Development-Examples-Login

Simple Login

Here is a little example on how to redirect a request to a "login" page if necessary:

```
public void appendToResponse(WOResponse aResponse, WOContext aContext) {
   String       aPageName = LoginPage.pageWithContext( aContext );

   if ( ( aPageName != null ) && ( this.name().equals( aPageName ) == false ) ) {
     WOComponent        aComponent = this.pageWithName( aPageName );

     this.currentSession().setPage( aPageName );
     aContext._setPageComponent( aComponent );
     aContext._setCurrentComponent( aComponent );
     aComponent.appendToResponse( aResponse, aContext);

     return;
   }

   this.currentSession().setPage( this.name() );
   super.appendToResponse( aResponse, aContext);
 }
```

WOContext._setPageComponent() and WOContext._setCurrentComponent() are undocumented methods, but are required in order to update the WOContext with the current page and component information.

The LoginPage class (not shown) will decide if a login is necessary or not.

Avoiding Session Timeouts at Login

A good login should not timeout when no user is logged in. This can be a problem if your application uses a session in the page generation.

Fortunately Apple has provided us with a way of doing this correctly: "DirectAction";. You should implement your login page as a DirectAction and be very careful not to create a "lazy" session otherwise you will have worked for nothing. WO create a session when you call certain methods so you have to be careful. Use this.context().hasSession() to check that everything is OK. After successful login you create a session by calling this.session() in your component and store the user in it; you are in business.

On logout you destroy the session and return the direct action page that way the user has a new login screen instead of an empty screen. The easiest way of generating a login page on logout is to return a redirect page (302) to the main URL of the application it is transparent to the user and does exactly what you need.

You will find the code sample there after. I place it in the Application and when I want to logout I just call handleLogout with the context.

```
private WOResponse responseForPageWithName(String aPageName, WOContext aContext) {
   if ( aPageName != null ) {
       WOResponse       aResponse = new WOResponse();
       String adaptorPrefix = aContext.request().adaptorPrefix();
       String applicationName = aContext.request().applicationName();
       String anURL = adaptorPrefix + "/" + applicationName + ".woa";
       aResponse.setHeader(anURL, "Location");
       aResponse.setHeader("text/html", "content-type");
       aResponse.setHeader("0","content-length");
       aResponse.setStatus(302);
       return aResponse;
   }
   return null;
 }

 public WOResponse handleLogout(WOContext aContext) {
   WOResponse aResponse = this.responseForPageWithName("Login", aContext);
   if ( ! aContext.session().isTerminating() ) {
       ((Session)aContext.session()).setUser(null);
       aContext.session().terminate();
   }
   return aResponse;
 }
```