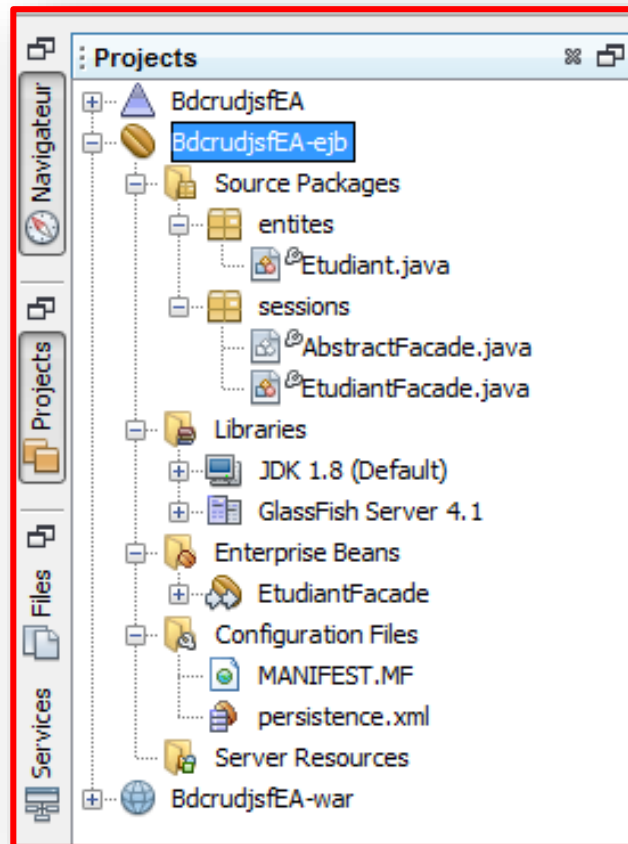
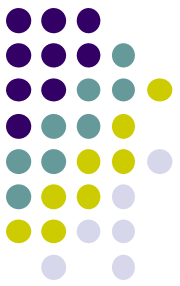
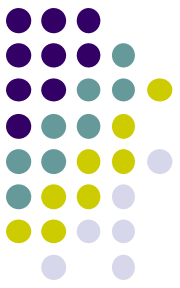




Exemple CRUD Complet JSF, JPA et EJB pour gestion table étudiant

Première étape: création de la base de données «bdcrudjsf»



- Création de la base de données

Bases de données

Créer une base de données ?

Ne pas oublier de spécifier l'Id comme un champ auto-incrémenté

- Création table «etudiant»

Nom de la table: Ajouter colonne(s)

Structure ?

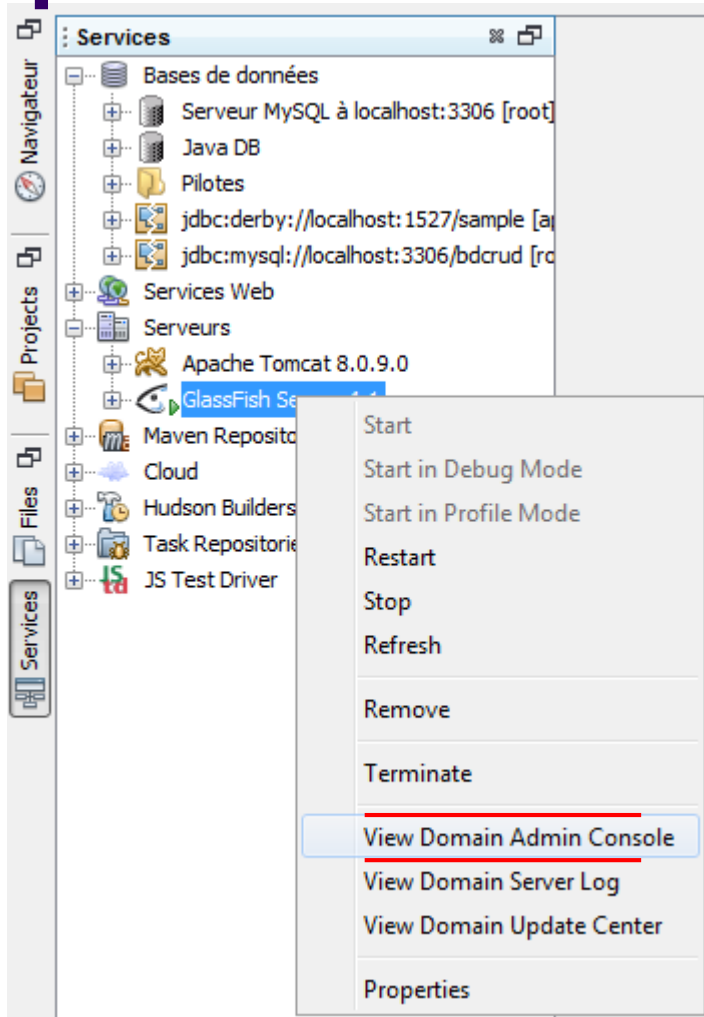
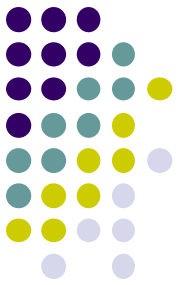
Nom	Type ?	Taille/Valeurs* ?	Défaut ?	Interclassement	Attributs	Null	Index	A_1 Commentaires
id	INT		Aucune			☐	PRIMARY	☒
nom	VARCHAR	50	Aucune			☐	---	☐
prenom	VARCHAR	50	Aucune			☐	---	☐
datenaiss	DATE		Aucune			☐	---	☐

Commentaires sur la table :

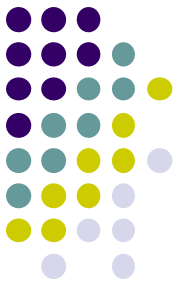
Moteur de stockage : Interclassement :

Définition de PARTITION : ?

Deuxième étape: création d'une pool de connexion et DataSource



Deuxième étape: création d'une pool de connexion(1)



GlassFish™ Server Open Source Edition

Common Tasks

- Domain
 - server (Admin Server)
 - Clusters
 - Standalone Instances
 - Nodes
 - Applications
 - Lifecycle Modules
 - Monitoring Data
 - Resources
 - Concurrent Resources
 - Connectors
 - JDBC
 - JDBC Resources
 - JDBC Connection Pools

New JDBC Connection Pool (Step 1 of 2)

Identify the general settings for the connection pool.

General Settings

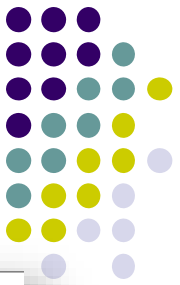
Pool Name: *

Resource Type: Must be specified if the datasource class implements more than 1 of the interface.

Database Driver Vendor: Select or enter a database driver vendor

Introspect: ☐ Enabled If enabled, data source or driver implementation class names will enable introspection.

Deuxième étape: création d'une pool de connexion(2)



New JDBC Connection Pool (Step 2 of 2)

Identify the general settings for the connection pool. Datasource Classname or Driver Classname must be specified for the connection pool.

General Settings

Pool Name: bdcrudjsfpool

Resource Type: javax.sql.DataSource

Database Driver Vendor: MySql

Datasource Classname:

Select or enter vendor-specific classname that implements the DataSource and/or XADataSource APIs

Driver Classname:



Select or enter vendor-specific classname that implements the java.sql.Driver interface.

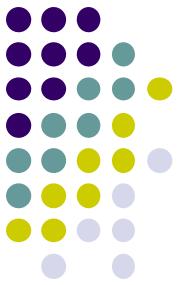
Ping: ☒ **Enabled**

When enabled, the pool is pinged during creation or reconfiguration to identify and warn of any erroneous values for its attributes



☐	URL	jdbc:mysql://localhost:3306/bdcrudjsf
☐	Url	jdbc:mysql://localhost:3306/bdcrudjsf
☐	ServerName	localhost
☐	DatabaseName	bdcrudjsf
☐	User	ufrset
☐	Password	ufrset
☐	driverClass	com.mysql.jdbc.Driver

Deuxième étape: création d'une source de données(3)



Common Tasks

- Domain
- server (Admin Server)
- Clusters
- Standalone Instances
- Nodes
- Applications
- Lifecycle Modules
- Monitoring Data
- Resources
 - Concurrent Resources
 - Connectors
 - JDBC
 - JDBC Resources
 - jdbc/__TimerPool
 - jdbc/__default
 - jdbc/bdcrudjsfPool
 - jdbc/sample
 - testcrudResource
 - JDBC Connection Pools
 - DerbyPool
 - SamplePool
 - __TimerPool
 - bdcrudjsfpool
 - connectionPool
 - testcrud

New JDBC Resource

Specify a unique JNDI name that identifies the JDBC resource you want to create.

JNDI Name: *

Pool Name: Use the JDBC Connection Pools page

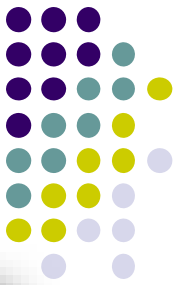
Description:

Status: ☒ Enabled

Additional Properties (0)

Select	Name
--------	------

Troisième étape: création d'un projet Enterprise Application



New Enterprise Application

Etapes

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

Name and Location

Project Name: BdcrudjsfEA

Project Location: C:\Users\Hpg7\Documents\NetBeansProjects **Browse...**

Project Folder: C:\Users\Hpg7\Documents\NetBeansProjects\BdcrudjsfEA

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: **Browse...**

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

< Précédent **Suivant >** Terminer Annuler Help

Troisième étape: création d'un projet Enterprise Application



New Enterprise Application

Etapes

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

Server and Settings

Server: GlassFish Server 4.1 Add...

Java EE Version: Java EE 7

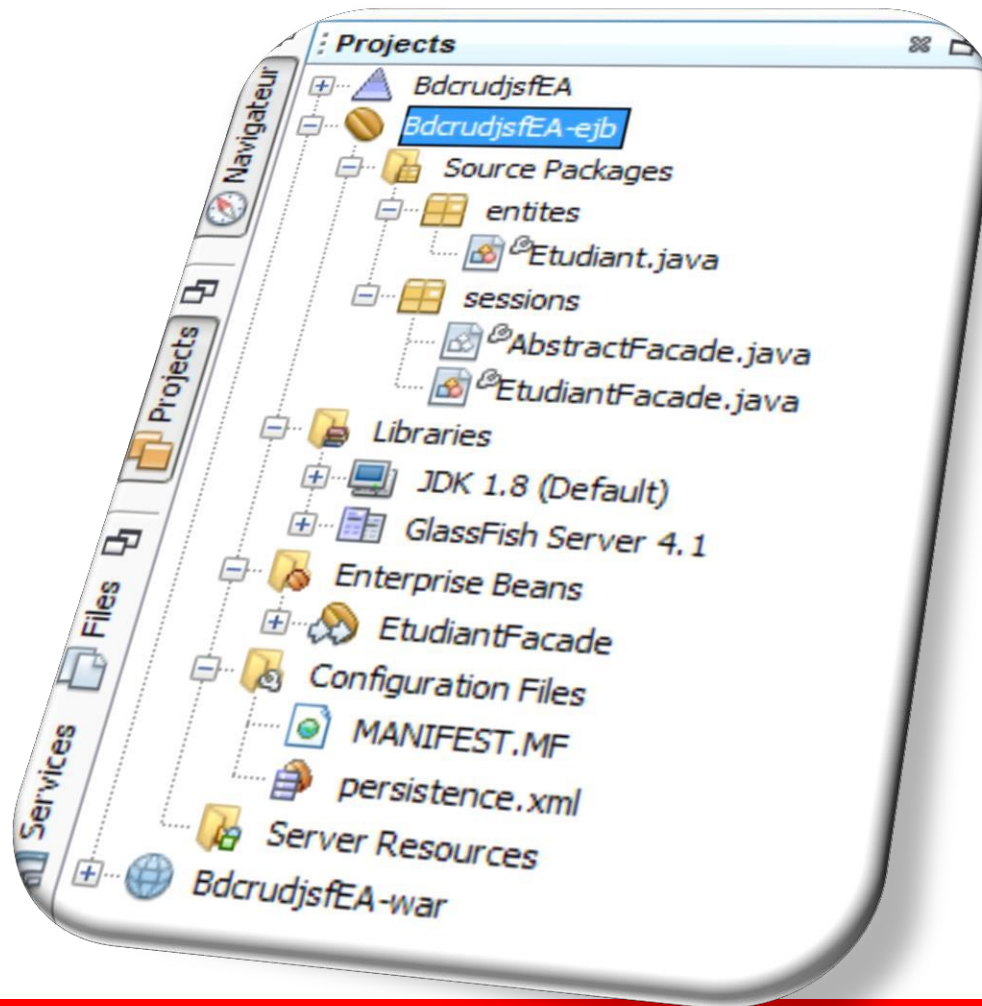
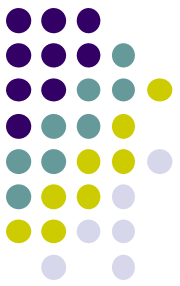
☒ Create EJB Module: BdcrudjsfEA-ejb

☒ Create Web Application Module: BdcrudjsfEA-war

< Précédent Suivant > **Terminer** Annuler Help

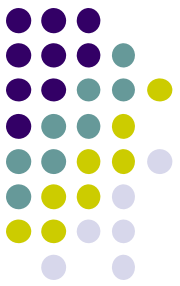
Projects

- BdcrudjsfEA
 - Java EE Modules
 - BdcrudjsfEA-war.war
 - BdcrudjsfEA-ejb.jar
 - Configuration Files
 - MANIFEST.MF
 - Server Resources
 - BdcrudjsfEA-ejb
 - Source Packages
 - <package par défaut>
 - Libraries
 - JDK 1.8 (Default)
 - GlassFish Server 4.1
 - Enterprise Beans
 - Configuration Files
 - MANIFEST.MF
 - Server Resources
 - BdcrudjsfEA-war
 - Web Pages
 - WEB-INF
 - index.html
 - Source Packages
 - <package par défaut>
 - Libraries
 - BdcrudjsfEA-ejb - dist/BdcrudjsfEA
 - JDK 1.8 (Default)
 - GlassFish Server 4.1
 - Configuration Files
 - MANIFEST.MF

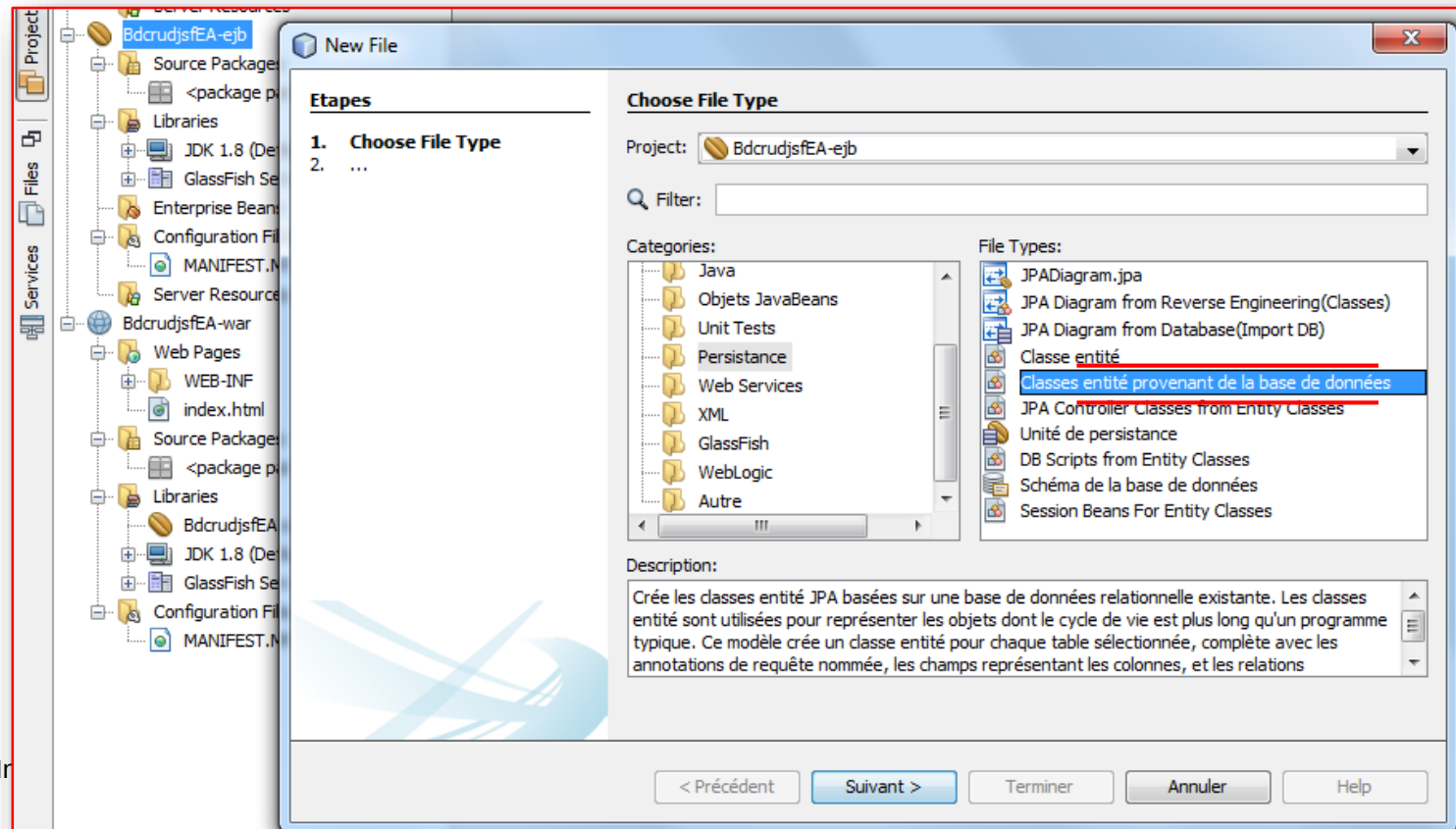


Créer de l'application dans le module EJB

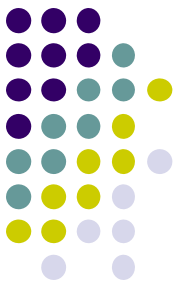
Quatrième étape: création des éléments métiers module EJB



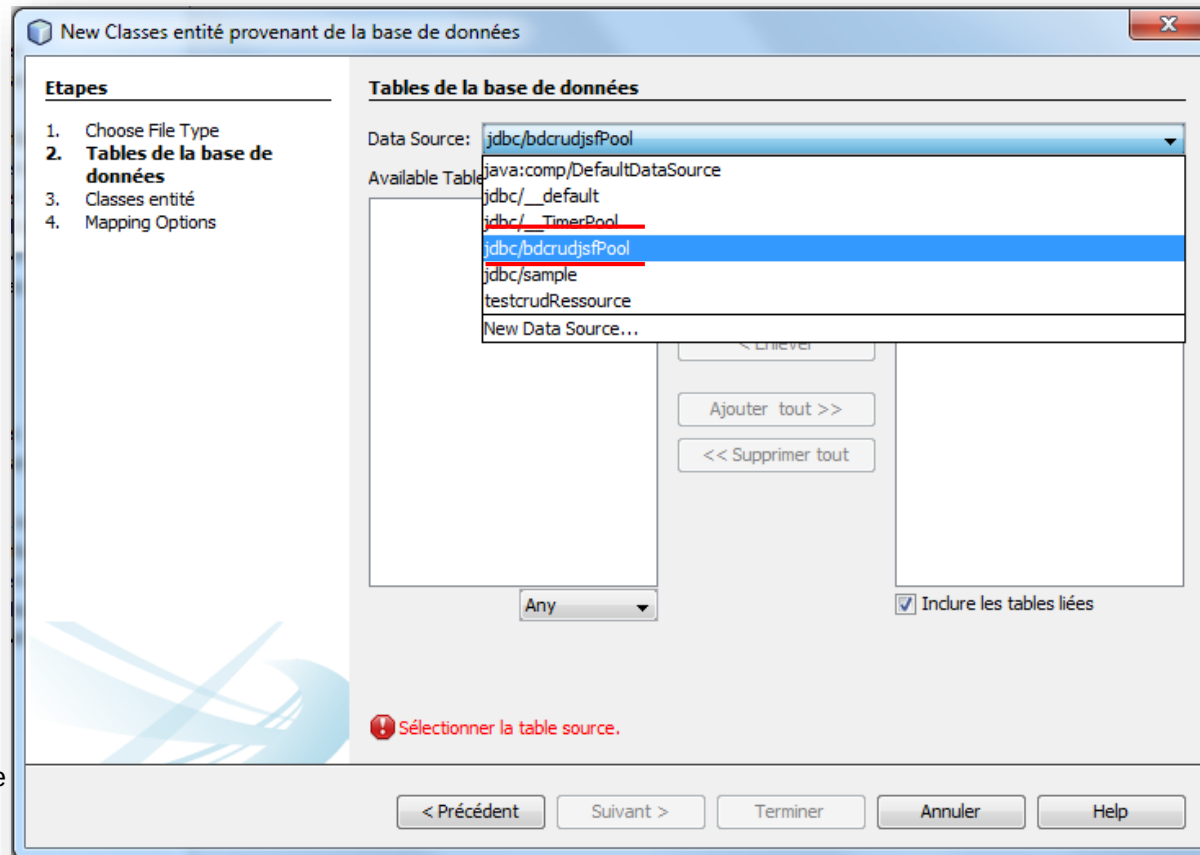
- Création d'une « entité bean » provenant de la base de données



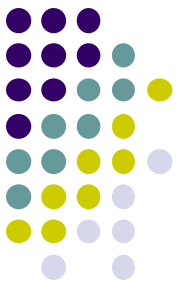
Quatrième étape: création des éléments métiers module EJB



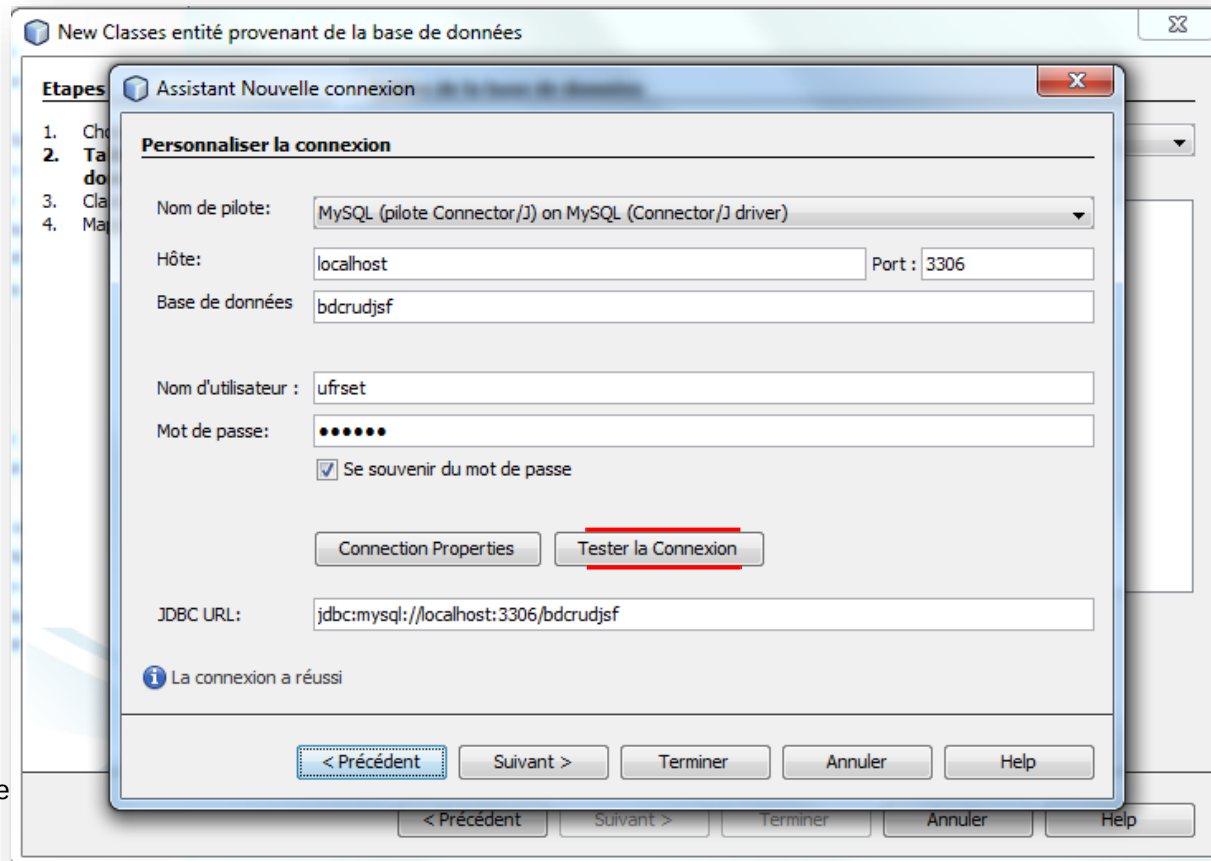
- Création d'une « entité bean » provenant de la base de données



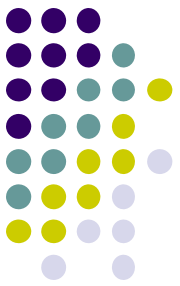
Quatrième étape: création des éléments métiers module EJB



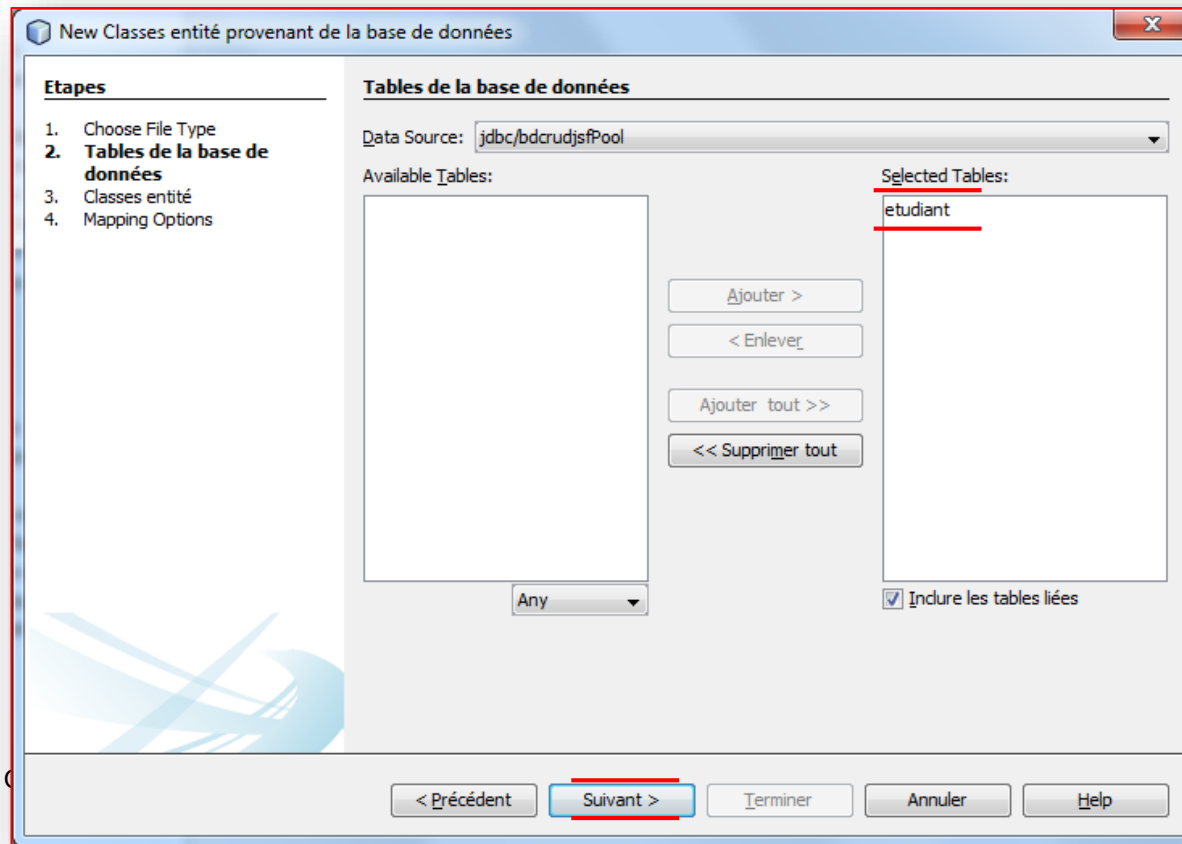
- Création d'une « entité bean » provenant de la base de données



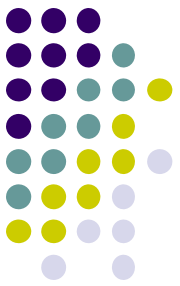
Quatrième étape: création des éléments métiers module EJB



- Création d'une « entité bean » provenant de la base de données



Quatrième étape: création des éléments métiers module EJB



- Création d'une « entité bean » provenant de la base de données

New Classes entité provenant de la base de données

Etapes

1. Choose File Type
2. Tables de la base de données
3. **Classes entité**
4. Mapping Options

Classes entité

Specify the names and the location of the entity classes.

Class Names:

Table de la base de données	Nom de la classe	Generation Type
etudiant	Etudiant	Nouveau

Projet: BdcrujsfEA-ejb

Emplacement: Source Packages

Package : entites

☒ Générer des annotations de requête nommée pour les champs persistants

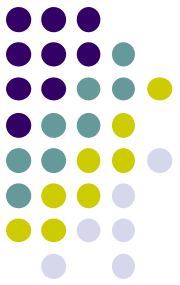
☒ Generate JAXB Annotations

☐ Generate MappedSuperclasses instead of Entities

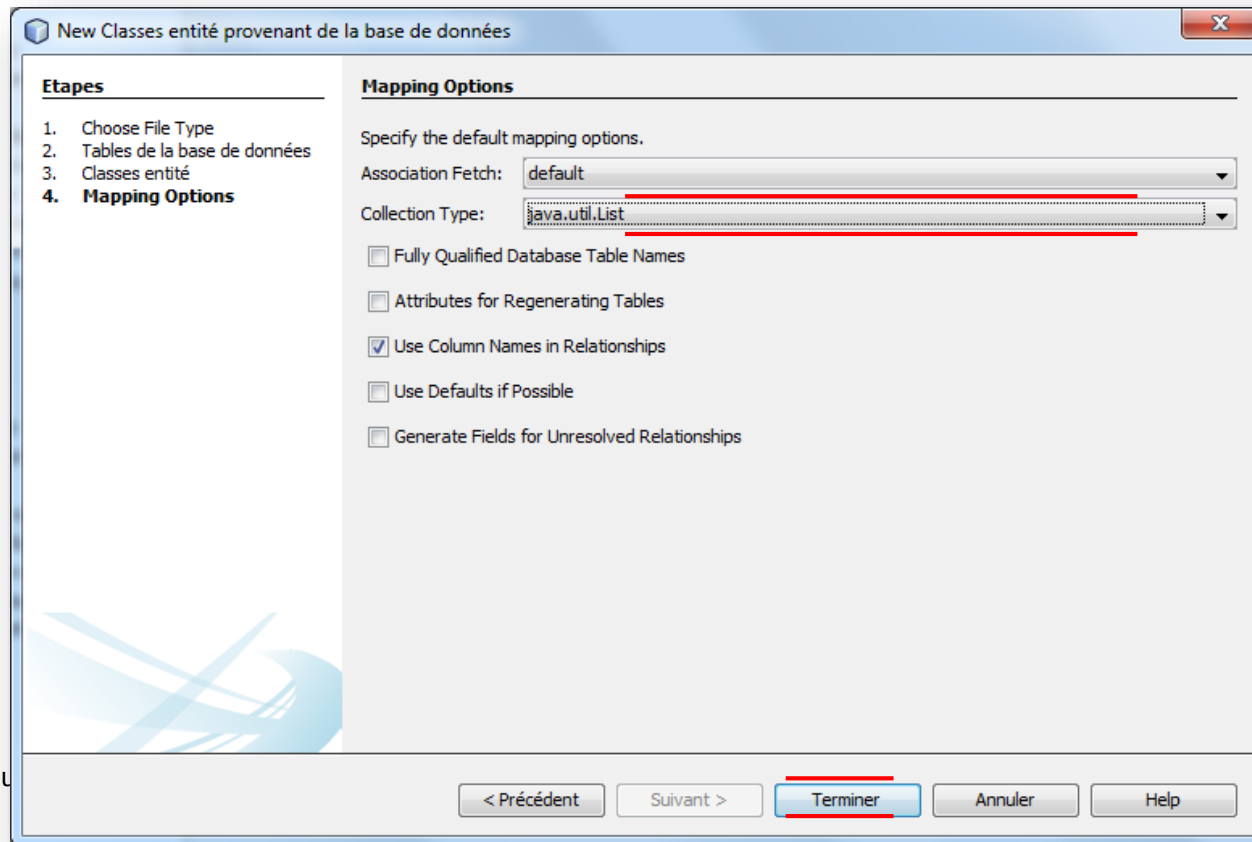
☒ Create Persistence Unit

< Précédent Suivant > Terminer Annuler Help

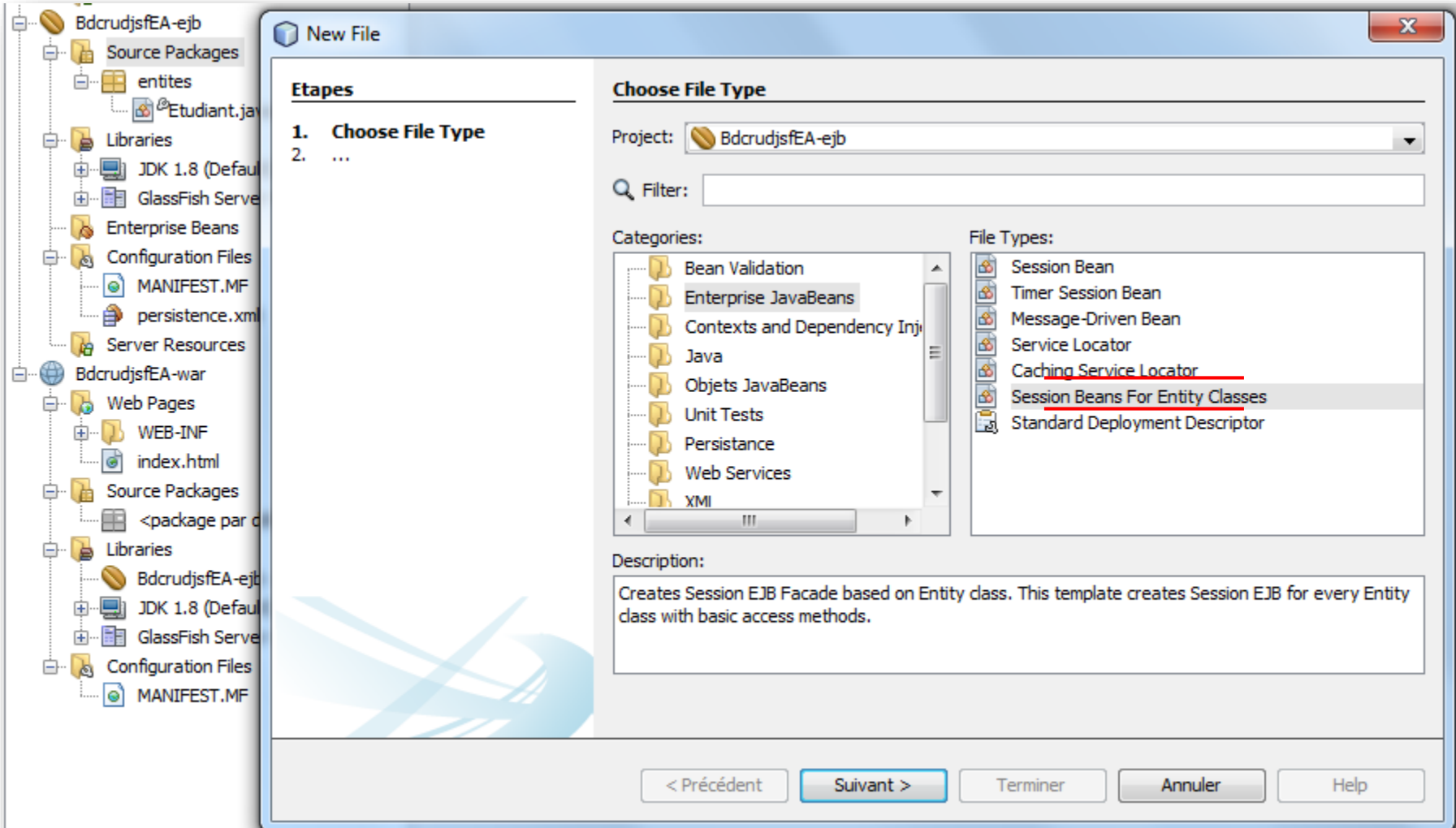
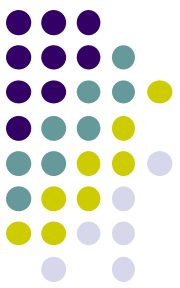
Quatrième étape: création des éléments métiers module EJB



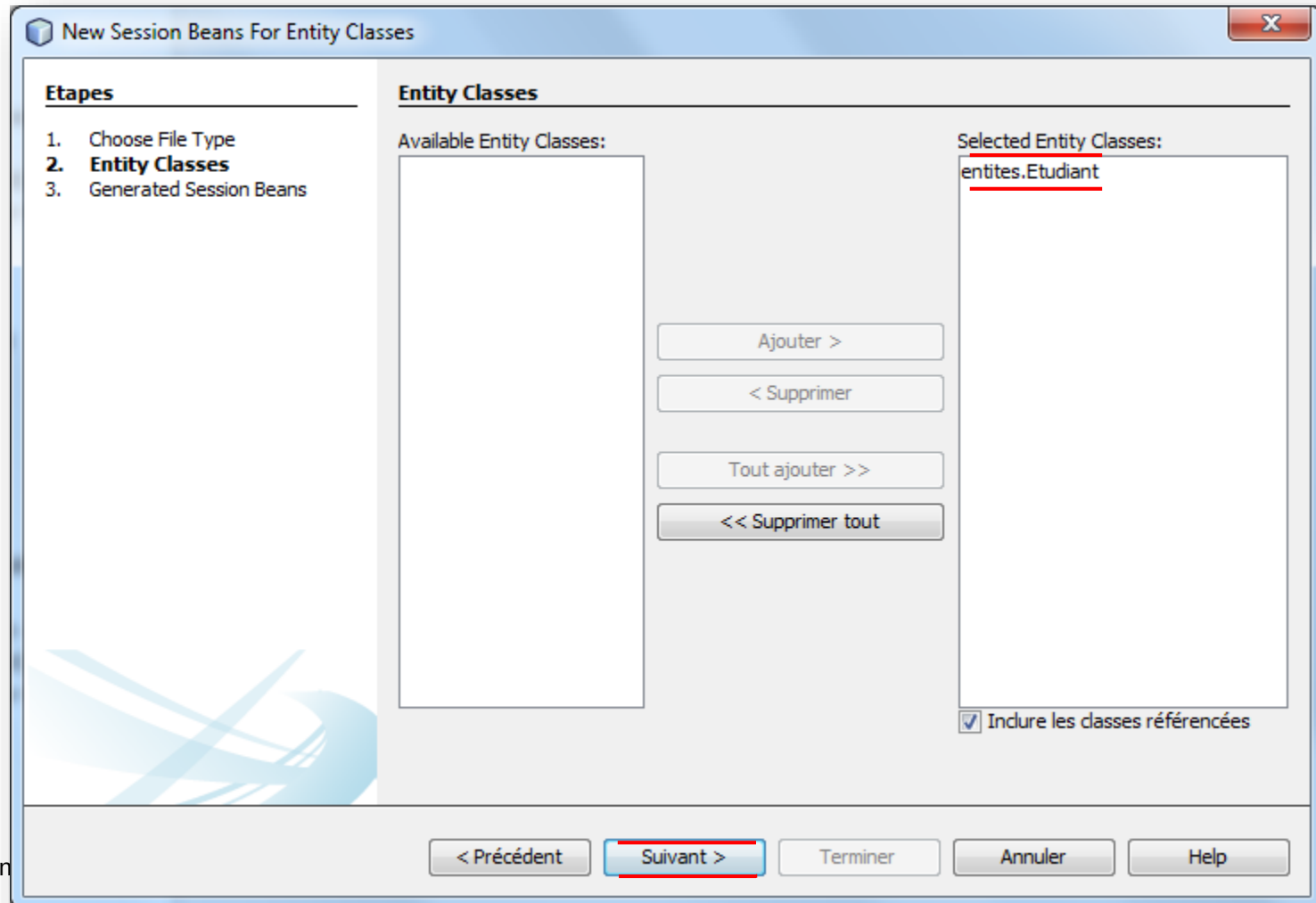
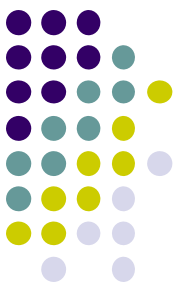
- Création d'une « entité bean » provenant de la base de données



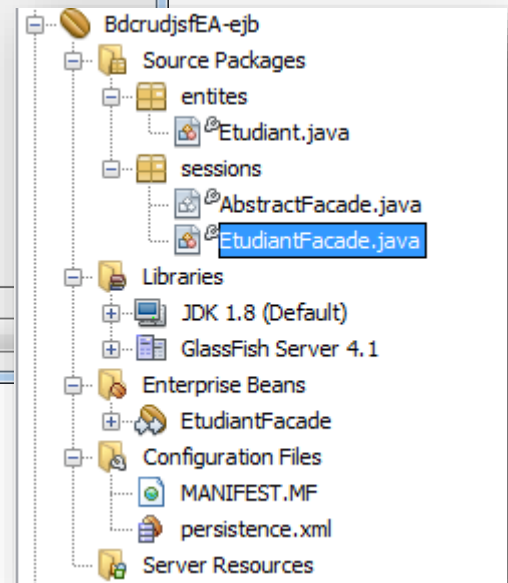
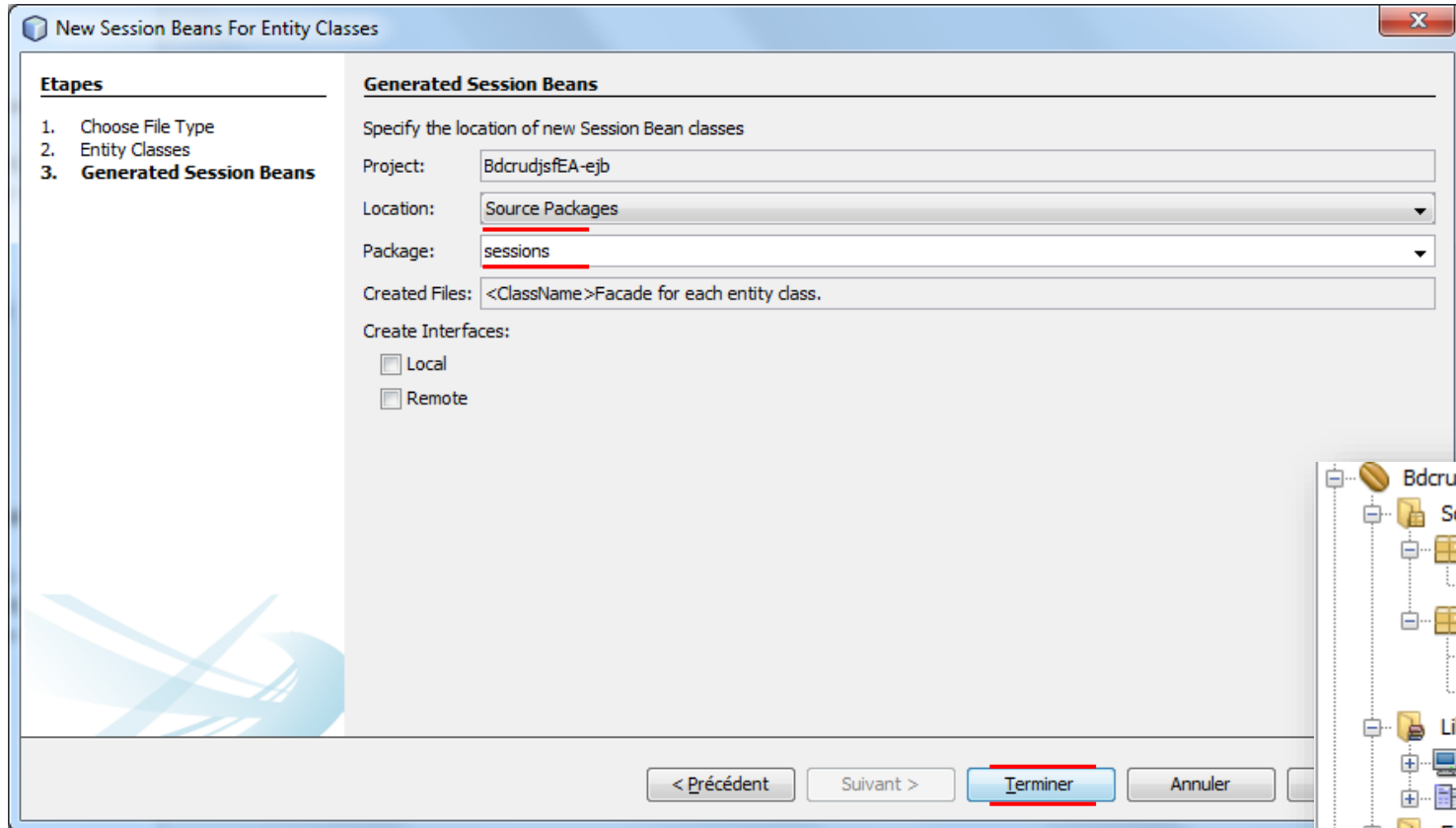
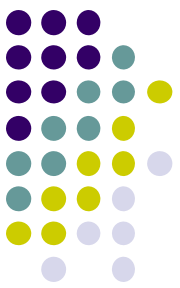
Cinquième étape: création d'une EJB Session pour l'EJB Entité

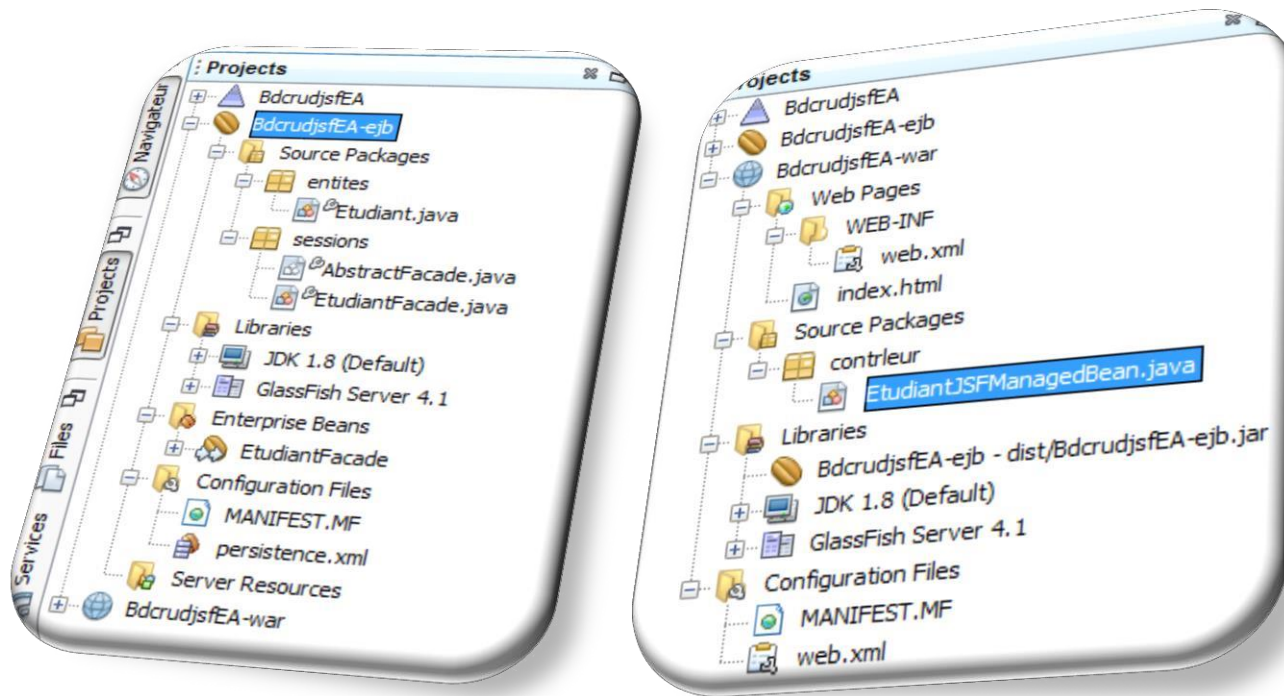
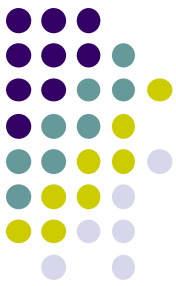


Cinquième étape: création d'une EJB Session pour l'EJB Entité



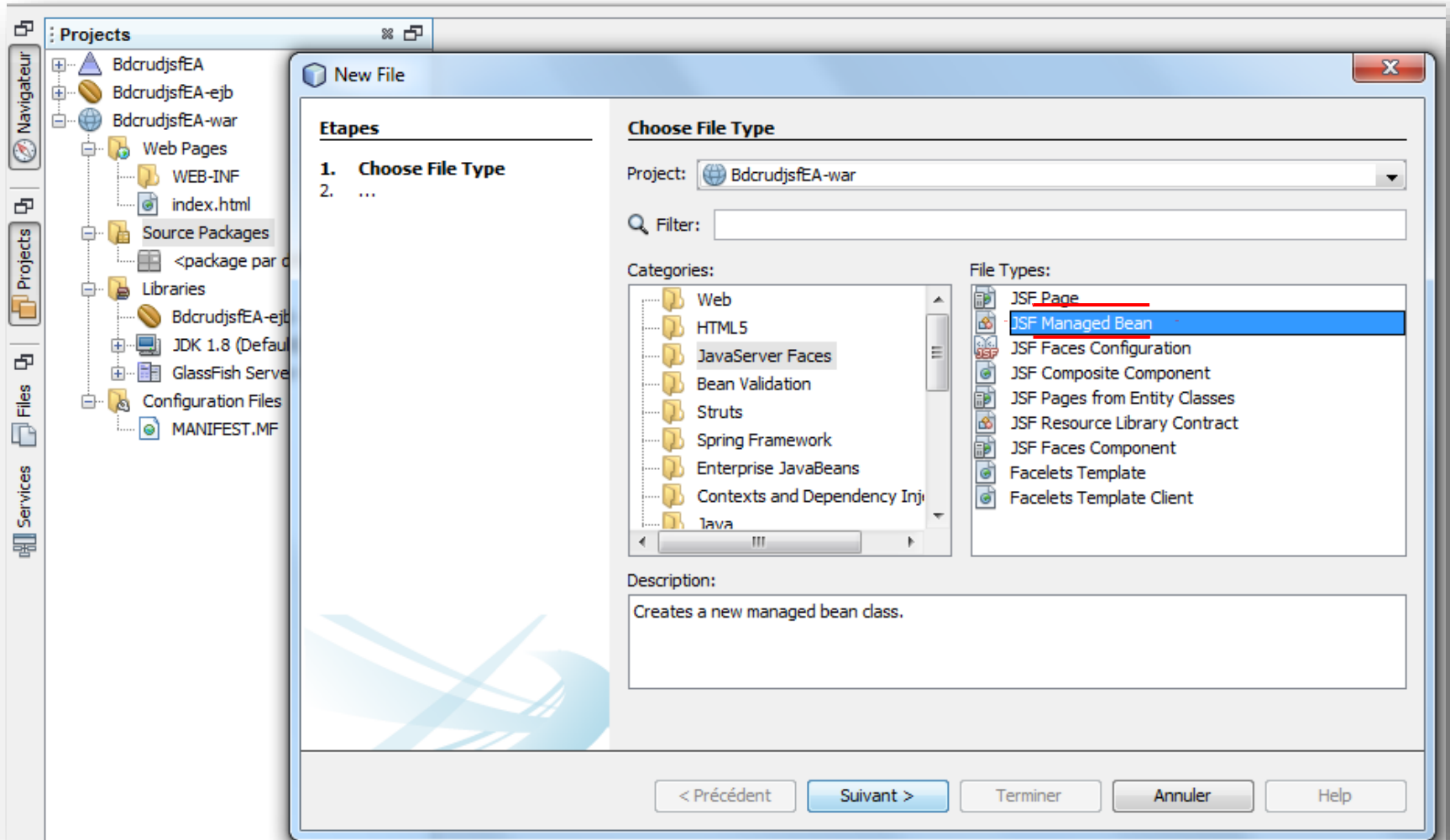
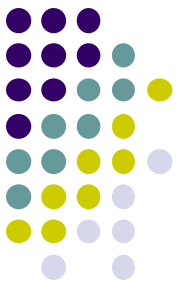
Cinquième étape: création d'une EJB Session pour l'EJB Entité



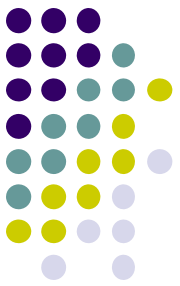


Création des éléments du module web: controleur et facelets

Sixième étape: création d'un ManagedBean



Sixième étape: création d'un ManagedBean



New JSF Managed Bean

Etapes

1. Choose File Type
2. **Nom et emplacement**

Nom et emplacement

Class Name: EtudiantJSFManagedBean

Projet: BdcrudjsfEA-war

Emplacement: Source Packages

Package: contrleur

Fichier créé: objects\BdcrudjsfEA\BdcrudjsfEA-war\src\java\contrleur\EtudiantJSFManagedBean.java

☐ Add data to configuration file

Configuration File:

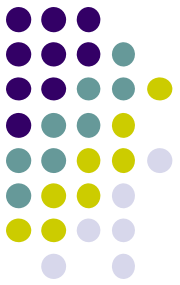
Name: etudiantJSFManagedBean

Scope: request

< Précédent Suivant > Terminer Annuler

Projects

- BdcrudjsfEA
 - BdcrudjsfEA-ejb
 - BdcrudjsfEA-war
 - Web Pages
 - WEB-INF
 - web.xml
 - index.html
 - Source Packages
 - contrleur
 - EtudiantJSFManagedBean.java**
 - Libraries
 - BdcrudjsfEA-ejb - dist/BdcrudjsfEA-ejb.jar
 - JDK 1.8 (Default)
 - GlassFish Server 4.1
 - Configuration Files
 - MANIFEST.MF
 - web.xml



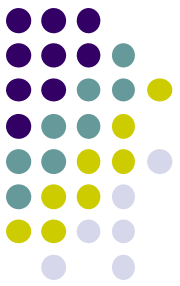
Sixième étape: création d'un ManagedBean

- Invoquer l'EJB et déclarer une instance ainsi qu'une liste d'étudiants

Ne pas oublier
d'ajouter les
modificateurs

```
@ManagedBean
@RequestScoped
public class EtudiantJSFManagedBean {
    @EJB
    private EtudiantFacade etudiantFacade;
    private Etudiant etudiant=new Etudiant();
    private List<Etudiant> listeEtudiants=new ArrayList<Etudiant>();

    /**
     * Creates a new instance of EtudiantJSFManagedBean
     */
    public EtudiantJSFManagedBean() {
    }
}
```



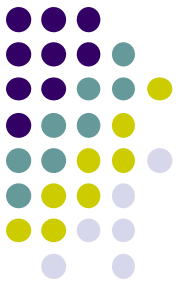
Sixième étape: création d'un ManagedBean

- Modifier la méthode **getListeEtudiants** qui retourne la liste des étudiants à partir de la base de données

```
public List<Etudiant> getListeEtudiants() {  
    return etudiantFacade.findAll();  
}
```

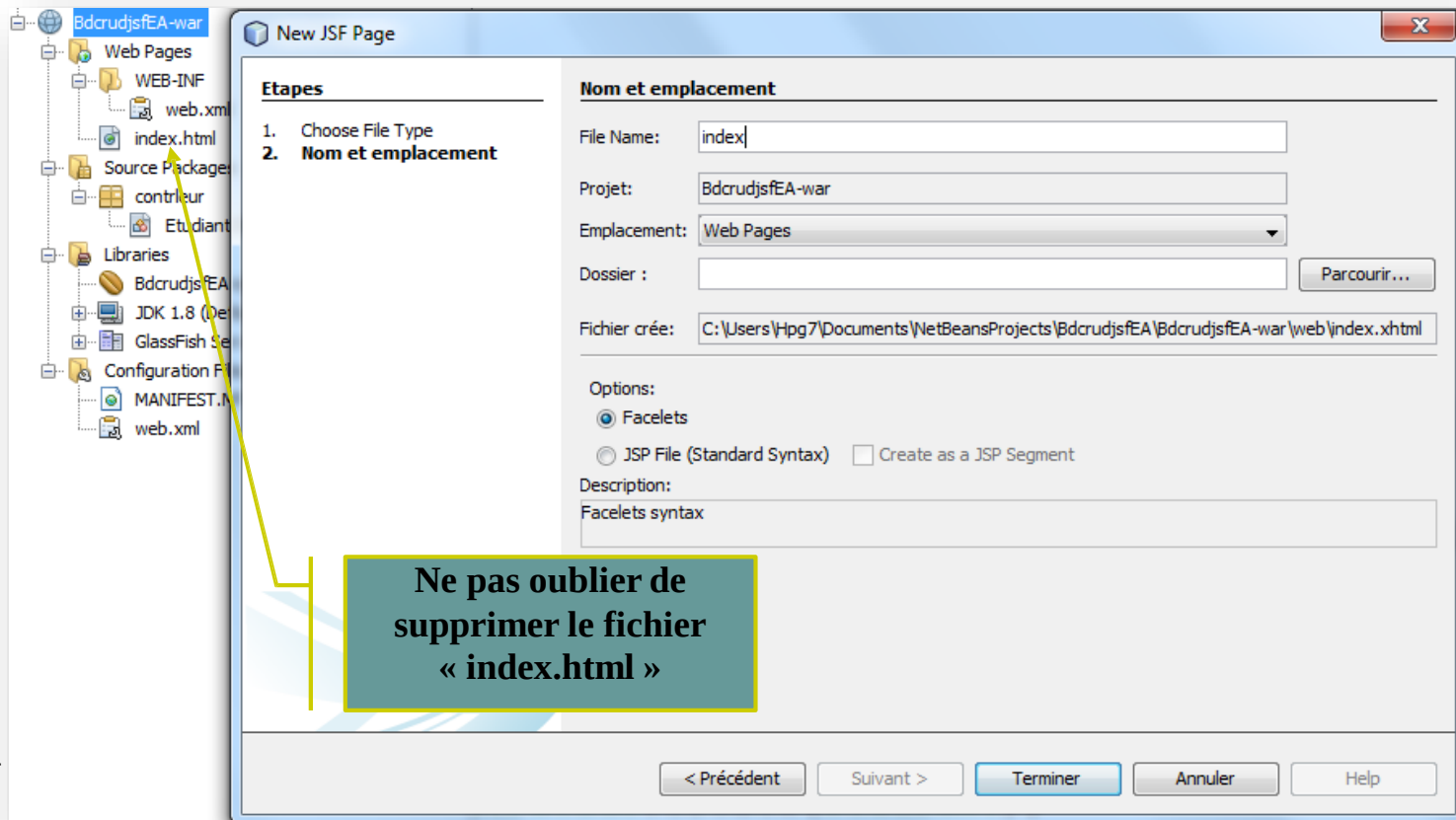
- Ajouter une méthode **ajouterEtudiant** qui insère un nouveau étudiant dans la base de données

```
public String ajouterEtudiant() {  
    etudiantFacade.create(etudiant);  
    listeEtudiants = etudiantFacade.findAll();  
    return "index.xhtml";  
}
```



Septième étape: création des vues

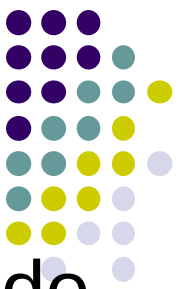
- Créer une JSF index.xhtml pour afficher tous les étudiants de la base données.



- 



Septième étape: création des vues



- Insérer une « JSF Data Table » à partir de de l'entité « etudiant »

The screenshot shows the NetBeans IDE with the `index.xhtml` file open. The code is in XML format, showing the beginning of an XHTML document. The `<h1>Liste des étudiants</h1>` tag is highlighted. A dialog box titled "JSF Table From Entity" is open, showing the configuration for creating a JSF Data Table from the `entites.Etudiant` entity. The "Managed Bean Property" is set to `etudiantJSFManagedBean.listeEtudiants`. The "Templates Style" is set to "Standard JavaServer Fa...". The "OK" button is highlighted. In the background, the "Palette" window shows the "HTML" category with options like "Table", "Liste classée", and "Liste non classée". The "Services" window shows the "JSF" category with options like "JSF Form", "Metadata", "JSF Form From Entity", "JSF Data Table", and "JSF Data Table From Entity".

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:h="http://xmlns.jcp.org/xmlns/java/org.apacheFaces.jsptemplates.html"
    <h:head>
        <title>Facelet Title</title>
    </h:head>
    <h:body>
        <h1>Liste des étudiants</h1>
    </h:body>
</html>
```

Entity: `entites.Etudiant`

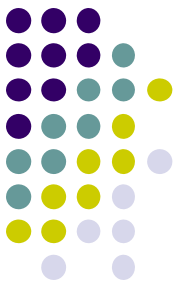
Managed Bean Property: `etudiantJSFManagedBean.listeEtudiants`

(the managed bean you are referencing must contain a method that returns a List<entites.Etudiant>)

Templates Style: `Standard JavaServer Fa...` [Customize Template](#)

OK Cancel

Septième étape: création des vues



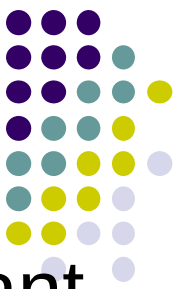
- Supprimer du formulaire les lignes concernant la saisie de l'Id qui est un champ autoincrément

```
<h:outputLabel value="Id:" for="id" />
<h:inputText id="id" value="#{etudiantJSFManagedBean.etudiant.id}"
             title="Id" required="true" requiredMessage="The Id field is required."/>
```

- Vous pouvez aussi améliorer la présentation du tableau

```
<h:dataTable value="#{etudiantJSFManagedBean.listeEtudiants}" var="item"
             border="1" bgcolor="red" width="50%"
             style="margin: auto; text-align: left">
```

Septième étape: création des vues



- Insérer un formulaire pour insérer un étudiant

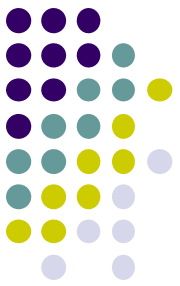
The screenshot shows an IDE with a project named 'EtudiantJSFManagedBean.java' and a file named 'index.xhtml'. The code in the editor is as follows:

```
33 <f:facet name="..."
34 <h:outputText value="..."
35
36 </f:facet>
37 <h:outputText value="..."
38 <f:commandAction value="..."
39 </h:outputText>
40 </h:column>
41 </h:dataTable>
42 </h:form>
43 </f:view>
44 </h:body>
45 </html>
46
```

A dialog box titled 'JSF Form From Entity' is open, showing the following configuration:

- Entity: entites.Etudiant
- Managed Bean Property: etudiantJSFManagedBean.etudiant
- (the managed bean you are referencing must contain a method that returns a entites.Etudiant)
- ☐ Generate read only view
- Templates Style: Standard JavaServer Faces
- Buttons: OK, Cancel

The 'JSF Form From Entity' dialog is also visible in the bottom right corner of the IDE.



Septième étape: création des vues

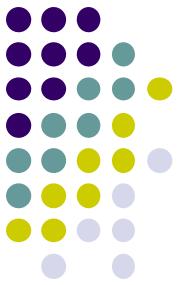
- Compléter le formulaire en ajoutant le bouton de commande qui appelle la méthode permettant d'ajouter l'étudiant dont les informations viennent d'être saisies et ensuite recharger la page index

```
<h:commandButton value="Ajouter un étudiant"  
                 action="#{etudiantJSFManagedBean.ajouterEtudiant()}" />
```

Créer/Editer un Etudiant

Nom:	Wade
Prenom:	Abdoulaye
Datenaiss:	12/20/1908
Ajouter un étudiant	

Septième étape: création des vues



Liste des étudiants

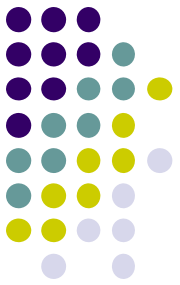
Id	Nom	Prenom	Datenaiss
1	Wade	Abdoulaye	12/20/1908
2	Sall	Macky	01/01/1945

Créer/Editer un Etudiant

Nom:

Prenom:

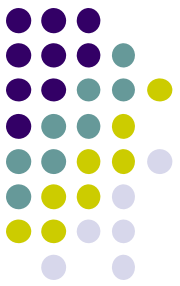
Datenaiss:



Septième étape: Ajout de liens de suppression et modification

- Ajouter au formulaire une nouvelle colonne pour les opérations de suppression et de modification

```
<h:column>
  <f:facet name="header">
    <h:outputText value="Opérations" />
  </f:facet>
  <h:commandLink value="Supprimer"
    action="#{etudiantJSFManagedBean.supprimerEtudiant(item)}" />
  &nbsp; &nbsp; &nbsp; &nbsp;
  <h:commandLink value="Mettre à jour" action="" />
</h:column>
```



Septième étape: Ajout de liens de suppression et modification

- Ajouter au ManagerBean de la méthode permettant de gérer l'opération de suppression

```
public String supprimerEtudiant(Etudiant etu) {  
    etudiantFacade.remove(etu);  
    listeEtudiants = etudiantFacade.findAll();  
    return "index.xhtml";  
}
```

Liste des étudiants

Id	Nom	Prenom	Datenaiss	Opérations
2	Sall	Macky	01/01/1945	Supprimer Mettre à jour

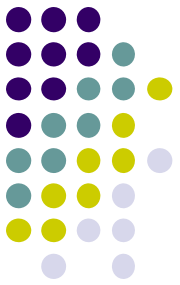
Créer/Editer un Etudiant

Nom:

Prenom:

Datenaiss:

Septième étape: Ajout de liens de suppression et modification



Liste des étudiants

Id	Nom	Prenom	Datenaiss	Opérations
2	Sall	Macky	01/01/1945	Supprimer Mettre à jour
3	Wade	Addoulaye	11/30/1908	Supprimer Mettre à jour
4	Diop	Addoulaye	11/18/1945	Supprimer Mettre à jour

Créer/Editer un Etudiant

Nom:

Prenom:

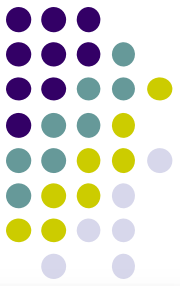
Datenaiss:

Septième étape: Ajout de liens de suppression et modification



- Proposez une solution pour la mise à jour et implémentez-la.

Septième étape: Ajout de liens de suppression et modification



▼ Affichage des étudiants

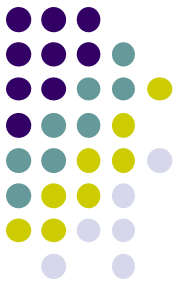
Liste des étudiants

Id	Nom	Prenom	Datenaiss	Opérations	
2	Sall	Macky	01/01/1945	Supprimer	Mettre à jour
17	Wade	Abdoulaye	30/11/1908	Supprimer	Mettre à jour
19	*****	Ousmane	09/07/0015	Supprimer	Mettre à jour

Créer/Editer un Etudiant

Id:	
Nom:	
Prenom:	
Datenaiss:	
Ajouter un étudiant	

Septième étape: Ajout de liens de suppression et modification



▼ Affichage des étudiants

Liste des étudiants

Id	Nom	Prenom	Datenaiss	Opérations	
2	Sall	Macky	01/01/1945	Supprimer	Mettre à jour
17	Wade	Abdoulaye	30/11/1908	Supprimer	Mettre à jour
19	*****	Ousmane	09/07/0015	Supprimer	Mettre à jour

Créer/Editer un Etudiant

Id:	19
Nom:	*****
Prenom:	Ousmane
Datenaiss:	09/07/0015 
Modifier un étudiant	

Septième étape: Ajout de liens de suppression et modification



▼ Affichage des étudiants

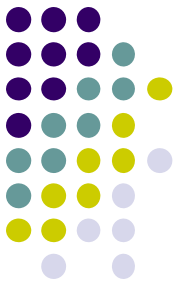
Liste des étudiants

Id	Nom	Prenom	Datenaiss	Opérations	
2	Sall	Macky	01/01/1945	Supprimer	Mettre à jour
17	Wade	Abdoulaye	30/11/1908	Supprimer	Mettre à jour
19	*****	Ousmane	09/07/0015	Supprimer	Mettre à jour

Créer/Editer un Etudiant

Id:	19
Nom:	DIOP
Prenom:	Ousmane
Datenaiss:	09/07/0015 
Modifier un étudiant	

Septième étape: Ajout de liens de suppression et modification



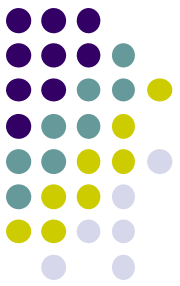
▼ Affichage des étudiants

Liste des étudiants

Id	Nom	Prenom	Datenaiss	Opérations	
2	Sall	Macky	01/01/1945	Supprimer	Mettre à jour
17	Wade	Abdoulaye	30/11/1908	Supprimer	Mettre à jour
19	DIOP	Ousmane	09/07/0015	Supprimer	Mettre à jour

Créer/Editer un Etudiant

Id:	
Nom:	
Prenom:	
Datenaiss:	
Ajouter un étudiant	



Autre alternative

▼ Affichage des étudiants

Liste des étudiants

Id	Nom
2	Sall
17	Wade
19	DIOP

Opérations	
Supprimer	Mettre à jour
Supprimer	Mettre à jour
Supprimer	Mettre à jour

Créer/Editer un Etudiant

Id:	19
Nom:	DIOP
Prenom:	Ousmane
Datenaiss:	09/07/0015
<button>Modifier un étudiant</button>	

Modifier un étudiant ✕

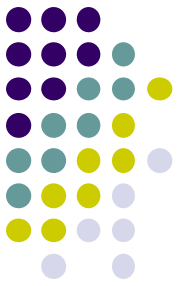
Editer un Etudiant

Id:	19
Nom:	DIOP
Prenom:	Ousmane
Datenaiss:	09/07/0015
Modifier	

- Ajouter un bouton pour mettre à jour les informations du formulaire

```
<h:commandButton value="Ajouter un étudiant"
    action="#{etudiantJSFManagedBean.ajouterEtudiant}"
    rendered="#{not etudiantJSFManagedBean.modifiable}"/>
<h:commandButton value="Modifier un étudiant"
    rendered="#{etudiantJSFManagedBean.modifiable}"
    action="#{etudiantJSFManagedBean.modifier}" />
```


Septième étape: Ajout de liens de suppression et modification



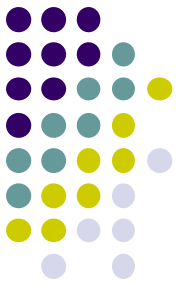
- Ajouter un attribut au managerBean afin de savoir s'il faut afficher un bouton ajouter ou modifier sur le formulaire

```
private boolean modifiable;

public boolean isModifiable() {
    return modifiable;
}

public void setModifiable(boolean modifiable) {
    this.modifiable = modifiable;
}
```

Septième étape: Ajout de liens de suppression et modification



- Enfin ajouter les méthodes suivantes

```
public String modifierEtudiant(Etudiant etu) {  
    modifiable=true;  
    etudiant = etu;  
    System.out.println("### Appel " + etudiant.toString());  
    return null;  
}  
  
public String modifier() {  
    System.out.println("### " + etudiant.toString());  
    etudiantFacade.edit(etudiant);  
    listeEtudiants = etudiantFacade.findAll();  
    modifiable=false;  
    return "index";  
}
```

Jeux de composants additionnels à JSF

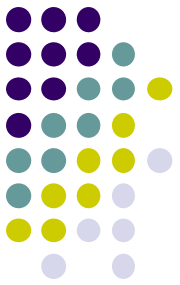


Les deux jeux de composants standards de JSF s'avèrent trop limités et insuffisants pour le développement d'applications d'entreprise

- Il est possible d'utiliser des jeux de composants additionnels offrant de nouveaux composants plus riches. On peut citer par exemple:
 - [Primefaces](#), un jeu de composants open-source supportant Ajax, Jsf 2, Push
 - [ICEfaces](#), un jeu de composants open-source supportant Ajax.
 - [jBoss RichFaces et Ajax4JSF](#), un jeu de composants open-source supportant Ajax
 - [Apache Tomahawk](#), un jeu de composants très riche et open-source
 - [RCFaces](#), un jeu de composants très riche AJAX et open-source

https://fr.wikipedia.org/wiki/JavaServer_Faces

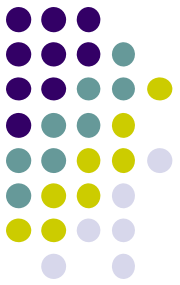
PrimeFaces



Original author(s)	PrimeTek
Stable release	5.2.1 ^[1] / April 17, 2015
Written in	Java
Operating system	Cross-platform
Type	Java, JSF, Ajax, Framework
License	Apache License 2.0 ^[2]
Website	primefaces.org 

http://www.primefaces.org/docs/guide/primefaces_user_guide_5_1.pdf

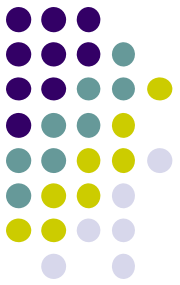
PrimeFaces



- Très Nombreux champs supplémentaires : **ColorPicker, Calendar, TagCloud...**
- **Panneaux, graphiques, multimédia, gestion de fichiers...**
- Gestion par JSF d'**images dynamiques** (en JSF pures, gérées par... une servlet).
- Extension de la spécification : langage plus riche pour désigner des éléments,

http://www.primefaces.org/docs/guide/primefaces_user_guide_5_1.pdf

PrimeFaces

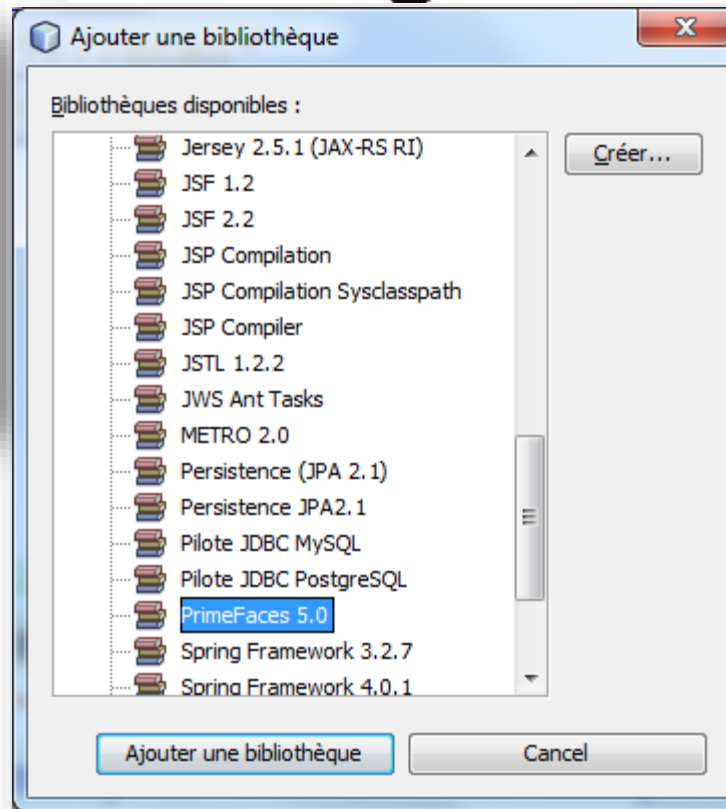
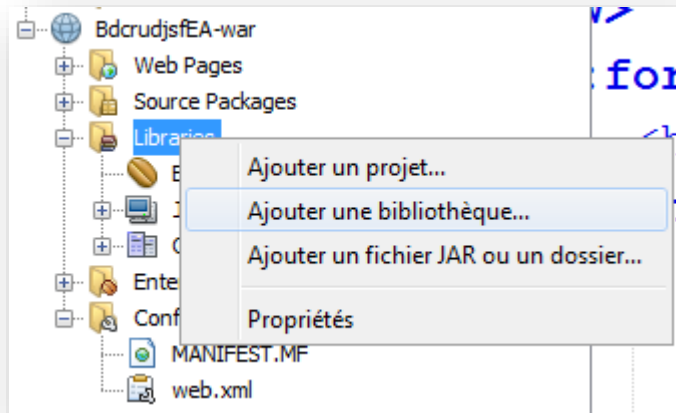
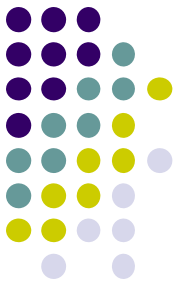


- Les concurrents de PRIMEFACES :

- RichFaces
- ICEfaces
- Tomahawk
- RC Faces
- Open Faces
-

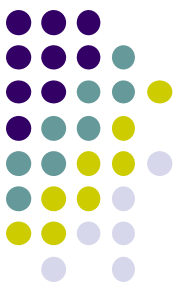


PrimeFaces



- Prime Faces namespace est nécessaire pour ajouter les composants Prime Faces à votre page : **`xmlns:p="http://primefaces.org/ui"`**

Quelques Composants et fonctions PrimeFaces



<http://www.primefaces.org/docs/vdl/3.5/alltags-noframe.html>

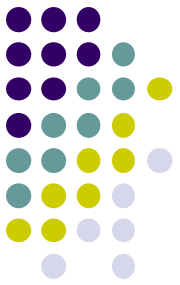
primefaces-p:component()
primefaces-p:ifAllGranted()
primefaces-p:ifAnyGranted()
primefaces-p:ifGranted()
primefaces-p:ifNoneGranted()
primefaces-p:remoteUser()
primefaces-p:userPrincipal()
primefaces-p:widgetVar()
primefaces-p:accordionPanel
primefaces-p:ajax
primefaces-p:ajaxStatus
primefaces-p:autoComplete
primefaces-p:barChart
primefaces-p:blockUI
primefaces-p:breadcrumb
primefaces-p:bubbleChart
primefaces-p:button
primefaces-p:calendar
primefaces-p:captcha
primefaces-p:carousel
primefaces-p:cellEditor
primefaces-p:clock
primefaces-p:collector
primefaces-p:colorPicker
primefaces-p:column
primefaces-p:columnGroup
primefaces-p:columns

primefaces-p:columnGroup
primefaces-p:columns
primefaces-p:commandButton
primefaces-p:commandLink
primefaces-p:confirmDialog
primefaces-p:contextMenu
primefaces-p:dashboard
primefaces-p:dataExporter
primefaces-p:dataGrid
primefaces-p:dataList
primefaces-p:dataTable
primefaces-p:defaultCommand
primefaces-p:dialog
primefaces-p:dock
primefaces-p:donutChart
primefaces-p:draggable
primefaces-p:droppable
primefaces-p:editor
primefaces-p:effect
primefaces-p:feedReader
primefaces-p:fieldset
primefaces-p:fileDownload
primefaces-p:fileUpload
primefaces-p:focus
primefaces-p:galleria
primefaces-p:gmap
primefaces-p:gmapInfoWindow
primefaces-p:graphicImage
primefaces-p:growl
primefaces-p:hotkey
primefaces-p:idleMonitor

primefaces-p:imageCompare
primefaces-p:imageCropper
primefaces-p:imageSwitch
primefaces-p:inplace
primefaces-p:inputMask
primefaces-p:inputText
primefaces-p:inputTextarea
primefaces-p:keyboard
primefaces-p:layout
primefaces-p:layoutUnit
primefaces-p:lightbox
primefaces-p:lineChart
primefaces-p:log
primefaces-p:media
primefaces-p:megaMenu
primefaces-p:menu
primefaces-p:menubar
primefaces-p:menuButton
primefaces-p:menuItem
primefaces-p:message
primefaces-p:messages
primefaces-p:meterGaugeChart
primefaces-p:mindmap
primefaces-p:notificationBar
primefaces-p:ohlcChart
primefaces-p:orderList
primefaces-p:outputLabel
primefaces-p:outputPanel
primefaces-p:overlayPanel
primefaces-p:panel
primefaces-p:panelGrid
primefaces-p:panelMenu
primefaces-p:password
primefaces-p:photoCam
primefaces-p:pickList

primefaces-p:pieChart
primefaces-p:poll
primefaces-p:printer
primefaces-p:progressBar
primefaces-p:radioButton
primefaces-p:rating
primefaces-p:remoteCommand
primefaces-p:resetInput
primefaces-p:resizable
primefaces-p:ring
primefaces-p:row
primefaces-p:rowEditor
primefaces-p:rowExpansion
primefaces-p:rowToggler
primefaces-p:schedule
primefaces-p:scrollPanel
primefaces-p:selectBooleanButton
primefaces-p:selectBooleanCheckbox
primefaces-p:selectCheckboxMenu
primefaces-p:selectManyButton
primefaces-p:selectManyCheckbox
primefaces-p:selectManyMenu
primefaces-p:selectOneButton
primefaces-p:selectOneListbox
primefaces-p:selectOneMenu
primefaces-p:selectOneRadio
primefaces-p:separator
primefaces-p:slideMenu
primefaces-p:slider
primefaces-p:socket
primefaces-p:spacer
primefaces-p:spinner
primefaces-p:splitButton

primefaces-p:stack
primefaces-p:submenu
primefaces-p:subTable
primefaces-p:summaryRow
primefaces-p:tab
primefaces-p:tabMenu
primefaces-p:tableView
primefaces-p:tagCloud
primefaces-p:terminal
primefaces-p:themeSwitcher
primefaces-p:tieredMenu
primefaces-p:toolbar
primefaces-p:toolbarGroup
primefaces-p:tooltip
primefaces-p:tree
primefaces-p:TreeNode
primefaces-p:treeTable
primefaces-p:watermark
primefaces-p:wizard

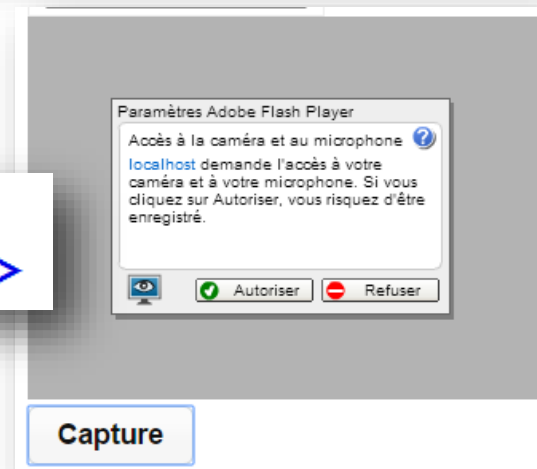


- Exemple du calendrier du formulaire

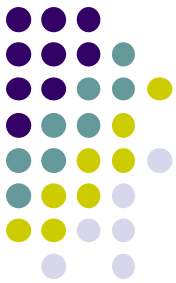
```
<p:calendar id="datenaiss" value="#{etudiantJSFManagedBean.etudiant.datenaiss}"  
            title="Datenaiss" required="true"  
            requiredMessage="The Datenaiss field is required."  
            showOn="both">  
    <f:convertDateTime pattern="dd/MM/yyyy" />  
</p:calendar>
```

- Simple accès au web cam

```
<p:photoCam widgetVar="pc" />  
<p:commandButton type="button" value="Capture"/>
```

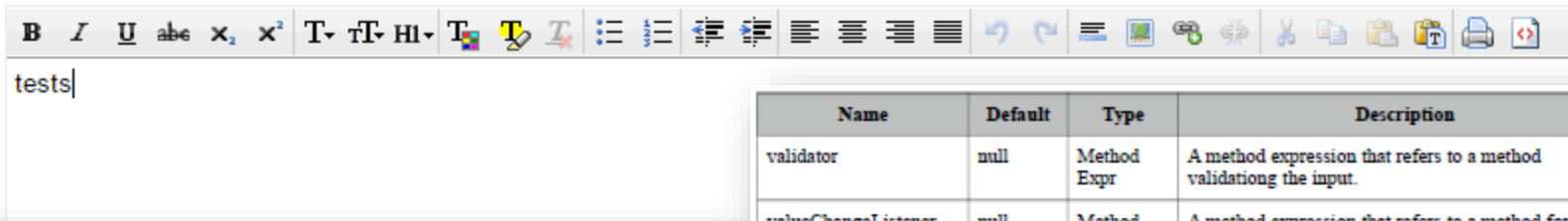


PrimeFaces



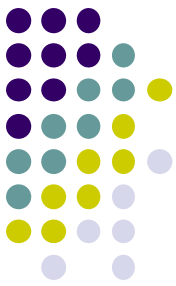
● Exemple Editor

```
<p:editor value="tests" />
```



Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component.
binding	null	Object	An el expression that maps to a server side UIComponent instance in a backing bean.
value	null	Object	Value of the component than can be either an EL expression of a literal text.
converter	null	Converter/ String	An el expression or a literal text that defines a converter for the component. When it's an EL expression, it's resolved to a converter instance. In case it's a static text, it must refer to a converter id.
immediate	false	Boolean	When set true, process validations logic is executed at apply request values phase for this component.
required	false	Boolean	Marks component as required.

Name	Default	Type	Description
validator	null	Method Expr	A method expression that refers to a method validating the input.
valueChangeListener	null	Method Expr	A method expression that refers to a method for handling a valuchangeevent.
requiredMessage	null	String	Message to be displayed when required field validation fails.
converterMessage	null	String	Message to be displayed when conversion fails.
validatorMessage	null	String	Message to be displayed when validation fails.
widgetVar	null	String	Name of the client side widget.
controls	null	String	List of controls to customize toolbar.
height	null	Integer	Height of the editor.
width	null	Integer	Width of the editor.
disabled	false	Boolean	Disables editor.
style	null	String	Inline style of the editor container.
styleClass	null	String	Style class of the editor container.
onchange	null	String	Client side callback to execute when editor data changes.
maxlength	null	Integer	Maximum length of the raw input.



● Exemple AccordionPanel

▼ First Tab Title

Lorem ...More content for first tab

▶ Second Tab Title

```
<p:accordionPanel>
  <p:tab title="First Tab Title">
    <h:outputText value= "Lorem"/>
    ...More content for first tab
  </p:tab>
  <p:tab title="Second Tab Title">
    <h:outputText value="Ipsum" />
  </p:tab>
  // any number of tabs
</p:accordionPanel>
```

PrimeFaces



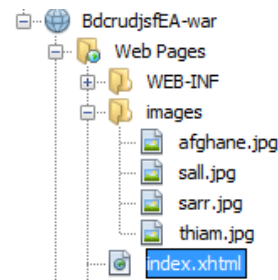
- Exemple Carousel

Carousel is a multi purpose component to display a set of data or general content with slide effects.

```
<p:carousel autoPlayInterval="2000" effect="easeInStrong" circular="true"
            itemStyle="width:200px" >
    <p:graphicImage value="/images/sarr.jpg"/>
    <p:graphicImage value="/images/thiam.jpg"/>
    <p:graphicImage value="/images/sall.jpg"/>
    <p:graphicImage value="/images/afghane.jpg"/>
</p:carousel>
```



PrimeFaces



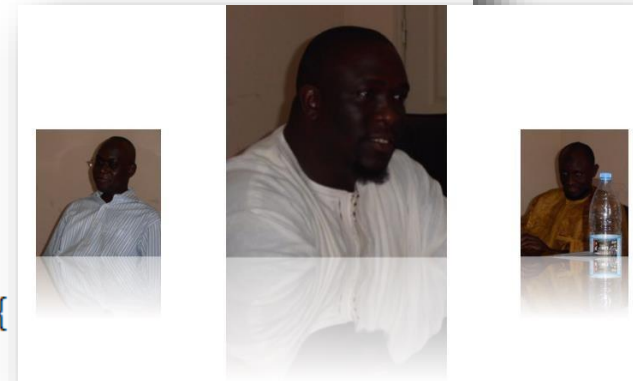
- Exemple ContentFlow

ContentFlow is a horizontal content gallery component with a slide animation.

```
private List<String> images = new ArrayList<>(Arrays.asList(  
    "thiam.jpg", "sarr.jpg", "sall.jpg", "afghane.jpg"));
```

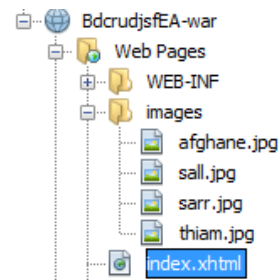
```
public List<String> getImages() {  
    return images;  
}
```

```
public void setImages(List<String> images) {  
    this.images = images;  
}
```



```
<p:contentFlow var="image" value="#{etudiantJSFManagedBean.images}">  
    <p:graphicImage value="/images/#{image}" styleClass="content" />  
</p:contentFlow>
```

PrimeFaces



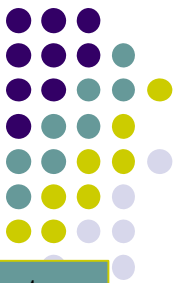
Galleria is used to display a set of images.

- Exemple Galleria

```
<p:galleria value="#{etudiantJSFManagedBean.images}" var="image" >
    <p:graphicImage value="/images/#{image}"
        title="titre" alt="description" />
</p:galleria>
```



PrimeFaces



● Exemple Color Picker

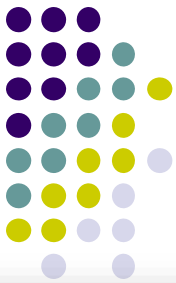
ColorPicker is an input component with a color palette.

```
<h:body style="background-color: #{etudiantJSFManagedBean.color}">
  <h:form>
    <p:colorPicker value="#{etudiantJSFManagedBean.color}" mode="inline"/>
    <h:commandButton value="Valider"/>
    Couleur:<h:outputText value="#{etudiantJSFManagedBean.color}" />
  </h:form>
</h:body>
```



Name	Default	Type	Description
valueChangeListener	null	ValueChange Listener	A method binding expression that refers to a method for handling a valuchangeevent.
requiredMessage	null	String	Message to be displayed when required field validation fails.
converterMessage	null	String	Message to be displayed when conversion fails.
validatorMessage	null	String	Message to be displayed when validation fields.
widgetVar	null	String	Name of the client side widget.
mode	popup	String	Display mode, valid values are "popup" and "inline".
style	null	String	Inline style of the component.
styleClass	null	String	Style class of the component.

PrimeFaces



● Exemple Dock

Dock component mimics the well known dock interface of Mac OS X.



`<p:dock>`

```
<p:menuItem value="Home" icon="/images/dock/home.png" url="#" />
<p:menuItem value="Music" icon="/images/dock/music.png" url="#" />
<p:menuItem value="Video" icon="/images/dock/video.png" url="#" />
<p:menuItem value="Email" icon="/images/dock/email.png" url="#" />
<p:menuItem value="Link" icon="/images/dock/link.png" url="#" />
<p:menuItem value="RSS" icon="/images/dock/rss.png" url="#" />
<p:menuItem value="History" icon="/images/dock/history.png" url="#" />
```

`</p:dock>`

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	Boolean value to specify the rendering of the component, when set to false component will not be rendered.
binding	null	Object	An el expression that maps to a server side UIComponent instance in a backing bean
model	null	MenuModel	MenuModel instance to create menus programmatically
position	bottom	String	Position of the dock, <i>bottom</i> or <i>top</i> .
itemWidth	40	Integer	Initial width of items.
maxWidth	50	Integer	Maximum width of items.
proximity	90	Integer	Distance to enlarge.
halign	center	String	Horizontal alignment,
widgetVar	null	String	Name of the client side widget.

PrimeFaces



- Exemple GMap

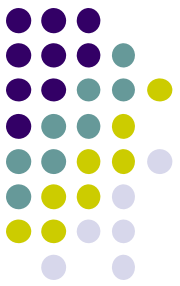
GMap is a map component integrated with Google Maps API V3.

```
<script src="http://maps.google.com/maps/api/js?sensor=true|false"
type="text/javascript"></script>

<p:gmap center="14.7727624, -16.95356909999998" zoom="15" type="hybrid"
style="width:600px;height:400px" />
```



PrimeFaces



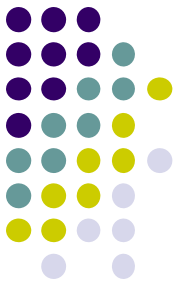
- Exemple Keyboard

```
<p:keyboard value="value" />
```

Keyboard is an input component that uses a virtual keyboard to provide the input. Notable features are the customizable layouts and skinning capabilities.



PrimeFaces

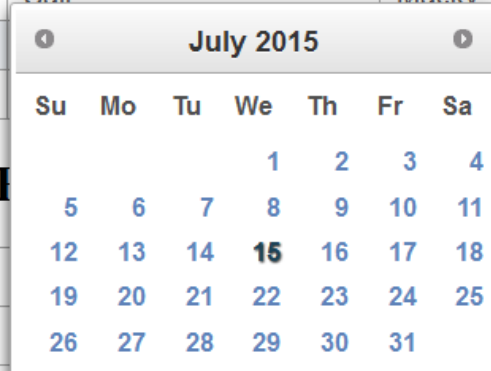


Liste des étudiants

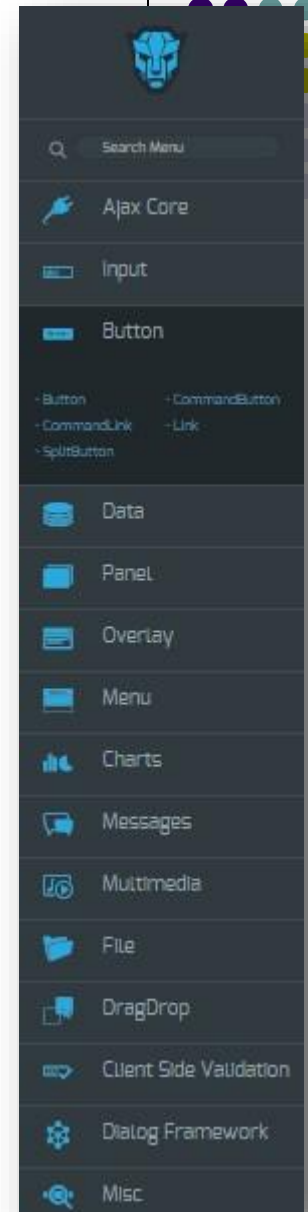
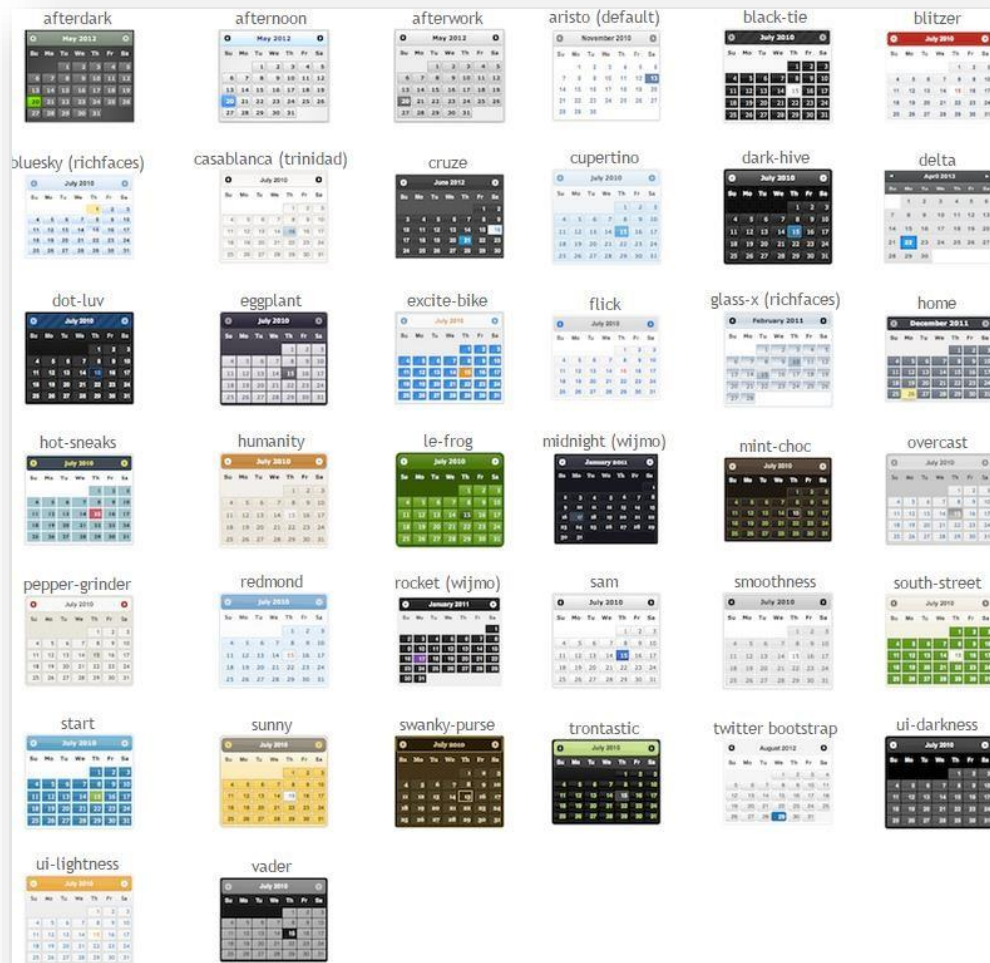
Id	Nom	Prenom	Datenaiss	Opérations	
2	Sall	Mackv	01/01/1945	Supprimer	Mettre à jour
3		ye	30/11/1908	Supprimer	Mettre à jour
4		ye	18/11/1945	Supprimer	Mettre à jour

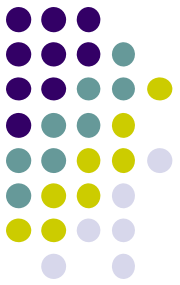
Créer/Editer un I

Nom:	
Prenom:	
Datenaiss:	
Ajouter un étudiant	

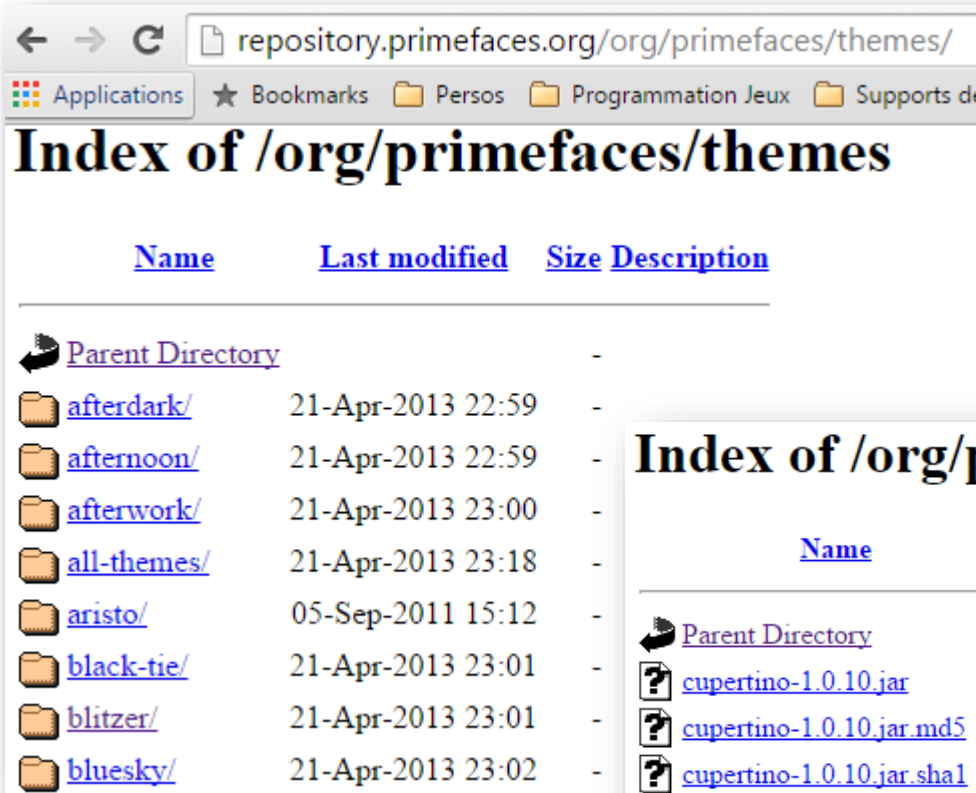


THÈMES (+30 THÈMES PRÉDÉFINIS) :





- Télécharger le jar d'un des thèmes et l'ajouter au projet <http://primefaces.org/themes>



← → ↻ repository.primefaces.org/org/primefaces/themes/

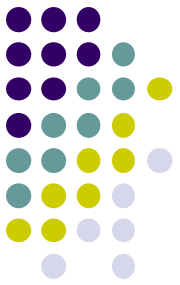
Applications ★ Bookmarks Persos Programmation Jeux Supports de

Index of /org/primefaces/themes

Name	Last modified	Size	Description
Parent Directory	-	-	-
afterdark/	21-Apr-2013 22:59	-	-
afternoon/	21-Apr-2013 22:59	-	-
afterwork/	21-Apr-2013 23:00	-	-
all-themes/	21-Apr-2013 23:18	-	-
aristo/	05-Sep-2011 15:12	-	-
black-tie/	21-Apr-2013 23:01	-	-
blitzer/	21-Apr-2013 23:01	-	-
bluesky/	21-Apr-2013 23:02	-	-

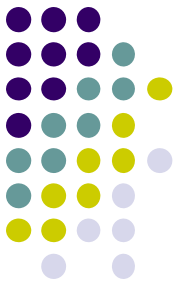
Index of /org/primefaces/themes/cupertino/1.0.10

Name	Last modified	Size	Description
Parent Directory	-	-	-
 cupertino-1.0.10.jar	21-Apr-2013 23:04	30K	
 cupertino-1.0.10.jar.md5	21-Apr-2013 23:04	32	
 cupertino-1.0.10.jar.sha1	21-Apr-2013 23:04	40	
 cupertino-1.0.10.pom	21-Apr-2013 23:04	627	
 cupertino-1.0.10.pom.md5	21-Apr-2013 23:04	32	
 cupertino-1.0.10.pom.sha1	21-Apr-2013 23:04	40	

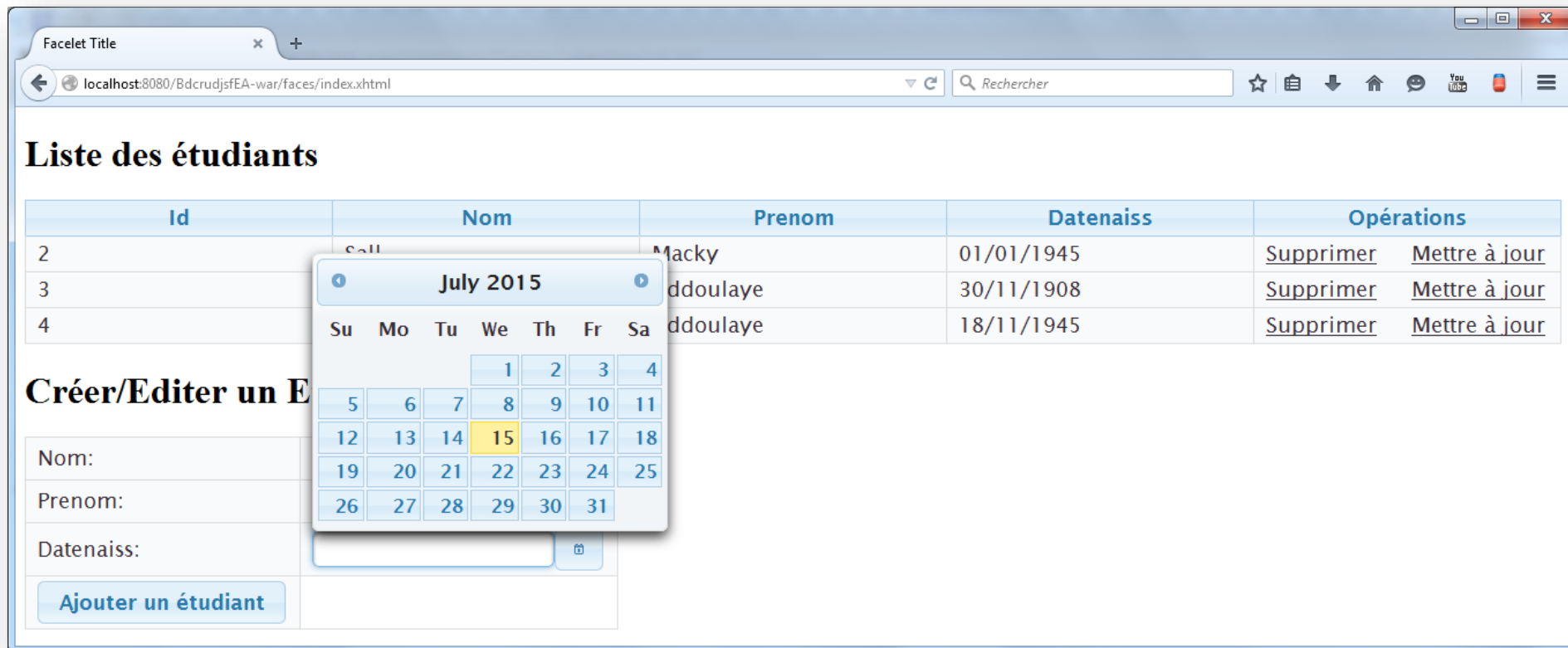


- Ajouter le nom du thème approprié dans le fichier « web.xml »

```
<context-param>  
    <param-name>primefaces.THEME</param-name>  
    <param-value>cupertino</param-value>  
</context-param>
```



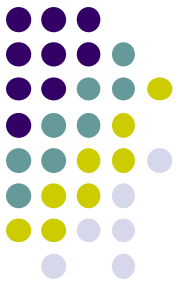
- Aspect du thème « cupertino »



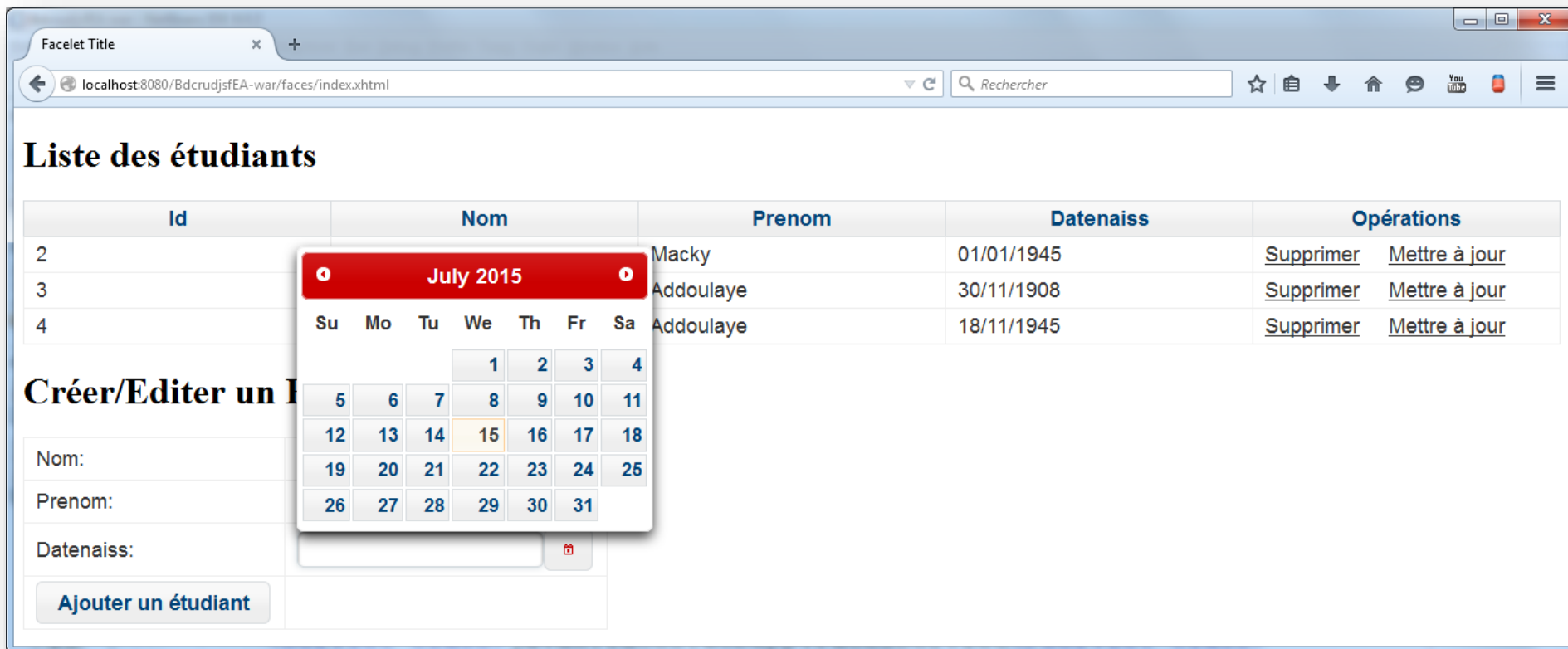
The screenshot shows a web browser window with the address bar displaying 'localhost:8080/BdcrudjsfEA-war/faces/index.xhtml'. The page title is 'Facelet Title'. The main content area is titled 'Liste des étudiants' and contains a table with the following data:

Id	Nom	Prenom	Datenaiss	Opérations
2	Sell	Macky	01/01/1945	Supprimer Mettre à jour
3		ddoulaye	30/11/1908	Supprimer Mettre à jour
4		ddoulaye	18/11/1945	Supprimer Mettre à jour

Below the table, there is a section titled 'Créer/Editer un E' with form fields for 'Nom:', 'Prenom:', and 'Datenaiss:'. A blue button labeled 'Ajouter un étudiant' is at the bottom. A date picker calendar is open, showing 'July 2015' with the date '15' selected.



- Aspect du thème « blitzer »

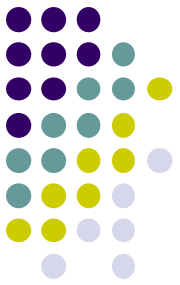


The screenshot shows a web browser window with the title "Facelet Title". The address bar shows "localhost:8080/BdcrudjsfEA-war/faces/index.xhtml". The page content includes a search bar with the text "Rechercher". Below the search bar is a section titled "Liste des étudiants" containing a table with the following data:

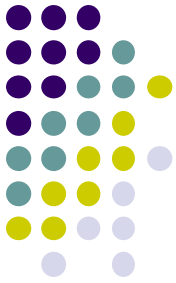
Id	Nom	Prenom	Datenaiss	Opérations
2		Macky	01/01/1945	Supprimer Mettre à jour
3		Addoulaye	30/11/1908	Supprimer Mettre à jour
4		Addoulaye	18/11/1945	Supprimer Mettre à jour

Below the table is a section titled "Créer/Editer un I" with form fields for "Nom:", "Prenom:", and "Datenaiss:". A date picker is open, showing the month of July 2015. The date picker has a header "July 2015" and a grid of dates. The date 15 is highlighted. Below the date picker is a button labeled "Ajouter un étudiant".

PrimeFaces



```
<p:accordionPanel>
  <p:tab title="Affichage des étudiants">
    <f:view>
      <h:form>
        <h1><h:outputText value="Liste des étudiants"/></h1>
        <p:dataTable value="#{etudiantJSFManagedBean.listeEtudiants}" var="item"
          style="margin: auto; text-align: left" class="table1">
          <p:column ...6 lines />
          <p:column ...6 lines />
          <p:column ...6 lines />
          <p:column ...8 lines />
          <p:column ...9 lines />
        </p:dataTable>
      </h:form>
    </f:view>
  </p:tab>
</p:accordionPanel>
```



► Affichage des étudiants

Créer/Editer un Etudiant

Nom:	
Prenom:	
Datenaiss:	
Ajouter un étudiant	

▼ Affichage des étudiants

Liste des étudiants

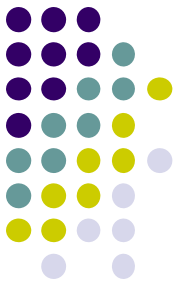
Id	Nom	Prenom	Datenaiss	Opérations	
2	Sall	Macky	01/01/1945	Supprimer	Mettre à jour
3	Wade	Addoulaye	30/11/1908	Supprimer	Mettre à jour
4	Diop	Addoulaye	18/11/1945	Supprimer	Mettre à jour

Créer/Editer un Etudiant

Nom:	
Prenom:	
Datenaiss:	
Ajouter un étudiant	

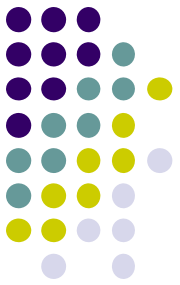


```
<p:accordionPanel>
  <p:tab title="Affichage des étudiants">
    <f:view>
      <h:form>
        <h1><h:outputText value="Liste des étudiants"/></h1>
        <p:dataTable value="#{etudiantJSFManagedBean.listeEtudiants}" var="item"
          style="margin: auto; text-align: left" class="table1">
          <p:column ...6 lines />
          <p:column ...6 lines />
          <p:column ...6 lines />
          <p:column ...8 lines />
          <p:column ...9 lines />
        </p:dataTable>
      </h:form>
    </f:view>
  </p:tab>
</p:accordionPanel>
```



Bibliographie

- The Java EE 6 Tutorial,
<http://docs.oracle.com/javaee/6/tutorial/doc/javaeetutorial6.pdf>
- <http://www.ibm.com/developerworks/library/j-jsf1/>
- <http://www.ibm.com/developerworks/library/j-jsf2/>
- <http://www.ibm.com/developerworks/library/j-jsf3/>



Bibliographie

- <http://www.ibm.com/developerworks/library/j-jsf4/>
- <http://www.ibm.com/developerworks/java/library/j-facelets/>
- Série « JSF-2Fu » sur <http://www.ibm.com/developerworks/> :
- <http://www.ibm.com/developerworks/java/library/j-jsf2fu1/index.html>
- <http://www.jsftutorials.net/>