

# Une introduction à JSF

---

## 1 Introduction

Le framework JSF (*Java Server faces*) a pour ambition de faciliter l'écriture d'applications WEB en offrant d'une part, une architecture MVC et d'autre part, une approche composant qui rappelle les interfaces graphiques natives. L'objectif est de remplacer l'approche « *je traite la requête HTTP et je renvoie une page HTML* » par « *j'envoie des composants (boutons, champs de texte, etc..) et je réagis aux actions de l'utilisateur* ».

JSF est composé de trois parties :

- une servlet générique qui va traiter les dialogues,
- une librairie de balises destinée à mettre en place les composants graphiques,
- une librairie de balises destinée à gérer ces composants et le dialogue avec le serveur.

Quelques liens :

**Site officiel de JSF** <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>

**API JSF 2.0** <https://jaserverfaces.java.net/nonav/docs/2.0/javadocs/>

**Tags JSF 2.0** [http://jaserverfaces.java.net/nonav/docs/2.0/vldocs/facelets/](http://jaserverfaces.java.net/nonav/docs/2.0/vlddocs/facelets/)

**myFaces : l'implantation libre proposée par Apache** <http://myfaces.apache.org/>

Des **cours** sur JSF :

- Un cours complet en français<sup>1</sup> (et pas seulement sur JSF),
- 176 très bons transparents en français sur JSF<sup>2</sup>,
- Une liste de cours sur le sujet<sup>3</sup>,
- Un site associé à un livre<sup>4</sup>.
- Des cours en ligne<sup>5</sup> (pas seulement sur JSF).

## 2 Mise en place

Suivez les étapes dans l'ordre pour obtenir une petite application JSF :

- Créez une application WEB dynamique avec **Eclipse JEE** et associez cette application à un serveur **TomEE**. Nous allons utiliser l'implantation MyFaces<sup>6</sup> de JSF intégrée à **TomEE**.
- Ajoutez à votre projet la facette **JavaServer faces** en version 2.2 (menu *Properties/Project Facets*). N'utilisez aucune librairie additionnelle (*Further configuration available*).
- Créez le fichier **WEB-INF/web.xml** listé ci-dessous :

1. <http://www.jmdoudoux.fr/java/dej/chap-jsf.htm#jsf>
2. <http://mbaron.developpez.com/javaee/jsf/>
3. <http://www.jsftutorials.net/>
4. <http://corejsf.com/>
5. <http://www.coreservlets.com/JSF-Tutorial/>
6. <http://myfaces.apache.org>

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  id="WebApp_ID" version="3.0">
  <display-name>JSF Application</display-name>
  <description>Exemple JSF</description>

  <!-- Déclaration de la servlet générique JSF -->
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <!-- URL traitées par cette servlet -->
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>/faces/*</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>*.xhtml</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>*.jsf</url-pattern>
  </servlet-mapping>

  <!-- Paramétrage du projet -->
  <context-param>
    <param-name>javax.faces.PROJECT_STAGE</param-name>
    <param-value>Development</param-value>
    <!-- <param-value>Production</param-value> -->
  </context-param>
  <context-param>
    <param-name>org.apache.myfaces.annotation.SCAN_PACKAGES</param-name>
    <param-value>monapp</param-value>
  </context-param>
</web-app>

```

- Créez (ou vérifiez) le fichier `WEB-INF/faces-config.xml` listé ci-dessous :

```

<?xml version="1.0" encoding="UTF-8"?>
<faces-config xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-facesconfig_2_2.xsd"
  version="2.2">

</faces-config>

```

- Créez la page JSP `WebContent/index.jsp` listée ci-dessous

```

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:redirect url="/hello.xhtml"/>

```

- Créez la page `WebContent/hello.xhtml` listée ci-dessous

```
<?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:f="http://java.sun.com/jsf/core"
  xmlns:h="http://java.sun.com/jsf/html">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Hello page</title>
</head>
<body>
  <f:view>
    <h1>
      <h:outputText value="Hello, world" />
    </h1>
  </f:view>
</body>
</html>
```

**Remarque :** Une page JSF (aussi appelée **Facelet**) est un document XML basé sur le langage XHTML. Les espaces de noms permettent identifier les balises que seront traitées avant l'envoi du résultat à l'utilisateur (dans notre exemple `<h:outputText .../>`).

**Remarque :** Vous pouvez accéder à cette page en utilisant les adresses : `/faces/hello.xhtml`, `/hello.xhtml` et `/hello.jsf`

**Remarque :** Il est à noter que la servlet récupère la main à **chaque requête** et prépare le contexte pour que la vue puisse produire le résultat.

## 2.1 Feuille de style

Dans JSF les ressources statiques (fichier CSS et/ou JavaScript) sont localisées dans des répertoires particuliers. Créez les répertoires

```
Racine de votre application
+ WebContent
|   + resources
|   |   + css
```

et créez le fichier `WebContent/resources/css/style.css`

```

h1 {
    border-bottom: solid 1px blue;
}

table {
    border-collapse: collapse;
    border: 1px solid black;
}

td, th {
    border: 1px solid black;
    padding: 3px;
}

#page {
    width: 800px;
    margin: 0 auto;
}

#header {
    border: solid 1px blue;
    padding: 5px;
    size: large;
    color: blue;
    background-color: #c8bfbf;
    margin-bottom: 20px;
}

#menu {
    float: right;
    width: 150px;
    border: solid 1px gray;
    padding: 5px;
    color: blue;
    background-color: white;
    margin-left: 20px;
}

#content {

}

#footer {
    border: solid 1px blue;
    padding: 5px;
    size: small;
    color: blue;
    background-color: #c8bfbf;
    margin-top: 20px;
}

.alert {
    color: red;
    border: solid 2px red;
}

```

Nous pouvons maintenant préparer une deuxième version de `hello.xhtml` qui utilise cette feuille de style :

```

<?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:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html">
<h:head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Hello page</title>
</h:head>
<h:body>
  <h:outputStylesheet library="css" name="style.css" />
  <f:view>
    <h1>
      <h:outputText value="Hello, world" />
      <h:graphicImage library="theme1" name="calanques.jpg" />
      <h:outputScript library="js" name="hello.js" />
    </h1>
  </f:view>
</h:body>
</html>

```

**Travail à faire :** Testez cette deuxième version et analysez le code source XHTML renvoyé par le serveur.

**Travail à faire :** En utilisant le tag `<h:graphicImage/>` introduisez une image dans votre page.

```

<h:graphicImage library="theme1" name="minimage.jpg" />

```

**Travail à faire :** Faites de même avec `<h:outputScript/>` et le script ci-dessous. Observez le code source généré.

```

alert('Hello');

```

## 2.2 Les templates JSF

La technologie JSF permet de définir un (ou plusieurs) gabarit(s) de page et de le réutiliser quand cela est nécessaire.

- Commencez par créer le fichier `WebContent/template/mylayout.xhtml` :

```

<?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:f="http://java.sun.com/jsf/core"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:c="http://java.sun.com/jsp/jstl/core"
  xmlns:ui="http://java.sun.com/jsf/facelets">
<h:head>
  <title><ui:insert name="title">My application</ui:insert></title>
</h:head>
<h:body>

  <h:outputStylesheet library="css" name="style.css" />

  <div id="page">
    <div id="header">
      <ui:insert name="header">
        <span>My application</span>
      </ui:insert>
    </div>
    <div id="menu">
      <ui:insert name="menu">
        <ui:include src="/template/menu.xhtml" />
      </ui:insert>
    </div>
    <ui:debug />
    <div id="content">
      <ui:insert name="content">
        <p>CONTENT</p>
      </ui:insert>
      <h:messages />
    </div>
    <div id="footer">
      <ui:insert name="footer">
        <span>(c) 2015</span>
      </ui:insert>
    </div>
  </div>
</h:body>
</html>

```

et WebContent/template/menu.xhtml :

```

<?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:f="http://java.sun.com/jsf/core"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:ui="http://java.sun.com/jsf/facelets">

<ui:composition>
  <div>Menu :</div>
  <ul>
    <li>lien 1</li>
    <li>lien 2</li>
  </ul>
</ui:composition>

</html>

```

Ceci est un exemple, cliquez sur le lien de téléchargement pour obtenir le cours complet.

