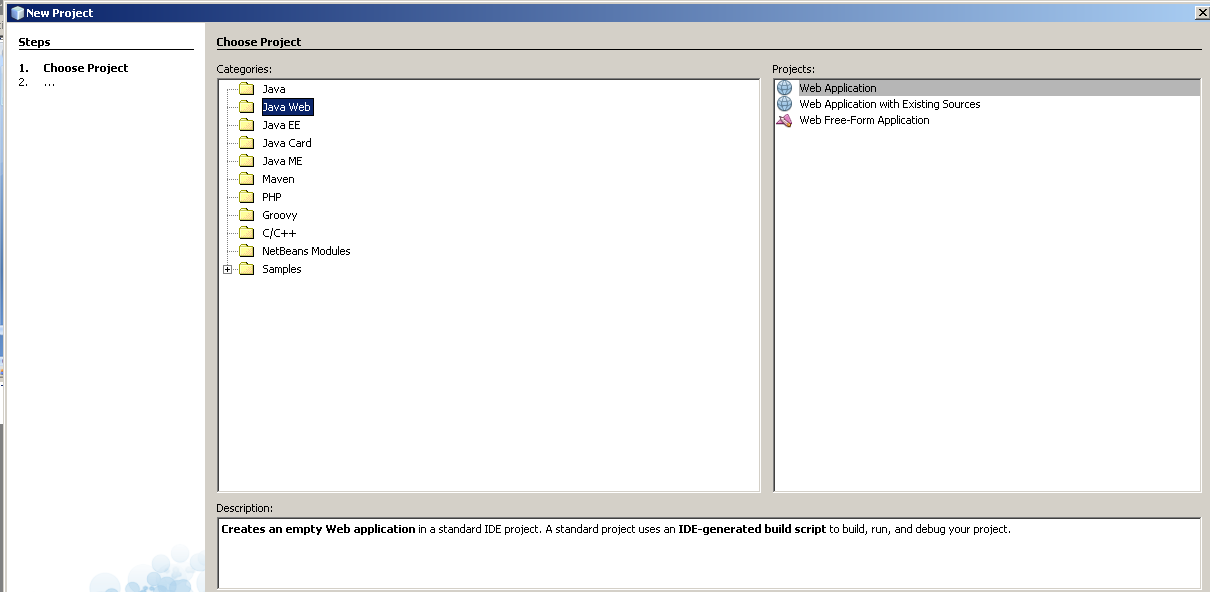
**Travaux pratiques**

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

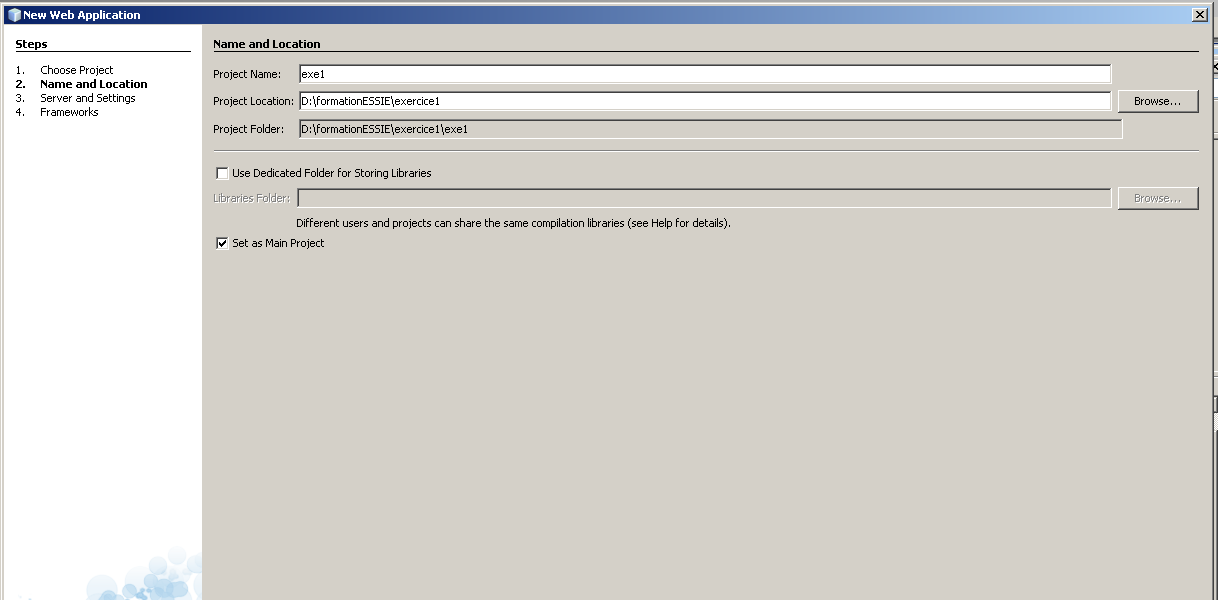
# Exercice 1 : Création d’un projet web

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

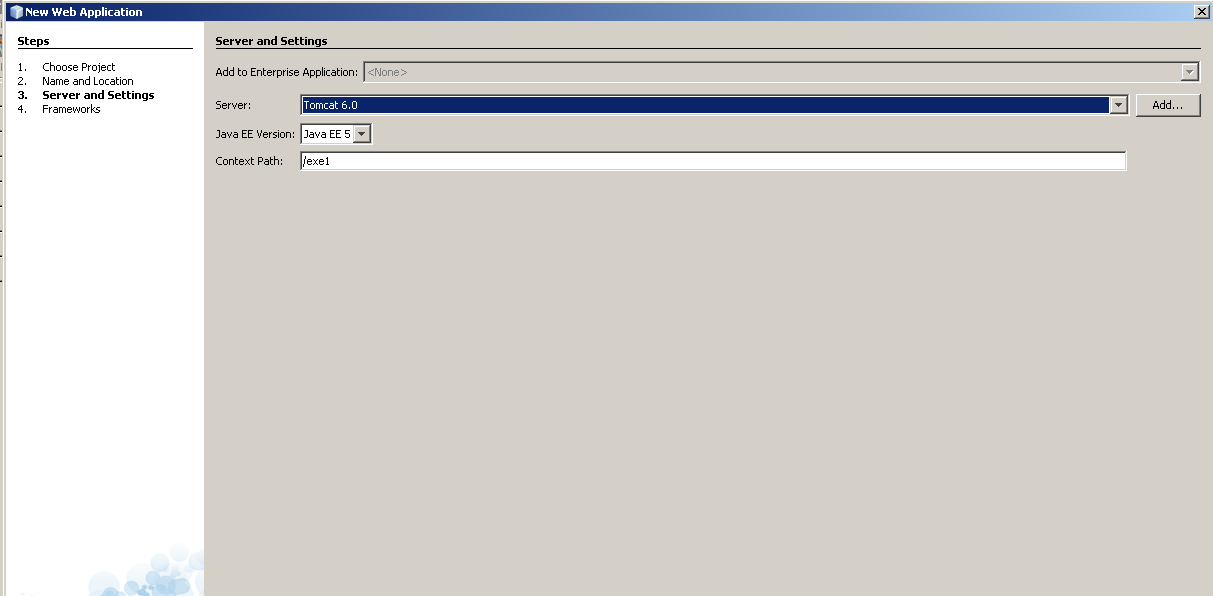
File->New Project



Java Web 🡪 Web Application



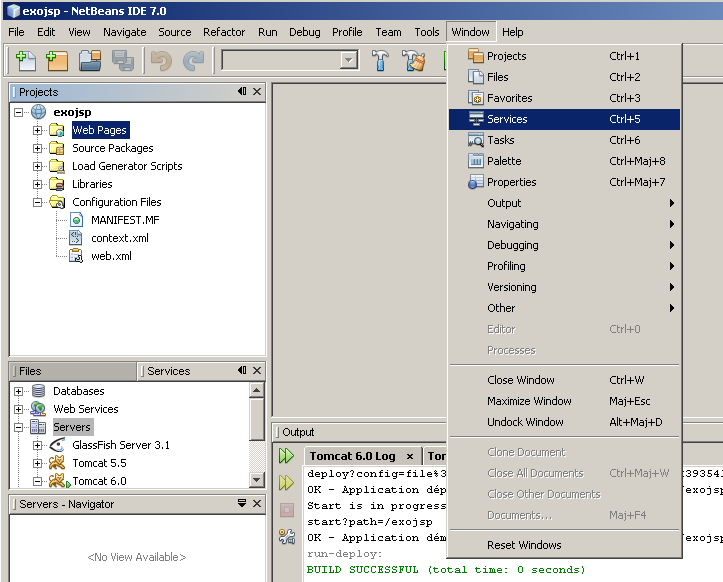
Sélection : serveur Tomcat 6 et Java EE 5



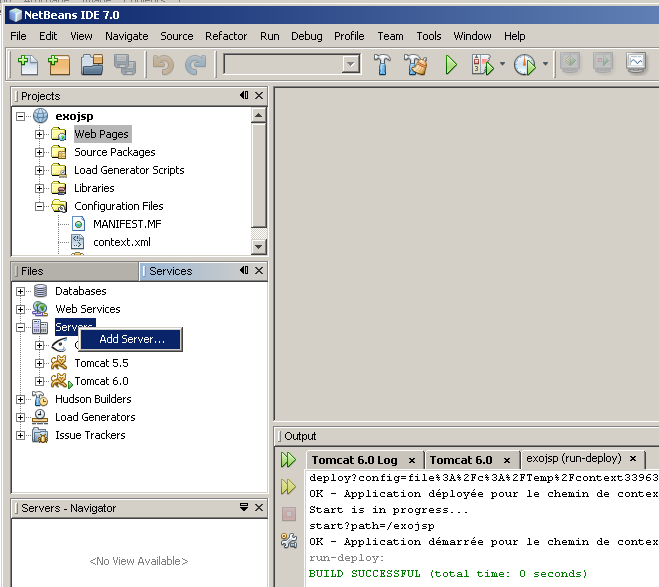
**Etape 1 : Tester l’application**

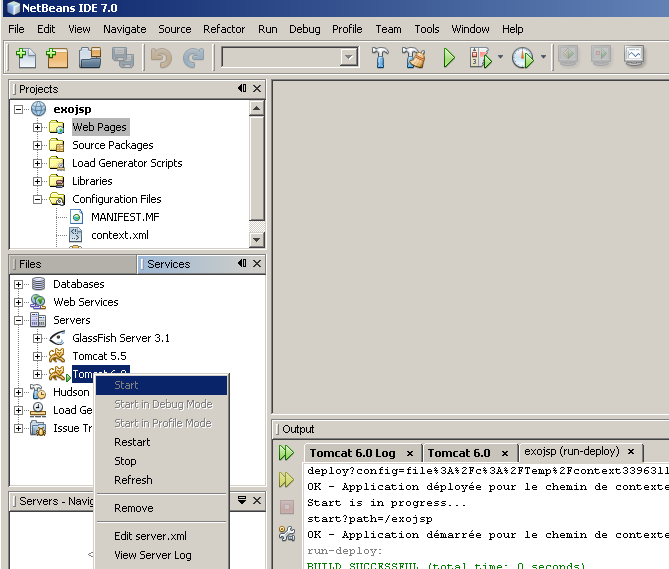
Démarrer Tomcat :

* dans le menu sélectionner : Windows/services



* dans la fenêtre services : sélectionner la ressource « server »
* ajouter tomcat (add server), et démarrer tomcat (start).





Compiler le projet exe1 : dans le menu sélectionner « Run/Clean and Build MainProject »

Déployer le projet exe1 : clic droit sur le projet avec le souris et sélectionner « deploy »

Cliquer sur l'URL suivante : <http://localhost:8083/exe1/index.jsp> et vérifier l’affichage de **Hello World !**

**Etape 2 :**

Tous les exercices de ce TP supposent l'existence du fichier «commun.jsp» suivant :

|  |
| --- |
| <%!  String entete(String titre) {  return "<HTML><HEAD>\n<TITLE>Premiers exercices de JSP : "  +titre  +"</TITLE>\n</HEAD><BODY>\n"  +"<H1>Premiers exercices de JSP&nbsp;: "+titre+"</H1>";  }  String pied() {  return "</BODY></HTML>";  }  %> |

Créer le fichier JSP « commun.jsp » dans le répertoire web du projet exe1:

* sélectionner le répertoire « Web Page » et avec le clic droit de la souris sélectionner (New->JSP) et nommer le JSP : « commun.jsp».
* copier/coller le code ci-dessus dans commun.jsp.
* Supprimer le code html du fichier index.jsp, et déclarer dans index.jsp le fichier « commun.jsp » en inclusion statique.
* Utiliser « les JSP Expressions » :
  + pour utiliser la méthode « String entete(String titre) » et afficher le titre « Etape2 »
  + pour insérer dans index.jsp la méthode « String pied() ».
* Ajouter dans commun.jsp, une méthode globale qui retourne l’id de la session et vous la nommer: String getSessionId() ;
* Utiliser la méthode getSessionId() dans index.jsp pour afficher l’id de la session.

# Exercice 1 : Réalisation de scriptlets, utilisation des variables request et out

Écrire un formulaire qui demande le nom et l'âge de l'utilisateur. Le bouton *submit* de ce formulaire provoquera l'affichage d'une page qui saluera l'utilisateur avec cette phrase : «Bonjour nom\_*xx*, vous avez *yy* ans…».

# Exercice 2 :

Écrire un formulaire qui demande le nom et le sexe de l'utilisateur (M ou Mme). Ce formulaire appelle une page qui affichera «Bonjour monsieur Truc» ou «Bonjour madame Bidule» suivant le cas.

# Exercice 3 :

Écrire un formulaire «calculatrice» : 2 cases pour la saisie des opérandes, un groupe de 4 cases à cocher (ou une liste déroulante) pour le choix de l'opération, et affichage du résultat de l'opération.

**Exercice 4 : Mise en session d’une liste d’utilisateurs**

Les expressions JSP permettent d'extraire ou de calculer des valeurs dynamiquement pour les insérer dans la page JSP. Dans cet exercice une expression extrait un utilisateur (nom, prénom, login, mot de passe) d’une entité Bean utilisateur et l'insère dans une liste.

Pour visualiser la liste des utilisateur, la liste est mise dans une variable de session extrait de la variable de session.

* 1ière jsp : exo4.jsp
  + formulaire et ajout d’un utilisateur : à chaque ajout d’un user, le user est ajouté dans une liste et ensuite la liste est remise en session.
* 2ième jsp : exo4Bis.jsp
  + le lien html «Visualiser les users » forwarde la demande d’affichage vers le 3ième jsp qui traite l’affichage de la liste des utilisateurs extraite de la session.
* 3ième jsp : exo4-1.jsp : extrait de la session la liste des utilisateurs ajoutés et affiche la liste

**Exo4.jsp :**

|  |  |
| --- | --- |
| **Ajouter un user** | |
| **nom** |  |
| **prénom** |  |
| **login** |  |
| **mot de passe** |  |
| Vider  Ajouter  **Visualiser users** | |

**Exo4Bis.jsp :**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ include file="commun.jsp"%>

<%=entete("4ième exercice")%>

<jsp:forward page="exo4-1.jsp"/>

<%=pied()%>

**Exo4-1.jsp : récupère la liste de la session et affiche le tableau suivant :**

|  |  |  |  |
| --- | --- | --- | --- |
| **nom** | **prénom** | **login** | **Mot de passe** |
| NOM\_1 | PRENOM\_1 | NOM\_PRENOM\_1 | MDP\_1 |
| NOM\_2 | PRENOM\_2 | NOM\_PRENOM\_2 | MDP\_2 |
| … | … | … | … |

**Exercice 5 : JSP et JavaBeans**

Utiliser Java Bean pour afficher « Hello world ».

* Créer une classe java et la nommer exo5.MyBean avec un seul attribut : name de type String
* Créer un jsp et le nommer : exo5.jsp

Insérer dans exo5.jsp :

<jsp:useBean id= « mybean » class= “exo5.MyBean » scope= “session”>

<jsp:setProperty name=”mybean” property=”name” value=”Hello world” />

</jsp:useBean>

**Exercice 6 : JSP cookies**

1 - Ecrire une jsp qui :

* Affiche la liste des n cookies reçues par le serveur
* Ajoute un cookie Cookie+1 ayant la valeur n+1

**Exercice 7 : Travail à préparer pour le prochain TP**

**Mettre en place un compteur pour compter le nombre d’accès http.**

Construire une jsp qui affiche un compteur qui est incrémenté par un lien, à chaque accès sur ce jsp pendant une session. En fait ce compteur est dans l'HttpSession.

• Indication : on pourra utiliser la classe Integer qui modélise le type int.

• Faire passer le session id de sorte à le récupérer lors de la prochaine interaction



**Optionnel : Récupérer id session par le cookie**

Par défaut le session id est passé comme cookie du serveur au client web (qui le lui renvoie à chaque requête cf. gestion des cookies), à condition que le client web autorise les cookies,

***Remarque importante*** : il faut dans le navigateur activer les cookies