CS 683 Emerging Technologies
Fall Semester, 2005
Doc 5 Web App Intro

Contents

Reading Assignment .................................. 2
About Web Applications .............................. 3
    Basic HTTP (Web) ................................. 5
    HTTP is Stateless ................................ 5
    CGI, Server Pages & Servlets .................. 6
    Multipage Transactions ......................... 7
    Hidden Fields ................................. 8

Copyright ©, All rights reserved. 2005 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (http://www.opencontent.org/opl.shtml) license defines the copyright on this document.
References

Past CS683 lecture notes

Reading Assignment

<table>
<thead>
<tr>
<th>Date</th>
<th>Chapters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sept 20</td>
<td>2, 3, 4</td>
</tr>
<tr>
<td>Sept 22</td>
<td>5, 6, 7</td>
</tr>
<tr>
<td>Sept 27</td>
<td>8, 9, 10</td>
</tr>
<tr>
<td>Sept 29</td>
<td>11, 12, 13</td>
</tr>
<tr>
<td>Oct 3</td>
<td>14, 15, 16</td>
</tr>
<tr>
<td>Oct 6</td>
<td>17, 18, 19</td>
</tr>
</tbody>
</table>

The chapter numbers above refer to chapters in the book Web Component Development with Zope 3, von Weitershausen, Springer-Verlag, 2005
About Web Applications

Datastore

Persistent storage of data
Files, Relational database, Object database
**Data - Datastore**
Find all data relevant to a request
Convert data from/to Datastore format to usable form
Handle concurrency
Transactions - rollback of a transaction

**Business logic**
Handle users request
Validate data
Handle concurrent requests
Handle transactions

**View**
Make users request accessible to program
Convert response in to html, CSS, Javascript etc

**Webserver**
Intermediary between client and Web app
Stateless
Basic HTTP (Web)
HTTP is Stateless

- Web browser connects to Web server with request
- Web Server handles request
  - Web Server connects (starts) to program (cgi, etc)
  - Program gets request
    - Program frequently connects to database
    - Program handles requests
    - Program returns response
- Web Server returns response
- All connections closed

All requests from Web Browser repeat this process
CGI, Server Pages & Servlets

Common ways to dynamically generate web pages

CGI & Servlets
Web request is passed to a program

```ruby
VeryBasicServlet>>doGet: aRequest response: aResponse

aResponse write: '<HTML><BODY>
Hello world</BODY></HTML>'.
```

Server Pages
Code is embedded in html pages

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
  <title>Hi</title>
</head>
<body>
  <% response write: 'Hello World'. %> 
</body>
</html>
```
Multipage Transactions

Information (state) from one page must be saved for next page

Information can be stored in

- Hidden Fields
- Sessions
- Database
Hidden Fields

Data is stored by client
Server can forget about client

<form action="fooBar" method="post" name="Sample">
    <input type="hidden" name="name" value="Whitney">
    <input type="hidden" name="cart" value="world peace">
    <input type="text" name="Credit Card Number" size="40">
    <input type="submit" name="submit">
</form>

<a href="bar/foo/index.html" title="Help information">Help</a>
Problems

Must insure all paths retain data

Pages are coupled

  Code depends on order of pages

  Reduces code reuse

Name clashes

  Field names must be different all other pages in transaction

Security Issues

Server can not track user

Presentation & domain logic mixed

<form action="fooBar" method="post" name="Sample">
  <% response
    write: 'input type="hidden" name="name" value="'.
    userName ifNotNil: [ response write: userName printString].
    response
    write: '">
    write: '<input type="hidden" name="cart" value="'
    cartItems
    do: [:each | response write: each printString]
    separatedBy: [response write: ',', '].
    response
    write: '">
  %>
  etc.
</form>