

# Indexation et Recherche d'Information

## Interrogation d'une base de données SQL via une Servlet

LO 17

### 1 Introduction

Vous allez dans ce TD utiliser des servlets Java pour interroger une base de données Postgresql à partir d'une page html.

Le principe de fonctionnement est le suivant : une application Java, une *servlet*, dont la fonction est de prendre en compte des requêtes SQL reçues d'un client web, de les transmettre à un serveur de base de données et de renvoyer au client web les résultats obtenus, doit être réalisée et déployée en tant que *webapp* dans un conteneur Tomcat sur votre poste de travail. **Tomcat traitera des requêtes HTTP que vous lui ferez parvenir en utilisant un URL commençant par `http://localhost:8080/`.**

### 2 Préparation de l'environnement

#### Notes :

- Si vous utilisez votre propre matériel informatique, sachez que l'Eclipse déployé sur les comptes LO17 est assorti du '*Plugin Tomcat*'.
- 'apache-tomcat-8.5.23' a été déployé sur chacun des comptes étudiants 'lo17aXYZ' et le plugin 'Eclipse Tomcat Plugin 9.1.2' a été déployé sur chaque poste 'smestaens24xy' de la salle BF-B100, dans le répertoire du disque local '/local2/eclipse/dropins/'.
- Vous trouverez ci-dessous les manipulations à suivre du côté utilisateur pour configurer votre environnement.

#### 2.1 Configuration de l'environnement

- Lancer 'eclipse' en tâche de fond : `eclipsejeedev &`
- Configuration du plugin 'Eclipse Tomcat Plugin' :
  - Sous Eclipse, Menu 'Window→Preferences → Tomcat'
  - 'Version de Tomcat' : 'Version 8.x'
  - 'Répertoire de Tomcat' : '/volsme/users/lo17a001/apache-tomcat'
  - 'Apply' et 'OK'
  - Ouvrir une nouvelle fenêtre : Menu 'Window → New Window'
  - Dans cette nouvelle fenêtre :
    - Ouvrir l'onglet 'Console' : Menu 'Window → Show Console'
    - Demarrer le serveur 'Tomcat' : Menu 'Tomcat → Demarrer Tomcat' (constater que le serveur est démarré : le dernier message de la console doit commencer par 'INFOS : Server startup in')
- Créer un nouveau projet, par exemple 'LO17-Test-Tomcat' :  
Menu 'File → New → Project... → fenêtre New Project → Java → Project Tomcat', puis 'Next'. Dans la fenêtre 'Nouveau projet Tomcat' :
  - 'Project Name' : 'LO17-Test-Tomcat'
  - Cocher la case 'Use default location' et 'Next'
  - 'Nom du contexte' : '/LO17-Test-Tomcat'
  - Cocher la case 'Autoriser la mise à jour de la définition du contexte'
  - 'Sous-répertoire racine de l'application web (optionnel)' : '/' et 'Finish'
  - Dans la fenêtre 'Open Associated Perspective' qui s'ouvre : cocher la case 'Remember my decision', et  
'Do you want to open this perspective now' : 'Yes'

- Un nouveau projet 'LO17-Test-Tomcat' est créé dans la vue 'Package Explorer'
- **Ne pas quitter 'Eclipse'**

## 2.2 Projet exemple

- Lancer 'Firefox' en tâche de fond : `firefox &`
- Recopier les fichiers d'exemple du TD du projet 'LanceRequete' dans les bons répertoires (voir ci-dessous '**Explications**') du workspace 'Eclipse' du projet précédemment créé, soit : `./workspace/LO17-Test-Tomcat/`
- Sous 'Eclipse' : dans le fenêtre avec la vue 'Package Explorer' : sélectionner 'LO17-Test-Tomcat', puis 'bouton droit souris → Refresh'
- Sélectionner 'LO17-Test-Tomcat → Web-INF → lib → postgresql-9.0-801.jdbc4.jar', puis 'bouton droit souris' → Build Path → Add to Build Path
- **Ne pas quitter 'Eclipse'**
- Sous 'Firefox' :
  - Aller à l'URL `'http://localhost:8080'` : une page 'Apache Tomcat' s'affiche
  - Sur cette page, sélectionner le bouton en haut à droite 'Manager App' et s'authentifier avec 'lo17mgr' et 'lo17tom4'
  - Constaté que, parmi la liste des 'Application', l'application '/LO17-Test-Tomcat' est présente
  - Entrer l'URL `'http://localhost:8080/LO17-Test-Tomcat/LanceRequete.html'`
  - Dans la zone de saisie de la page 'Frame de lance requete', saisir une requête SQL, par exemple : `'select * from titretext'` et valider via le bouton 'Submit Query' et constater le résultat de la requête qui s'affiche sur la frame de droite
- **A la fin de votre projet**
  - Quitter 'Firefox'
  - Sous 'Eclipse' : arrêter 'Apache-Tomcat' : Menu 'Tomcat → Arrêter Tomcat', puis quitter 'Eclipse'

## 3 Explications sur la configuration et les exemples

### 3.1 Les fichiers et répertoires qui constituent votre *webapp*

- Votre projet Tomcat a été déployé en *webapp* par votre serveur Tomcat au démarrage de ce dernier
- Les documents que vous placez directement en dessous du répertoire du projet (c'est à dire dans `<votre workspace>/<nom de votre projet>`) seront accessibles tels quels au client web, et **c'est donc à ce niveau que vous devez mettre vos fichiers .html, etc. (exemples LanceRequete.html, FormLeft.html)**
  - Le sous-répertoire WEB-INF contient les éléments que Tomcat utilisera pour construire des pages dynamiquement, notamment des servlet.
    - **Les fichiers source de vos servlet (fichiers .java) sont à placer dans WEB-INF/src** (exemple **LanceRequete.java**).
    - Les classes Java correspondantes (fichiers .class) sont créées de manière transparente par Eclipse dans WEB-INF/classes (que par défaut le '*Plugin Tomcat*' n'affiche pas).
    - **Toutes bibliothèques** dont vos servlet pourraient avoir besoin (en dehors de celles qui sont automatiquement rendues disponibles par le conteneur Tomcat) **sont à placer dans WEB-INF/lib**. Pour vous il s'agira notamment du fichier **postgresql-9.0-801.jdbc4.jar**, que vous avez téléchargé depuis le site web de l'UV et avez déployé en activant le *Chemin de Génération*.
  - **Vous devez également créer, dans le répertoire WEB-INF, un fichier web.xml** sur le modèle de celui que vous pouvez télécharger depuis le site web de l'UV. Ce fichier fait correspondre les servlet que vous pourriez mettre à disposition à des chemins d'URL permettant à des clients web distants de les invoquer.

### 3.2 Préparation de votre page html

Les fichiers html que vous avez téléchargé sont un exemple de page html qui accepte une requête en entrée et affiche le résultat dans un cadre (frame) (il faut changer les extensions « `.txt` » en « `.html` » pour les deux fichiers concernés). Inspirez-vous de cet exemple pour créer votre propre application de

façon à ce qu'elle permette, par exemple, de mettre en colonnes (en tableau) les différentes réponses si la requête porte sur des items multiples (auteur et année et fichier) et que lorsque la réponse contient des noms de fichiers, ceux-ci soient des liens *clickables* vers les fichiers en question.

Pour ce qui est des URL à utiliser, le nom que vous choisissez pour votre projet fera partie du chemin. Si vous déployez un fichier `LanceRequete.html` dans un projet que vous avez appelé `monProjetTomcatLCI`, l'URL correspondant sera `http://localhost:8080/monProjetTomcatLCI/LanceRequete.html`. Faites attention aux *chemins relatifs*. Dans `web.xml`, un chemin renseigné par la balise `<url-pattern>`, même si le premier caractère est `/`, reste *relatif à la racine de votre projet*, et par conséquent

```
<url-pattern>/servlet/LanceRequete</url-pattern>
```

désigne l'URL `http://localhost:8080/monProjetTomcatLCI/servlet/LanceRequete` (notez que `/servlet` ne correspond pas à un sous-répertoire, mais dans le contexte d'un `<url-pattern>` est interprété par Tomcat comme une chaîne de caractères quelconque). En revanche, dans une page html faisant partie de votre projet, le caractère `/` initial dans un URL indique que l'URL est *relatif à la racine de votre serveur*. Si vous désirez qu'il soit relatif à la racine de votre *projet* le caractère `/` doit être omis. Ainsi :

```
<form action = "/servlet/LanceRequete"> → http://localhost:8080/servlet/LanceRequete  
(mauvais URL)
```

```
<form action = "servlet/LanceRequete"> → http://localhost:8080/  
monProjetTomcatLCI/servlet/LanceRequete  
(bon URL)
```

**Configuration de votre serveur html :** de façon à ce que l'adresse `localhost` soit correctement traitée vous devez modifier les paramètres réseau de votre serveur html (*Firefox*).

- choisissez Préférences → Avancé → Réseau → Paramètres
- cochez Configuration manuelle du proxy, Proxy HTTP : `proxyweb.utc.fr`, Port 3128
- Pas de proxy pour : `localhost`, `127.0.0.1`

## 4 Déclaration et mise à jour de vos servlet

### 4.1 Première déclaration de la servlet

Quand vous avez fini d'écrire le code source de votre servlet, pour la première fois vous devez la déclarer dans le manager de Tomcat, par un click droit sur votre projet, puis menu local Projet Tomcat sous menu Mise à jour du contexte.

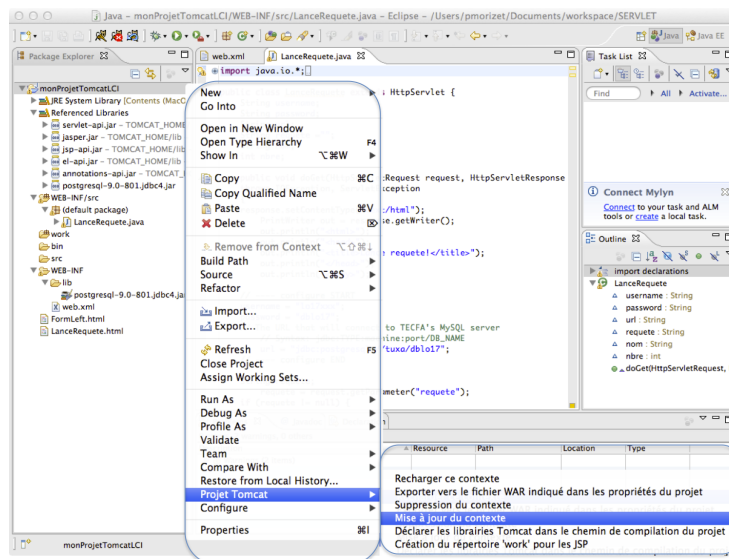


FIGURE 1 – Première déclaration de la Servlet

## 5 Modifications d'une servlet

Pour chaque modification de la Servlet, vous devez déclarer au manager de Tomcat que l'application a été mise à jour par un click droit sur votre projet, puis menu local Projet Tomcat sous menu Recharger ce contexte.

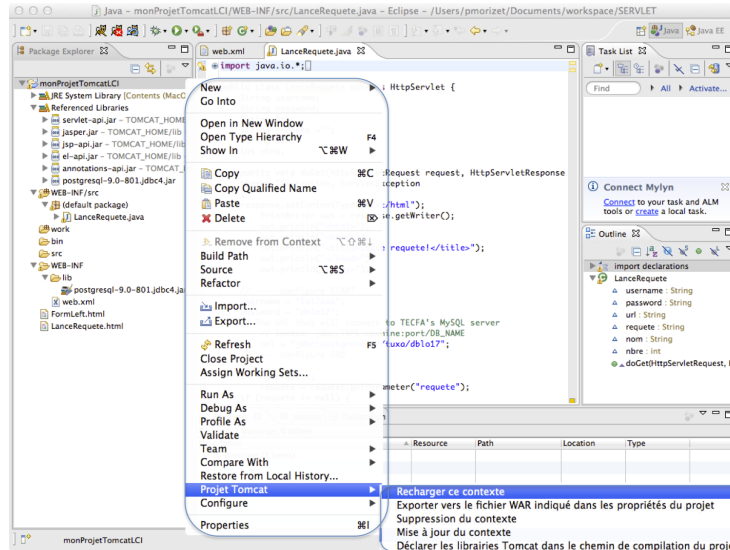


FIGURE 2 – Mise à jour de la Servlet