

<XYZ EDITOR> Changes and Process

Return Page from Login not working

Date: 11/30/01

Problem: Log in screen not returning user to right place if user tried to check out a file without logging in first (and was then sent to login screen)

Solution: changed line in checkOut.jhtml

changed

```
<DROPLET BEAN="/dancomp/<XYZ Editor>Authentication">  
    <PARAM NAME="url" VALUE="" request.getRequest("URI")`">  
</DROPLET>
```

to

```
<DROPLET BEAN="/dancomp/<XYZ Editor>Authentication">  
    <PARAM NAME="url" VALUE="" request.getHeader("Referer")`">  
</DROPLET>
```

Why:

The original code looks like it should have worked, but it simply didn't work as intended, I guess. It seems like it was picking up checkOut.jhtml as the URI instead of nodeContent.jhtml ... which ... maybe that's right, but I am not sure (API not clear to me).

The users were looking at content in nodeContent.jhtml, and clicking on the Check Out button. They hadn't logged in, so they'd get the log in screen. They'd log in and then get sent to what looked like a blank page. That blank page was actually **checkOut.jhtml** which doesn't have any kind of html display.

I tried putting a hard coded URL in place of `request.getRequest("URI")` and that worked. So, then I spent a lot of time trying to figure out how to pass a parameter along to checkOut.jhtml from nodeContent.jhtml and nodeProperties.jhtml to set the origin of where the user came from so the log in screen could send them back.

I didn't get anywhere with this. Doesn't seem like I can pass a parameter from one page to another ... if I set the parameter on nodeContent.jhtml then it's just set for the display of nodeContent.jhtml. There was no way to actively pass it along to another page. Tried using a blank `<form action=post><input type=hidden value=xyz></form>` inside the `<a href>` around the Check Out button on the content page. No go.

nodeContent.jhtml does set some parameters, but they are then used on the same page, not passed to another page. (`request.setParameter("previewLink", previewLink)`; which is previously declared and assigned, then it's retrieved on a new page with `<AHREF="" request.getParameter("previewLink")" target=_BLANK>` or on the current page with code like `<VALUEOF PARAM="templateName">`)

Finally, I tried using `VALUE=""`request.getHeader("Referer")`"` on the `checkOut.jhtml` page, which worked. I had avoided trying this because this didn't work for me when I tried to use it in the key.com Tell A Friend project (I ended up passing parameters through the form for that one.)

The background on how the check out works is below:

It is possible for a user to look at the contents of a content file or a property file in <XYZ EDITOR> without first logging into <XYZ EDITOR>. One can do this by accessing <XYZ EDITOR>, clicking on one of the icons on the <XYZ EDITOR> home page to build a site tree, then clicking on a **content** or **property** link from one of the items listed in the site tree.

The user is taken to either `nodeContent.jhtml` or `nodeProperties.jhtml` (depending on what kind of item they selected for viewing). The user can then click on the **Check Out** button at the top of the page to try to check out the page.

If the user has not logged in the Log In screen appears. The user then logs in and is returned to `nodeContent.jhtml` or `nodeProperties.jhtml` with the page the person was viewing originally.

The code path is as follows from `nodeContent.jhtml` (similar for `nodeProperties.jhtml`):

`nodeContent.jhtml`'s **Check Out** button contains the following code:

```
<A  
HREF="checkOut.jhtml?nodeID=`request.getParameter("nodeID")`&type=content"><IMG  
G SRC="gfx/btn_checkoutbig.jpg" WIDTH="77" HEIGHT="40" BORDER="0"></A>
```

The user clicks on the button and moves along to `checkOut.jhtml`, carrying along the `nodeID` of the page they were viewing (such as A-2) and the type of page (such as content).

`checkOut.jhtml` does not display anything to the user. It contains the lines

```
<DROPLET BEAN="/dancomp/<XYZ Editor>Authentication">  
    <PARAM NAME="url" VALUE=""`request.getHeader("Referer")`">  
</DROPLET>
```

which creates a parameter called **url** and sets that to the URL of the referring page (such as `/<XYZ EDITOR>/nodeContent.jhtml?nodeID=A-2`

`<XYZ Editor>Authentication` runs, and since the user did not log in, it gets the information from the parameter **url** and creates a session object called **string2** from that. The code is as follows for this, in `<XYZ Editor>Authentication.java`:

```
{  
String string2 = dynamoHttpServletRequest.getParameter("url");  
httpSession.putValue("url", string2);  
dynamoHttpServletResponse.sendRedirect("/<XYZ EDITOR>/authenticateUser.jhtml");  
}
```

And the user is sent along to the authenticateUser.jhtml page, which is the log in screen. The session object obviously continues to exist while the user is at this new page.

authenticateUser.jhtml is a form, and that form is processed by LoginFormHandler.java.

The user types in a userid and password. LoginFormHandler.java processes that and then looks up the session object **url** to see where to redirect the user to upon successfully logging in. That piece of code is:

```
if (userAdminManager.login(userId, password, string1))
{
    String string2 = (String)httpSession.getValue("url");
    if (string2 == null)
        string2 = "/<XYZ EDITOR>/" ;
    dynamoHttpServletResponse.sendLocalRedirect(string2,
dynamoHttpServletRequest);
    return;
}
```

If the user logged in from the <XYZ EDITOR> home page, the user will be returned to the <XYZ EDITOR> home page because there is no **url** session object. Otherwise, the person will be returned to nodeContent.jhtml or nodeProperties.jhtml either of which will display the page the person had originally pulled up for viewing and check out.

So, the path taken is nodeContent.jhtml (or nodeProperties.jhtml) > checkOut.jhtml > authenticateUser.jhtml > LoggingFormHandler.java > nodeContent.jhtml (or nodeProperties.jhtml)

The files are located in the following cvs repositories under /<XYZ EDITOR>

jhtml contains

nodeContent.jhtml
nodeProperties.jhtml
checkOut.jhtml
authenticateUser.jhtml

all those files should be placed in /<XYZ EDITOR>/

<XYZ Editor>Servlets contains

LoginFormHandler.java

the compiled file should be placed in /FRkeycomsite/lib, above the /<XYZ EDITOR> directory