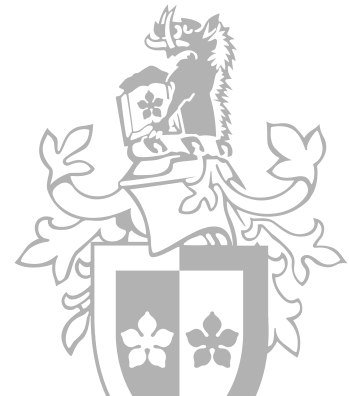


COS30041 Creating Secure and Scalable Software

Lecture 08a: **Realm & 7.1P**



SWIN
BUR
* NE *

SWINBURNE
UNIVERSITY OF
TECHNOLOGY

Commonwealth of Australia
Copyright Act 1968

Notice for paragraph 135ZXA (a) of the *Copyright Act 1968*

Warning

This material has been reproduced and communicated to you by or on behalf of Swinburne University of Technology under Part VB of the *Copyright Act 1968* (the Act).

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice.

Realm, User and Group [Java EE]

- A Realm = A security policy domain defined for a web or application server
 - Contain a collection of users with or without groups
- A User = An individual that has been defined in the [GlassFish] server
- A Group = A set of authenticated users defined in the [GlassFish] server

Different types of Realm [JavaEE]

■ File-realm*

- ☐ **Stores user credentials in a file**

■ Admin-realm

- ☐ A file realm that stores administrators' user credentials

■ JDBC-realm [Java EE]

- ☐ **Store user credentials in database records**

■ Certificate

- ☐ Store user credentials in a certificate database

GlassFish™ Server Open Source Edition



Tree

- ED Container
 - HTTP Service
 - JVM Settings
 - Java Message Service
 - Logger Settings
 - Monitoring
 - Network Config
 - ORB
 - Security
 - Realms
 - admin-realm
 - certificate
 - file
 - Audit Modules
 - JACC Providers
 - Message Security
 - System Properties
 - Thread Pools
 - Transaction Service

File Users

Back

Manage user accounts for the currently selected security realm.

Configuration Name: server-config

Realm Name: file

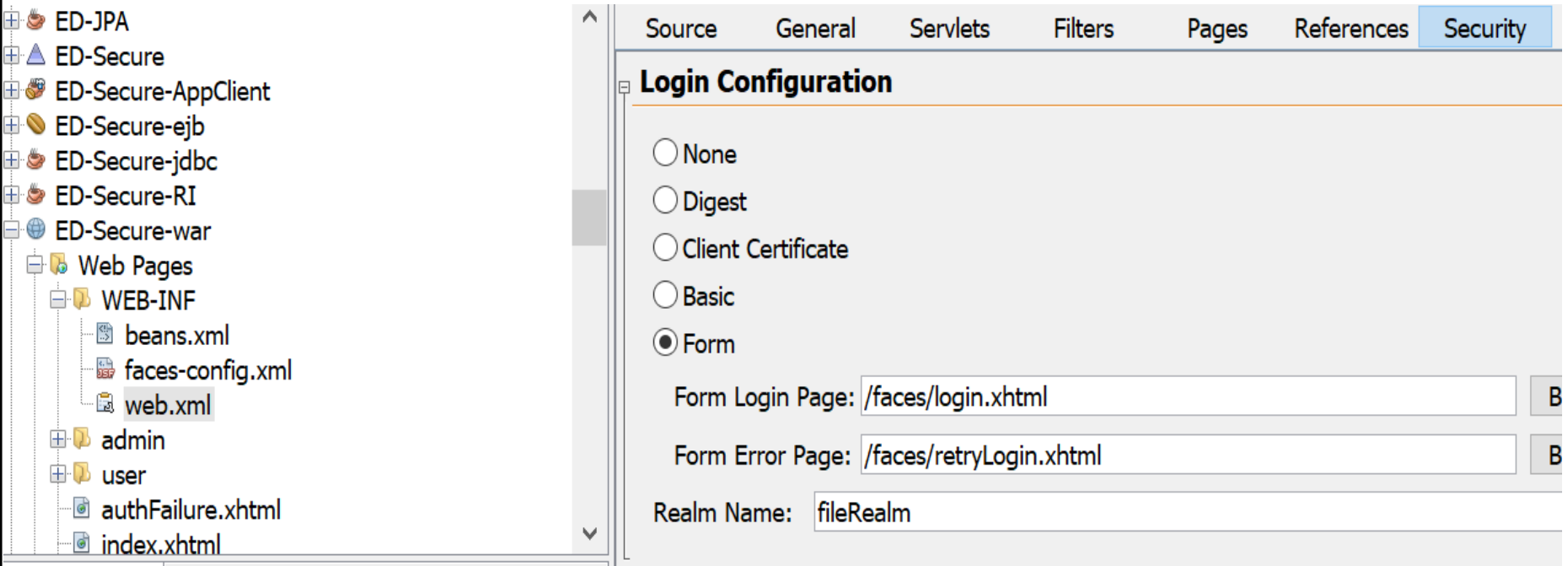
File Users (6)

New...

Delete

Select	User ID	Group List:
<input type="checkbox"/>	ed-none	ED-NONE
<input type="checkbox"/>	ed-guest	ED-APP-GUEST
<input type="checkbox"/>	ed-admin1	ED-APP-ADMIN
<input type="checkbox"/>	ed-admin2	ED-APP-ADMIN
<input type="checkbox"/>	ed-user1	ED-APP-USERS
<input type="checkbox"/>	ed-user2	ED-APP-USERS

LT6.3 – Lab 07b



The screenshot displays an IDE interface. On the left, a project tree shows a web application structure with files like `beans.xml`, `faces-config.xml`, `web.xml`, and `index.xhtml`. The main editor area is titled 'Login Configuration' and has tabs for 'Source', 'General', 'Servlets', 'Filters', 'Pages', 'References', and 'Security'. The 'Security' tab is active, showing radio buttons for authentication methods: 'None', 'Digest', 'Client Certificate', 'Basic', and 'Form'. The 'Form' option is selected. Below these, there are text fields for 'Form Login Page' (set to `/faces/login.xhtml`) and 'Form Error Page' (set to `/faces/retryLogin.xhtml`), each with a 'Browse' button. At the bottom, the 'Realm Name' is set to `fileRealm`.

Login using the file realm

file realm

- The file realm is NOT related to those records in the table in the database.

- You could look at another example program under Week 7

[EX-SLSB-SecInfo.zip](#)

It uses the file realm. This sample program does not perform any work of accessing the table in the database.

- Which part in the resources is allowed to access?
 - ☐ Declarative Security
 - ☐ Programmatic Security

Declarative

vs

Programmatic

Declarative Security

- Specify the application component's security requirements by using either
 - Deployment descriptors, or
 - Annotations
- `@DeclareRoles`
- `@RolesAllowed`
- `@PermitAll`
- See EX-SLSB-SecInfo

Programmatic Security

- Security decisions were programmed in the source code
- Useful when declarative security is not sufficient to express the security model of an application
- `SessionContext.getCallerPrincipal()`
- `SessionContext.isCallerInRole()`
- `Principal.getName()`
- See EX-SLSB-SecInfo

7.1P

← → ↻ ⓘ localhost:8080/ED-Secure-war/faces/index.xhtml

Secure Company Ltd Home Page

Welcome to our company

We aim at developing secure enterprise application solutions to corporations.

[Employee Management System \(Amin\)](#)

[Employee Management System \(Employee\)](#)

7.1P

← → ↻ ⓘ localhost:8080/ED-Secure-war/faces/user/mainmenu.xhtml

SECURE Company Ltd

Employee Management System

Main Menu

1. [Display employee's details](#)
2. [Change an employee's details](#)
3. [Change an employee's password](#)

Click

7.1P

set up welcome page for employees

Source General Servlets Filters **Pages** References Security History Welcome Files

Welcome Files

Welcome Files:

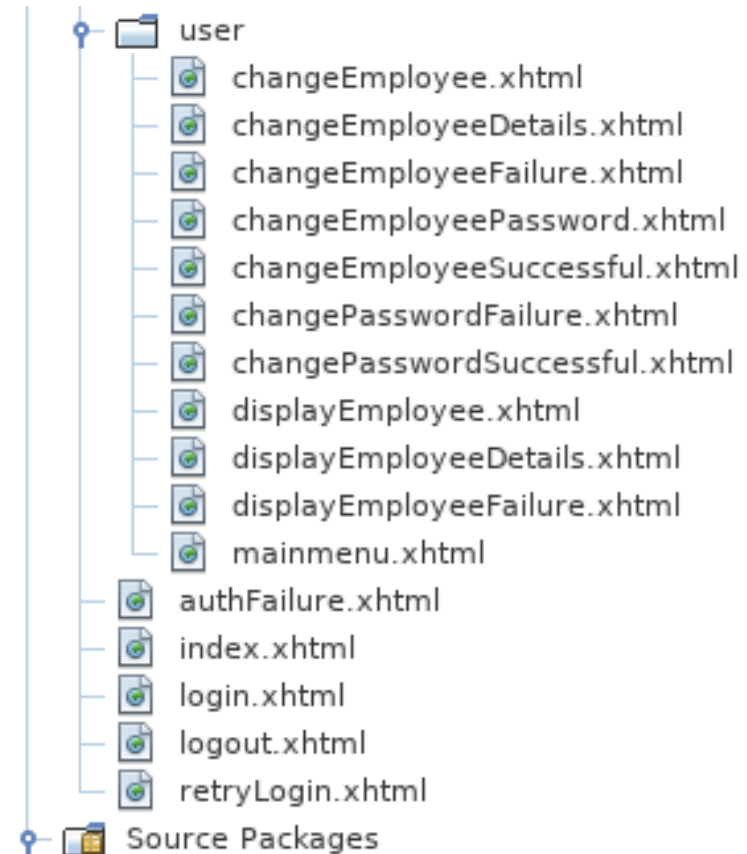
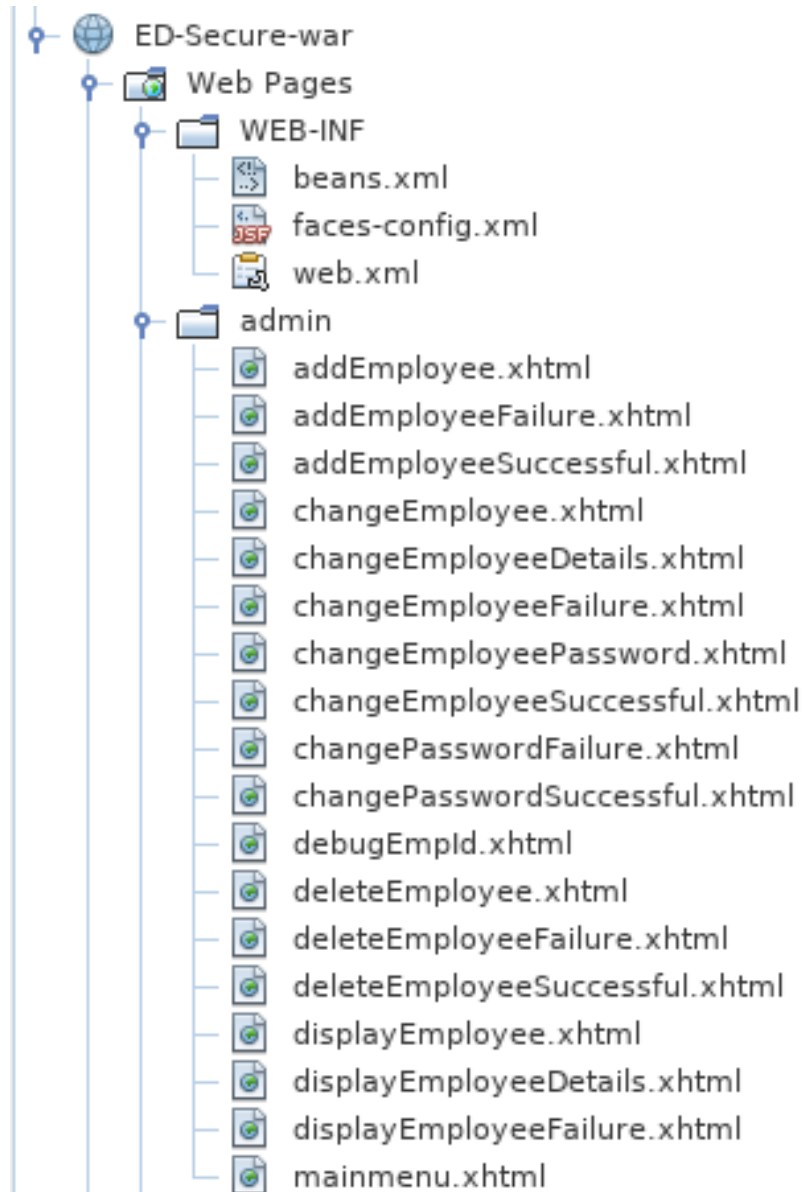
Use comma(,) to separate multiple welcome files.

[Go To Source\(s\)](#)

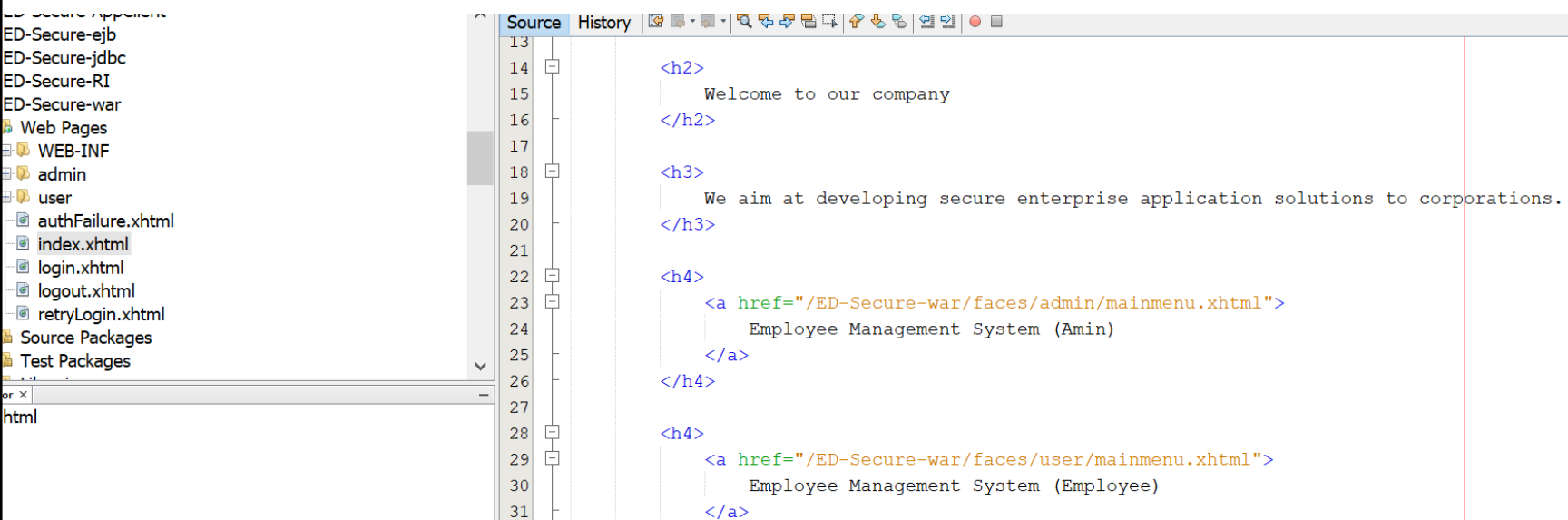
Error Pages

Error Page Location	Error Code	Exception Type
/faces/authFailure.xhtml	403	

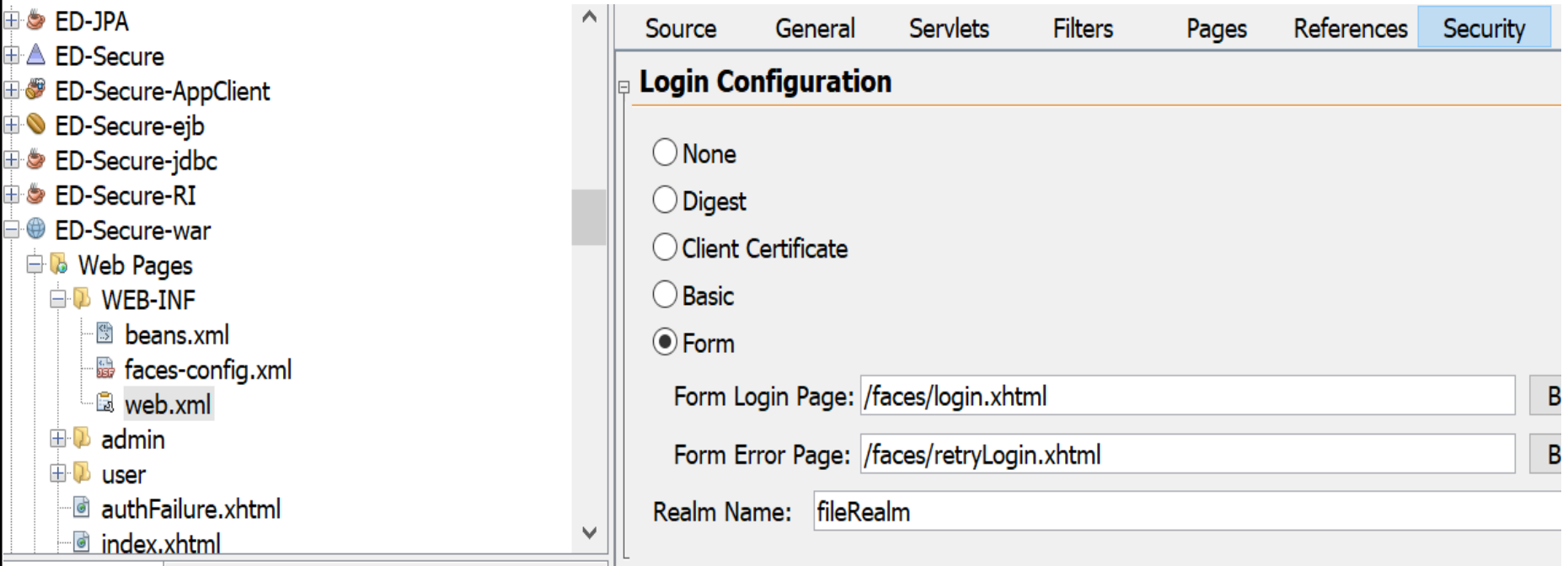
Create web pages for employees



Index.xhtml - modified



LT6.3 (Lab 07b) – it has been done



Login using the file realm

Secure Company Ltd Home Page

Welcome to our company

We aim at developing secure enterprise application solutions to corporations.

[Employee Management System \(ADMIN\)](#)

[Employee Management System \(EMPLOYEE\)](#)

SECURE Company Ltd

Employee Management System

Login Page

Username

Password

Login

Reset

Login using the file realm

SECURE Company Ltd

Employee Management System

Main Menu

1. [Display employee's details](#)
2. [Change an employee's details](#)
3. [Change an employee's password](#)

Click

After selecting 1:

Note that: “ed-user1” in file realm is not linked to empid (e.g. 00003) in the table in database.



localhost:8080/ED-Secure-war/faces/user/displayEmployee.xhtml

Search for an employee

Please enter the employee id and password

Employee Id:

Password:

file realm (previous slide)

- The file realm is NOT related to those records in the table in the database.
- You could look at another example program under Week 7

[EX-SLSB-SecInfo.zip](#)

It uses the file realm. This sample program does not perform any work of accessing the table in the database.

To access a record in the database

- if a user wants access a record in the table, the user should enter empid (00003) & password (pwd)
- the program to be developed should
 - 1) get the record (name, password, ...) by using the empid (00003).
 - 2) compare the password inputted and the password got from database, if they are the same, the record for to empid (00003) can be shown up / can be updated

tips: Employee employee = employeeFacade.find(empId);

*check: employee.getAppGroup().equals("ED-APP-USERS")
 employee.getPassword().equals(pwd))*

The user with empid (00003) can only access his/her own record not others.



localhost:8080/ED-Secure-war/faces/user/displayEmployee.xhtml

Search for an employee

Please enter the employee id and password

Employee Id:

Password:



localhost:8080/ED-Secure-war/faces/user/displayEmployee.xhtml

Details of Employee 00003

Name:	Ceci
Phone:	3456789012
Address:	3 Mary Street, Hawthorn
Email:	ceci@secure.com.au
User Group:	ED-APP-USERS
Bank Account No:	210-987654-3
Salary:	75000.0
Active:	true

Click [here](#) to return to the Main Menu

Another Approach

- **Use jdbcRealm: choose employee id and password to access the Employee Management System.**
 - For 7.1P: you can use plain text password by defining “NONE” for plain password in jdbc Realm
 - For 7.2C (credit task): you should use “SHA-256” for encrypted password.

See jdbcRealm set up under Lab 8

User (employee) main menu

- Then use empld (00003) to get the record in the database.
- For the user section, you could display record for empld (e.g. 00003) or update it.
- For update password, you may ask only for new password.
 - ☐ You may not need to check the password