Altering a Page Flow

Table of contents

1. Introduction

In the previous pages, you were shown how to stitch together a controller class and a set of JSPs. One of the most powerful features of the Beehive way of building applications is being able to change the flow by editing only the controller class file, without having to touch the JSPs.

2. Logical Flow Change

Sometime after deploying the application, it is decided that before allowing a user to login, they must first be presented a page describing the terms of service for using the application. Additionally, a decision is made to remove the `thanks.jsp` file and have the "Sign Up" action automatically log-in the new user so they don't have to traverse the login screen. In short, after (successfully) completing signup.jsp, the user is taken directly to mypage.jsp.

The original logical flow looked like this:

logical flow, before alteration

After these changes, the logical flow now resembles:

logical flow, after alteration

3. Implementation Flow Change

To accommodate the new logical flow, the implementation flow model originally looked like this:

logical flow, after alteration

After removing `thanks.jsp`, adding `terms.jsp`, adding an `acceptTerms()` method, and re-routing the post-signup flow, looks like:

logical flow, after alteration

4. Changes Required in the JSPs

Strictly speaking, no changes to JSP content are required. Only deletion of the unused `thanks.jsp` and creation of the `terms.jsp` is all that is required.

Note:, The `signup.jsp` page that previously would direct a user to `thanks.jsp` but will now send him to `mypage.jsp` requires *absolutely no changes*.

The `terms.jsp` page would include a link through a new `acceptTerms()` controller method, similar to:

```
  <netui:anchor action="acceptTerms">Accept these Terms of
Service</netui:anchor>
```

5. Changes Required in the Controller Class

To change the flow, a few simple edits of the controller class are all that is required.

5.1. Redirect login()

Previously, the `login()` controller method was defined to statically return the forward to `login.jsp`.

```
    @Jpf.Action(
        forwards={
            @Jpf.Forward(name="success", path="login.jsp")
        }
    )
    public Forward login()
    {
        return new Forward( "success" );
    }
```

By changing the `path` property of the `Jpf.Forward` associated with this page, all links that previously took the user to `login.jsp` will now direct him to `terms.jsp`.

```
    @Jpf.Action(
        forwards={
            @Jpf.Forward(name="success", path="terms.jsp")
        }
    )
    public Forward login()
    {
        return new Forward( "success" );
    }
```

The `terms.jsp` page links through a new `acceptTerms()` controller method which simply is a static forward to `login.jsp`.

```
    @Jpf.Action(
        forwards={
            @Jpf.Forward(name="success", path="login.jsp")
        }
    )
    public Forward acceptTerms()
    {
        return new Forward( "success" );
```

```
        }
```

This biggest change in the controller class is in the `processSignUp()` controller method.
Instead of just recording the user's information, it now must also perform the logic of logging
in the user automatically. Additionally, the forward returned by it will send the user directly
to `mypage.jsp` instead of the now unused `thanks.jsp` page.

```
    @Jpf.Action(
        forwards={
            @Jpf.Forward(name="success", path="mypage.jsp")
        }
    )
    public Forward processSignUp(SignUpForm form)
    {
        // record the user's sign-up information.
        ...

        // perform automatic login for the user.
        getSession().setAttribute( "authenticated_user", form.getUsername()
);

        return new Forward( "success" );
    }
```

The flow has successfully be altered to now include a jump through the site's
terms-of-service and automatically logging-in newly signed-up users.

Java, J2EE, and JCP are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other
countries.

© 2004, Apache Software Foundation