

## 8.11 Using Cookies to Remember User Preferences

One of the most common applications of cookies is to use them to "remember" user preferences. For simple user settings, as here, the preferences can be stored directly in the cookies. For more complex applications, the cookie typically contains a unique user identifier and the preferences are stored in a database.

[Listing 8.7](#) presents a servlet that creates an input form with the following characteristics.

- **The form is redisplayed if it is incomplete when submitted.** The form sends data to a second servlet ([Listing 8.8](#)) that checks whether any of the designated request parameters is missing, then stores the parameter values in cookies. If no parameter is missing, the second servlet displays the parameter values. If a parameter is missing, the second servlet redirects the user to the original servlet so that the form can be redisplayed. The original servlet maintains the user's previously entered values by extracting them from the cookies.
- **The form remembers previous entries.** The fields are prepopulated with whatever values the user entered on the most recent request.

[Figures 8-11](#) through [8-13](#) show some typical results.

### Listing 8.7 RegistrationForm.java

```
package coreservlets;

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

/** Servlet that displays an HTML form to collect user's
 *  * first name, last name, and email address. Uses cookies
 *  * to determine the initial values of each of those
 *  * form fields.
 *  */

public class RegistrationForm extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String actionURL =
            "/servlet/coreservlets.RegistrationServlet";
        String firstName =
            CookieUtilities.getCookieValue(request, "firstName", "");
        String lastName =
            CookieUtilities.getCookieValue(request, "lastName", "");
        String emailAddress =
            CookieUtilities.getCookieValue(request, "emailAddress",
                "");

        String docType =
            "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 \" +
            \"Transitional//EN\">\n";
        String title = "Please Register";
        out.println
            (docType +
            "<HTML>\n" +
```

```

        "<HEAD><TITLE>" + title + "</TITLE></HEAD>\n" +
        "<BODY BGCOLOR=\"#FDF5E6\">\n" +
        "<CENTER>\n" +
        "<H1>" + title + "</H1>\n" +
        "<FORM ACTION=\"" + actionURL + "\">\n" +
        "First Name:\n" +
        "  <INPUT TYPE=\"TEXT\" NAME=\"firstName\" " +
        "    VALUE=\"" + firstName + "\"><BR>\n" +
        "Last Name:\n" +
        "  <INPUT TYPE=\"TEXT\" NAME=\"lastName\" " +
        "    VALUE=\"" + lastName + "\"><BR>\n" +
        "Email Address: \n" +
        "  <INPUT TYPE=\"TEXT\" NAME=\"emailAddress\" " +
        "    VALUE=\"" + emailAddress + "\"><P>\n" +
        "<INPUT TYPE=\"SUBMIT\" VALUE=\"Register\">\n" +
        "</FORM></CENTER></BODY></HTML>");
    }
}

```

## Listing 8.8 RegistrationServlet.java

```

package coreservlets;

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

/** Servlet that processes a registration form containing
 *  a user's first name, last name, and email address.
 *  If all the values are present, the servlet displays the
 *  values. If any of the values are missing, the input
 *  form is redisplayed. Either way, the values are put
 *  into cookies so that the input form can use the
 *  previous values.
 */

public class RegistrationServlet extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        boolean isMissingValue = false;
        String firstName = request.getParameter("firstName");
        if (isMissing(firstName)) {
            firstName = "Missing first name";
            isMissingValue = true;
        }
        String lastName = request.getParameter("lastName");
        if (isMissing(lastName)) {
            lastName = "Missing last name";
            isMissingValue = true;
        }
        String emailAddress = request.getParameter("emailAddress");
        if (isMissing(emailAddress)) {
            emailAddress = "Missing email address";
            isMissingValue = true;
        }
        Cookie c1 = new LongLivedCookie("firstName", firstName);
        response.addCookie(c1);
        Cookie c2 = new LongLivedCookie("lastName", lastName);
        response.addCookie(c2);
        Cookie c3 = new LongLivedCookie("emailAddress",
                                         emailAddress);
        response.addCookie(c3);
    }
}

```

```

String formAddress =
    "/servlet/coreservlets.RegistrationForm";
if (isMissingValue) {
    response.sendRedirect(formAddress);
} else {
    PrintWriter out = response.getWriter();
    String docType =
        "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 " +
        "Transitional//EN">\n";
    String title = "Thanks for Registering";
    out.println
        (docType +
         "<HTML>\n" +
         "<HEAD><TITLE>" + title + "</TITLE></HEAD>\n" +
         "<BODY BGCOLOR=\"#FDF5E6\">\n" +
         "<CENTER>\n" +
         "<H1 ALIGN>" + title + "</H1>\n" +
         "<UL>\n" +
         "  <LI><B>First Name</B>: " +
         "    firstName + "\n" +
         "  <LI><B>Last Name</B>: " +
         "    lastName + "\n" +
         "  <LI><B>Email address</B>: " +
         "    emailAddress + "\n" +
         "</UL>\n" +
         "</CENTER></BODY></HTML>");
    }
}

/** Determines if value is null or empty. */
private boolean isMissing(String param) {
    return((param == null) ||
           (param.trim().equals("")));
}
}

```

**Figure 8-11. Initial result of `RegistrationForm` servlet.**



**Figure 8-13. When the input form is completely filled in (top), the `RegistrationServlet` (bottom) simply displays the request parameter values. The input form shown here (top) is also representative of how the form will look when the user revisits the input form at some later date: form is prepopulated with the most recently used values.**



**Figure 8-12.** When the input form is incompletely filled in (top), the `RegistrationServlet` redirects the user to the `RegistrationForm` (bottom). The `RegistrationForm` uses cookies to determine the values of the form fields that were already filled in.

Please Register - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://localhost/servlet/coreservlets.RegistrationForm> Go

## Please Register

First Name:

Last Name:

Email Address:

Done Local intranet

Please Register - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://localhost/servlet/coreservlets.RegistrationForm> Go

## Please Register

First Name:

Last Name:

Email Address:

Done Local intranet

[\[ Team LiB \]](#)

[◀ PREVIOUS](#) [NEXT ▶](#)