

## Handling Forms

HTML *forms* are for browsers to send queries and data to the server.

A form may contain a series of fields including:

- Buttons
- Check boxes
- Radio buttons
- Text input fields and areas
- Drop down menus

## HTML Forms

HTML forms are defined by the `form` tag:

```
<form method=m action=uri >
  ... fields ...
</form>
```

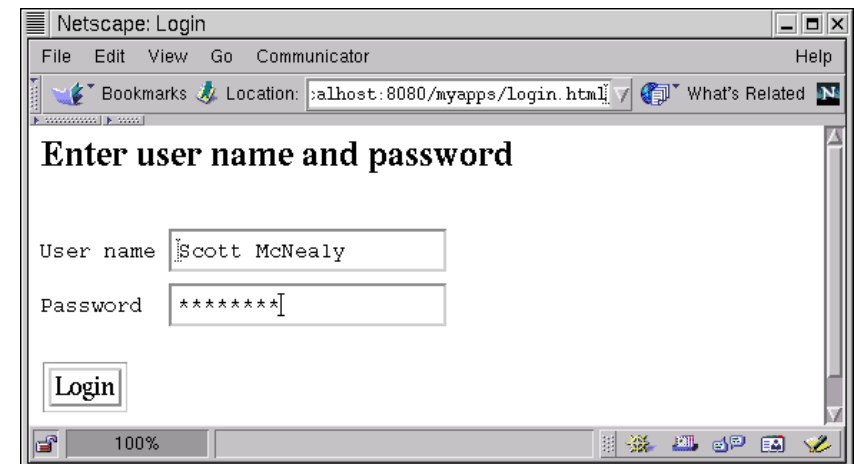
- The *method* attribute specifies the request method (`GET` or `POST`) used to send the form data.
- The *action* attribute identifies the URI of the servlet to be executed to process the form.

## Request Methods

The form data are sent to the server in the request as a sequence of *URL-encoded* name-value pairs.

- `GET` method  
The key-value pairs are encoded and appended to the request URI.
- `POST` method  
The key-value pairs are encoded and included in the message body.

## A Login Form



# Login Form -- HTML Source

```
<!-- login.html -->
<html >
<head><title>Login</title></head>
<body bgcolor="white">
<form method="post"
      action="/myapps/servlet/forms.Login">
<h1>Enter user name and password</h1>
<pre>
User name <input name="username" size="20" />
Password  <input type="password" name="password"
              size="20" />
</pre>
<input type="submit" value="Login" />
</form>
</body>
</html >
```

# Request Parameters

For HTTP servlets, form data can be retrieved as *request parameters*.

## String getParameter(String name)

- ◆ Returns the *URL-decoded* value of the first occurrence of the parameters with the specified name.
- ◆ Works identically for **GET** and **POST** requests.
- ◆ Returns **null** if no such parameter is present.

# A Login Servlet

```
package forms;

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Login extends HttpServlet {
    protected Map users = new HashMap();
    public void init() {
        // username and password
        users.put("Scott McNealy", "lavender");
        users.put("Steve Jobs", "aqua");
        users.put("Bill Gates", "blue");
    }
}
```

# A Login Servlet (cont'd)

```
public void doPost(HttpServletRequest request,
                  HttpServletResponse response)
    throws ServletException, IOException {
    String username =
        request.getParameter("username");
    String password =
        request.getParameter("password");
    if (username != null) {
        username = username.trim();
    }
    if (password != null) {
        password = password.trim();
    }
}
```

## A Login Servlet (cont'd)

```
if (username != null &&
    username.length() > 0) {
    if (password != null &&
        password.length() > 0) {
        String pw = (String) users.get(username);
        if (pw != null) {
            if (pw.equals(password)) {
                String firstname = username;
                int i = username.indexOf(' ');
                if (i > 0) {
                    firstname = username.substring(0,
i);
                }
                sendPage(response, "Login successful ",
                    "Hello " + firstname + "!");
            } else {
                //...
            }
        }
    }
}
```

## A Login Servlet (cont'd)

```
void sendPage(HttpServletRequest response,
               String title, String message)
    throws IOException{
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html>");
    if (title != null) {
        out.println("<head><title>" + title +
            "</title></head>");
    }
    out.println("<body>");
    if (message != null) {
        out.println("<h1>" + message + "</h1>");
    }
    out.println("</body>");
    out.println("</html>");
}
```

## A Login Servlet (cont'd)

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse
response)
    throws ServletException, IOException {
    doPost(request, response);
}
```

## Deploy Login Servlet

### Deploy:

- ♦ copy `login.html` to:  
`<myapps home>/`
- ♦ copy `Login.class` to:  
`<myapps home>/WEB-INF/classes/forms/`

### Invoke:

`http://localhost:8080/myapps/login.html`

# A Registration Form

Enterprise Java Expo Registration

◆Mr. ◆Mrs. ◆Ms. ◆Dr.

First name  Last name

E-mail  Organization

Primary area of business

Mailing address:  
Street  City   
State  Country   
Postal Code

Interest areas:  
 Servlet  JSP  EJB  JDBC  
 JavaMail  JNDI  JMS  JTA  
 JTS  IDL  RMI-IIOP  J2EE Connector

Comments:

Copyright © 2001-2002, Xiaoping Jia.

3-13/35

# HTML Form: Radio Button

◆Mr. ◆Mrs. ◆Ms. ◆Dr.

- ◆ Only one of the radio buttons can be selected.

```
<input type="radio" checked name="salutation"
value="Mr.">Mr. </input>
<input type="radio" name="salutation"
value="Mrs.">Mrs. </input>
<input type="radio" name="salutation"
value="Ms.">Ms. </input>
<input type="radio" name="salutation"
value="Dr.">Dr. </input>
```

Copyright © 2001-2002, Xiaoping Jia.

3-14/35

# HTML Form: Text Input

First name  Last name

E-mail  Organization

```
First name
<input type="text" size="20" name="firstname">
Last name
<input type="text" size="20" name="lastname">
E-mail
<input type="text" size="20" name="email">
Organization
<input type="text" size="20" name="company">
```

Copyright © 2001-2002, Xiaoping Jia.

3-15/35

# HTML Form: Drop-Down Menu

Primary area of business

```
<select name="business">
  <option value="Software Development">
    Software Development</option>
  <option value="Consulting">Consulting</option>
  <option value="E-Commerce">E-Commerce</option>
  <option value="Financial">Financial</option>
  <option value="Education">Education</option>
  <option value="Government">Government</option>
  <option value="Research">Research</option>
  <option value="Unknown">Other</option>
</select>
```

Copyright © 2001-2002, Xiaoping Jia.

3-16/35

# HTML Form: More Text Input

Mailing address:

Street	<input type="text" value="901 San Antonio Road"/>	City	<input type="text" value="Palo Alto"/>
State	<input type="text" value="CA"/>	Country	<input type="text" value="USA"/>
Postal Code	<input type="text" value="94303"/>		

```
Street
<input type="text" size="20" name="street">
City
<input type="text" size="20" name="city">

State
<input type="text" size="20" name="state">
Country
<input type="text" size="20" name="country">

Postal Code
<input type="text" size="20" name="postal">
```

# HTML Form: Checkbox

Interest areas:

<input checked="" type="checkbox"/> Servlet	<input checked="" type="checkbox"/> JSP	<input checked="" type="checkbox"/> EJB	<input type="checkbox"/> JDBC
<input checked="" type="checkbox"/> JavaMail	<input type="checkbox"/> JNDI	<input type="checkbox"/> JMS	<input type="checkbox"/> JTA
<input type="checkbox"/> JTS	<input type="checkbox"/> IDL	<input type="checkbox"/> RMI-IIOP	<input type="checkbox"/> J2EE Connector

```
<input type="checkbox" name="interests"
value="Servlets">Servlet</input>
<input type="checkbox" name="interests"
value="JSP">JSP</input>
<input type="checkbox" name="interests"
value="EJB">EJB</input>
<input type="checkbox" name="interests"
value="JDBC">JDBC</input>
...
```

# HTML Form: Text Area

Comments:

```
Comments:
<textarea cols="80" rows="3"
name="comments">
</textarea>
```

# More Methods for Retrieving Parameters

`String[] getParameterValues(String name)`

- Returns an array of the URL-decoded values of all occurrences of the parameters with the specified name.
- Returns an one-element array if the parameter is not repeated
- Returns `null` if no such parameter is present.

`Enumeration getParameterNames()`

- Returns all names of request parameters.

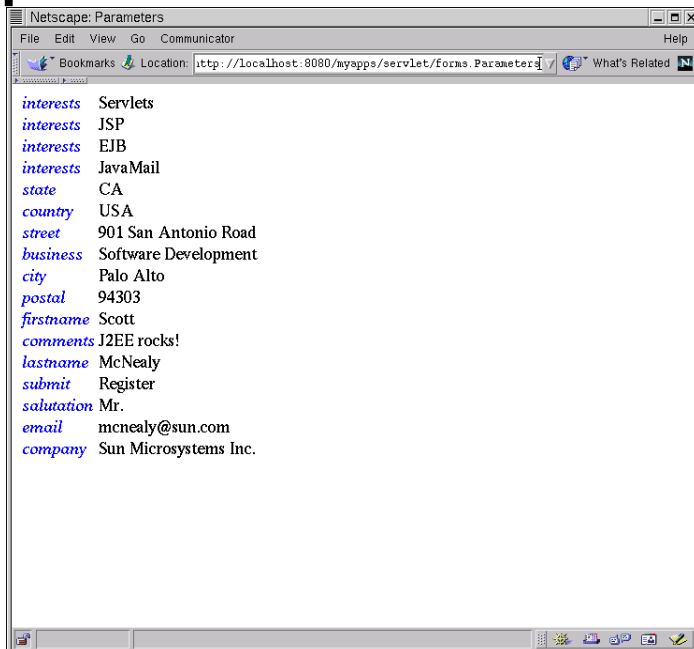
# Retrieving All Parameters

```
public class Parameters extends HttpServlet {  
    public void doPost(HttpServletRequest request,  
                       HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/html");  
        PrintWriter out = response.getWriter();  
        out.println("<html >");  
        out.println(  
            "<head><title>Parameters</title></head>");  
        out.println("<body>");  
        out.println("<table border=0>");
```

# Retrieving All Parameters (cont'd)

```
Enumeration e = request.getParameterNames();  
while (e.hasMoreElements()) {  
    String name = (String) e.nextElement();  
    String values[] =  
        request.getParameterValues(name);  
    if (values != null) {  
        for (int i = 0; i < values.length; i++) {  
            out.println("<tr><td><i><font color=blue>"  
                + name + "<font></i></td><td>"  
                + values[i] + "</td></tr>");  
        }  
    }  
}  
out.println("</table>");  
out.println("</body>");  
out.println("</html >");  
}
```

# Response of Parameters Servlet



# Using ECS

## ECS (Element Construction Set)

- ♦ An open source project by Apache
  - ♦ <http://jakarta.apache.org/ecs/>
  - ♦ current version: 1.4.1
- ♦ A Java API for generating elements for various markup languages:
  - ♦ HTML 4.0
  - ♦ XHTML
  - ♦ XML
  - ♦ WML
- ♦ It is easily extensible

## Construct HTML Documents

- ◆ Using ECS, an HTML document is constructed as a tree of elements.
- ◆ The root of the tree is an element of class `Html`
- ◆ Each tag in HTML has a corresponding element class

```
<head>    Head
<body>    Body
<table>   Table
<tr>     TR
<td>     TD
<ul>     UL
<li>     LI
```

etc.

## Construct HTML Documents (cont'd)

- ◆ Each element class usually has following methods:
  - `addElement(child)`
    - ◆ adds a child element, if the element allows children
    - ◆ returns the (parent) element itself
  - `setXYZ(value)`
    - ◆ sets the attribute `xyz` to `value`
    - e.g. `setSize(20)`  
`setValue("Login")`
    - ◆ returns the element itself
- ◆ See API for details.

## Getting Parameters (ECS Version)

```
package forms;

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.apache.ecs.*;
import org.apache.ecs.html.*;

public class Parameters2 extends HttpServlet {

    <doPost() and doGet() methods>

}
```

```
public void doPost(HttpServletRequest request,
                    HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    Table table = new Table(0); // border=0
    Html html = new Html()
        .addElement(new Head()
            .addElement(new Title("Parameters")))
        .addElement(new Body()
            .addElement(table));
    Enumeration e = request.getParameterNames();
    while (e.hasMoreElements()) {
        String name = (String) e.nextElement();
        String values[] =
            request.getParameterValues(name);
    }
}
```

(doPost() method continued)

```
if (values != null) {
    for (int i = 0; i < values.length; i++) {
        table.addElement(new TR()
            .addElement(new TD()
                .addElement(new I()
                    .addElement(new Font()
                        .setColor(HtmlColor.BLUE)
                        .addElement(name))))
            .addElement(new TD()
                .addElement(values[i])));
    }
}
out.println(html.toString());
}
```

## Load Additional JAR Files

There are several ways to load additional jar files:

- add jar files in `<webapp>/WEB-INF/lib/`
  - only available to one web app
- add jar files in `<tomcat home>/lib/`
  - available to all web apps in the servlet container
- add jar files in `<tomcat home>/common/lib/`
  - available to all web apps in the servlet container as well as the Tomcat engine
- add jar files in `<Java home>/jre/lib/ext/`
  - available to all Java programs

To compile your program, the jar files must be included in the classpath.

## Registration Servlet

```
public void doPost(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    Body body = new Body();
    Html html = new Html()
        .addElement(new Head()
            .addElement(new Title("Registration")))
        .addElement(body);

    body.addElement(new H1(
        "Thank you! Your registration is confirmed."));
}
```

(doPost() method continued)

```
body.addElement(
    request.getParameter("salutation") + " " +
    request.getParameter("lastname") + " " +
    request.getParameter("firstname"))
    .addElement(new BR());
body.addElement(request.getParameter("company"))
    .addElement(new BR());
body.addElement(
    request.getParameter("street"))
    .addElement(new BR());
body.addElement(
    request.getParameter("city") + " " +
    request.getParameter("state") + " " +
    request.getParameter("postal"))
    .addElement(new BR());
```

(doPost() method continued)

```
body.addElement(  
    request.getParameter("country"))  
    .addElement(new BR());  
body.addElement("Email: " +  
    request.getParameter("email"))  
    .addElement(new BR()).addElement(new BR());  
body.addElement("Primary area of business: " +  
    request.getParameter("business"))  
    .addElement(new BR());  
body.addElement("Interest areas");
```

(doPost() method continued)

```
String values[] =  
    request.getParameterValues("interests");  
if (values != null) {  
    UL list = new UL();  
    for (int i = 0; i < values.length; i++) {  
        list.addElement(new LI(values[i]));  
    }  
    body.addElement(list);  
}  
  
body.addElement("Comments: " +  
    request.getParameter("comments"));  
  
out.println(html.toString());  
}
```

## Registration Servlet Response

