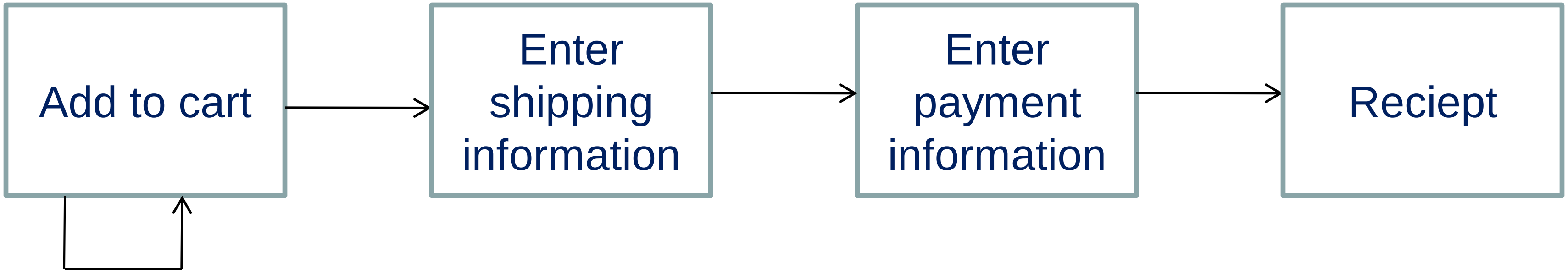
Server-side Web Programming

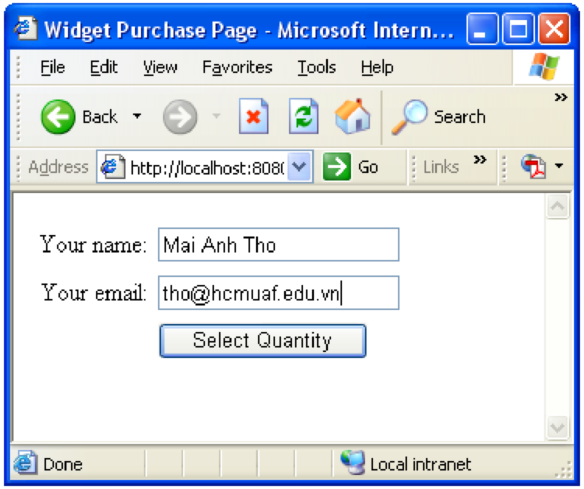
Lecture 7:
Introduction to Sessions

thoma@hcmute.edu.vn

Sessions

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

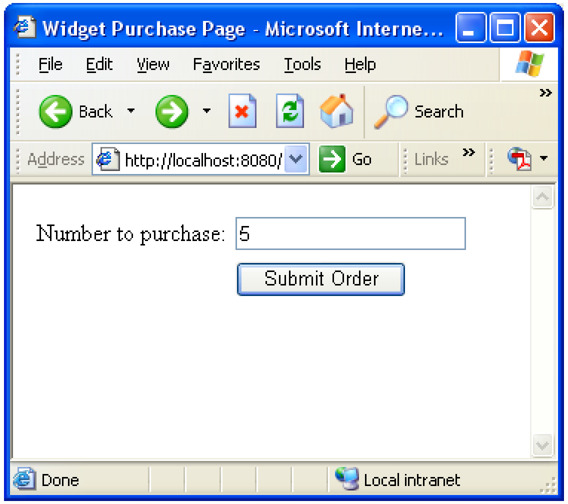



Widget Purchase Page - Microsoft Intern...
File Edit View Favorites Tools Help
Back Search
Address http://localhost:808(Go Links
Done Local intranet

Your name: Mai Anh Tho

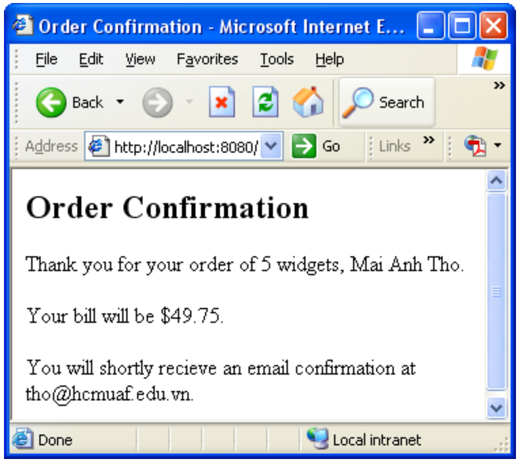
Your email: tho@hcmuaf.edu.vn

Select Quantity


Widget Purchase Page - Microsoft Interne...
File Edit View Favorites Tools Help
Back Search
Address http://localhost:8080/ Go Links
Done Local intranet

Number to purchase: 5

Submit Order


Order Confirmation - Microsoft Internet E...
File Edit View Favorites Tools Help
Back Search
Address http://localhost:8080/ Go Links
Done Local intranet

Order Confirmation

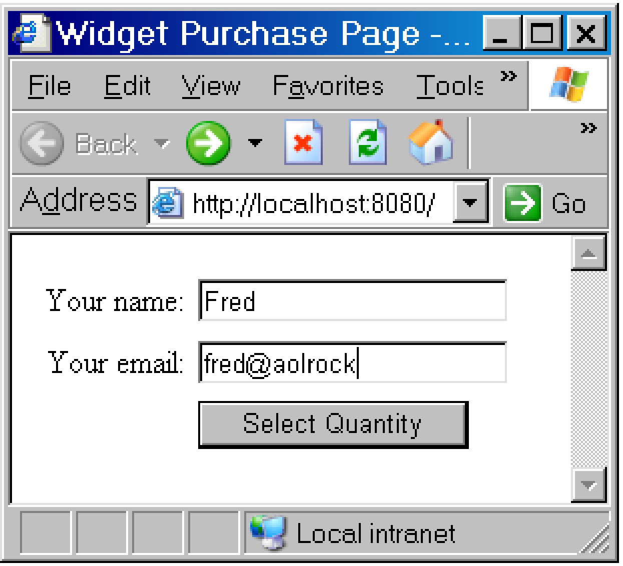
Thank you for your order of 5 widgets, Mai Anh Tho.

Your bill will be $49.75.

You will shortly recieve an email confirmation at tho@hcmuaf.edu.vn.

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.

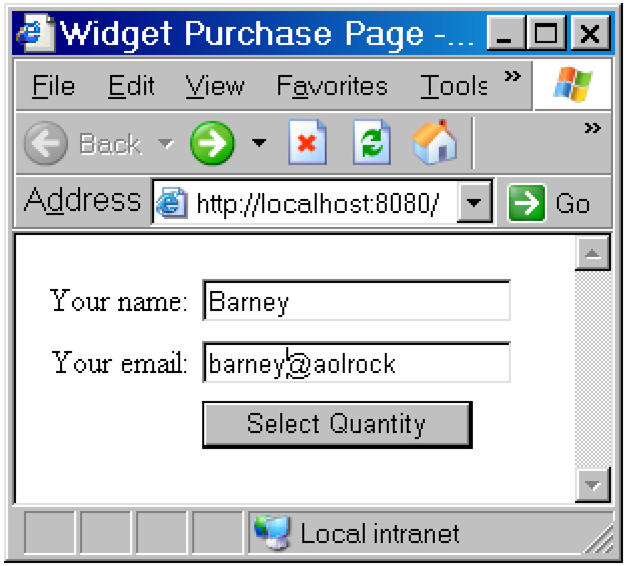


http://localhost:8080/

Your name: Fred

Your email: fred@aolrock

Select Quantity

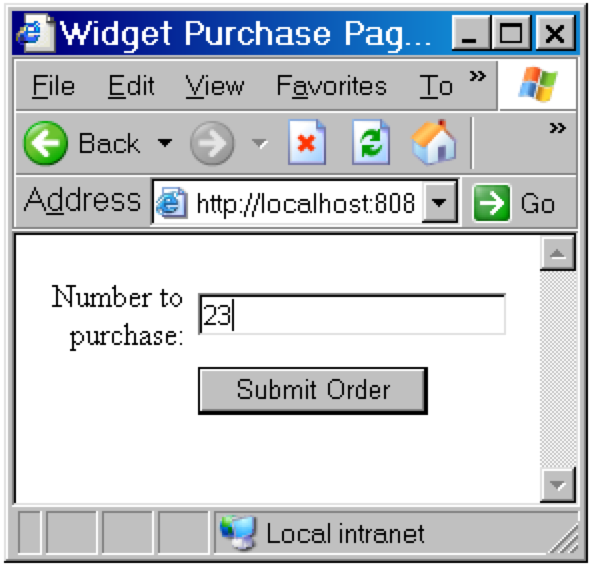


http://localhost:8080/

Your name: Barney

Your email: barney@aolrock

Select Quantity



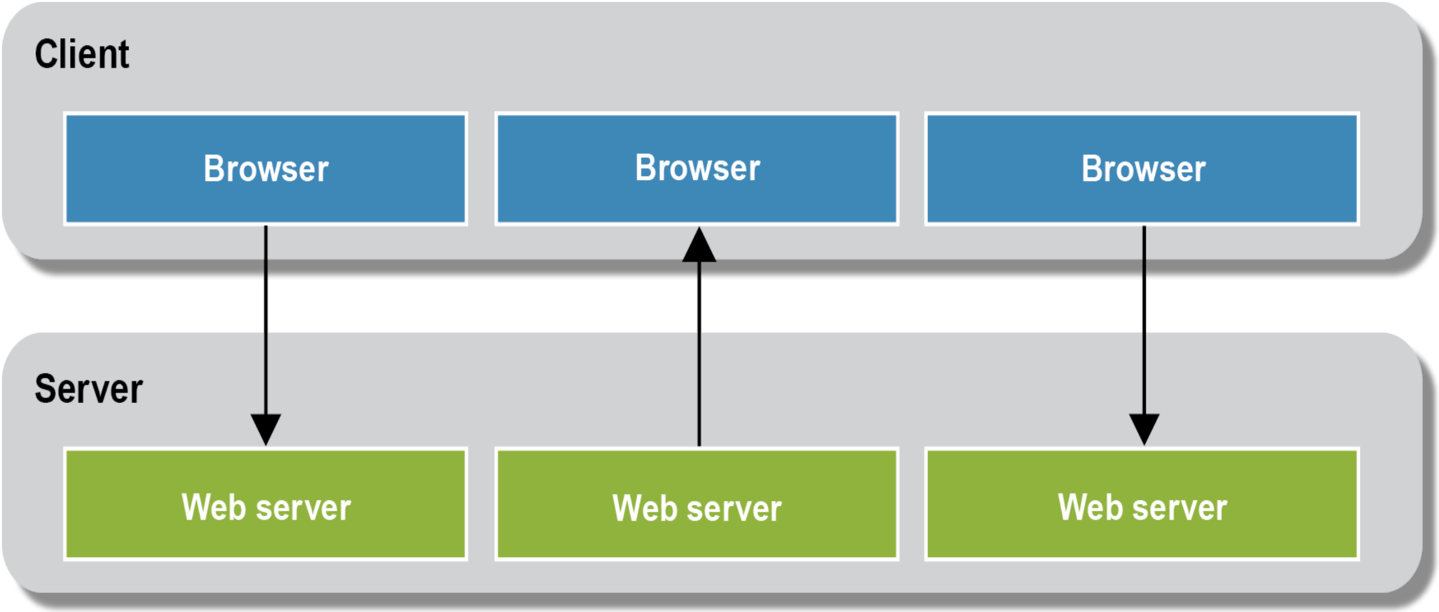
http://localhost:808

Number to purchase: 23

Submit Order

Who submitted this request?

Why session tracking is difficult with HTTP



Client

Browser

Browser

Browser

Server

Web server

Web server

Web server

First HTTP Request:

The browser requests a page.

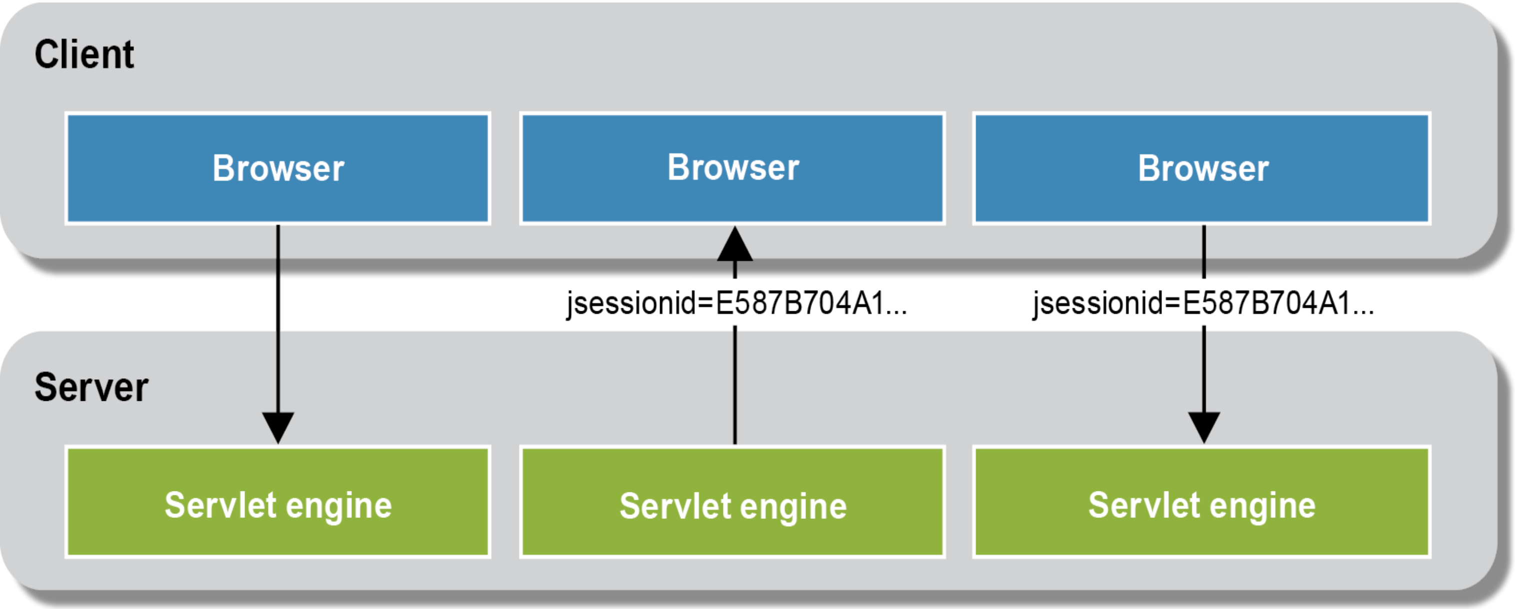
First HTTP Response:

The server returns the requested page and drops the connection.

Following HTTP Requests:

The browser requests a page. The web server has no way to associate the browser with its previous request.

How Java keeps track of sessions



First HTTP Request:

The browser requests a JSP or servlet. The servlet engine creates a session object and assigns an ID for the session.

First HTTP Response:

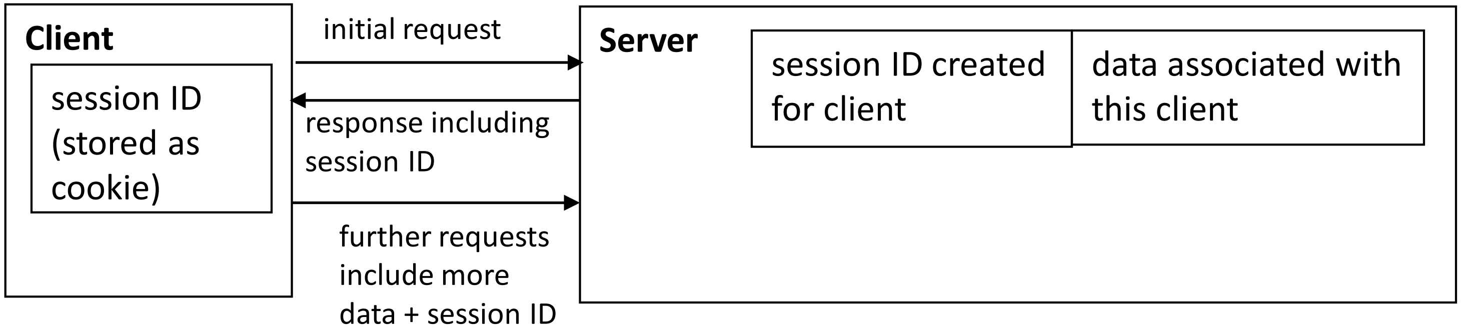
The server returns the requested page and the ID for the session.

Following HTTP Requests:

The browser requests a JSP or servlet. The servlet engine uses the session ID to associate the browser with its session object.

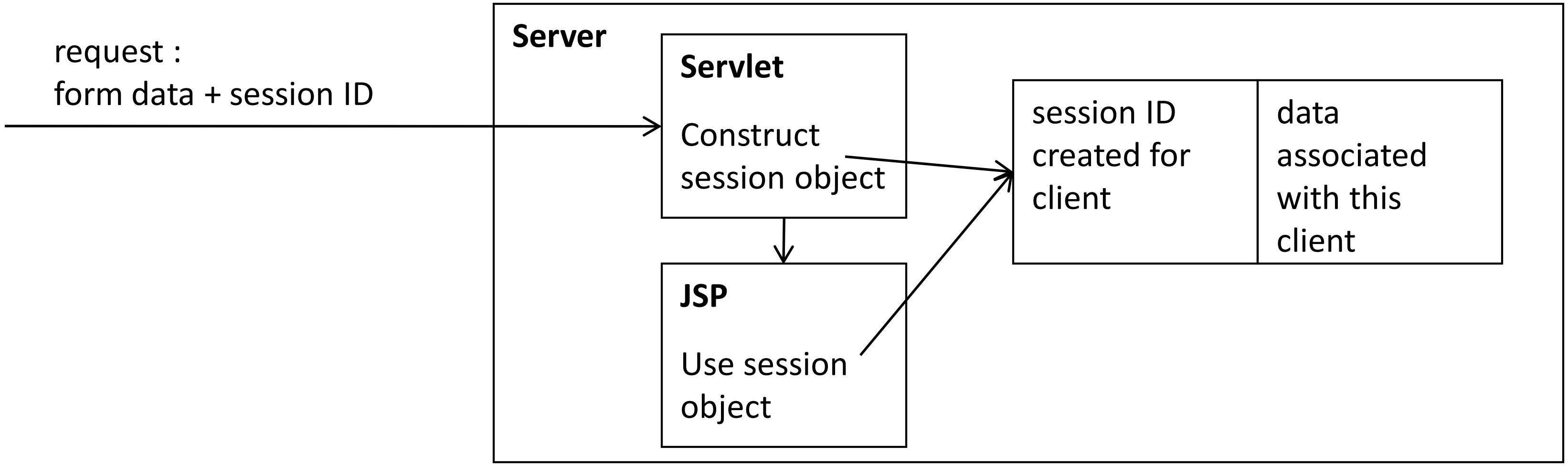
Session Handling

- 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


Client
session ID (stored as cookie)
initial request
response including session ID
further requests include more data + session ID
Server
session ID created for client
data associated with this client

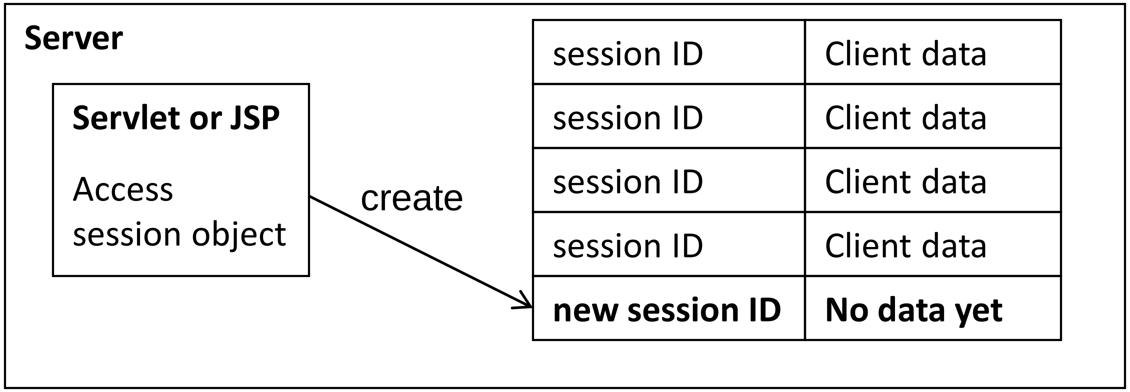
Session Handling

- 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)



Creating a New Session

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



Server

Servlet or JSP

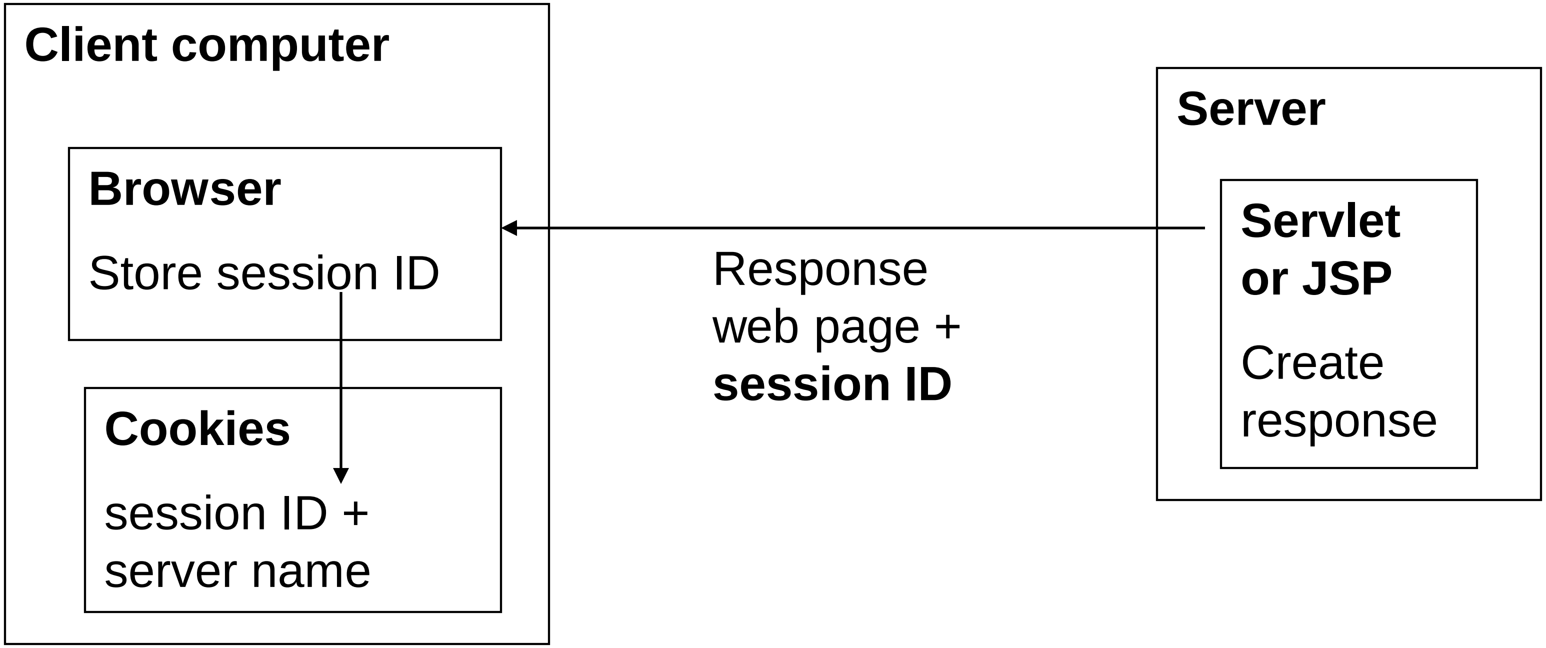
Access session object

create

session ID	Client data
session ID	Client data
session ID	Client data
session ID	Client data
new session ID	**No data yet**

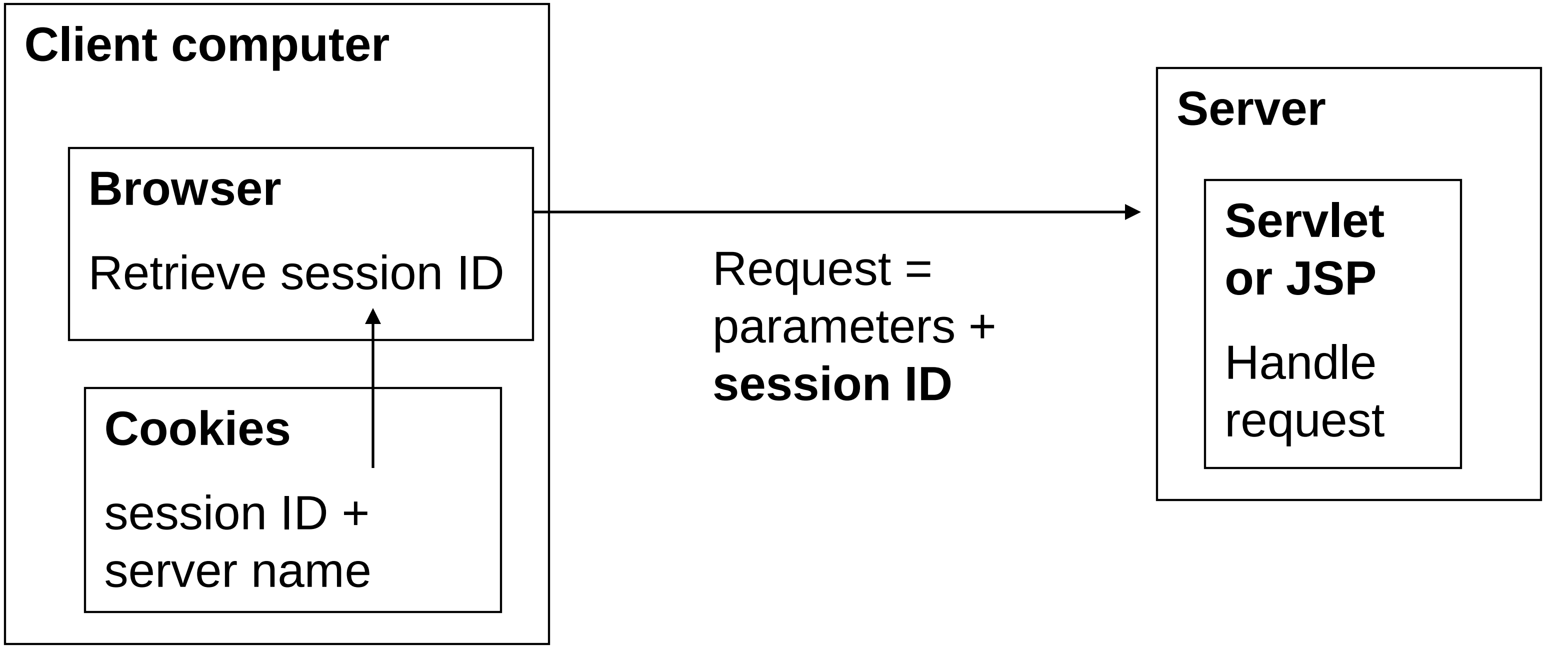
Passing Session IDs

- 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!



Client computer

Browser

Retrieve session ID

Cookies

session ID +
server name

Request =
parameters +
session ID

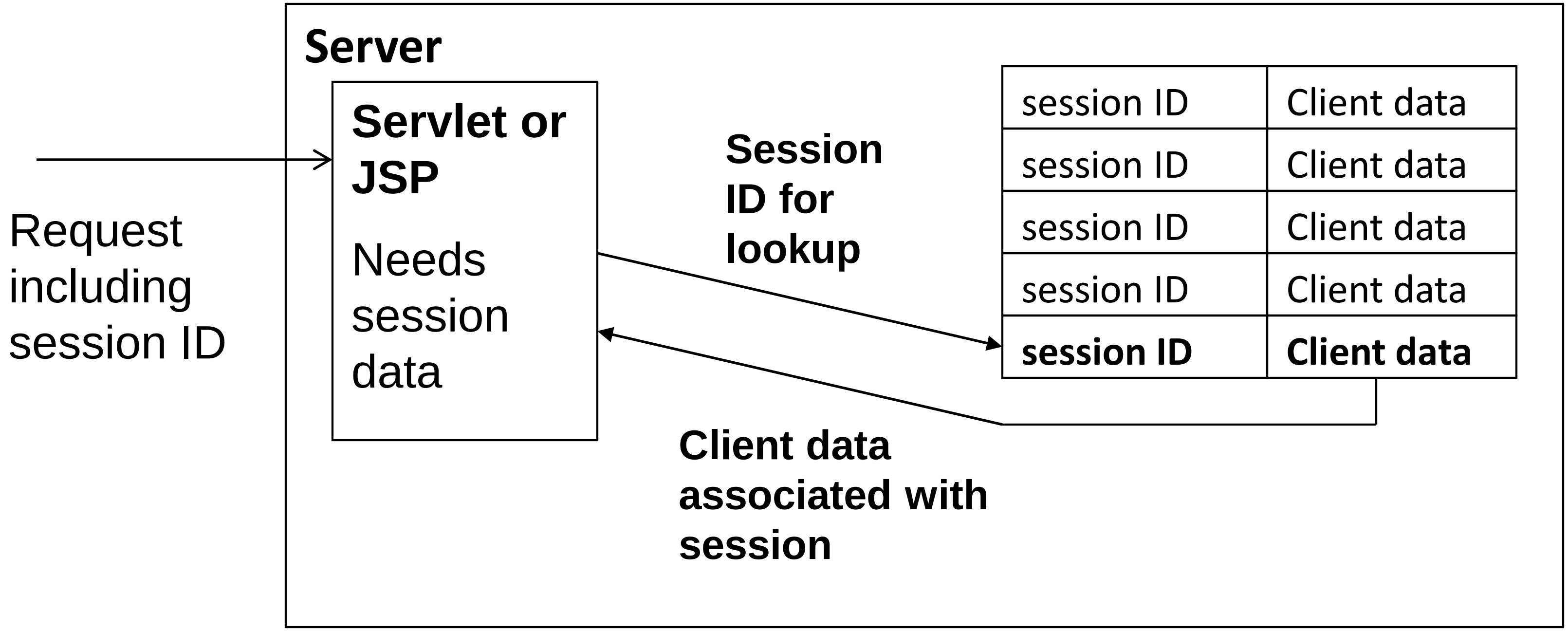
Server

Servlet or JSP

Handle request

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



Request including session ID

Server

Servlet or JSP

Needs session data

Session ID for lookup

session ID	Client data
session ID	Client data
session ID	Client data
session ID	Client data
session ID	**Client data**

Client data associated with session

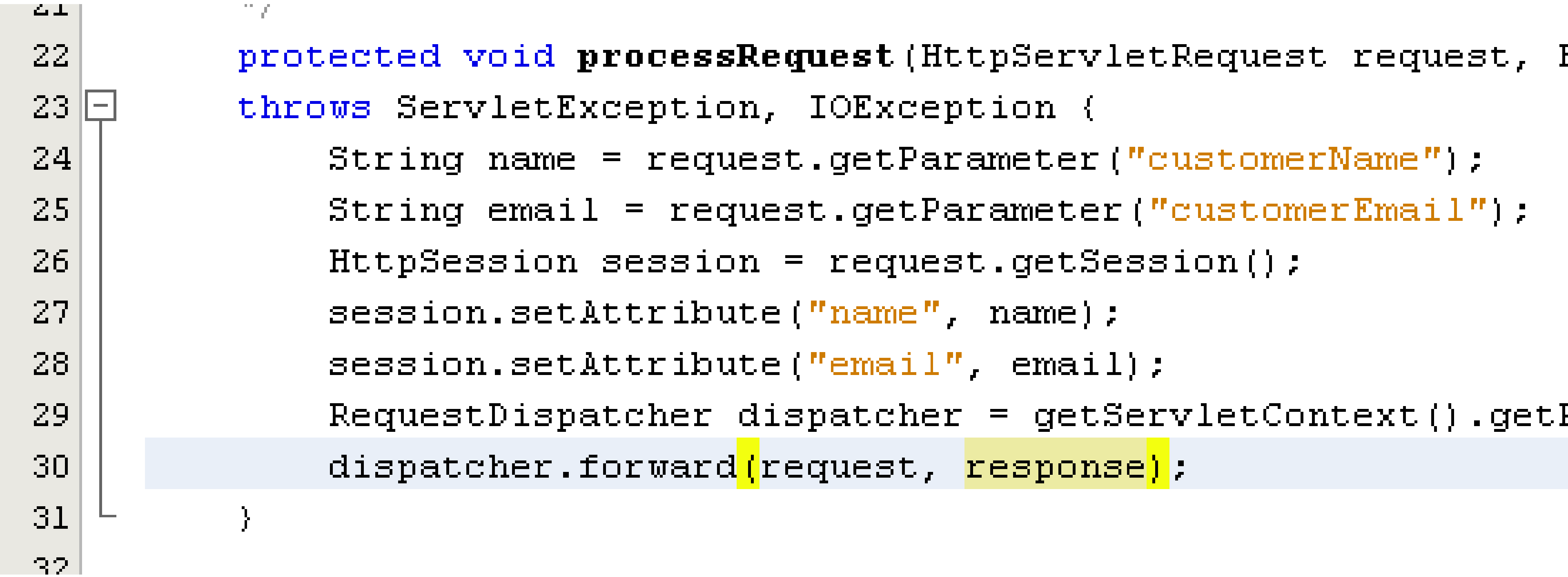
Storing Session Data

- **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

<table>
<tr><td>…</td><td>…</td></tr>
<tr><td>Session ID =
fieh4K39Rdk</td><td>Session data
<table>
<tr><td>name</td><td>“Fred”</td></tr>
<tr><td>email</td><td>“fred@aolrock”</td></tr>
</table>
</td></tr>
<tr><td>…</td><td>…</td></tr>
</table>

Storing Session Data



```
    */
    protected void processRequest(HttpServletRequest request, H
    throws ServletException, IOException {
        String name = request.getParameter("customerName");
        String email = request.getParameter("customerEmail");
        HttpSession session = request.getSession();
        session.setAttribute("name", name);
        session.setAttribute("email", email);
        RequestDispatcher dispatcher = getServletContext().getR
        dispatcher.forward(request, response);
    }
```

Retrieving Session Data

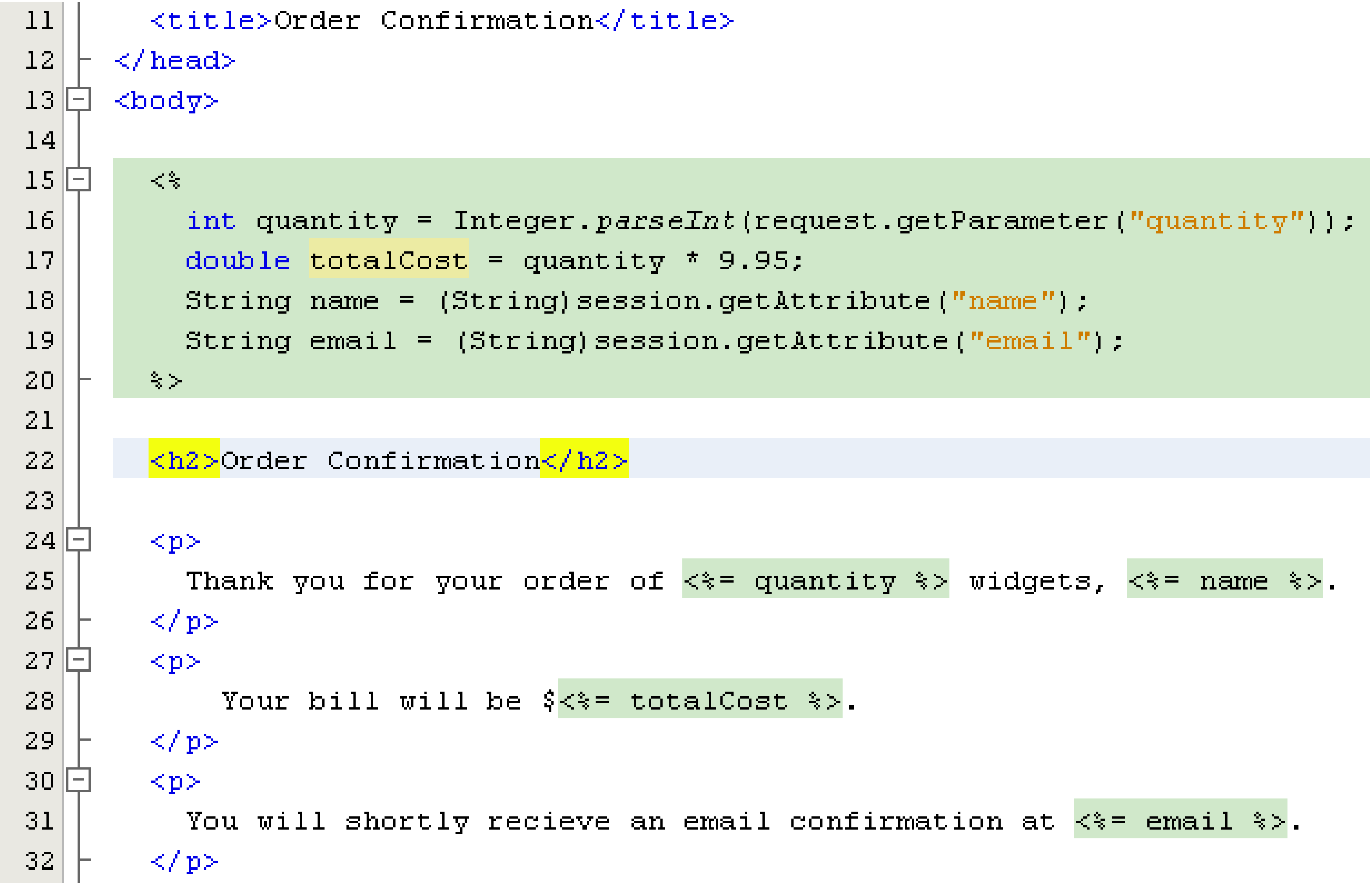
- Syntax:

```
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



```
  <title>Order Confirmation</title>
</head>
<body>

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

  <h2>Order Confirmation</h2>

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

URL Encoding

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

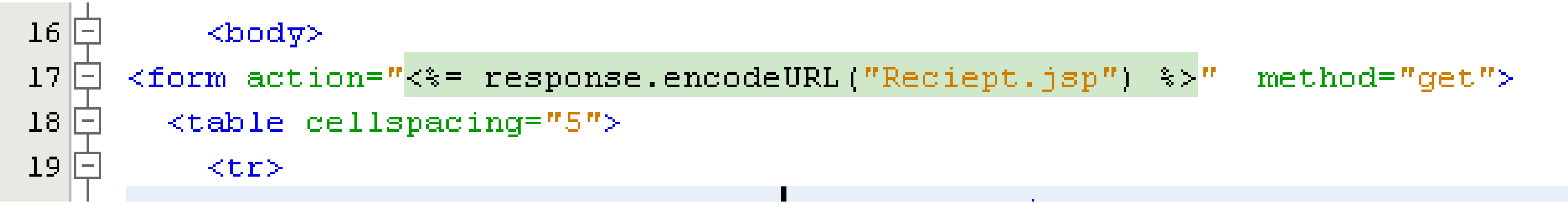
Solution:

- 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

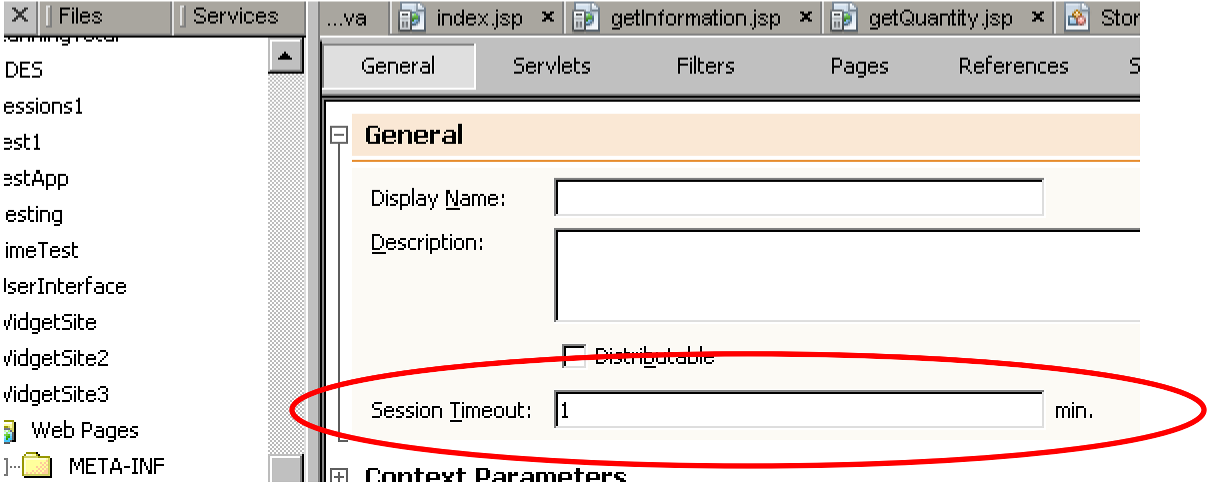
- If browser detects cookies not enabled, it appends the session ID to the request
  - Like other form data



```
    <body>
<form action="<%= response.encodeURL("Reciept.jsp") %>"  method="get">
  <table cellspacing="5">
    <tr>
```

Session Expiration

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

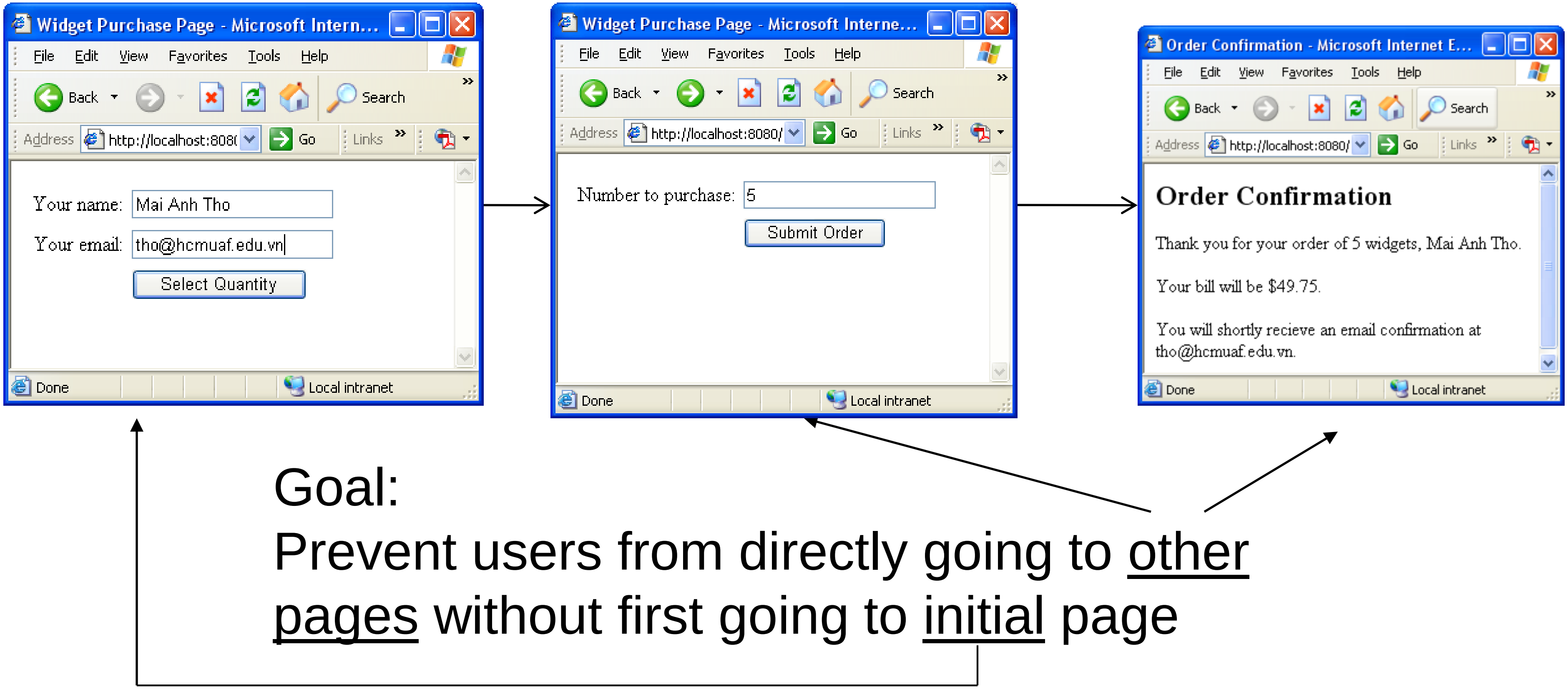


Session Timeout: 1 min.

- 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 <u>skip</u> pages in a sequence
  - Bookmarked page in middle


Widget Purchase Page - Microsoft Intern...
File Edit View Favorites Tools Help
Back Search
Address http://localhost:808(Go Links
Done
Local intranet

Your name: Mai Anh Tho

Your email: tho@hcmuaf.edu.vn

Select Quantity
Widget Purchase Page - Microsoft Interne...
File Edit View Favorites Tools Help
Back Search
Address http://localhost:8080/ Go Links
Done
Local intranet

Number to purchase: 5

Submit Order
Order Confirmation - Microsoft Internet E...
File Edit View Favorites Tools Help
Back Search
Address http://localhost:8080/ Go Links
Done
Local intranet

Order Confirmation

Thank you for your order of 5 widgets, Mai Anh Tho.

Your bill will be $49.75.

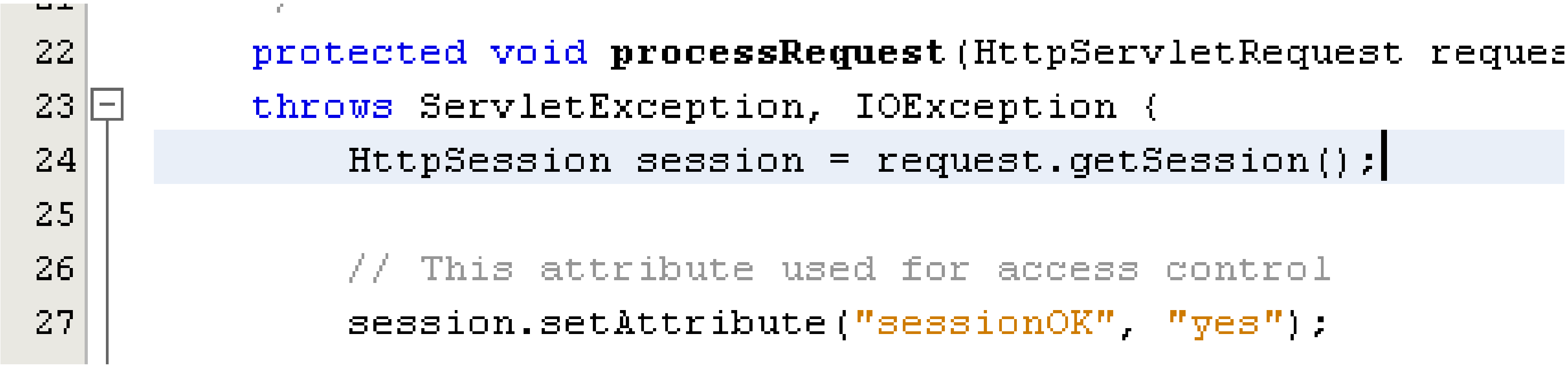
You will shortly recieve an email confirmation at tho@hcmuaf.edu.vn.

Goal:
Prevent users from directly going to <u>other pages</u> without first going to <u>initial</u> 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



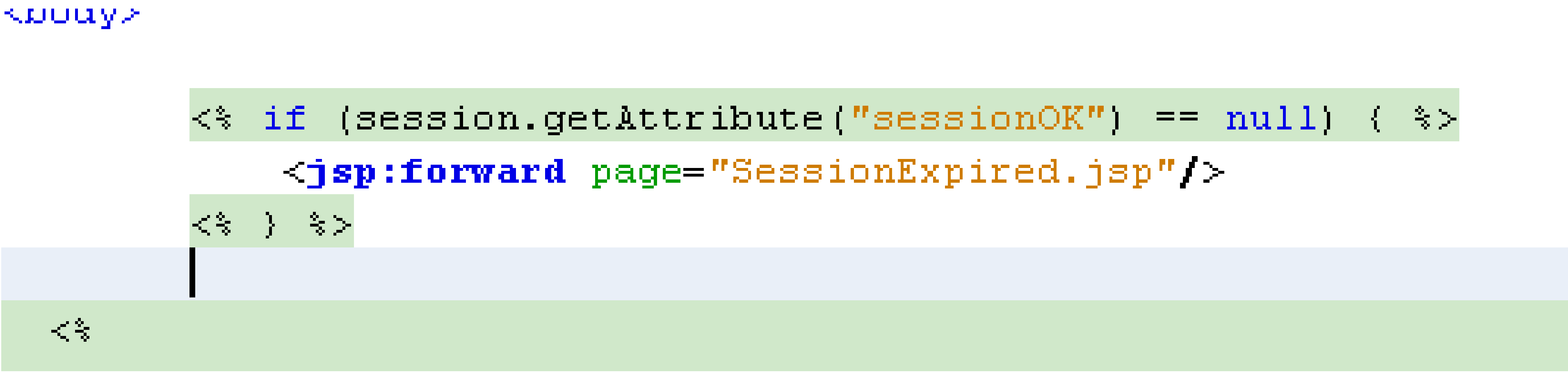
```
protected void processRequest(HttpServletRequest reques
throws ServletException, IOException {
    HttpSession session = request.getSession();

    // 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_**”/>**

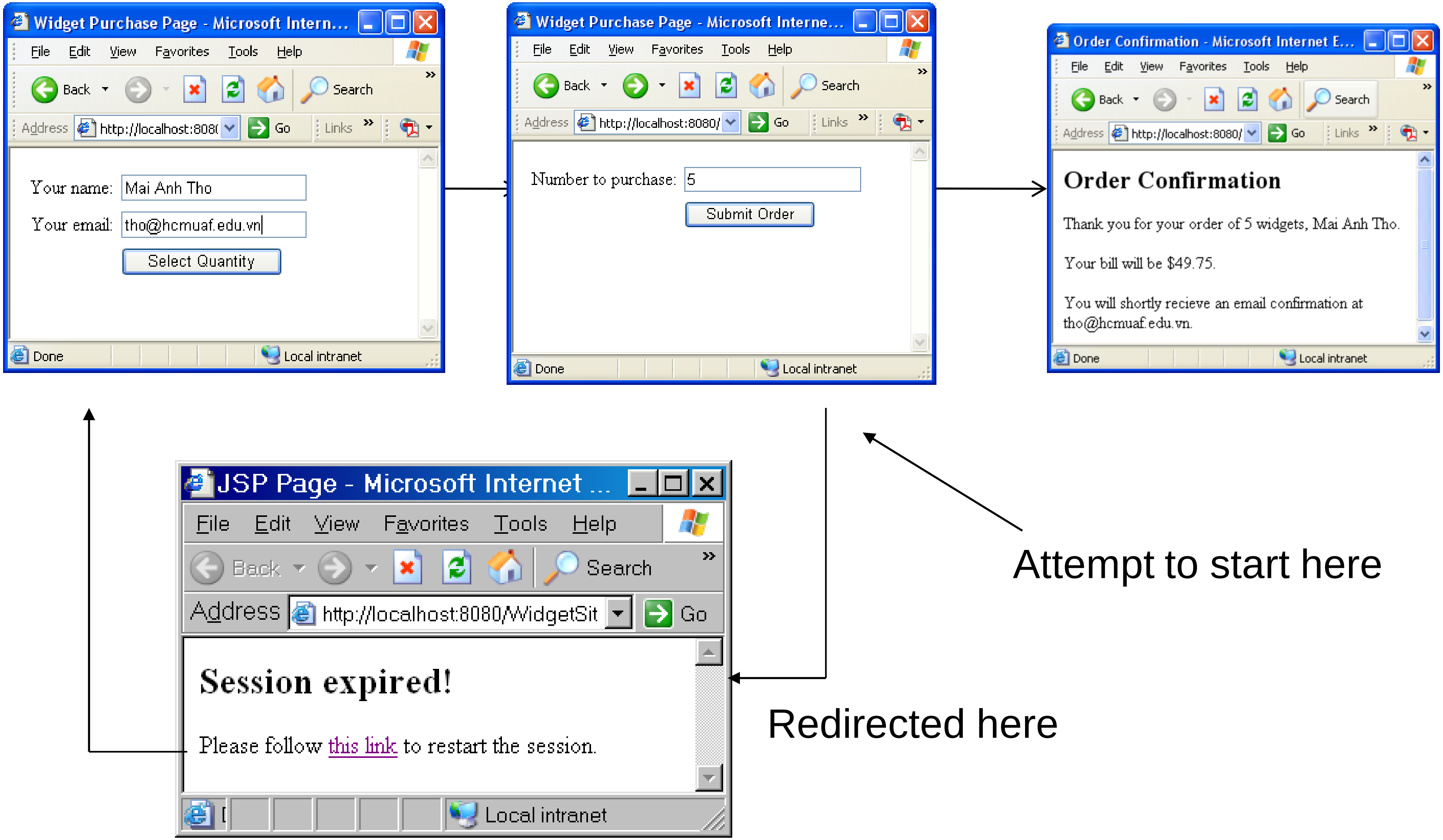


```
<body>

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

<%
```

Sessions for Access Control



Widget Purchase Page - Microsoft Intern...
File Edit View Favorites Tools Help
Back Search
Address http://localhost:808(Go Links
Done Local intranet

Your name: Mai Anh Tho

Your email: tho@hcmuaf.edu.vn

Widget Purchase Page - Microsoft Interne...
File Edit View Favorites Tools Help
Back Search
Address http://localhost:8080/ Go Links
Done Local intranet

Number to purchase: 5

Order Confirmation - Microsoft Internet E...
File Edit View Favorites Tools Help
Back Search
Address http://localhost:8080/ Go Links
Done Local intranet

Order Confirmation

Thank you for your order of 5 widgets, Mai Anh Tho.

Your bill will be $49.75.

You will shortly recieve an email confirmation at tho@hcmuaf.edu.vn.

JSP Page - Microsoft Internet ...
File Edit View Favorites Tools Help
Back Search
Address http://localhost:8080/WidgetSit Go
Local intranet

Session expired!

Please follow this link to restart the session.

Attempt to start here

Redirected here