

Contents

TD 1 : Servlets TOMCAT	1
1.1 Questions de Cours (sur feuille à part à partir du TD 2)	1
1.2 Un premier servlet	2
1.3 Concurrence dans les servlets	5
1.4 Spécifications du projet	6

TD 1 : Servlets TOMCAT

1.1 Questions de Cours (sur feuille à part à partir du TD 2)

Question 1. Rappelez l'architecture matérielle/logicielle du projet à réaliser

Question 2. Qu'est-ce que TOMCAT ?

Question 3. Qu'est-ce qu'une servlet ?

Question 4. Qu'est-ce que le mapping d'URL dans TOMCAT ?

Question 5. La classe Servlet de TOMCAT est définie ainsi:

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

    // méthodes répondant aux différents 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 côté client
    protected long getLastModified(HttpServletRequest req)

    // gère le dispatching en fonction du type de requête}
    protected void service(HttpServletRequest req, HttpServletResponse resp)
    protected void service(ServletRequest req, ServletResponse res)
}
```

Rappelez l'usage des différentes méthodes

Question 6. Quelle est la méthode qui nous concerne ? En décrire les paramètres

Question 7. Quel est le principe de protocole GET ?

1.2 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. Quelle est la procédure pour déployer cette servlet sur TOMCAT ? Quels en sont les inconvénients ?

Question 5. 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**

1.3 Concurrency 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 son premier appel

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

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

Question 4. Voici 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 ?

1.4 Spécifications du projet

Nous allons commencer à réfléchir aux différents services que nous prévoyons d'implémenter dans le projet. **Les services seront plus amplement décrits lors du Cours 2.** Basiquement, un service est une méthode qui prend des paramètres et renvoie une ou plusieurs valeurs. Les spécification de chacun des services sera faite préalablement à son implémentation sous la forme suivante:

Nom du web service	
Description du service	
Paramètres en entrée	
Format de sortie	
Exemple de sortie	
Erreurs possibles	
Avancement du Service	
Classes JAVA en rapport avec le Web service	
Informations additionnelles	

La spécification des services sous cette forme sera **préalable à leur implémentation**. Chaque binôme devra toujours maintenir à jour un fichier/cahier décrivant les services implémentés/à implémenter que les enseignants pourront consulter à la demande.

Question 1. Spécifiez le service permettant d'enregistrer un nouvel utilisateur

Question 2. Spécifiez le service permettant de se logger sur le site.

Question 3. Ecrivez une méthode java qui permette de générer automatiquement une chaîne de caractères aléatoires (entre A-Z et 0-9) de taille définie

Question 4. Spécifiez le service permettant de se délogger du site