Google code

☆ Google App Engine

Using the Users Service

Google App Engine provides several useful services based on Google infrastructure, accessible by applications using libraries included with the SDK. One such service is the Users service, which lets your application integrate with Google user accounts. With the Users service, your users can use the Google accounts they already have to sign in to your application.

Let's use the Users service to personalize this application's greeting.

Using Users

Edit `src/guestbook/GuestbookServlet.java` as indicated to resemble the following:

```java
package guestbook;

import java.io.IOException;
import javax.servlet.http.*;
import com.google.appengine.api.users.User;
import com.google.appengine.api.users.UserService;
import com.google.appengine.api.users.UserServiceFactory;

public class GuestbookServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse resp)
              throws IOException {
        UserService userService = UserServiceFactory.getUserService();
        User user = userService.getCurrentUser();

        if (user != null) {
            resp.setContentType("text/plain");
            resp.getWriter().println("Hello, " + user.getNickname());
        } else {
            resp.sendRedirect(userService.createLoginURL(req.getRequestURI()));
        }
    }
}
```

If you are using Eclipse and your development server is running in the debugger, when you save your changes to this file, Eclipse compiles the new code automatically, then attempts to insert the new code into the already-running server. Changes to classes, JSPs, static files and `appengine-web.xml` are reflected immediately in the running server without needing to restart. If you change `web.xml` or other configuration files, you must stop and start the server to see the changes.

If you are using Ant, you must stop the server and rebuild the project to see changes made to source code. Changes to JSPs and static files do not require restarting the server.

Rebuild your project and restart the server, if necessary. Test the application by visiting the servlet URL in your browser:

http://localhost:8888/guestbook

Instead of displaying the message, the server prompts you for an email address. Enter any email address (such as `alfred@example.com`, then click "Log In." The app displays a message, this time containing the email address you entered.

The new code for the `GuestbookServlet` class uses the Users API to check if the user is signed in with a Google Account. If not, the user is redirected to the Google Accounts sign-in screen. `userService.createLoginURL(...)` returns the URL of the sign-in screen. The sign-in facility knows to redirect the user back to the app by the URL passed to `createLoginURL(...)`, which in this case is the URL of the current page.

The development server knows how to simulate the Google Accounts sign-in facility. When run on your local machine, the redirect goes to the page where you can enter any email address to simulate an account sign-in. When run on App Engine, the redirect goes to the actual Google Accounts screen.

You are now signed in to your test application. If you reload the page, the message will display again.

To allow the user to sign out, provide a link to the sign-out screen, generated by the method `createLogoutURL()`.

Google Code offered in: English - Español - 日本語 - 한국어 - Português - Русский - 中文(简体) - 中文(繁體)