

# New Web Application



## Steps

1. Choose Project
- 2. Name and Location**
3. Server and Settings
4. Frameworks

## Name and Location

Project Name:

Project Location:

Project Folder:

Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

# New Web Application



## Steps

1. Choose Project
2. Name and Location
- 3. Server and Settings**
4. Frameworks

## Server and Settings

Add to Enterprise Application:

Server:

Java EE Version:

Context Path:

Steps

- 1. Choose Project
- 2. Name and Location
- 3. Server and Settings
- 4. Frameworks

1

Frameworks

Select the frameworks you want to use in your web application.

- Spring Web MVC
- JavaServer Faces
- Struts 1.3.10
- Hibernate 3.6.10

JavaServer Faces Configuration

Libraries Configuration Components

Server Library: JSF 2.2

Registered Libraries: JSF 2.2

Create New Library

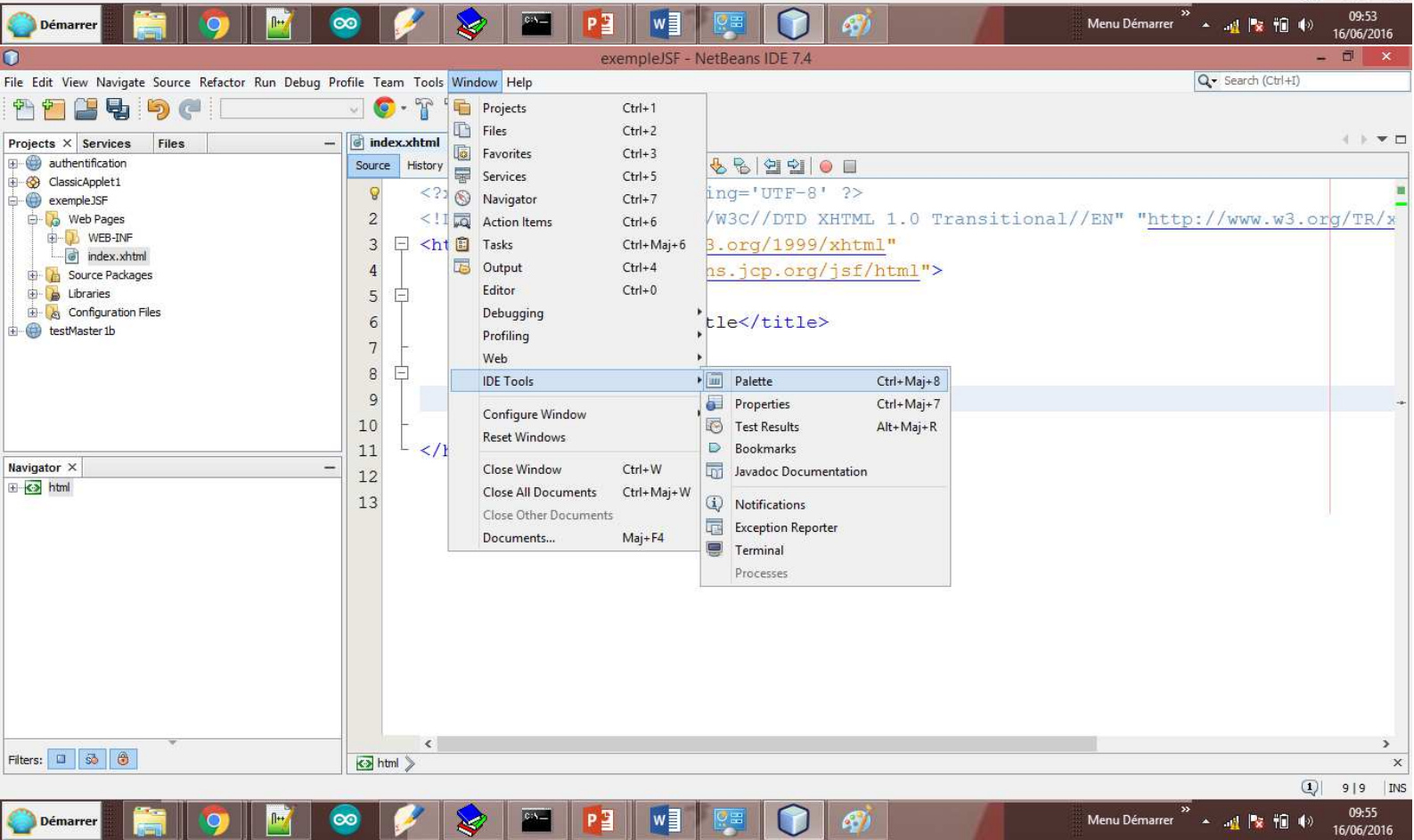
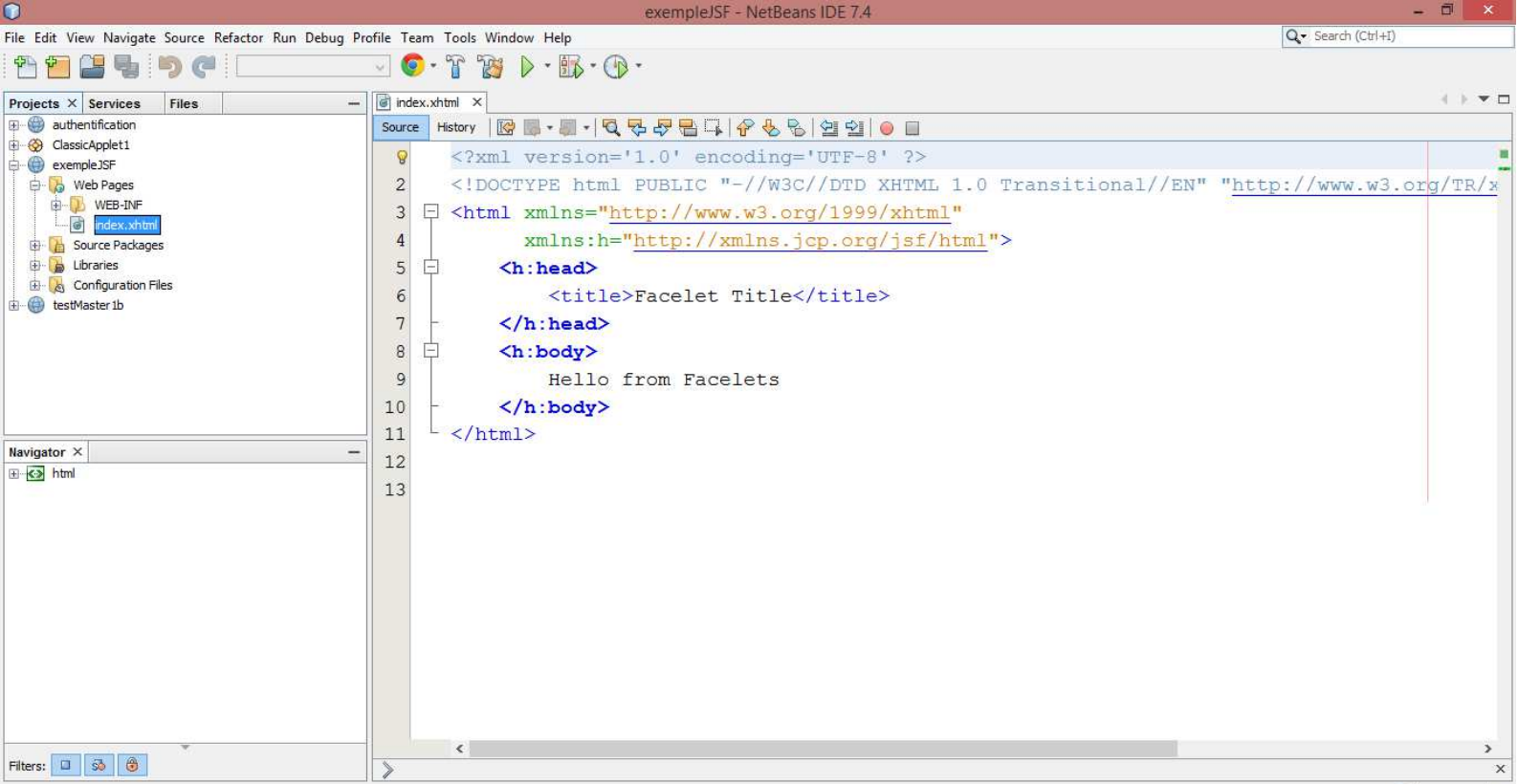
JSF Folder or JAR:  Browse...

Library Name:

2

- < Back
- Next >
- Finish
- Cancel
- Help

The screenshot shows the NetBeans IDE interface. On the left, the 'Projects' pane displays a tree view of the project structure. The 'exempleJSF' project is expanded, showing 'Web Pages' and 'WEB-INF' folders. The 'index.xhtml' file is selected and highlighted, with a handwritten '1' next to it. Below the 'Projects' pane is the 'Navigator' pane, which shows the 'html' file. The main editor area is currently empty. The bottom of the screen shows the Windows taskbar with various application icons and the system tray displaying the time as 09:53 and the date as 16/06/2016.



exempleJSF - NetBeans IDE 7.4

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Services Files

authentication  
ClassicApplet1  
exempleJSF  
Web Pages  
WEB-INF  
index.xhtml  
Source Packages  
Libraries  
Configuration Files  
testMaster1b

index.xhtml

```

1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional...
3 <html xmlns="http://www.w3.org/1999/xhtml"
4   xmlns:h="http://xmlns.jcp.org/jsf/html">
5   <h:head>
6     <title>Facelet Title</title>
7   </h:head>
8   <h:body>
9
10
11 </h:body>
12 </html>
13

```

Palette

- HTML
  - Table
  - Image
  - Link
  - Meta data
- HTML Forms
- JSF
  - Metadata
  - JSF Form
  - JSF Form From Entity
  - JSF Data Table
  - JSF Data Table From Entity

Navigator

- html

Filters

9 | 9 INS

Démarrer

Menu Démarrer

09:55  
16/06/2016

exempleJSF - NetBeans IDE 7.4

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Services Files

authentication  
ClassicApplet1  
exempleJSF  
Web Pages  
WEB-INF  
index.xhtml  
Source Packages  
Libraries  
Configuration Files  
testMaster1b

index.xhtml

```

1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional...
3 <html xmlns="http://www.w3.org/1999/xhtml"
4   xmlns:h="http://xmlns.jcp.org/jsf/html"
5   xmlns:f="http://java.sun.com/jsf/html">
6   <h:head>
7     <title>
8   </h:head>
9   <h:body>
10     <f:view>
11       <h:ou
12
13     </h:ou
14     </f:vie
15
16   </h:body>
17 </html>
18
19

```

Library: http://xmlns.jcp.org/jsf/html,  
http://java.sun.com/jsf/html (Html Basic)

**outputLabel**

Renders an HTML "label" element. Render the current value of the component as label text if it is specified. If a "for" attribute is specified, find the component specified by the value of the "for" attribute, and render its client id as the value of the "for" attribute. If

<h:outputFormat>	Html Basic
<h:outputLabel>	Html Basic
<h:outputLink>	Html Basic
<h:outputScript>	Html Basic
<h:outputStylesheet>	Html Basic
<h:outputText>	Html Basic

Navigator

- html
  - title
  - output

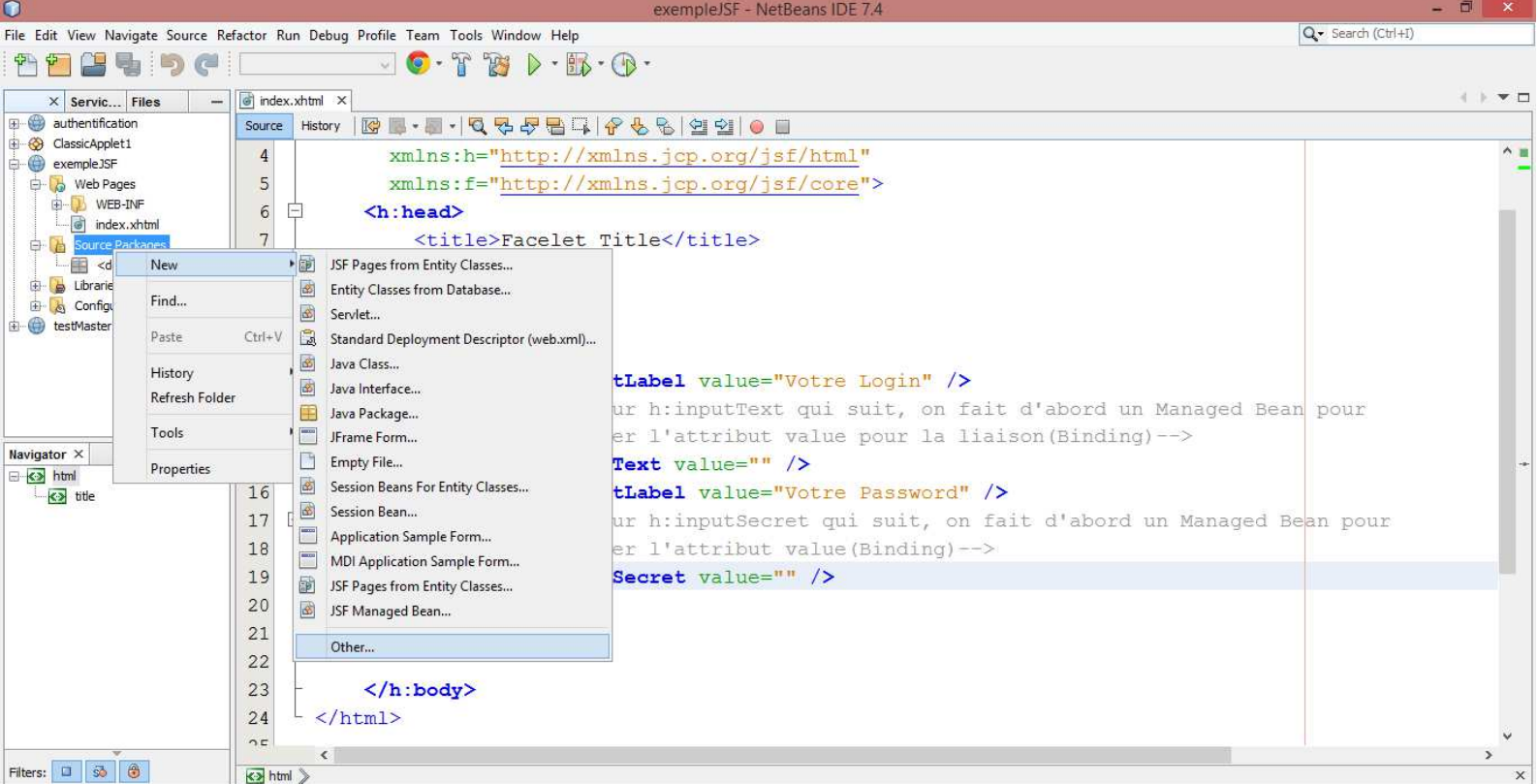
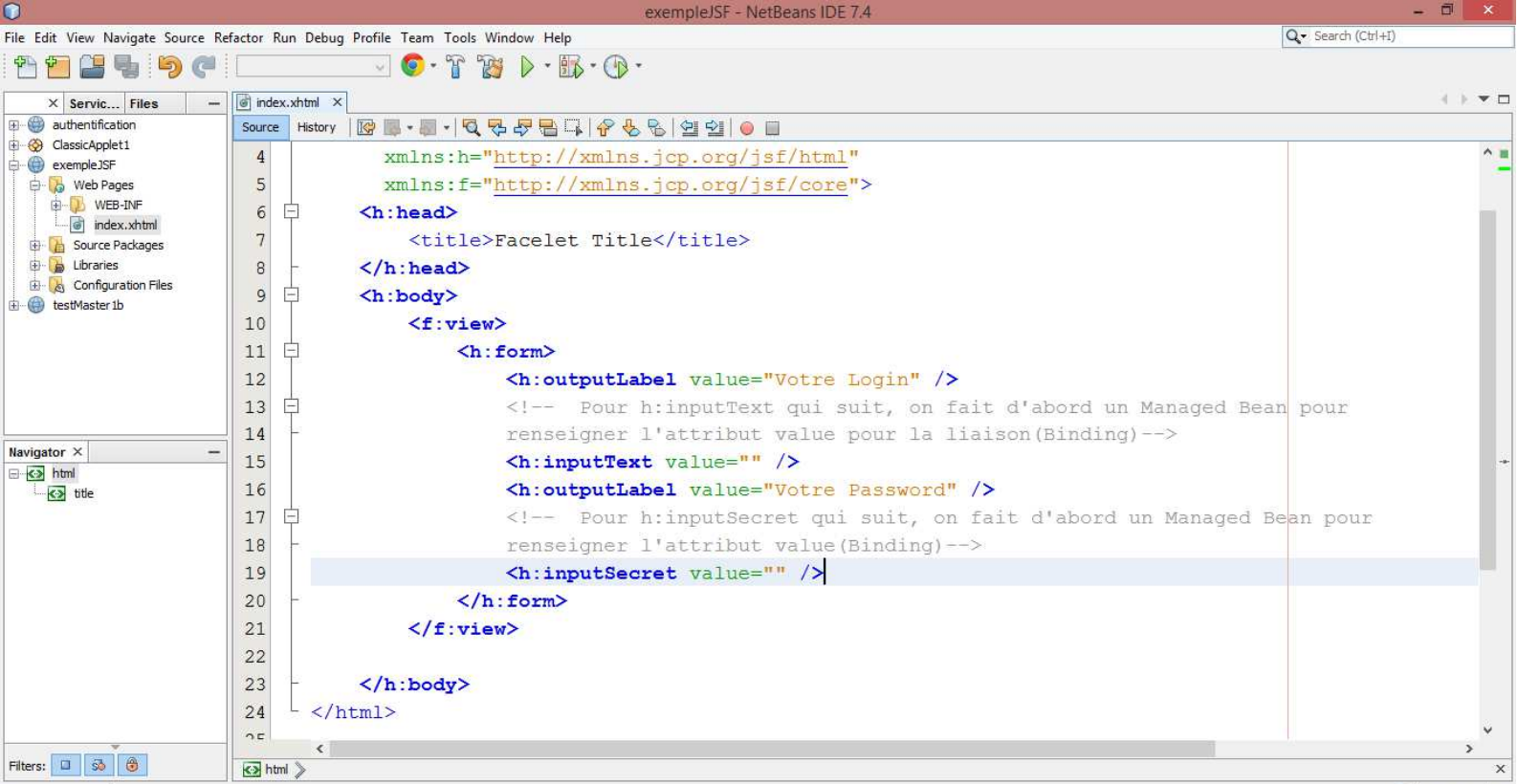
Filters

12 | 22 INS

Démarrer

Menu Démarrer

09:56  
16/06/2016



**Steps**

1. Choose File Type
2. ...

**Choose File Type**Project: Filter: 

## Categories:

- Web
- HTML5
- JavaServer Faces
- Bean Validation
- Struts
- Spring Framework
- Enterprise JavaBeans
- Contexts and Dependency Inj
- Java

## File Types:

- JSF Page
- JSF Managed Bean**
- JSF Faces Configuration
- JSF Composite Component
- JSF Pages from Entity Classes
- JSF Resource Library Contract
- JSF Faces Component
- Facelets Template
- Facelets Template Client

## Description:

Creates a new managed bean class.

&lt; Back

Next &gt;

Finish

Cancel

Help

## New JSF Managed Bean

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**Class Name: Project: Location: Package: Created File:  Add data to configuration fileConfiguration File: Name: Scope: 

&lt; Back

Next &gt;

Finish

Cancel

Help

The screenshot shows the NetBeans IDE interface with the source code of `UserManagedBean.java` open. The code is as follows:

```

7 package com.tdsi;
8
9 import javax.faces.bean.ManagedBean;
10 import javax.faces.bean.SessionScoped;
11
12 /**
13  *
14  * Annotations from the package javax.faces.bean will be deprecated in the next JSF version. CDI ones are recommended instead.
15  * (Alt-Enter shows hints)
16  */
17 @ManagedBean
18 @SessionScoped
19 public class UserManageredBean {
20
21     /**
22      * Creates a new instance of UserManageredBean
23      */
24     public UserManageredBean() {
25     }
26 }
27

```

A tooltip is displayed over the `@ManagedBean` annotation, containing the text: "Annotations from the package javax.faces.bean will be deprecated in the next JSF version. CDI ones are recommended instead. (Alt-Enter shows hints)".

The Windows taskbar is visible at the bottom of the screen, showing the Start button, several application icons, and the system tray with the date 16/06/2016 and time 10:06.

The screenshot shows the NetBeans IDE interface with the source code of `UserManagedBean.java` open. The code is as follows:

```

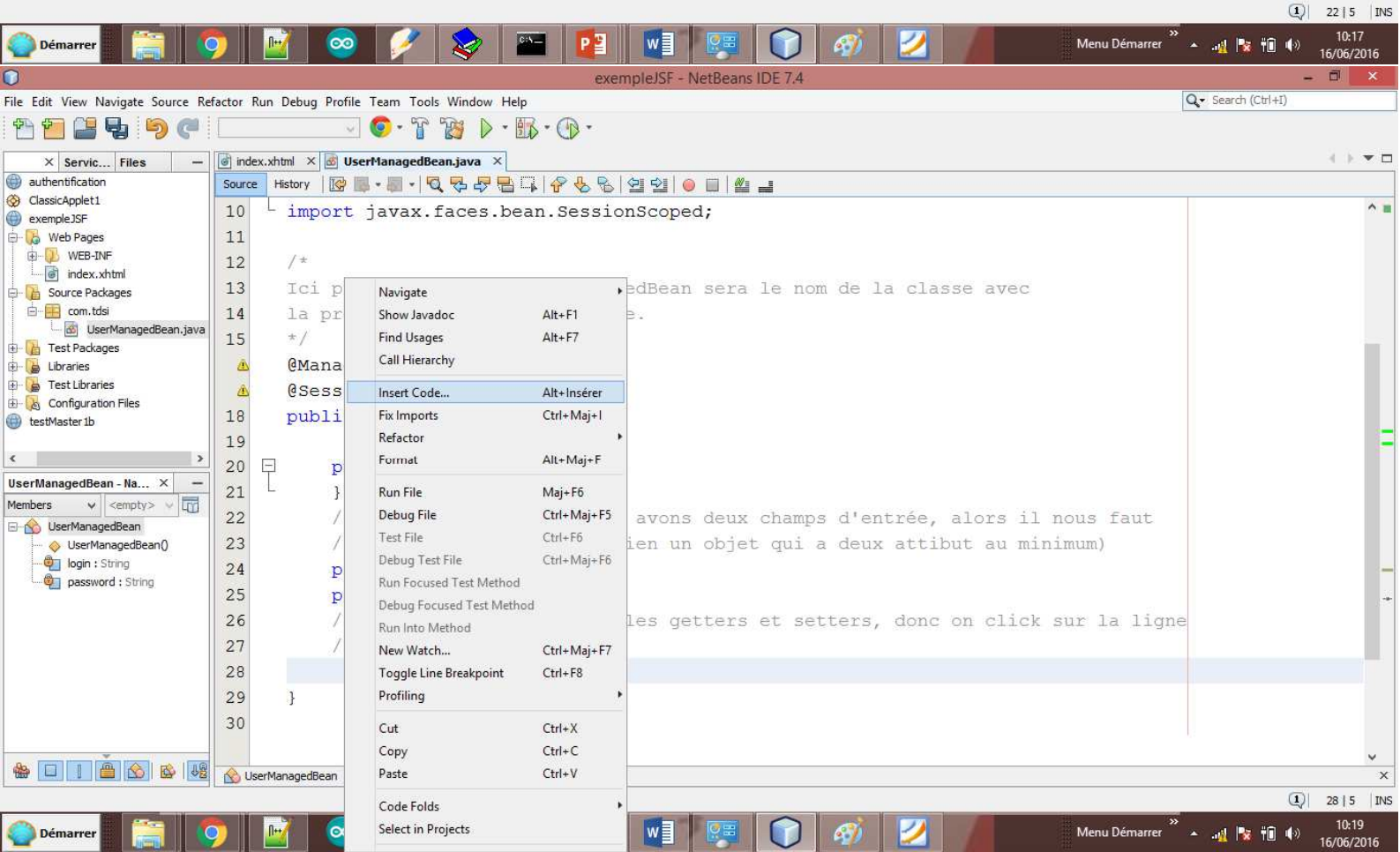
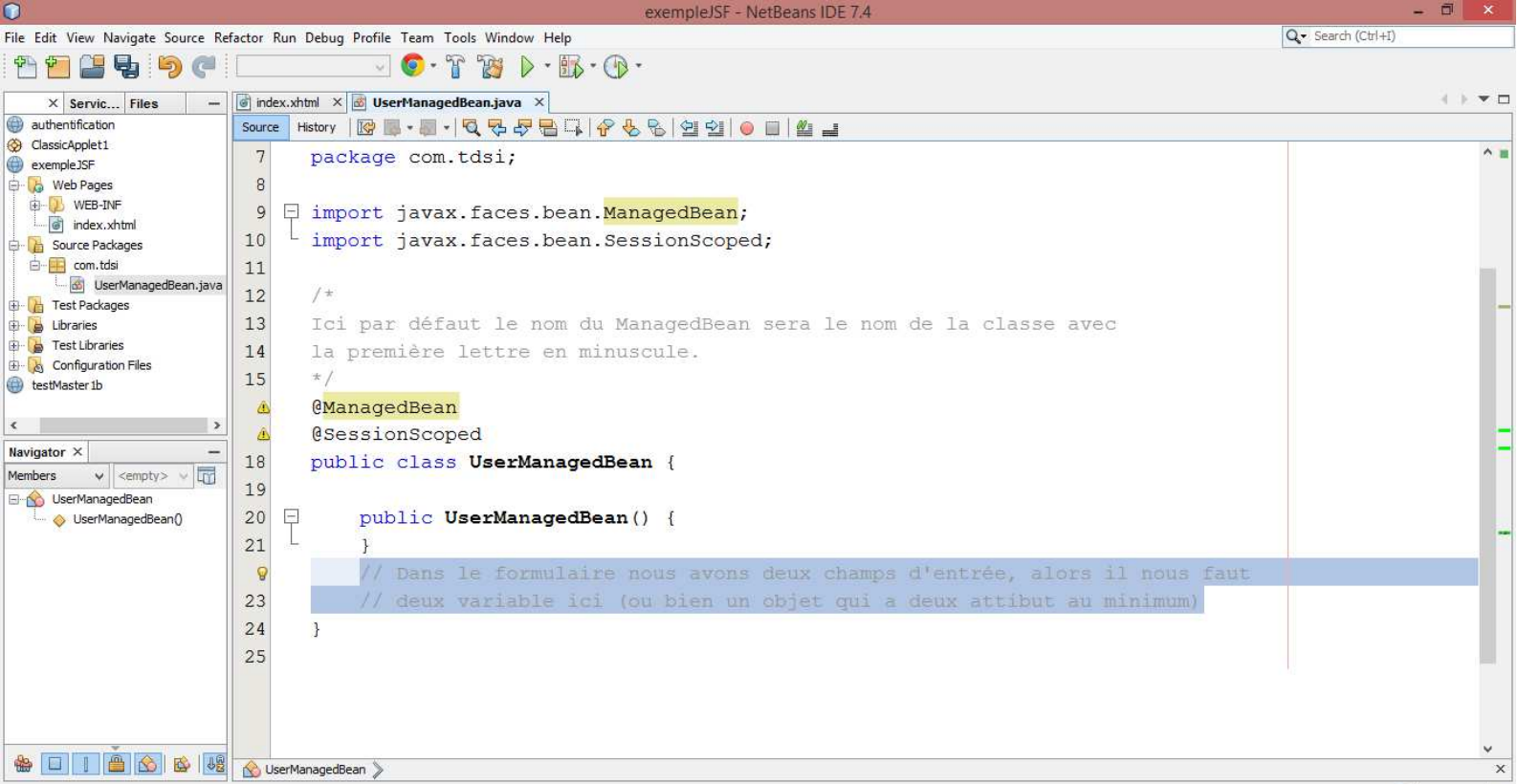
7 package com.tdsi;
8
9 import javax.faces.bean.ManagedBean;
10 import javax.faces.bean.SessionScoped;
11
12 /**
13  * Ici par défaut le nom du ManagedBean sera le nom de la classe avec
14  * la première lettre en minuscule.
15  */
16 @ManagedBean
17 @SessionScoped
18 public class UserManageredBean {
19
20     /**
21      * Creates a new instance of UserManageredBean
22      */
23     public UserManageredBean() {
24     }
25 }
26 }
27

```

The `@ManagedBean` annotation is highlighted in yellow.

The Windows taskbar is visible at the bottom of the screen, showing the Start button, several application icons, and the system tray with the date 16/06/2016 and time 10:14.





File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

index.xhtml x UserManagerBean.java x

Source History

```

10 import javax.faces.bean.SessionScoped;
11
12 /*
13 Ici par défaut le nom du ManagedBean sera le nom de la classe avec
14 la première lettre en minuscule.
15 */
16 @ManagedBean
17 @SessionScoped
18 public class UserManagerBean {
19
20     public UserManagerBean() {
21
22     }
23
24     // Dans le formulaire nous avons deux champs d'entrée, alors il nous faut
25     // deux variables (login et password) (au minimum un attribut au minimum)
26     private String login;
27     private String password;
28
29     // il faut obligatoirement les getters et setters, donc on click sur la ligne
30     // suivante
31 }
    
```

Members

- UserManagedBean
  - UserManagedBean()
  - login : String
  - password : String

UserManagedBean - Na... x

Members

- UserManagedBean
  - UserManagedBean()
  - login : String
  - password : String

UserManagedBean

- Generate
- Constructor...
- Logger...
- Getter...
- Setter...
- Getter and Setter...
- equals() and hashCode()...
- toString()...
- Delegate Method...
- Override Method...
- Add Property...
- Call Enterprise Bean...
- Use Database...
- Send JMS Message...
- Send E-mail...
- Call Web Service Operation...
- Generate REST Client...

exempleJSF - NetBeans IDE 7.4

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

index.xhtml x UserManagerBean.java x

Source History

```

10 import javax.faces
11
12 /*
13 Ici par défaut le
14 la première lettre
15 */
16 @ManagedBean
17 @SessionScoped
18 public class UserM
19
20 public UserMa
21 }
22
23 // Dans le fo
24 // deux varia
25 private Strin
26 private Strin
27 // il faut obl
28 // suivante
29 }
30
    
```

Members

- UserManagedBean
  - UserManagedBean()
  - login : String
  - password : String

UserManagedBean - Na... x

Members

- UserManagedBean
  - UserManagedBean()
  - login : String
  - password : String

UserManagedBean

Generate Getters and Setters

Select fields to generate getters and setters for:

- UserManagerBean
  - login : String
  - password : String

Encapsulate Fields

Generate Cancel

```

4      xmlns:h="http://xmlns.jcp.org/
5      xmlns:f="http://xmlns.jcp.org/
6      <h:head>
7          <title>Facelet Title</title>
8      </h:head>
9      <h:body>
10         <f:view>
11             <h:form>
12                 <h:outputLabel value="
13                 <!-- Pour h:inputText qui suit, on fait d'abord un Managed Bean pour
14                 renseigner l'attribut value (Binding)-->
15                 <!-- Nous sommes de retour ici, car on connait le nom du Managedbean -->
16                 <h:inputText value="#{use}" />
17                 <h:outputLabel value="userManagedBean UserManagedBean"
18                 <!-- Pour h:inputSecret qui suit, on fait d'abord un Managed Bean pour
19                 renseigner l'attribut value (Binding)-->
20                 <h:inputSecret value="" />
21             </h:form>
22         </f:view>
23     </h:body>
24 </html>

```

```

public class UserManagedBean extends Object
{
    Javadoc not found. Either Javadoc
    documentation for this item does not exist
    or you have not added specified Javadoc in
    the Java Platform Manager or the Library
    Manager.
}

```

```

4      xmlns:h="http://xmlns.jcp.org/jsf/html"
5      xmlns:f="http://xmlns.jcp.org/jsf/core">
6      <h:head>
7          <title>Facelet Title</title>
8      </h:head>
9      <h:body>
10         <f:view>
11             <h:form>
12                 <h:outputLabel value="Votre Login" />
13                 <!-- Pour h:inputText qui suit, on fait d'abord un Managed Bean pour
14                 renseigner l'attribut value pour la liaison(Binding)-->
15                 <!-- Nous sommes de retour ici, car on connait le nom du Managedbean -->
16                 <h:inputText value="#{userManagedBean.login}" />
17                 <h:outputLabel value="Votre Password" login String
18                 <!-- Pour h:inputSecret qui suit, on fait d'abord un Managed Bean pour
19                 renseigner l'attribut value (Binding)-->
20                 <h:inputSecret value="" />
21             </h:form>
22         </f:view>
23     </h:body>
24 </html>

```

```

public class UserManagedBean extends Object
{
    Javadoc not found. Either Javadoc
    documentation for this item does not exist
    or you have not added specified Javadoc in
    the Java Platform Manager or the Library
    Manager.
}

```

```

7      <title>Facelet Title</title>
8      </h:head>
9      <h:body>
10     <f:view>
11     <h:form>
12     <h:outputLabel value="Votre Login" />
13     <!-- Pour h:inputText qui suit, on fait d'abord un Managed Bean pour
14     renseigner l'attribut value pour la liaison(Binding)-->
15     <!-- Nous sommes de retour ici, car on connaît le nom du Managedbean -->
16     <h:inputText value="#{userManagedBean.login}" />
17     <h:outputLabel value="Votre Password" />
18     <!-- Pour h:inputSecret qui suit, on fait d'abord un Managed Bean pour
19     renseigner l'attribut value(Binding)-->
20     <h:inputSecret value="#{userManagedBean.password}" />
21     <!-- Il reste à ajouter le bouton de validation ici, et fait la
22     méthode d'authentification dans le ManagedBean-->
23     <h:commandButton value="Cliquer Ici" action="#" />
24     </h:form>
25     </f:view>
26
27     </h:body>
28 </html>

```

```

31    }
32
33    public void setLogin(String login) {
34        this.login = login;
35    }
36
37    public String getPassword() {
38        return password;
39    }
40
41    public void setPassword(String password) {
42        this.password = password;
43    }
44    //création de la méthode, la valeur retournée permet de faire la redirection
45    public String authenticier(){
46        if(login.equals("ouzu") && password.equals("tdsinekhna"))
47            return "accueil.xhtml";
48        else return "index.xhtml";
49    }
50    }
51

```

```

7      <title>Facelet Title</title>
8      </h:head>
9      <h:body>
10     <f:view>
11         <h:form>
12             <h:outputLabel value="Votre Login" />
13             <!-- Pour h:inputText qui suit, on fait d'abord un Managed Bean pour
14                  renseigner l'attribut value pour la liaison(Binding)-->
15             <!-- Nous sommes de retour ici, car on connait le nom du Managedbean -->
16             <h:inputText value="#{userManagedBean.login}" />
17             <h:outputLabel value="Votre Password" />
18             <!-- Pour h:inputSecret qui suit, on fait d'abord un Managed Bean pour
19                  renseigner l'attribut value(Binding)-->
20             <h:inputSecret value="#{userManagedBean.password}" />
21             <!-- Il reste à ajouter le bouton de validation ici, et fait la
22                  méthode d'authentification dans le ManagedBean-->
23             <!-- De retour car la méthode est créée, on s'occupe de "action"-->
24             <h:commandButton value="Cliquer Ici" action="#{userManagedBean.authentifier()}" />
25
26         </h:form>
27     </f:view>

```

index.xhtml saved.

25 | 17 | INS

```

7      <title>Facelet Title</title>
8      </h:head>
9      <h:body>
10     <f:view>
11         <h:form>
12             <h:outputLabel value="Votre Login" />
13             <!-- Pour h:inputText qui suit, on fait d'abord un Managed Bean pour
14                  renseigner l'attribut value pour la liaison(Binding)-->
15             <!-- Nous sommes de retour ici, car on connait le nom du Managedbean -->
16             <h:inputText value="#{userManagedBean.login}" />
17             <h:outputLabel value="Votre Password" />
18             <!-- Pour h:inputSecret qui suit, on fait d'abord un Managed Bean pour
19                  renseigner l'attribut value(Binding)-->
20             <h:inputSecret value="#{userManagedBean.password}" />
21             <!-- Il reste à ajouter le bouton de validation ici, et fait la
22                  méthode d'authentification dans le ManagedBean-->
23             <!-- De retour car la méthode est créée, on s'occupe de "action"-->
24             <h:commandButton value="Cliquer Ici" action="#{userManagedBean.authentifier()}" />
25             <!-- Il manque juste la création de la page accueil.xhtml-->
26         </h:form>
27     </f:view>

```

Convert to Composite Component Surround With Tag Remove Surrounding Tag

25 | 17 | INS

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

index.xhtml x UserManagedBean.java x

Source History

7 <title>Facelet Title</title>

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

Filters: [ ] [ ] [ ]

html

authenticat...  
ClassicApplet1  
exempleJSF  
Web Pages  
WEB-INF  
index.xhtml  
Source Packages  
com.tdsi  
UserManagedBean.java  
Test Packages  
Libraries  
Test Libraries  
Configuration Files  
testMaster Ib

Navigation: html

New JSF Managed Bean...  
JSF Pages from Entity Classes...  
Entity Classes from Database...  
Servlet...  
Standard Deployment Descriptor (web.xml)...  
Java Class...  
Java Interface...  
Java Package...  
JFrame Form...  
Empty File...  
Session Beans For Entity Classes...  
Session Bean...  
Application Sample Form...  
MDI Application Sample Form...  
JSF Pages from Entity Classes...  
Other...

```

<h:inputText value="Votre Login" />
<h:inputText qui suit, on fait d'abord un Managed Bean pour
l'attribut value pour la liaison(Binding)-->
comme de retour ici, car on connait le nom du Managedbean -->
<h:inputText value="#{userManagedBean.login}" />
<h:inputText value="Votre Password" />
<h:inputSecret qui suit, on fait d'abord un Managed Bean pour
l'attribut value(Binding)-->
<h:inputSecret value="#{userManagedBean.password}" />
Il faut ajouter le bouton de validation ici, et fait la
l'authentification dans le ManagedBean-->
<!-- De retour car la méthode est créée, on s'occupe de "action"-->
<h:commandButton value="Cliquer Ici" action="#{userManagedBean.authentifier()}" />
<!-- Il manque juste la création de la page accueil.xhtml-->
</h:form>
</f:view>
    
```

Démarrer [Icons] Menu Démarrer 10:33 16/06/2016

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

index.xhtml x UserManagedBean.java x

Source History

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

Filters: [ ] [ ] [ ]

html

New JSF Page

Steps

1. Choose File Type
2. Name and Location

Name and Location

File Name: accueil

Project: exempleJSF

Location: Web Pages

Folder: [ ] Browse...

Created File: C:\Users\ouz\Documents\NetBeansProjects\exempleJSF\web\accueil.xhtml

Options:

Facelets

JSP File (Standard Syntax)  Create as a JSP Segment

Description: Facelets syntax

< Back Next > Finish Cancel Help

authenticat...  
ClassicApplet1  
exempleJSF  
Web Pages  
WEB-INF  
index.xhtml  
Source Packages  
com.tdsi  
UserManagedBean.java  
Test Packages  
Libraries  
Test Libraries  
Configuration Files  
testMaster Ib

Navigation: html

```

</h:form>
</f:view>
    
```

Démarrer [Icons] Menu Démarrer 10:34 16/06/2016

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

index.xhtml x UserManagedBean.java x accueil.xhtml x

Source History

```

1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4     xmlns:h="http://xmlns.jcp.org/jsf/html">
5   <h:head>
6     <title>Facelet Title</title>
7   </h:head>
8   <h:body>
9     <h:outputText value="Bonjour #{userManagedBean.login}" />
10  </h:body>
11 </html>
12
13

```

Filters: [ ] [ ] [ ]

exampleJSF deployed.

Démarrer 9 | 64 | INS 10:43 16/06/2016

exempleJSF - NetBeans IDE 7.4

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

index.xhtml x UserManagedBean.java x accueil.xhtml x

Source History

```

1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4     xmlns:h="http://xmlns.jcp.org/jsf/html">
5   <h:head>
6     <title>Facelet Title</title>
7   </h:head>
8   <h:body>
9     <h:outputText value="Bonjour #{userManagedBean.login}" />
10  </h:body>
11 </html>
12
13

```

Filters: [ ]

- New
- Build
- Clean and Build
- Clean
- Verify
- Generate Javadoc
- Run
- Deploy
- Debug
- Profile
- Test RESTful Web Services
- Test Alt+F6
- Open Required Projects
- Close
- Rename...
- Move...
- Copy...
- Delete Supprimer
- Find... Ctrl+F
- Inspect and Transform...
- Versioning
- History
- Properties

Démarrer 9 | 64 | INS 10:44 16/06/2016