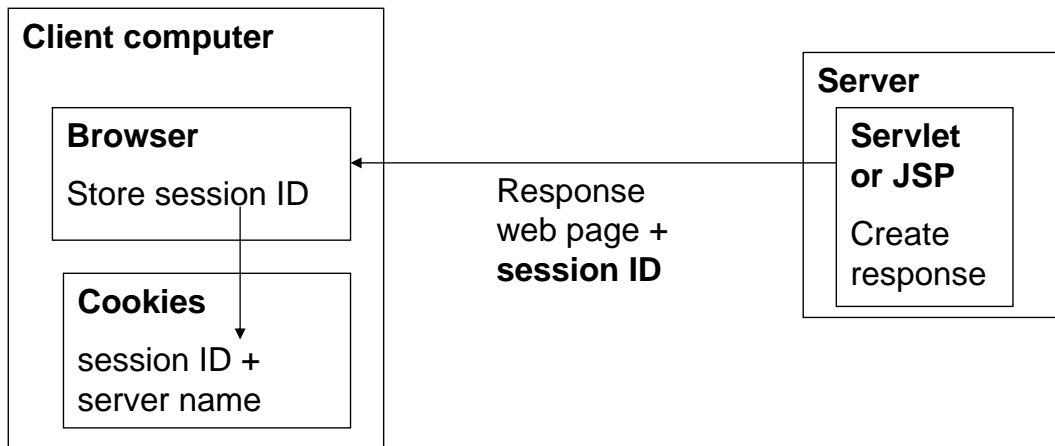
Creating a New Session

- Done automatically first time session requested by servlet or JSP
 - `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



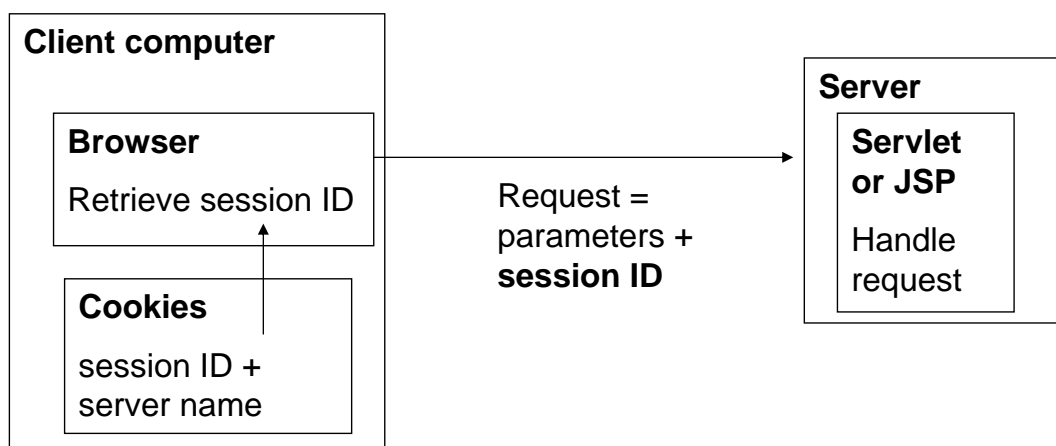
Passing Session IDs

- 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.



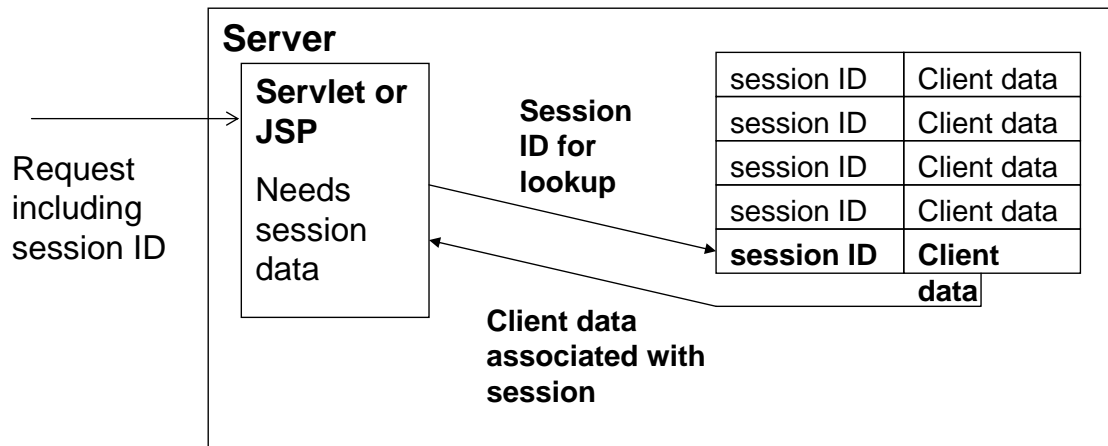
Passing Session IDs

- 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:**
`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

All session data

...	...				
Session ID = fieh4K39Rdk	Session data <table border="1"> <tr> <td>name</td> <td>"Fred"</td> </tr> <tr> <td>email</td> <td>"fred@aolrock"</td> </tr> </table>	name	"Fred"	email	"fred@aolrock"
name	"Fred"				
email	"fred@aolrock"				
...	...				

Storing Session Data

```
21 //
22 protected void processRequest(HttpServletRequest request, HttpServletResponse response)
23     throws ServletException, IOException {
24     String name = request.getParameter("customerName");
25     String email = request.getParameter("customerEmail");
26     HttpSession session = request.getSession();
27     session.setAttribute("name", name);
28     session.setAttribute("email", email);
29     RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/");
30     dispatcher.forward(request, response);
31 }
32
```

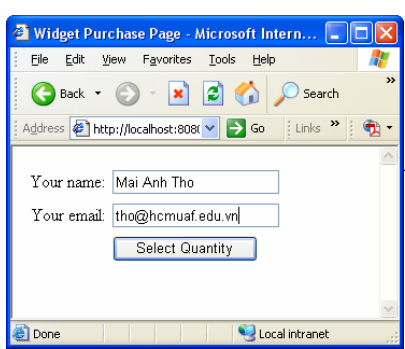
Retrieving Session Data

- Syntax:
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

```
11 <title>Order Confirmation</title>
12 </head>
13 <body>
14
15 <%
16     int quantity = Integer.parseInt(request.getParameter("quantity"));
17     double totalCost = quantity * 9.95;
18     String name = (String)session.getAttribute("name");
19     String email = (String)session.getAttribute("email");
20 %>
21
22 <h2>Order Confirmation</h2>
23
24 <p>
25     Thank you for your order of <%= quantity %> widgets, <%= name %>.
26 </p>
27 <p>
28     Your bill will be $<%= totalCost %>.
29 </p>
30 <p>
31     You will shortly receive an email confirmation at <%= email %>.
32 </p>
```

Session Example



"Mai Anh Tho",
tho@hcmuaf.edu.vn
passed to server

storeInfo servlet
creates session and
stores the information
in new session

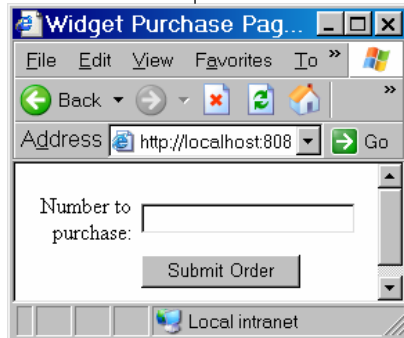
Session ID =
fieh4K39Rdk

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

Session Example

Cookies

ID= fieh4K39Rdk
server=www.widgets.com



Response =
page +
Session ID

StoreInfo servlet
adds session ID to
response

Session ID

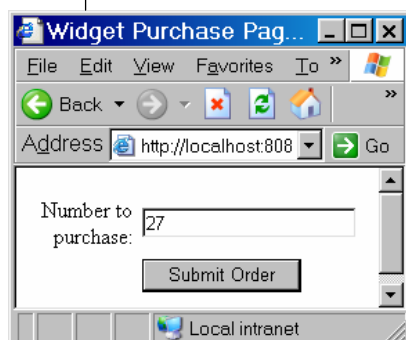
getQuantity JSP
sends session ID to
client as part of page

Session Example

Cookies

ID= fieh4K39Rdk
server=www.widgets.com

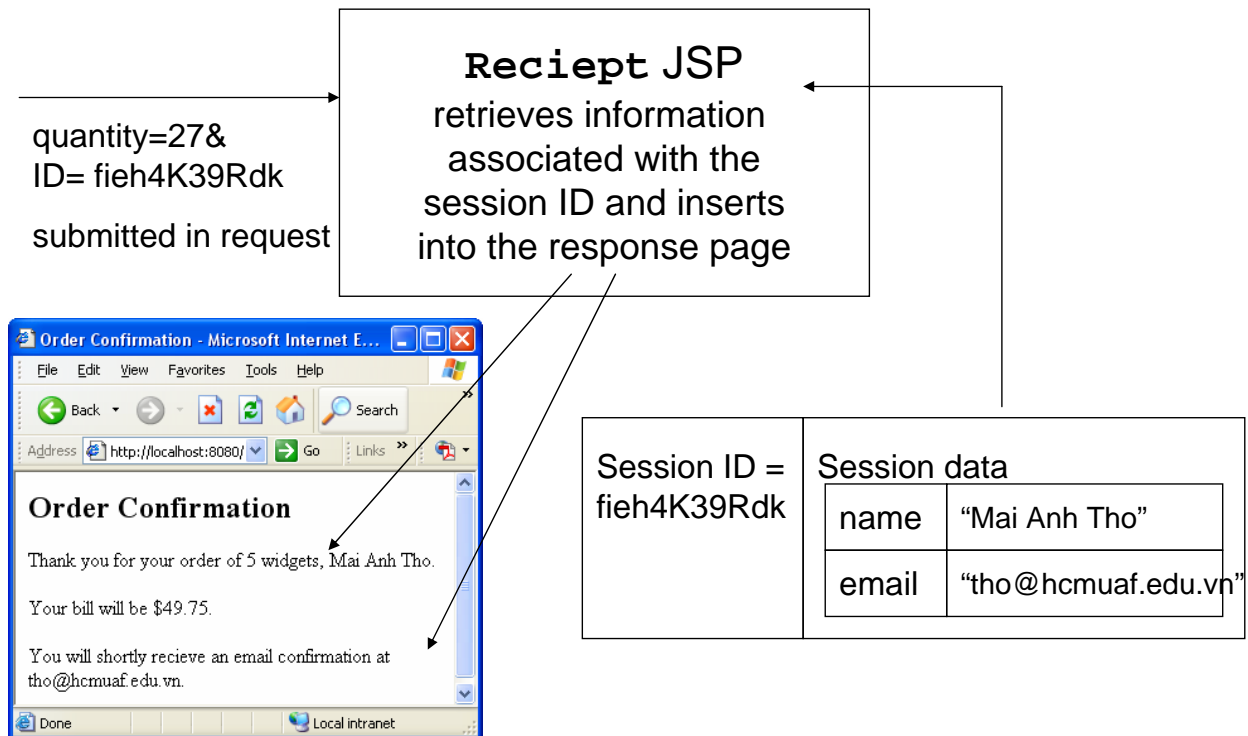
Sending request to
www.widgets.com, so
retrieve its cookies



quantity=27&
ID= fieh4K39Rdk
submitted in request

Server at
www.widgets.com

Session Example



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:

```
<form action=  
    "<%= response.encodeURL("url") %>"  
    method=...>
```

└── Page being requested

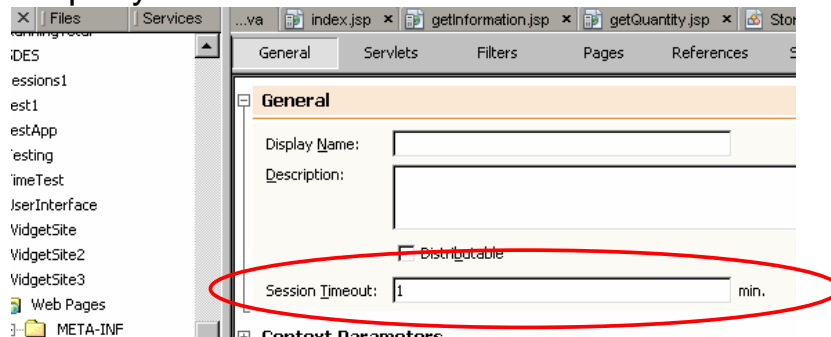
- If browser detects cookies not enabled, it appends the session ID to the request
 - Like other form data

```
16 <body>  
17 <form action="<%= response.encodeURL("Reciept.jsp") %>" method="get">  
18 <table cellpadding="5">  
19 <tr>
```

Session Expiration

- Can set time until session expiration

- Property of **web.xml** file

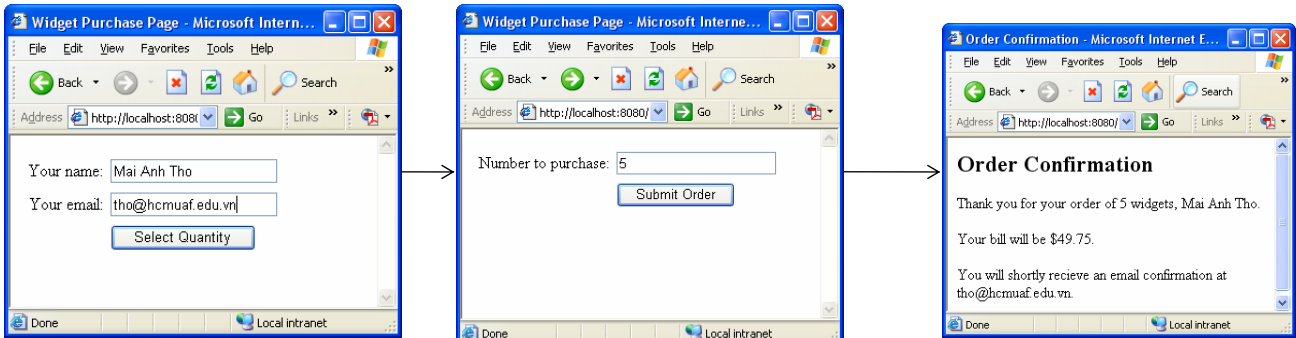


- Session expires if no request within time limit

- 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

Sessions for Access Control

Solution:

- Set session attribute at servlet called from first page
 - Use this in other pages to determine whether initial page requested in this session

```
22     protected void processRequest (HttpServletRequest request  
23     throws ServletException, IOException {  
24         HttpSession session = request.getSession();  
25  
26         // This attribute used for access control  
27         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"/>
```

```
<body>
```

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

```
<%
```

Sessions for Access Control

