

Serolets

Contents

- Introduction to HTML
- HTML Tables and Forms
- HttpServlet Class
 - doGet method
 - doPost method
- Generating and Processing Dynamic Forms with Java Servlets

Hypertext Markup Language (HTML)

- HTML documents consist of ordinary text and “tags”.
- Tags were originally designed to specify the *structure* of a document -- not its format.
- A tag might, for example, specify that some text is a “level 1 heading” rather than specify the font and style.

HTML Document Structure

```
<html>
  <head>
    <title>Example 1</title>
  </head>
  <body>
    <h1><center>Example 1</center></h1>
    <p>The first line of the first
      paragraph. <br />
      The second line.</p>
    <p>The second paragraph.</p>
    <a href='some_url'>click this link</a>
  </body>
</html>
```

HTML Tables

```
<table border='1'>
<tr>
  <th>Name</th> <th>Phone Number</th>
</tr>

<tr>
  <td>Bergstein</td> <td>999-9184</td>
</tr>

<tr>
  <td>Bergandy</td> <td>999-8293</td>
</tr>
</table>
```

HTML Forms

- <form action='url' method='POST'>
 :
</form>
- Forms may contain *input* elements:
 <input type='xxx' name='yyy' value='zzz'>
- When a form is submitted, name and value pairs are sent to the web server.
- The names are used to retrieve values on the server side.
- The exact meaning of *name* and *value* depends on the *type* of the control.

type='text'

- Text field where users can edit text.
- The *value*, if specified, is the initial text to display.
- The *name* can be used to retrieve the text entered by the user.

type='password'

- Same as *type='text'*, except that * characters are displayed in the text field instead of the actual text.

type='checkbox'

- Only name/value pairs of checkboxes that are checked are submitted with the form.
- Several checkboxes with different values may have the same name. On the server, the result may be a parameter with multiple values.

type='radio'

- Radio buttons with the same name are grouped, such that only one per group may be selected.
- When the form is submitted, there will only be one value for the parameter *name*.

type='submit'

- A button that causes the form to be submitted.
- The value is the label on the button.
- If a submit button is clicked, the name/value pair for the button is submitted with the form.
- We might give several submit buttons the same name and test the value on the server side to determine which button was clicked.

Other input types

- Reset
- File
- Hidden
- Image
- Button

Form Elements

- Form elements may contain text and markup in addition to input elements.
- Forms specify an action and a method to be used for submitting the form.
- The action is a url, where the form should be sent.
- If the method is “GET”, the form data is sent as part of the url. If it is “POST” the data is sent as part of the HTTP request.

Order Form (1)

Select Items

- Monitor
- Keyboard
- Mouse
- Speakers

Choose Color

- Red
- Orange
- Blue
- Yellow
- Purple

Credit Card Number

Order Form (2)

```
<html>
<head>
  <title>Form Example</title>
</head>
<body bgcolor='#FFFFFF' text='#000000'>

<form action='myFormServlet' method='POST'>

<b>Select Items</b><br>
<input type='checkbox' name='items'
  value='monitor'>Monitor <br>
<input type='checkbox' name='items'
  value='keyboard'>Keyboard <br>
<input type='checkbox' name='items'
  value='mouse'>Mouse <br>
<input type='checkbox' name='items'
  value='speakers'>Speakers <br>
```

```
<p><b>Choose Color</b><br>
<input type='radio' name='color'
  value='red' checked>Red <br>
<input type='radio' name='color'
  value='orange'>Orange <br>
<input type='radio' name='color'
  value='blue'>Blue <br>
<input type='radio' name='color'
  value='yellow'>Yellow <br>
<input type='radio' name='color'
  value='purple'>Purple <br>

<p><b>Credit Card Number</b>
<input type='text' name='cc_num'
  size='16' maxlength='16'><br>

<p><input type='submit' name='submit'
  value='Submit Order'>

</form>
</body>
</html>
```

Order Form (3)

HttpServlet Class

- Servlets generally extend HttpServlet and override the doGet and/or doPost methods.
- A web server may be configured so that certain url's map to servlets instead of static files.
- When a web server receives a GET or POST request for a url that maps to a servlet, it invokes the servlets doGet or doPost method, respectively.

doGet and doPost Parameters

```
public void doGet(HttpServletRequest req,  
    HttpServletResponse res)
```

```
public void doPost(HttpServletRequest req,  
    HttpServletResponse res)
```

- The HttpServletRequest can be used to get information that was sent from the browser to the server.
- The HttpServletResponse is used to send information back to the browser.

HttpServletRequest Methods

- *To get the current session, or create a new one*
`public HttpSession getSession();`
- *To get the value of a parameter*
`public String getParameter(String name);`
- *To get the values of a multivalued parameter, or null if the parameter does not exist.*
`public String[] getParameterValues(String name);`
- *To get a RequestDispatcher for forwarding or including. (If path starts with "/" it is relative to context root, otherwise relative to the request of calling servlet.)*
`public RequestDispatcher getRequestDispatcher(String path)`

More HttpServletRequest Methods (also in HttpSession)

- `public void setAttribute (String name, Object value)`
- `public Object getAttribute (String name)`
- `public void removeAttribute (String name)`

RequestDispatcher Methods

- `public void forward(ServletRequest req, ServletResponse res)`
- `public void include(ServletRequest req, ServletResponse res)`
 - If the resource is dynamic, its *output* is included in the response.

HttpServletResponse Methods

```
public void setContentType(String type);  
public ServletOutputStream getOutputStream();  
public PrintWriter getWriter();  
public String encodeRedirectURL(String url);  
public void sendRedirect(String url);
```

Form Generation Example

```
public void doGet(HttpServletRequest req,
    HttpServletResponse res)
{
    res.setContentType("text/html");
    PrintWriter out = res.getWriter();
    out.println("<html><body>");

    out.println("<form action='myServlet', "
        + "method='POST'>");

    out.println("<p><b>Choose a Color</b> <br>");

    // Query db for colors currently in stock
    ResultSet results =
        stmt.executeQuery("select * from colors");
```

Form Generation Example (2)

```
String prefix = "<input type='radio', "
    + "name='color', value=''";

while (results.next()) {
    String color = results.getString(1);
    out.println(prefix + color + "'>"
        + color + " <br>");
}

// ...

out.println("</form> </body> </html>");
}
```

Form Processing Example

```
public void doPost(HttpServletRequest req,
    HttpServletResponse res)
{
    String credit_card =
        req.getParameter("cc_num");

    String color = req.getParameter("color");

    String[] items =
        req.getParameterValues("items");

    // Store the form data in a database,
    // or whatever ...

    // Redirect the browser to the next page
    String url =
        res.encodeRedirectURL("thankYou.html");
    res.sendRedirect(url);
}
```