

Exercice 6 Servlet Simple

Créez un nouveau projet Web Application ServletSimple.

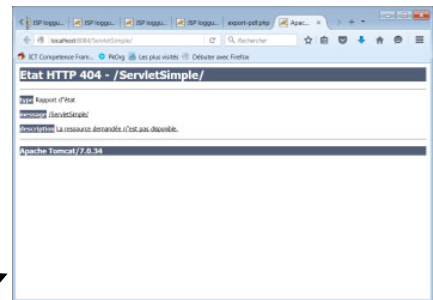
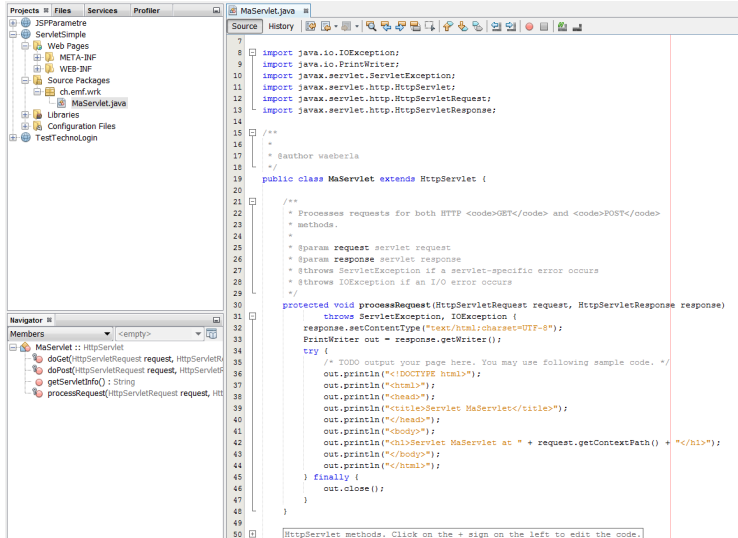
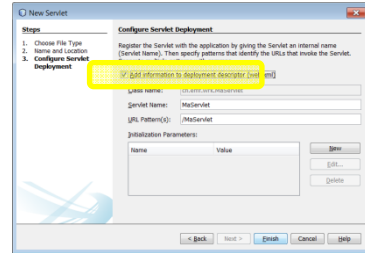
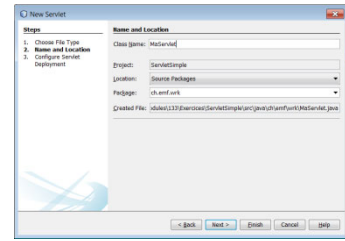
Supprimer index.html

Etape 1 Servlet simple

Créez une Servlet avec New -> Servlet dans le package ch.emf.m133

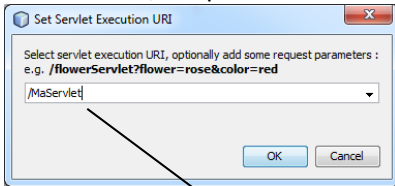
s'appelant MaServlet.

Ensuite n'oubliez pas de cocher Add information

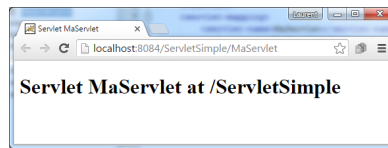


Modifier le texte pour le personnaliser et exécutez, ce qui donnera

Pour lancer la Servlet directement depuis NetBeans, sélectionnez la Servlet dans le package source, puis click droit et "Run File", ce qui devrait donner:



puis



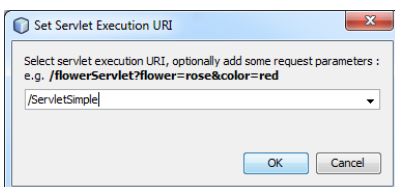
/MaServlet vient du fichier web.xml du dossier "Configuration Files"



Si on change le fichier web.xml balise <url-pattern>/MaServlet par /ServletSimple



, il faut donc changer également l'URI au lancement



ce qui donnera



Etape 2 Appel d'une servlet depuis une page Jsp

Dans le même projet, ajouter une page de login (idem exercice ultérieur) et password

```

1 <!--
2 Document : index
3 Created on : 10 mars 2016, 07:23:00
4 Author : waeberla
5 -->
6
7 <%%page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>JSP Page d'accueil</title>
13 </head>
14 <body>
15 <h1>Ma Jsp d'accueil!</h1>
16 <p>
17 Texte d'introduction...
18 </p>
19 <form method="post" action="/ServletSimple/MaServlet">
20 <input type="text" name="login" id="login" size="20" maxlength="50" />
21 <input type="password" name="password" id="password" size="20" maxlength="50" />
22 <input type="submit" value="Log in" />
23 </form>
24 </body>
25 </html>
26
27
28

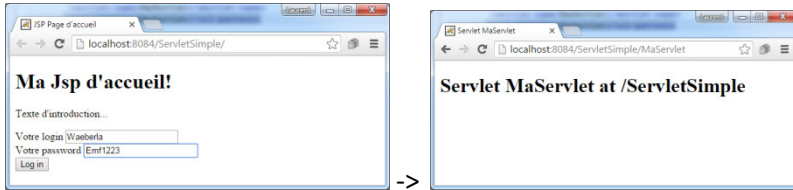
```

Appel de la Servlet depuis le formulaire

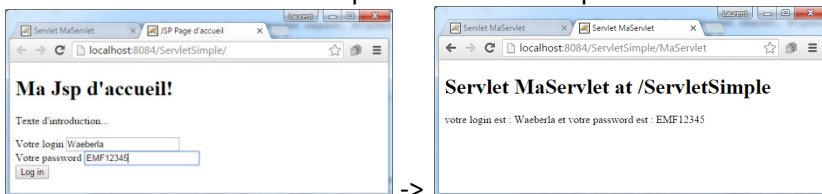
pour autant que dans le fichier web.xml, la balise `<url-pattern>/MaServlet</url-pattern>` pointe bien sur ce qui est écrit dans la Jsp.

Dans l'action du formulaire, on donne le contexte (`/ServletSimple`) puis l'url de la servlet que l'on veut lancer.

Exécutez =>



Maintenant il faut utiliser les paramètres de la Jsp dans la Servlet



Vous remarquerez que le code html de la servlet n'est pas exceptionnellement agréable à écrire et à lire!!!

```

public class MaServlet extends HttpServlet {
    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String login = request.getParameter("login");
            String password = request.getParameter("password");
            /* TODO output your page here. You may use following sample code. */
            out.println("<DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet MaServlet</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet MaServlet at " + request.getContextPath() + "</h1>");
            out.println("<p>votre login est : " + login + " et votre password est : " + password);
            out.println("</body>");
            out.println("</html>");
        }
    }
}

```

Par contre la récupération des paramètres est relativement simple et nous sommes dans du "PUR JAVA".

Conclusion

Nous arrivons donc à la conclusion que les pages Jsp sont idéales pour l'affichage et les Servlets idéales pour le code. Nous retrouvons donc quelque chose de connu, le MVC. **Différences des méthodes doGet et doPost ?**

Documentation :

Dans le chapitre <<Annexes>> de votre documentation de projet, votre travail doit y être intégré et expliqué selon l'exemple que vous avez réalisé.