

**Master Ingénierie de Données et
Développement Logiciel**
Architecture et Framework JEE

TP2 : JAVASERVER FACES 2.2

Année universitaire 2019-2020

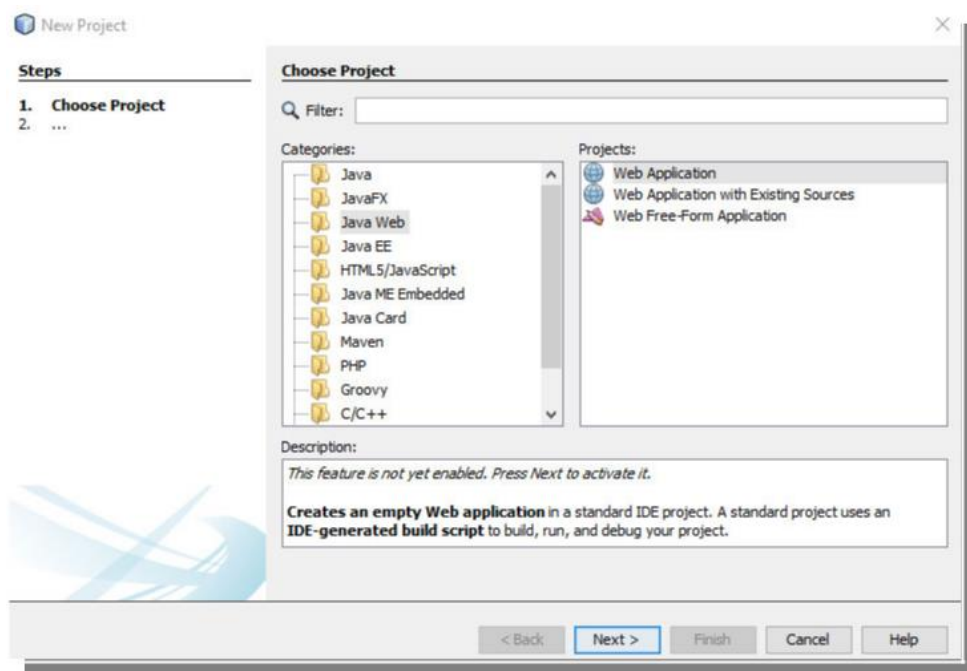
RÉALISER PAR : YOUNES MAHMAR

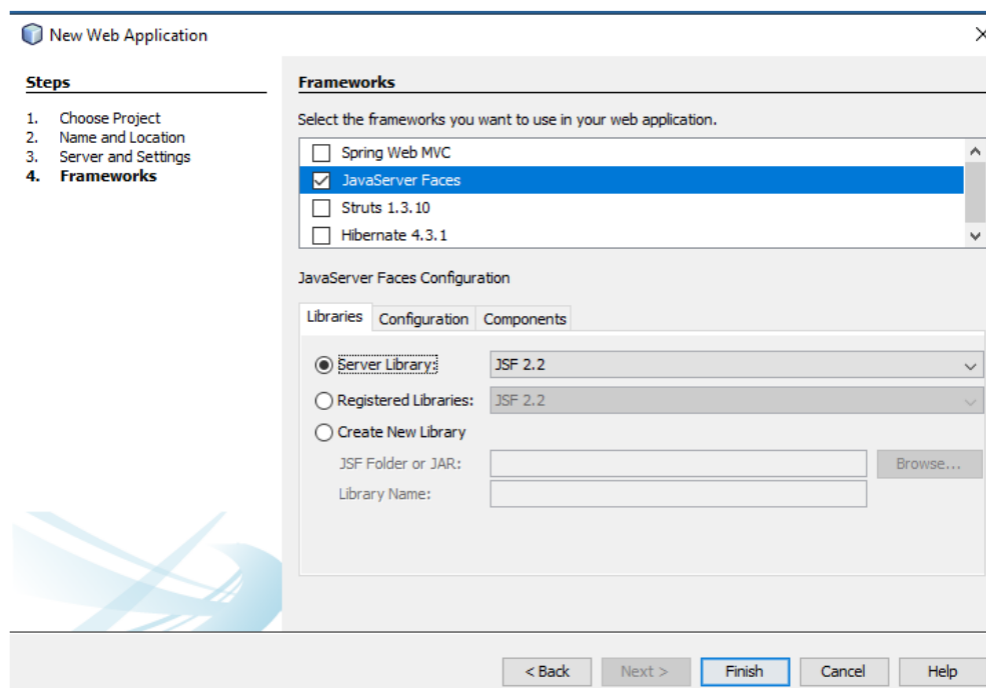
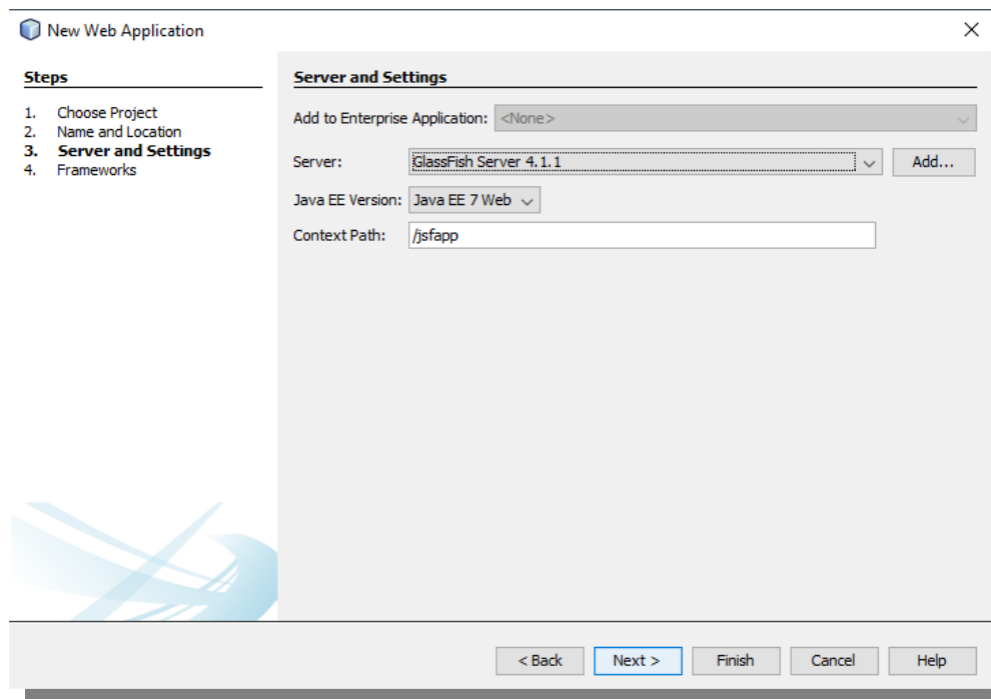
Objectif :

L'objectif de ce TP est le développement d'applications Web en utilisant JavaServer Faces 2.2. De nombreuses anciennes applications ont été écrites en utilisant les servlets et les JSP, les applications web modernes sont écrites en utilisant le Framework Java Server Faces (JSF). L'utilisation JSF permet de simplifier le développement des applications Web.

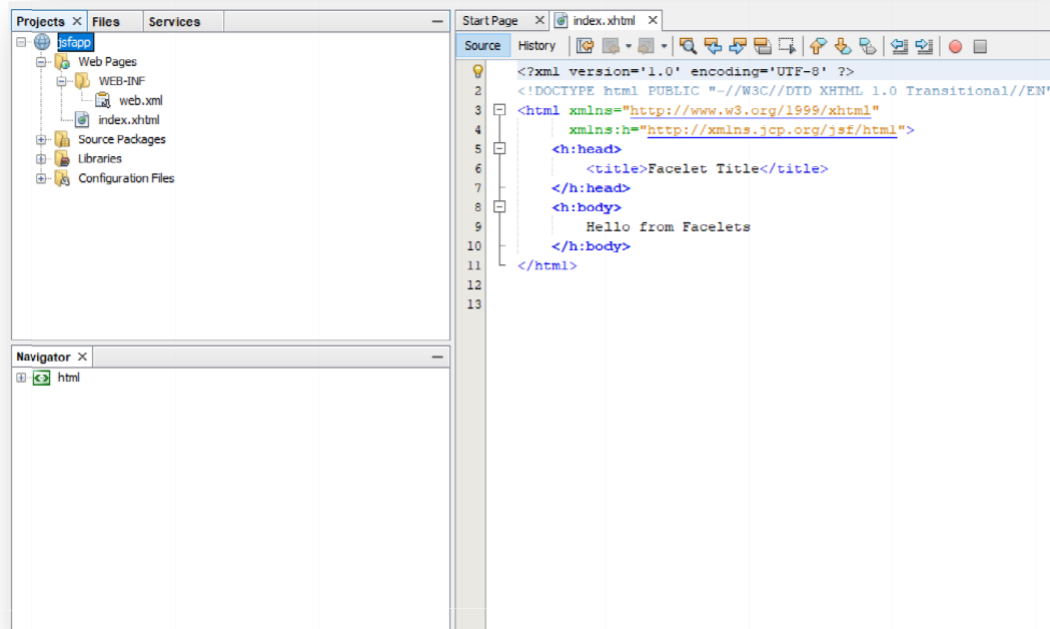
1 Créer un nouveau projet JSF

La démarche pour créer un nouveau projet jsf est très simple, il suffit de créer un nouveau projet de type java web et sélectionner le Framework javaServer Faces





Après cette étape, netbeans nous génère un squelette de projet JSF, composé d'un unique fichier facelet **index.xhtml**, un fichier de configuration **web.xml**.



On exécutant l'application, le serveur sera démarré et déployé.

```
Output - jsfapp (run)
Incrementally deploying jsfapp
Completed incremental distribution of jsfapp
run-deploy:
Browsing: http://localhost:8080/jsfapp
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 0 seconds)
```

← → ↻ ⓘ localhost:8080/jsfapp/

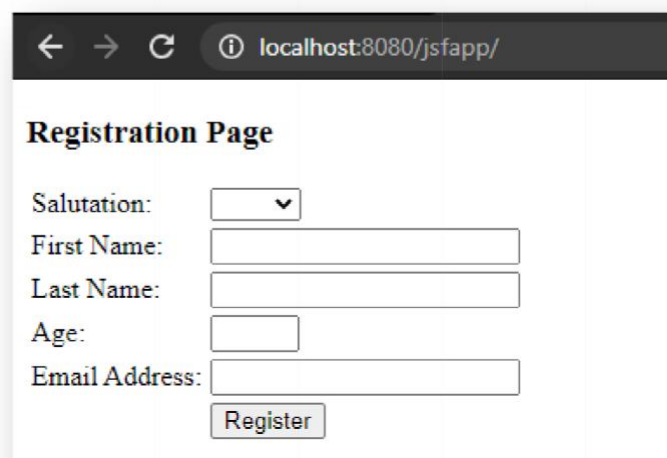
Hello from Facelets

2 Modifiez la page pour capturer les données utilisateur

Maintenant On va modifier le fichier d'accueil généré **index .xhtml** pour collecter des données de l'utilisateur.

On va utiliser le code proposer dans le tp.

```
<h:form>
<h:panelGrid columns="3"
columnClasses="rightalign,leftalign,leftalign">
<h:outputLabel value="Salutation: " for="salutation"/>
<h:selectOneMenu id="salutation" label="Salutation"
value="#{registrationBean.salutation}" >
<f:selectItem itemLabel="" itemValue="" />
<f:selectItem itemLabel="Mr." itemValue="MR"/>
<f:selectItem itemLabel="Mrs." itemValue="MRS"/>
<f:selectItem itemLabel="Miss" itemValue="MISS"/>
<f:selectItem itemLabel="Ms" itemValue="MS"/>
<f:selectItem itemLabel="Dr." itemValue="DR"/>
</h:selectOneMenu>
<h:message for="salutation"/>
<h:outputLabel value="First Name:" for="firstName"/>
<h:inputText id="firstName" label="First Name"
required="true"
value="#{registrationBean.firstName}" />
<h:message for="firstName" />
<h:outputLabel value="Last Name:" for="lastName"/>
<h:inputText id="lastName" label="Last Name"
required="true"
```



← → ↻ ⓘ localhost:8080/jsfapp/

Registration Page

Salutation:

First Name:

Last Name:

Age:

Email Address:

Les fichiers css et javascript doivent dans les répertoire resources->css et resources->js

Pour utilise un fichies css on utilise la balise **<h:outputStylesheet>** Pour utilise un fichies css on ut ilise la balise **<h:outputScript>**

3 Créer notre backing bean

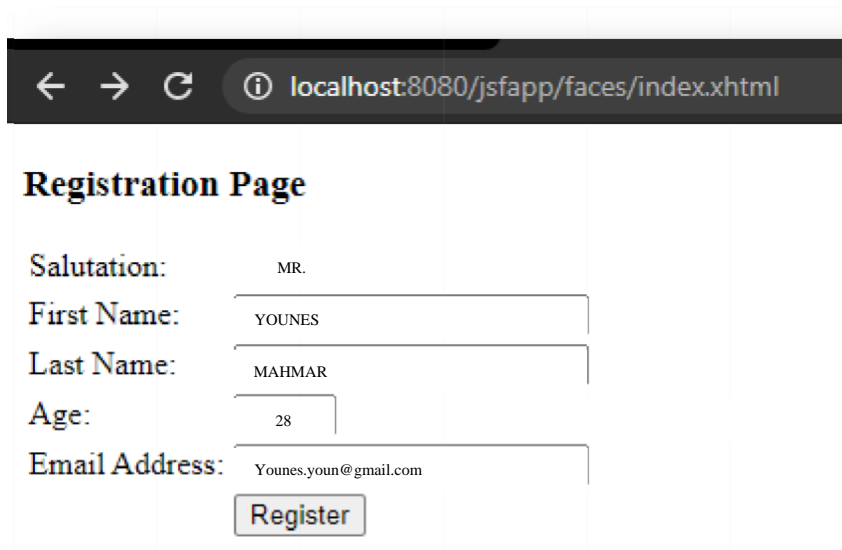
Les Beans només sont utilisés pour contenir les données entrées par les utilisateurs.

```
import javax.enterprise.context.RequestScoped;
import javax.inject.Named;

/**
 *
 * @author mohcine
 */
@Named
@RequestScoped
public class RegistrationBean {
    private String salutation;
    private String firstName;
    private String lastName;
    private Integer age;
    private String email;
    //getters and setters omitted for brevity
}
```

Pour ajouter les (getter et setter) il suffit de taper alt+inser

4) Implémenter la page de confirmation



← → ↻ ⓘ localhost:8080/jsfapp/faces/index.xhtml

Registration Page

Salutation: MR.

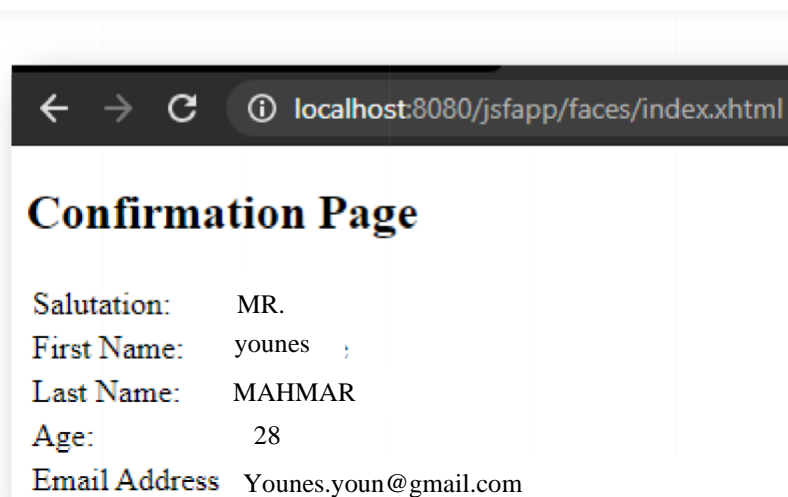
First Name: YOUNES

Last Name: MAHMAR

Age: 28

Email Address: Younes.youn@gmail.com

Register



← → ↻ ⓘ localhost:8080/jsfapp/faces/index.xhtml

Confirmation Page

Salutation: MR.

First Name: younes

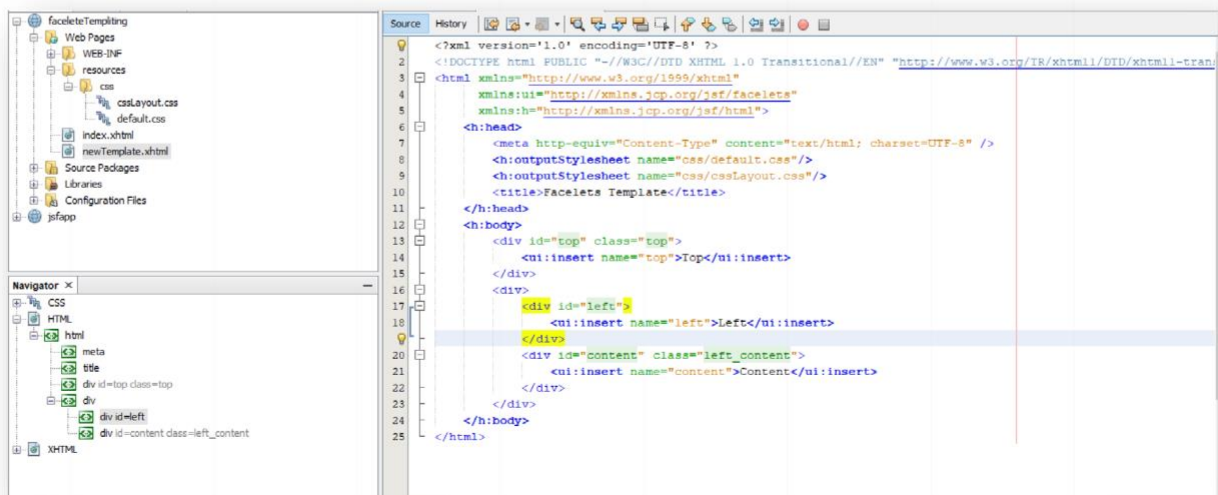
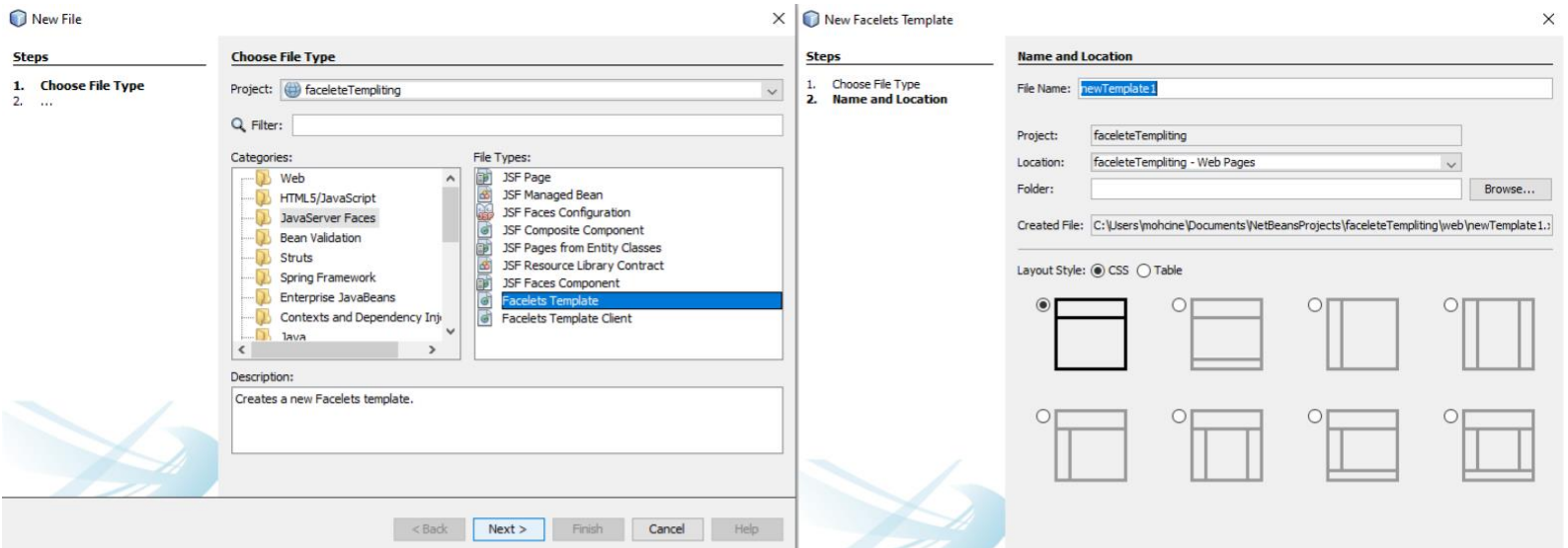
Last Name: MAHMAR

Age: 28

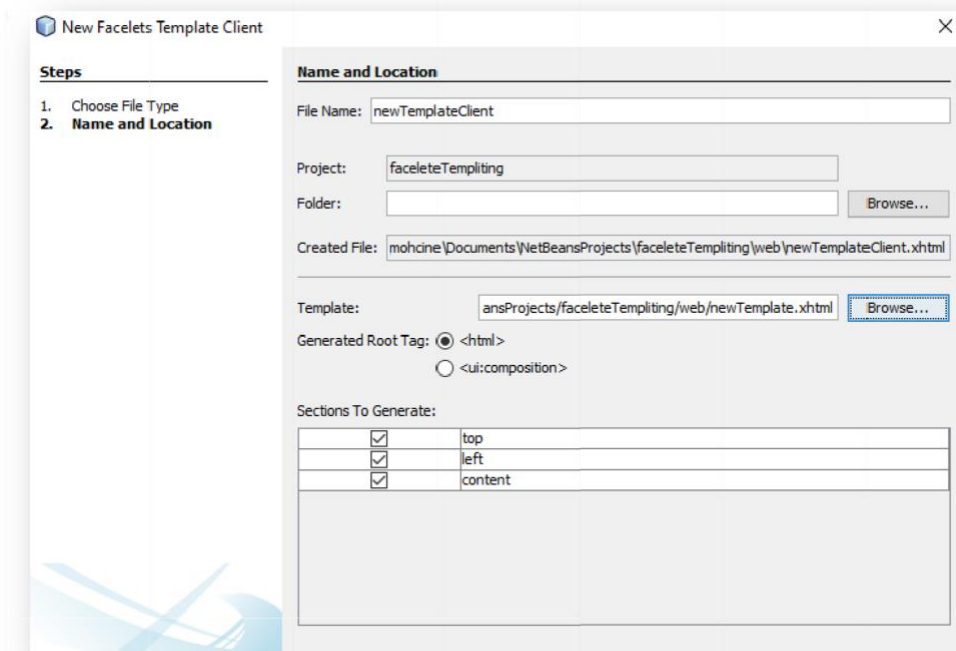
Email Address: Younes.youn@gmail.com

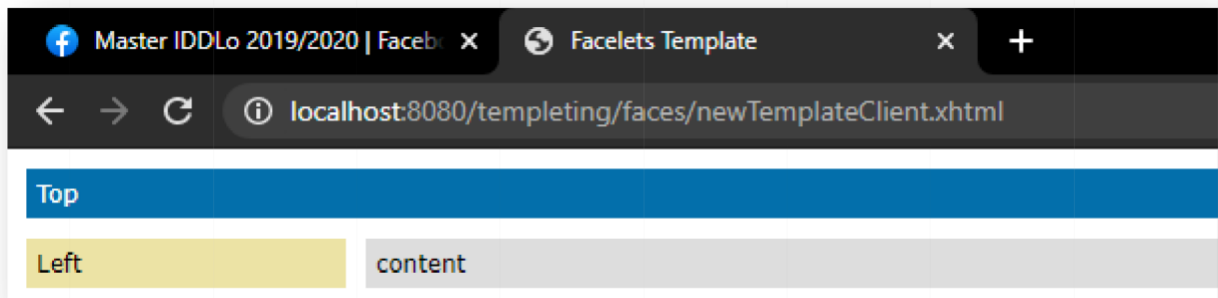
5 Les templates des Facelets

Un avantage des Facelets par rapport aux JS P est le mécanisme des templates (modèles). Les templates permettent de spécifier une page modèle et alors de l'utiliser pour créer des clients. L'utilisation d'un modèle simplifie la maintenance des applications, puisque les changements doivent être réalisés en un seul endroit (qui est le modèle). Si nous devons changer la disposition de nos pages (ajouter un pied, ou déplacer une colonne depuis le côté gauche de la page vers la droite, par exemple, on a à modifier seulement le modèle, et le changement se reflète dans toutes les pages clientes du modèle. NetBeans fournit un très bon support pour les templates des Facelets.



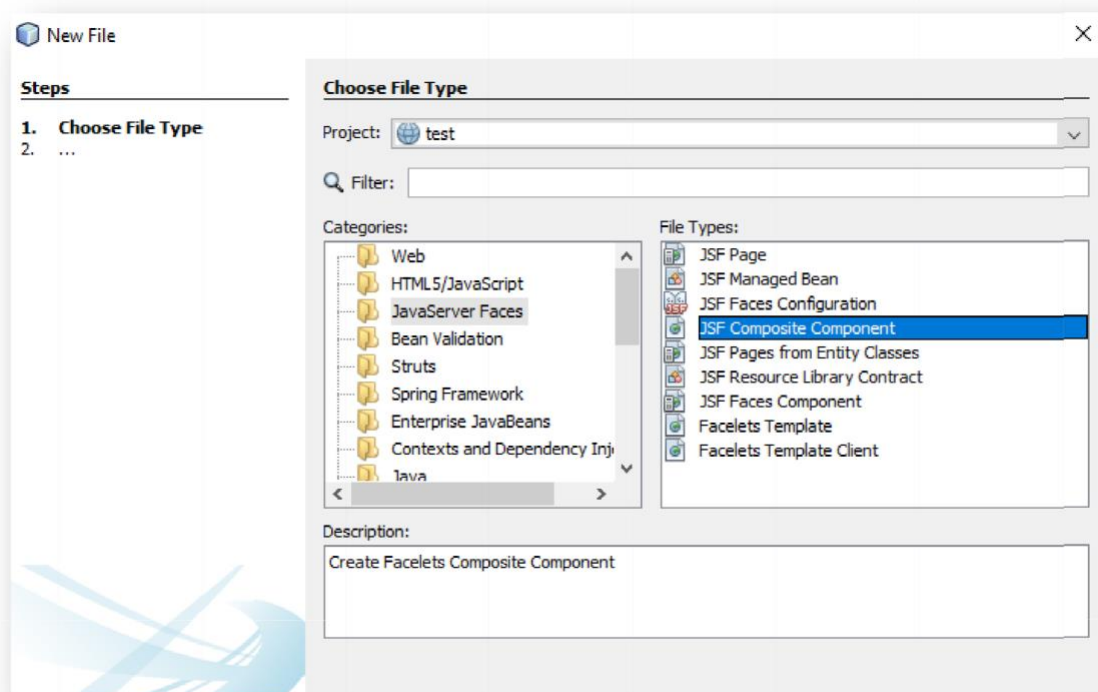
Ajout d'un modèle Facelets à notre projet

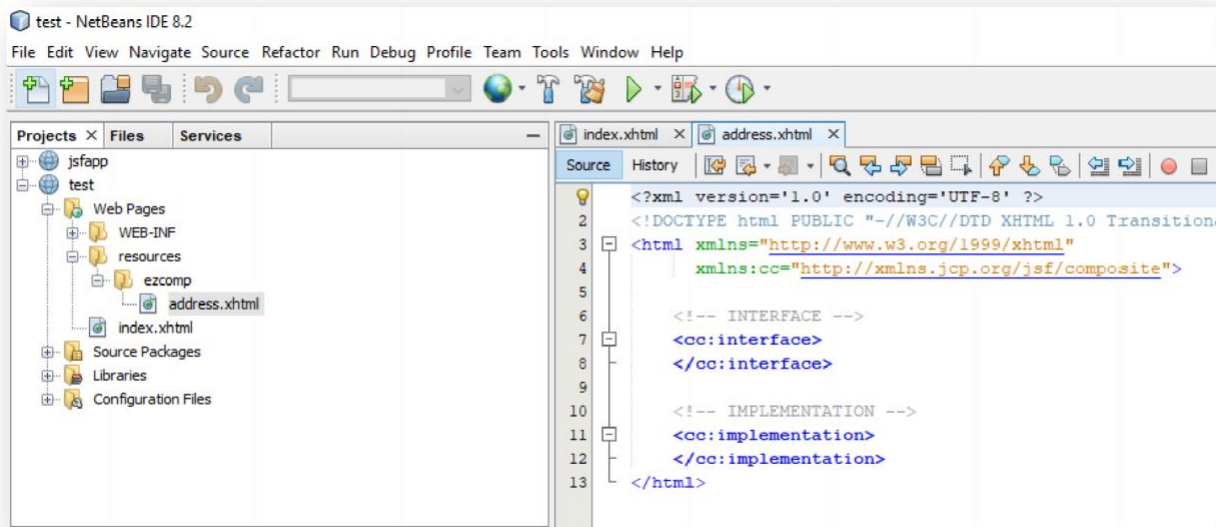




6 Composants composites

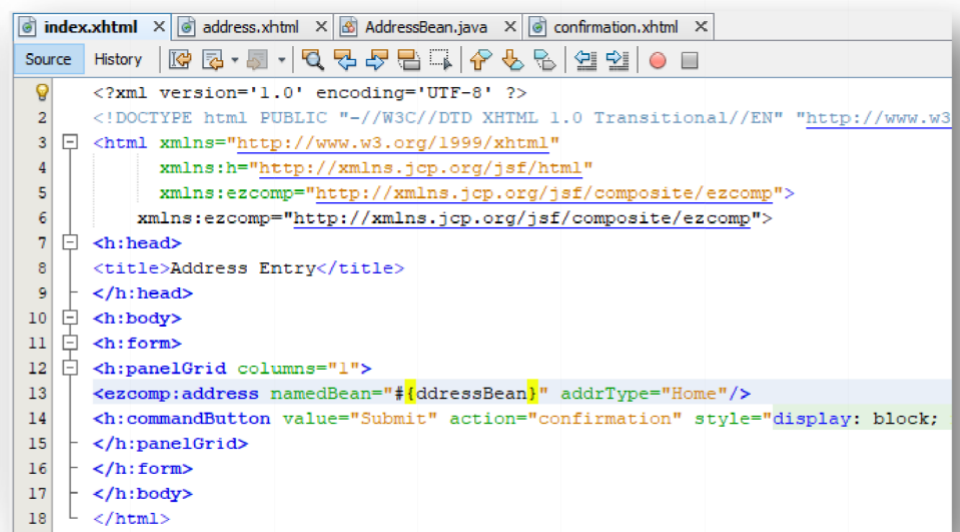
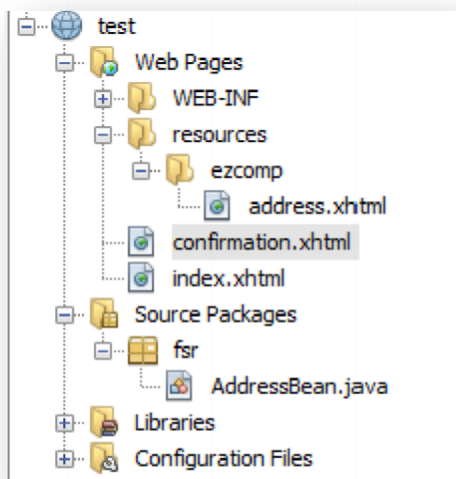
Une très belle fonctionnalité de JSF 2.2 est qu'il permet d'écrire facilement des composants personnalisés. Avec JSF 2, la création d'un composant personnalisé ressemble à la création d'une balise, sans aucun code Java et sans aucune configuration. Les composants personnalisés sont généralement composés des autres composants JSF, ils sont considérés comme des composants composites.

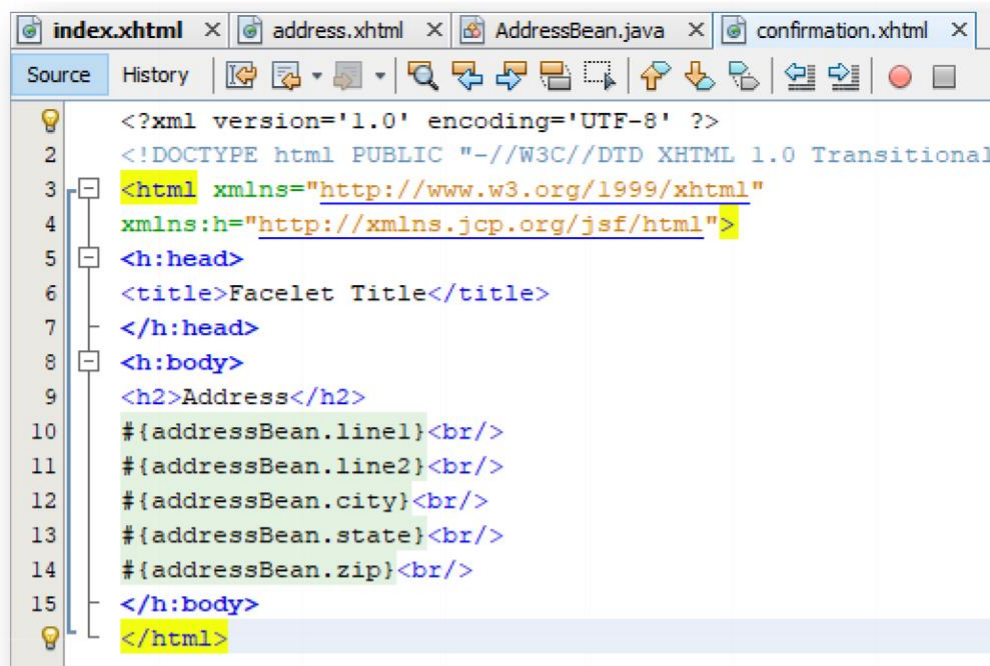




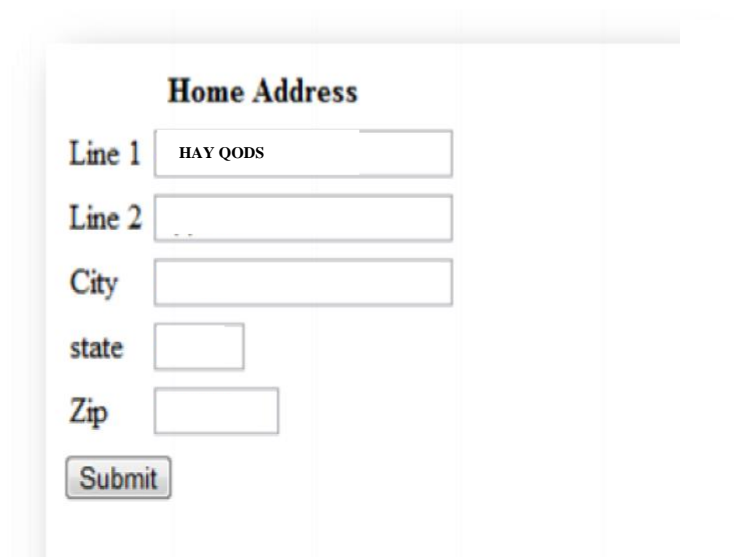
Dans notre exemple, on va développer un composant simple utilisé pour entrer l'adresse. De cette façon, si on doit entrer plusieurs adresses dans une appli cation, on peut encapsuler la logique et/ou l'affichage dans notre composant. Si plus tard, on doit changer l'entrée de l'adresse (par exemple, pour se conformer aux adresses internationales).

Après avoir rempli le fichier de confirmation, index.html et le bean addressBean on exécute notre application





```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://xmlns.jcp.org/jsf/html">
<h:head>
<title>Facelet Title</title>
</h:head>
<h:body>
<h2>Address</h2>
#{addressBean.line1}<br/>
#{addressBean.line2}<br/>
#{addressBean.city}<br/>
#{addressBean.state}<br/>
#{addressBean.zip}<br/>
</h:body>
</html>
```



Home Address

Line 1

Line 2

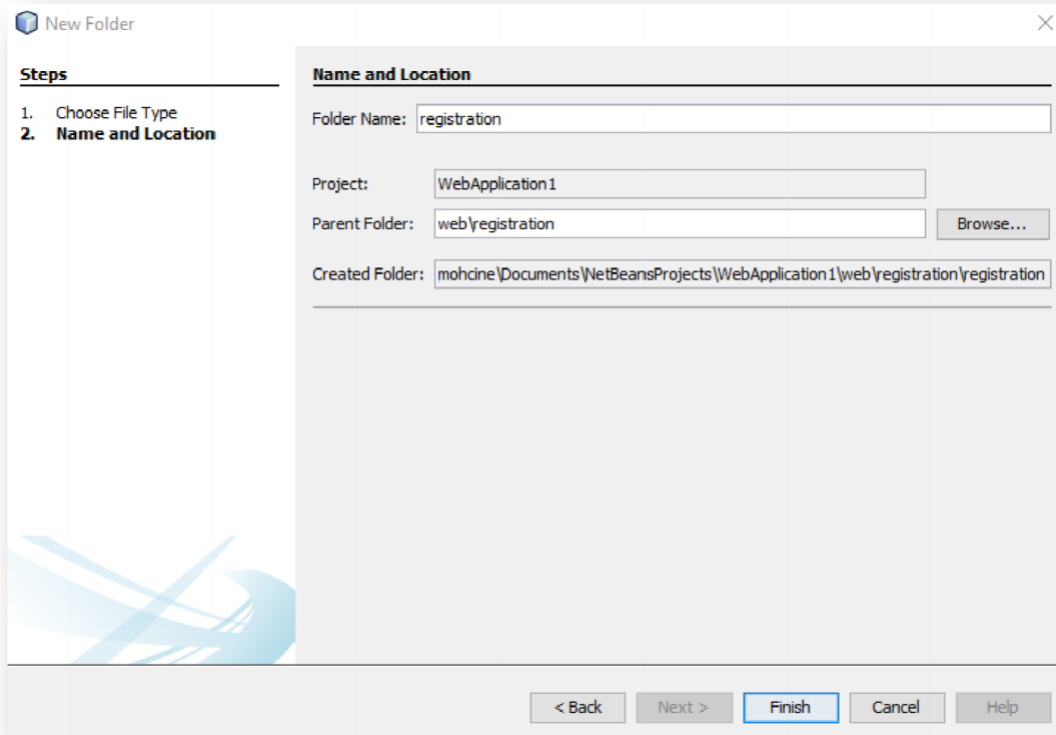
City

state

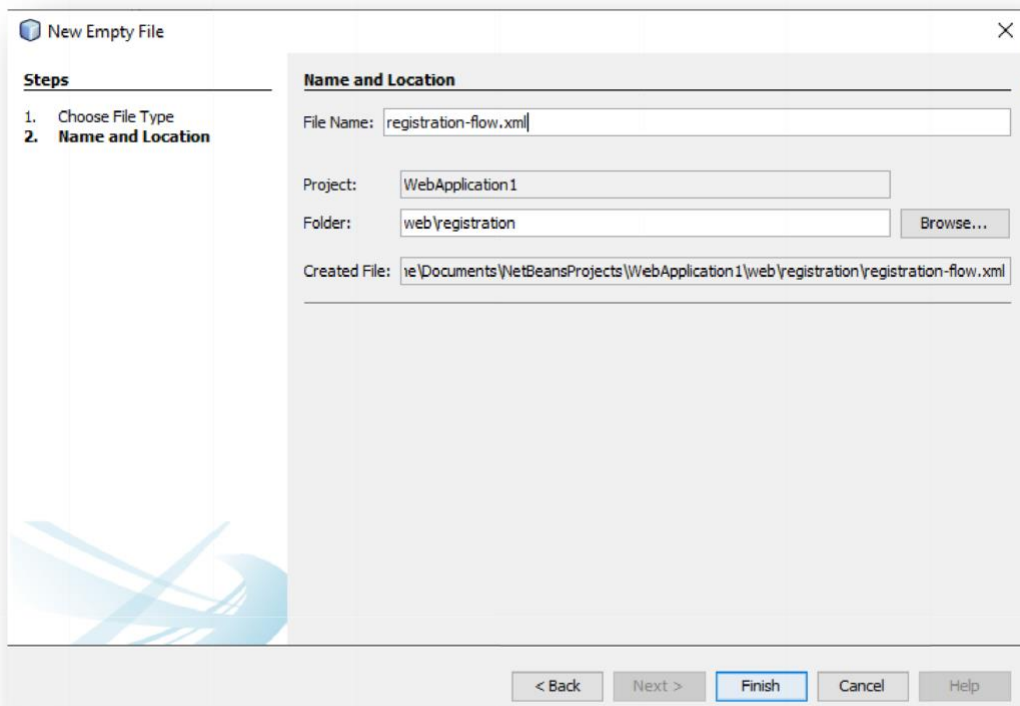
Zip

7 Les Flows faces

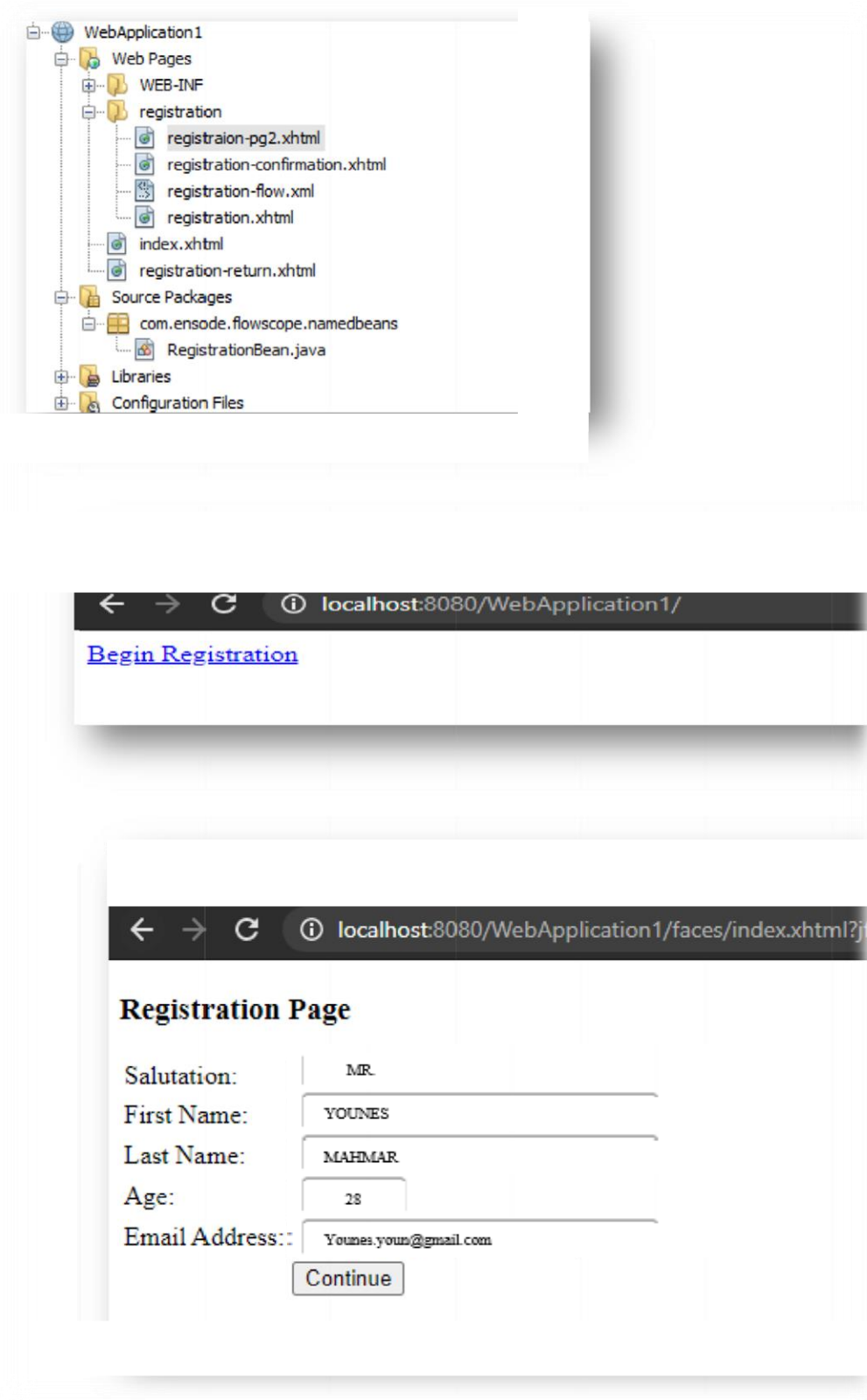
La première étape à faire et crée le répertoire registration qui va centenaire nous pages jsp.



Création de fichier de configuration.



Après la création des fichiers **registraion-pg2**, **registration-confirmation**, **registration-return** et le Bean **registrationBean** on exécute notre application.



On re marque que la navigation entre les pages marche bien.