ASSIGNMENT AUTHORIZATION

INTRODUCTION

In the stories W13 and W14 you modified the web shop to enable users to login and logout. This process is known as authentication. According to Wikipedia authentication is …

"The process of identifying an individual, usually based on a username and password.

*In security systems, **authentication** is distinct from **authorization**, which is the process of giving individuals access to system objects based on their identity..."*

In stories W10 and W11 you are asked to check whether an authenticated user has the proper role to access particular pages of the web application. Each time the user navigates to a page, the application should check whether this person is **authorized** to access the page.

STEP 1: ROLES – HIGH PRIORITY

Implement **StoryW15_AssignRoles**.

You will need to create a Java `enum`. For more information, check **Enum.pdf** and **ExampleUseOfEnumInJava.pdf**.

STEP 2: AUTHORIZATION – HIGH PRIORITY

For **StoryW16_ProtectAdminPages**, read the story and the rest of this assignment.

Depending on the role of a user, the user can perform certain actions:

Actions vs. Roles	Not logged in	Customer	Administrator
Open Home Page	V	V	V
Open User overview	X	X	V
Open Product overview	X	V	V
Add Product	X	X	V

When a user chooses to perform an action, the application will have to check if the person is allowed to perform that action. If the person is not allowed, an error message is given and the login form is shown.

WORKFLOW

Below you find the steps your controller-class will have to execute to handle if the user clicks on the Add Product link resp. the Products link:

Add Product	**Open product overview**
- read the parameter 'action' - if the value of 'action' = 'addProduct': - get the user from the session - check if there is a user and if his role is Role.ADMINISTRATOR - if not, throw a NotAuthorizedException - set the destination to addProduct.jsp - if a NotAuthorizedException is thrown - create an error message - add the error message to the request - set the destination to index.jsp - forward the request to the destination	- read the parameter 'action' - if the value of 'action' = 'productOverview': - get the user from the session - check if there is a user and if his role is Role.CUSTOMER or role.ADMINISTRATOR - if not, throw a NotAuthorizedException - get the list of products from the database - set the destination to productOverview.jsp - if a NotAuthorizedException is thrown - create an error message - add the error message to the request - set the destination to index.jsp - forward the request to the destination

BEFORE YOU BEGIN TO IMPLEMENT THESE STEPS...

We assume you have a front controller: one controller class (a servlet)...

- with a method (`processRequest()`), that is called from the `doGet()` and `doPost()` methods
- in this method `processRequest()`, you have an if-block that checks the value of a request-parameter `action`
- for each possible value of `action` you have a private method with the code that needs to be executed if the action has the corresponding value
- each private method returns the destination page
- the request is then forwarded to the destination page.

IMPLEMENTATION OF AUTHORIZATION

Modify your `Controller` class to implement authorization.

On the next page you can find some tips, but try it on your own first. Maybe you don't need the tips ☺.

TIPS

1. Create a class `NotAuthorizedException`. Make it an unchecked exception.

2. Create a method `checkRole(request:HttpServletRequest, roles:Role[])` in your controller. The method:
   - Gets the user from the session. If no user is found, the method throws a `NotAuthorizedException.`
   - If a user is found in the session, the method checks if the role of the user is in the list of roles passes as a parameter. If not, a `NotAuthorizedException` is thrown.

3. In your controller, you probably have private methods for each action: to add a product, navigate to the user overview, ... In these methods, you should first call the method `checkRole(request, <the roles required for this action>).`
   This will assure that the rest of the method is only executed if the user has the proper role.

4. Finally, you have to catch the exception if it was thrown.
   In your controller you probably have a method `processRequest()` with an if-block that checks the value of you parameter `action`. Surround the entire if-block with a try-catch, catching the `NotAuthorizedException`. In the catch-block you should
   - create an error message: *"Insufficient rights"*
   - add the error message to the request, as an attribute
   - set `index.jsp` as the destination

5. In the `index.jsp` page, make sure the error message and the login form is shown.

STEP 3: MANAGE ROLES – NORMAL PRIORITY

Implement **StoryW17_ManageRoles**.