Google code

☆ Google App Engine

Using JSPs

While we could output the HTML for our user interface directly from the Java servlet code, this would be difficult to maintain as the HTML gets complicated. It's better to use a template system, with the user interface designed and implemented in separate files with placeholders and logic to insert data provided by the application. There are many template systems available for Java, any of which would work with App Engine.

For this tutorial, we'll use JavaServer Pages (JSPs) to implement the user interface for the guest book. JSPs are part of the servlet standard. App Engine compiles JSP files in the application's WAR automatically, and maps them to URL paths.

Hello, JSP!

Our guest book app writes strings to an output stream, but this could also be written as a JSP. Let's begin by porting the latest version of the example to a JSP.

In the directory `war/`, create a file named `guestbook.jsp` with the following contents:

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ page import="com.google.appengine.api.users.User" %>
<%@ page import="com.google.appengine.api.users.UserService" %>
<%@ page import="com.google.appengine.api.users.UserServiceFactory" %>

<html>
  <body>

<%
    UserService userService = UserServiceFactory.getUserService();
    User user = userService.getCurrentUser();
    if (user != null) {
%>
<p>Hello, <%= user.getNickname() %>! (You can
<a href="<%= userService.createLogoutURL(request.getRequestURI()) %>">sign out</a>.)
</p>
<%
    } else {
%>
<p>Hello!
<a href="<%= userService.createLoginURL(request.getRequestURI()) %>">Sign in</a>
to include your name with greetings you post.</p>
<%
    }
%>

  </body>
</html>
```

By default, any file in `war/` or a subdirectory (other than `WEB-INF/`) whose name ends in `.jsp` is automatically mapped to a URL path. The URL path is the path to the `.jsp` file, including the filename. This JSP will be mapped automatically to the URL `/guestbook.jsp`.

For the guest book app, we want this to be the application's home page, displayed when someone accesses the URL `/`. An easy way to do this is to declare in `web.xml` that `guestbook.jsp` is the "welcome" servlet for that path.

Edit `war/WEB-INF/web.xml` and replace the current `<welcome-file>` element in the `<welcome-file-list>`. Be sure to remove `index.html` from the list, as static files take precedence over JSP and servlets.

```
    <welcome-file-list>
        <welcome-file>guestbook.jsp</welcome-file>
    </welcome-file-list>
```

Tip: If you are using Eclipse, the editor may open this file in "Design" mode. To edit this file as XML, select the "Source" tab at the bottom of the frame.

Stop then restart the development server. Visit the following URL: