Software Documentation : Java Custom Online Review Login 1.1

This page last changed on May 31, 2010 by volodymyrk.

1. Scope

1.1 Overview

The component provides the login and logout action support for the Online Review application. Assemblies will be integrated with this component to complete the authentication process.

Version 1.1 adds new functionality to support cookies for Online Review. The cookies will be used to keep users signed in between sessions. The new and modified requirements are in blue.

In addition to the functional changes, designers are responsible for migrating the design to TCUML format, and for updating it and the documentation to account for modifications that have taken place since version 1.0. Developers are responsible for correcting any existing tests that are now failing, as well as for adding tests for the new functionality.

1.1.1 Version

Version 1.1

1.2 Logic Requirements

1.2.1 Log In Action

Struts action should be provided to take a username and password as input. The action will use the TopCoder Security Manager EJB to log the user in. Upon successful login the session will be cleared and the user's subject identifier will be stored in session under a configurable attribute name.

There will be a new "Remember me" check box in the login form. If it is checked a cookie will be added to the HTTP response to save the user ID and hash of the password. The cookie age will be set to maximum value so that the cookie will never expire.

1.2.2 Log Out Action

Struts action should be provided to log the user out. The session attribute that stores the user's subject identifier must be removed. The session will be invalidated. The user authentication cookie(s) will be cleared.

1.2.3 Cookies

A public method will be provided that given the HTTP request reads the cookies from it and retrieves user ID. The method will verify that the password hash is correct.

The provided file (BasicAuthentication.java) contains implementation of cookies for the TopCoder website. The basic idea is that we save a one-way hash of the user ID and password from security_user table in the cookie. Please refer to this file to get an idea of how things need to be implemented.

It is up to the designer to decide where the new method(s) will be added to, it may be a new class (e.g. CookieManager) or an existing one. The requirement is that Online Review application needs to be able to create an instance of the class once and then use it to make calls to retrieve user ID from HTTP request.

1.2.4 Online Review changes

The designers are also responsible to provide a description of the changes in OR needed to integrate the new functionality. This should be a high-level plan for what methods need to be changed and/or added in the application code and how.

The code of the Online Review is located under this SVN path. Please request SVN access in the forum.

1.2.5 Thread Safety

The component must be thread safe. An instance of the cookie manager class will be created to serve multiple requests from the application at the same time.

1.3 Required Algorithms

No specific algorithms are required.

1.4 Example of the Software Usage

User will initiate a login request with username and password. The action invokes the login EJB to authenticate the user. The user will then be forwarded to the Online Review project listing page

1.5 Future Component Direction

None.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.

2.1.2 External Interfaces

None.

2.1.3 Environment Requirements

- Development language: Java 1.4
- Compile target: Java 1.4, Java 1.5

2.1.4 Package Structure

com.cronos.onlinereview.login

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

- Session attribute name for the logged user
- DB connection configuration

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

- Struts
- EJB

3.2.2 TopCoder Software Component Dependencies:

- Configuration Manager

- Security Manager
- DB Connection Factory

3.2.3 Third Party Component, Library, or Product Dependencies:

None.

3.2.4 QA Environment:

- Solaris 7
- RedHat Linux 7.1
- Windows 2000
- Windows 2003

3.3 Design Constraints

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines. Modifications to these guidelines for this component should be detailed below.

3.4 Required Documentation

3.4.1 Design Documentation

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification
- Online Review change plan

3.4.2 Help / User Documentation

- Design documents must clearly define intended component usage in the 'Documentation' tab of the TopCoder UML tool.