

Tutoriel 8: Les formulaires

L'objectif de ce tutoriel consiste à montrer comment récupérer par une Servlet les données d'un formulaire d'une page HTML, puis afficher ces données dans une autre. Il montre aussi via un simple exemple comment utiliser une feuille CSS pour mettre en forme une page web.

Pour plus de détail, n'hésitez pas à lire le tutoriel suivant: <http://sdz.tdct.org/sdz/creez-votre-application-web-avec-java-ee.html#IntroductionauJavaEE>

Etape 0: Création d'un nouveau projet

Si on n'a pas un projet web dynamique, on le crée selon les mêmes étapes vues dans le premier tutoriel, puis on ajoute ce projet au serveur Tomcat.

RQ1: Il ne faut pas oublier de modifier le *Build Path*.

Etape 1: Création de la première JSP et de la feuille CSS

Nous allons voir ici comment afficher le formulaire ci-dessous à l'aide d'une JSP, et comment le mettre en forme en utilisant une feuille CSS.



The screenshot shows a web browser window with the address bar displaying 'localhost:8081/formulaire/inscription'. The page content is titled 'Inscription' and contains the following text and form elements:

Vous pouvez vous inscrire via ce formulaire.

Adresse email *

Mot de passe *

Confirmation du mot de passe *

Nom d'utilisateur

1) Créer dans le répertoire WEB-INF, une JSP appelée *form.jsp*. Cette JSP contient le code suivant:

```
<%@ page pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Inscription</title>
    <link type="text/css" rel="stylesheet" href="form.css" />
  </head>
  <body>
    <form method="post" action="affichage">
      <fieldset>
        <legend>Inscription</legend>
        <p>Vous pouvez vous inscrire via ce formulaire.</p>
        <label for="email">Adresse email <span
class="requis">*</span></label>
        <input type="text" id="email" name="email" value=""
size="20" maxlength="60" />
        <br />
```

```

        <label for="motdepasse">Mot de passe <span
class="requis">*</span></label>
        <input type="password" id="motdepasse" name="motdepasse"
value="" size="20" maxlength="20" />
        <br />

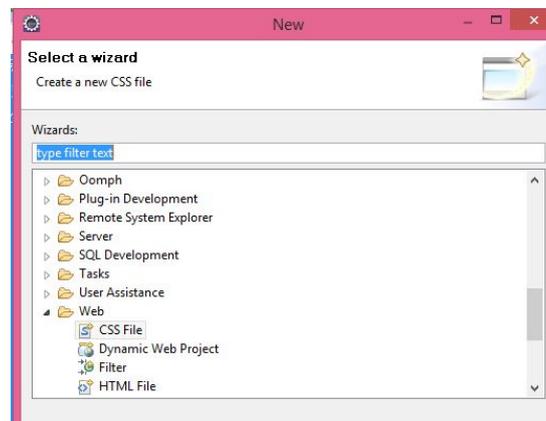
        <label for="confirmation">Confirmation du mot de passe <span
class="requis">*</span></label>
        <input type="password" id="confirmation" name="confirmation"
value="" size="20" maxlength="20" />
        <br />

        <label for="nom">Nom d'utilisateur</label>
        <input type="text" id="nom" name="nom" value="" size="20"
maxlength="20" />
        <br />

        <input type="submit" value="Inscription" class="sansLabel"
/>
    <br />
</fieldset>
</form>
</body>
</html>

```

2) Créer une feuille CSS appelé *form.css* dans le dossier *WebContent*. Pour cela, cliquer par bouton droit sur le dossier *WebContent*, puis sur *New*, puis sur *Other*. Puis cliquer sur *Web* puis sur *CSS File* (voir la figure ci-dessous).



3) Placer dans le fichier *form.css*, le code suivant:

```

/* Général -----
----- */

body, p, legend, label, input {
    font: normal 8pt verdana, helvetica, sans-serif;
}

fieldset {
    padding: 10px;
    border: 1px #0568CD solid;
}

legend {
    font-weight: bold;
    color: #0568CD;
}

```

```

/* Forms -----
----- */

form label {
    float: left;
    width: 200px;
    margin: 3px 0px 0px 0px;
}

form input {
    margin: 3px 3px 0px 0px;
    border: 1px #999 solid;
}

form input.sansLabel {
    margin-left: 200px;
}

form .requis {
    color: #c00;
}

```

Etape 2: Création de la première Servlet

1) Nous allons créer une servlet appelée *Form.java* dans un package appelé *servlets* par exemple (le package *servlets* doit être créé dans *src*). Cette servlet appelle le JSP précédent (*form.jsp*).

2) Remplacer la méthode *doGet* de classe *ServletRequest* par le code suivant:

```

protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    this.getRequestDispatcher( "/WEB-INF/form.jsp"
).forward( request, response );
}

```

3) Ajouter le code suivant au fichier *web.xml*

```

<servlet>
    <servlet-name>Inscription</servlet-name>
    <servlet-class>servlets.Form</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>Inscription</servlet-name>
    <url-pattern>/inscription</url-pattern>
</servlet-mapping>

```

L'appelle du servlet *Form.jsp* affiche le résultat ci-dessous:



Etape 3: Création de la deuxième Servlet

1) Nous allons créer une deuxième servlet appelée *RecupForm.java* dans le package *servlets*. Cette servlet permet de récupérer les données du formulaire précédent puis fait appel à une deuxième JSP appelé *affichage.jsp* pour afficher ces données. **La servlet *RecupForm.java* utilise la méthode *doPost* au lieu de *doGet*.**

2) Remplacer la méthode *doPost* de classe *RecupForm* par le code suivant:

```
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

    String email = request.getParameter( "email" );
    String motDePasse = request.getParameter("motdepasse");
    String confirmation = request.getParameter("confirmation");
    String nom = request.getParameter("nom");

    request.setAttribute( "emailAtt", email);
    request.setAttribute( "motDePasseAtt", motDePasse);
    request.setAttribute( "confirmationAtt", confirmation);
    request.setAttribute( "nomAtt", nom);

    this.getServletContext().getRequestDispatcher( "/WEB-
INF/affichage.jsp" ).forward( request, response );
}
}
```

3) Ajouter le code suivant au fichier web.xml

```
<servlet>
    <servlet-name>RecupForm</servlet-name>
    <servlet-class>servlets.RecupForm</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>RecupForm</servlet-name>
    <url-pattern>/affichage</url-pattern>
</servlet-mapping>
```

Etape4: Création de deuxième JSP

1) Nous allons créer dans le répertoire WEB-INF, une JSP appelée *affichage.jsp*. Cette JSP contient le code suivant:

```
<%@ page language="java" contentType="text/html; charset=windows-1256"
pageEncoding="windows-1256"%>
```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1256">
<title>Insert title here</title>
</head>
<body>

    <%
String email = (String) request.getAttribute("emailAtt");
out.println( email );
String motDePasse = (String) request.getAttribute("motDePasseAtt");
out.println( motDePasse );
String confirmation = (String) request.getAttribute("confirmationAtt");
out.println( confirmation );
String nom = (String) request.getAttribute("nomAtt");
out.println( nom );
    %>
</body>
</html>

```

Etape5: Exécution

Si tout va bien, on obtient le résultat ci-dessous. Pour cela, vous allez directement exécuter la *Servlet Form.java* à partir d'eclipse ou à partir d'un navigateur, en utilisant l'url suivant :

<http://localhost:8081/formulaire/inscription>

The screenshot shows a web browser window with the address bar displaying 'localhost:8081/formulaire/inscription'. The page content is titled 'Inscription' and includes the text 'Vous pouvez vous inscrire via ce formulaire.' Below this, there are four input fields: 'Adresse email *' containing 'adr@yahoo.com', 'Mot de passe *' containing three dots, 'Confirmation du mot de passe *' containing three dots, and 'Nom d'utilisateur' containing 'adr'. An 'Inscription' button is located at the bottom right of the form.

Si on remplit les champs du formulaire, puis en clique sur le bouton *Inscription*, on obtient le résultat suivant:

The screenshot shows a web browser window with the address bar displaying 'localhost:8081/formulaire/affichage'. The page content displays the submitted data: 'adr@yahoo.com 123 123 adr'.