**Travaux pratiques**

Cette fiche d'exercices aborde les Servlets. Les exercices sont réalisés avec environnement de développement (Netbeans 7, Tomcat 6, Java EE 5).

## Structure d’une application web

Une application web adopte une structure précise :

1. Elle est contenue dans un répertoire portant le nom de l'application (par exemple, exe1).

Ce répertoire possède un sous-répertoire WEB-INF qui contient les ressources composant l'application et non fournies directement au client.

1. Ce répertoire WEB-INF contient deux sous-répertoires : classes et lib. Le répertoire classes stocke les classes Java. Lors du développement on peut y inclure également les sources Java. Le répertoire lib contient les librairies nécessaires comme des connecteurs de base de données (JDBC, ODBC, etc.). On a donc la structure minimale ci-dessous :

exe1

exe1/WEB-INF

exe1/WEB-INF/classes

exe1/WEB-INF/lib

1. Les formulaires statiques html sont ajoutées sous le répertoire exe1.
2. Le répertoire exe1/WEB-INF doit contenir un fichier « web.xml » décrivant

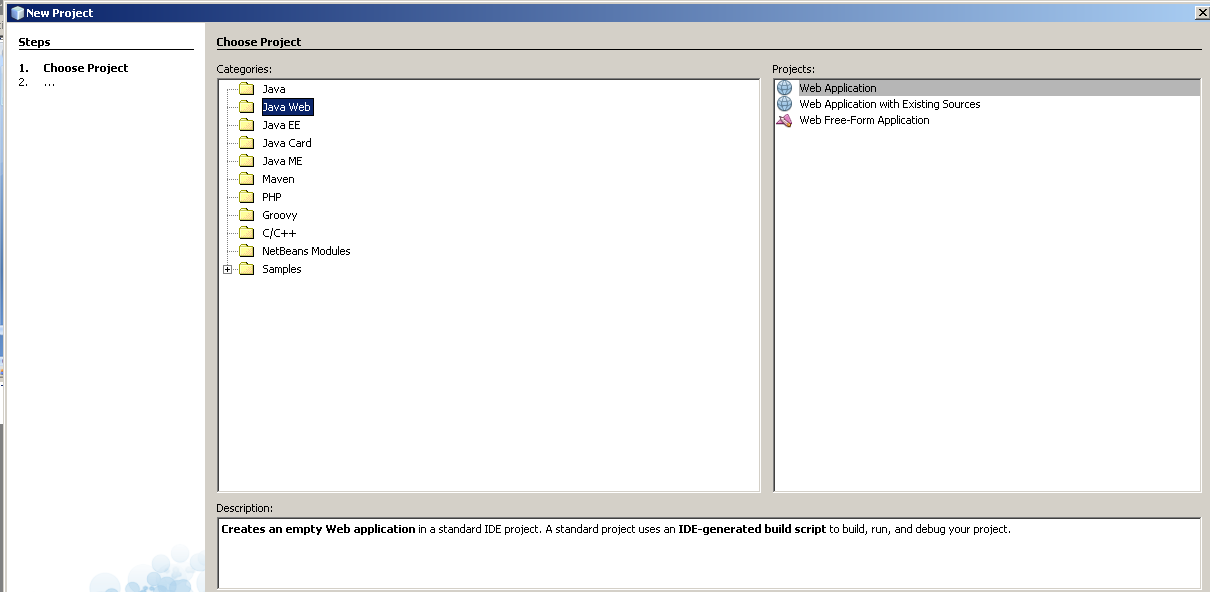
le déploiement de l'application.

# Exercice 1 : Première servlet

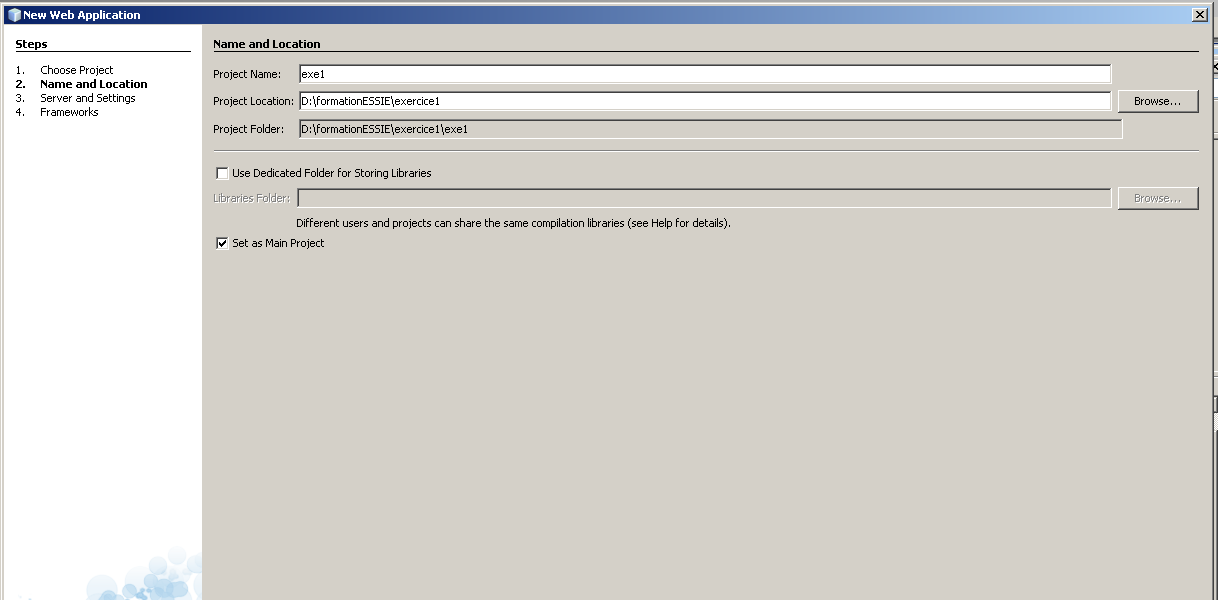
Prise en main Netbeans [et Tomcat](http://www.eclipsetotale.com/tomcatPlugin.html).: création d' une servlet **BonjourServlet** qui affiche la chaine de caractère : "Bonjour".

1. créer un nouveau projet appelé ex1 avec le serveur web Tomcat (dans un nouveau répertoire de nom exe1)

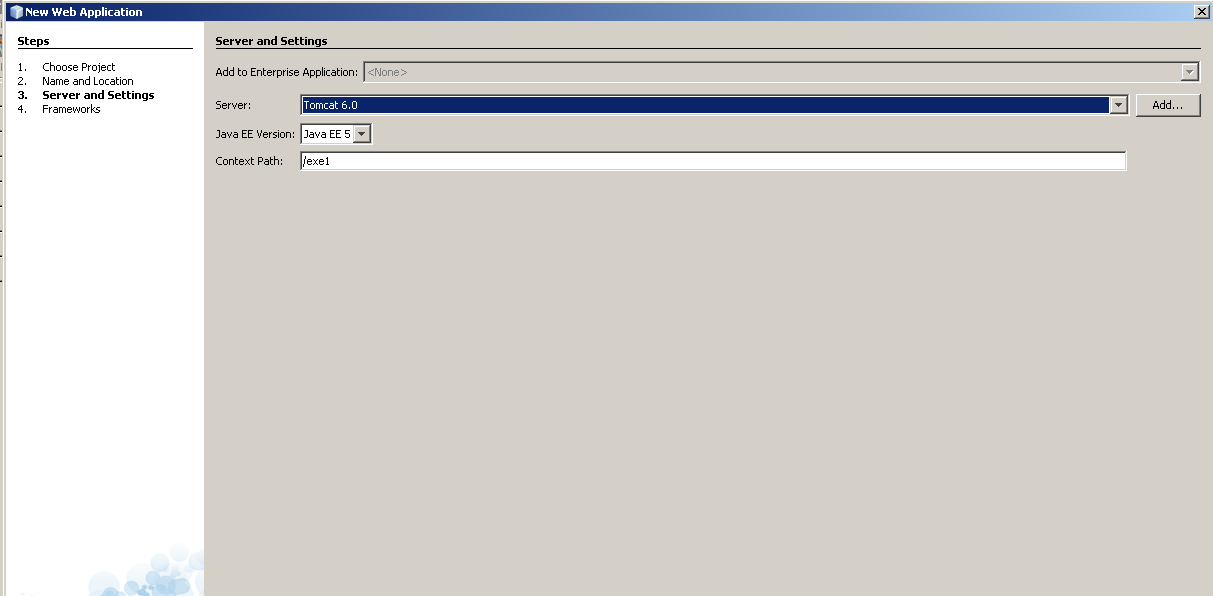
File->New Project



Java Web 🡪 Web Application



Sélection : serveur Tomcat 6 et Java EE 5



1. créer (File->New->File) le fichier web.xml minimaliste dans le répertoire WEB-INF à partir de ce [**fichier**](exercice1/ex1/web.xml) :
2. créer (File->New->Package) le package servlet dans exe1/src/java
3. créer (File->New->Class) la classe **BonjourServlet** dans exe1/src/java /servlet (préciser sa superclasse javax.servlet.http.HttpServlet)
4. implémenter la méthode doGet().

**Etape 1 :**

Pour tester l'application, cliquer sur l'URL suivante : http://localhost:8083/exe1/bonjour

**Etape 2 :**

Reprendre le code de l'étape 1 et ajouter des balises HTML afin de mettre en forme le texte affiché par la servlet.

# Exercice 2 : Servlet traitant les paramètres d'une requête

Créer la servlet **LoginServlet** (dans le package servlet). Cette servlet récupère 2 paramètres *login* et *password* et répond en créant dynamiquement une page HTML contenant la chaîne :

**Hello login :xx ; password : yy**

Pour tester l'application, cliquer sur l'URL suivante : [http://localhost:8080/exe1/login?login=Jules&password=Verne](http://localhost:8080/exe1/login?login=Jules&password=Verne%20)

# Exercice 3 : Servlet traitant les paramètres d'une formulaire html statique

Créer la servlet **LoginServletFormulaire** (dans le package servlet). Cette servlet récupère 2 paramètres *login* et *password* dans une formulaire de saisie LoginServlet.html et répond en créant dynamiquement une page HTML contenant la chaîne :

**Hello login :xx ; password : yy**

# Créer le formulaire LoginServlet.html à partir de ce [fichier](../enoncé/LoginServlet.html)1.

# Pour saisir les login et password dans le formulaire LoginServlet, Saisir dans le navigateur ie l’URL  <http://localhost:8083/exe1/LoginServlet.html>

# Exercice 4 : Liste arbitraire des paramètres d’invocation de la servlet avec leur numéro d’ordre

Ecrire une servlet (appelons-la ParamServlet) créant une page

HTML qui listera un nombre arbitraire de paramètres d’invocation de la servlet avec leur numéro d’ordre ; exemple

# <http://localhost:8083/exe1/ParamServlet?toto:EIVD&tata:TCOM&tutu:HES-SO>

