

# TP 1 Servlets

Matthieu Foucault  
matthieu.foucault@labri.fr  
<http://www.labri.fr/~mfoucaul>

11 février 2013

## Q1.1 Installation de Tomcat

- Accédez à votre machine virtuelle se trouvant sur morpheus via ssh.
- Assurez vous que le SDK Java est bien installé sur votre machine, en lançant la commande `javac -version`
- Téléchargez Tomcat à l'adresse suivante :  
`http://mirrors.linsrv.net/apache/tomcat/tomcat-7/v7.0.35/bin/apache-tomcat-7.0.35.tar.gz`
- Extraire l'archive tar.gz
- Editez le fichier `conf/server.xml` pour que Tomcat utilise le port 80 au lieu de 8080.
- Démarrez le serveur en exécutant le script `bin/startup.sh`
- Vérifiez que le serveur est bien accessible en entrant, dans la barre d'adresse de votre navigateur, l'adresse :  
`http://{votre-login}.rmorpheus.enseirb.fr`
- Vous pouvez stopper le serveur avec le script `bin/shutdown.sh`.

## Q1.2 Votre premier Servlet : Hello World!

Il faut tout d'abord créer l'arborescence des dossiers : à la racine du serveur Tomcat se trouve un dossier `webapps`. Dans ce dossier, créez un répertoire `hello`. Les fichiers de ce répertoire seront accessibles via `http://{votre-login}.rmorpheus.enseirb.fr/hello`.

Dans le dossier `hello`, créez un dossier `WEB-INF`. C'est ici que seront situées les informations liées à la configuration de l'application.

Dans le dossier `WEB-INF`, ajoutez un dossier `classes`. Comme son nom l'indique, c'est ici que devront être déployées les différentes classes utilisées par l'application.

Il est temps de déployer votre premier servlet : ajoutez le code suivant dans un fichier `HelloServlet.java` :

```
package fr.enseirb.servlets;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.*;

@WebServlet(urlPatterns={"/servlet"})
public class HelloServlet extends HttpServlet
{
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException
    {
        resp.setContentType("text/html");
        PrintWriter out = resp.getWriter();

        out.println("<html>");
        out.println("<body>");
        out.println("Hello World!");
        out.println("</body>");
        out.println("</html>");
    }
}
```

Compilez ce fichier avec `javac`, en ajoutant l'api `servlets` au classpath, située dans le répertoire `lib` du serveur.

Copiez le fichier `.class` obtenu dans `WEB-INF/classes/fr/enseirb/servlets/`. Notez que la fin de l'arborescence correspond au nom du package de `HelloServlet`.

Votre servlet est accessible depuis `http://{votre-login}.rmorpheus.enseirb.fr/hello/servlet`

### Q1.3 Traitement des paramètres d'une requête GET

Il est possible de passer des paramètres dans une requête HTTP GET. Modifiez la classe `HelloServlet` afin d'afficher "Hello Toto!" lorsque l'on ajoute "?name=Toto" à l'adresse du servlet.

La documentation de l'API servlet est disponible ici : <http://tomcat.apache.org/tomcat-5.5-doc/servletapi/>

### Q1.4 Requetes POST

Un autre type de requêtes pouvant être traitée par un servlet sont les requêtes POST. C'est le type de requêtes envoyées par les formulaires html.

Coder un simple formulaire html avec un champ de texte, qui envoie une requête POST. Cette requête sera traitée dans la méthode `doPost` de notre servlet.

### Q1.5 Sessions

Il est possible de conserver des données le temps d'une session. Une session se termine lorsque le timeout expire ou lorsque le serveur s'arrête. Modifiez la servlet de la question 3 pour que le nom de la dernière personne soit affiché si aucun nom n'est mis en paramètre.

### Q1.6 Forward

Différentes parties de l'application peuvent traiter une même requête. Cette opération s'appelle le forwarding, et elle est transparente du point de vue du client. Le forwarding se fait grâce à la méthode suivante : `req.getRequestDispatcher("/servletToto").forward(req, resp);`

Créez un servlet qui *forward* les requêtes GET et/ou POST vers un servlet différent.

### Q1.7 Redirection

Il est possible de rediriger le navigateur du client vers une autre URL, qui peut être externe au serveur (e.g. [www.google.fr](http://www.google.fr)), grâce à la méthode `sendRedirect` de la classe `HttpServletResponse`.

Créez un formulaire qui redirige le navigateur du client sur Google, en ayant lancé la recherche avec les mots clés contenus dans le champ de saisie.