TD 1- TME : Servlets TOMCAT

1.1 A LIRE - Minimum vital pour la réalisation du projet

Dans ce TME, nous allons configurer l'environnement éclipse et créer notre premier servlet. Pour la semaine suivante, il est obligatoire de s'assurer que :

- Eclipse Java EE Developper Tools est bien installé.
- Vous pouvez accéder à Tomcat
- Vous avez compris comment implémenter un servlet (Hello World)
- Vous savez déployer un servlet sur Tomcat
- Vous savez interroger le servlet sur un navigateur web

1.2 Prise en main de Eclipse

Prérequis pour le TP :

- Lancer Eclipse
- Regarder si il est possible de créer un nouveau "dynamic Web Project". Si ce n'est pas le cas :
  - Allez dans "Help"->"Install New Software"
  - Chosir "All available sites", puis taper "EE" dans la barre de recherche (et attendre)
  - Installer "Eclipse Java EE Developper Tools"
  - Redémarrer Eclipse, et ca devrait fonctionner
- Si problème : supprimer le répertoire ".eclipse" de votre compte, et recommencez.

Lors de la création d'un projet :

- Assurez vous que le projet est configuré pour utiliser JAVA 1.6 (Bouton droit, Properties, Java Compiler)
- Ajouter le fichier Attach :servlet-api.jar dans le dossier WebContent/WEB-INF/lib. Ce fichier permet d'avoir accès aux différentes classes des servlets. Pour cela, il faut télécharger le fichier et faire un "Glisser" du fichier dans le projet eclipse
- Créez un fichier web.xml dans WebContent/WEB-INF. Conseil : Afin d'éviter les erreurs, vous pouvez faire un copier/coller de web.xml depuis le site suivant : `http://lfe.developpez.com/Java/TomCat/?page=page_4`

Question 1. Télécharger le servlet HelloWorld.java et l'importer sous votre projet Eclipse (adapter le code si besoin). Comprendre les principaux éléments du code.

Question 2. Ecrivez le fichier de routage correspondant

Question 3. Quelle est la procédure pour déployer cette servlet sur TOMCAT ? Importer le servlet sur Tomcat et afficher son résultat sur le navigateur.

1.3 Un premier servlet

Nous allons nous intéresser à un premier servlet permettant d'effectuer une addition. Le servlet prendra deux paramètres en entrée a et b et retournera (sous forme $plain/text$) la réponse de $a+b$

Question 1. Implémentez l'applet en question

Question 2. Ecrivez le fichier de routage correspondant

Question 3. Etendez la servlet précédente pour qu'elle prenne en paramètre le type d'opération (*addition*, *multiplication* ou *division*) et qu'elle effectue l'opération demandée. En cas d'erreur, la servlet renvoie un contenu vide.

Question 4. Afin de pouvoir tester la servlet sans utiliser TOMCAT, il nous faut séparer le traitement de la servlet.
Dans le package `services`, créez une classe `Operation` qui contient les trois fonctions de calcul (`addition`, `multiplication` ou `division`) ainsi qu'une méthode `double calcul(double a,double b,String operation)`.

Implémentez une fonction `main` permettant le test de cette fonction. Implémentez la servlet correspondante. **C'est la procédure que vous suivrez lors du développement de vos servlets**

1.4 Concurrence dans les servlets

Question 1. Soit la servlet suivante :

```
import java.io.IOException;
import javax.servlet.*;
import javax.servlet.http.*;
import java.math.*;
public class SimpleServlet extends HttpServlet
{
 private int counter = 0;
 public void doGet(HttpServletRequest req, HttpServletResponse resp)
  throws ServletException, IOException
 {

  resp.getWriter().println("<HTML><BODY>");
  resp.getWriter().println(this + ": <br>");
  for (int c = 0; c < 10; c++)
  {
   resp.getWriter().println("Counter = " + counter + "<BR>");
   counter++;
  }
  resp.getWriter().println("</BODY></HTML>");
 }
}
```

Donnez la sortie de la servlet après sont premier appel

Question 2. Donnez la sortie de l'applet après sont deuxième appel

Question 3. Que se passe-t-il si la servlet est appelée au même moment par deux clients différents ?

Question 4. Voila une nouvelle servlet...

```
import java.io.IOException;
import javax.servlet.*;
import javax.servlet.http.*;
import java.math.*;
public class SimpleServlet extends HttpServlet
{
 //A variable that is NOT thread-safe!
 private int counter = 0;
 private String mutex = "";
 public void doGet(HttpServletRequest req, HttpServletResponse resp)
  throws ServletException, IOException
 {
   resp.getWriter().println("<HTML><BODY>");
  resp.getWriter().println(this + ": <br>");
  synchronized (mutex)
  {
   for (int c = 0; c < 10; c++)
   {
    resp.getWriter().println("Counter = " + counter + "<BR>");
    counter++;
   }
  }
  resp.getWriter().println("</BODY></HTML>");
 }
```

```
}
```

Donnez la sortie si deux servlets sont exécutées quasiment en même temps

Question 5. Quel est l'inconvénient d'une telle programmation ?

1.5 Notions de base à savoir - voir les supports de cours pour les réponses

Question 1. Qu'est-ce que TOMCAT ?

Question 2. Qu'est-ce qu'une servlet ?

Question 3. Qu'est-ce que le mapping d'URL dans TOMCAT ?

Question 4. La classe Servlet de TOMCAT est définie ainsi :

```
public abtract class HttpServlet extends GenericServlet
{
      HttpServlet()

      // m\'ethodes r\'epondant aux diff\'erents types de requêtes
      protected void doDelete(HttpServletRequest req, HttpServletResponse resp)
      protected void doGet(HttpServletRequest req, HttpServletResponse resp)
      protected void doHead(HttpServletRequest req, HttpServletResponse resp)
      protected void doOptions(HttpServletRequest req, HttpServletResponse resp)
      protected void doPost(HttpServletRequest req, HttpServletResponse resp)
      protected void doPut(HttpServletRequest req, HttpServletResponse resp)
      protected void doTrace(HttpServletRequest req, HttpServletResponse resp)

      // permet d'utiliser le cache côt\'e client
      protected long getLastModified(HttpServletRequest req)

      // g\'erent le dispatching en fontion du type de requête}
      protected void service(HttpServletRequest req, HttpServetResponse resp)
      protected void service(ServletRequest req, ServletResponse res)
}
```

Rappelez l'usage des différentes méthodes

Question 5. Quelle est la méthode qui nous concerne ? En décrire les paramètres

Question 6. Quel est le principe de protocole GET ?