

# Securing Java Web Applications-1

For both the exercises below, create the following roles in **tomcat-users.xml**

- customer
- admin

Also create the following users in **tomcat-users.xml**

- At least one user with the customer role.
- Create at least one user with the admin role

Create a new web project, and include a *web.xml* file in the project → Right click the project node, select *New Standard Deployment Descriptor (web.xml)* which is located under the Web category.

In the projects folder, create two folders, each with an html-file as sketched below:  
(Just add some dummy content to the html-pages, like "Customer-page" and "Admin-page")

- customer
  - customer.html
- admin
  - admin.html

In your *index.html*, create two links to the pages created above (you could also add a back button to both pages, to bring you back to the index page)





## Security with Basic Auth

Now add the necessary changes to *web.xml* in order to:

- Let the project use Basic Authentication
- Ensure that only users with the customer- or admin role, can access pages in the customer folder
- Ensure that only users with the admin role can access pages in the admin folder.

Test and verify that the security constraints given above is in place

Explain the following:

- Which HTTP messages is sent from server to client in order for this to work, That is how does the server tell the browser to pop up the login dialog (Chrome hides this, you must either use Firefox to monitor communication, or just Google to come up with the answer) 
- For subsequent calls to a protected resource, where is the information stored that tells we are already logged in (Server or Client)? 
- This relates to the question above. Is basic auth stateful or stateless (seen from the servers point of view)? 
- Why is it so problematic to log of using Basic Auth. 

## Form Based Authentication

In the same project as above, add two files as sketched below:

- Login.htm → Add the form from the slide Form Based Authentication to this slide.
- LoginFailed → Add a single paragraph with text like "Login failed" and a link with the text "try again" which should lead back to the login-page.

Change web.xml to use Form Based Authentication, using the two files created above.

Test and verify that the security constraints is (still) in place

Provide the main page (index.html) with log out button. In order to figure out how to log out, you might have to answer the questions below first.

Explain the following:

- How does the server know that users are logged-in for subsequent calls to a protected page after we have logged-in (why is it not necessary to retype the username/password)
- For subsequent calls to a protected resource, where is the information stored that tells we are already logged in (on the Server or in the Client)?
- This question relates to the question above. Is Form Based Authentication stateful or stateless (seen from the servers point of view)?