

## INSTALLATION

Vous pouvez installer très simplement un système serveur WEB-Servlets-JSP , et le lancer depuis votre station de travail. Ce serveur, programmé en Java pourra être lui-même client du serveur SQL Postgresql de l'ESSI.

**Vous devez positionner votre variable d'environnement JAVA\_HOME correctement vers une jdk (e.g., /usr/java/j2sdk1.4.2\_05/) ainsi que votre PATH qui doit avoir un accès vers \$JAVA\_HOME/bin (pour permettre l'accès à Resin au compilateur javac ).**

Exemple:

```
export JAVA_HOME='/usr/java/j2sdk1.4.2_05'  
export PATH=$PATH:'/usr/java/j2sdk1.4.2_05/bin'
```

Le serveur WEB, à copier chez vous, est une version allégée du serveur Resin. Cette version allégée est disponible sur: **~rueher/BDR/resin-2.0.2Lite**

Dans ce serveur vous trouverez les fichiers et répertoires suivants:

- **resin-2.0.2Lite/conf/resin.conf**  
Il contient les infos pour le lancement du serveur. Vous pouvez modifier ce fichier de configuration, et en particulier:
  - le port de lancement qui est prévu pour être **8091**
  - l'adresse (relative par rapport au répertoire **resin-2.0.2Lite** ) du répertoire racine des applications (**doc** dans la configuration livrée); ce répertoire contient tout ce qui constitue votre application Web: pages html, pages jsp , images, classes et packages Java), ...
- **resin-2.0.2Lite/doc**
  - l'application WEB réside dans ce répertoire: pages .html, pages .jsp, images .gif or .jpg, classes et/ou packages Java, ....  
Vous pouvez structurer ce répertoire en sous-répertoires.
  - **resin-2.0.2Lite/doc/WEB-INF/classes**  
Si vous voulez créer et utiliser du code java (classes et/ou packages) utilisées dans vos pages .jsp, il vous faut mettre les sources **xxx.java** dans ce sous répertoire. Le système se chargera de les compiler automatiquement en **xxx.class**
- Actuellement **doc** contient deux applications permettant d'effectuer des requêtes à une bases de données Postgresql: Tout ce qui constitue votre application :
  - **marques.jsp** est un exemple de requête à une table construite en fonction de certains critères entrés par l'utilisateur grâce à un formulaire de saisie. Il n'utilise pas d'objet persistant entre les pages du client.
  - **requete.jsp** permet de lancer n'importe quelle requête SQL. Il est programmé avec la notion de `ResultSetMetaData` de JDBC qui permet de connaître dynamiquement la liste des champs d'une requête dont on ne connaît pas le texte à l'avance.  
Cette page utilise un package **pgsql** qui se trouve dans le répertoire.../**WEB-INF/classes**, et utilise également l'objet «session» pour sauvegarder des données entre les pages du client (comme par exemple l'objet qui représente la connexion à SQL).

- **resin-2.0.2Lite/bin**  
Il contient le binaire pour lancer et arrêter le serveur:
  - **httpd.sh -h** pour connaître la syntaxe
  - **httpd.sh start** pour lancer le serveur
  - **httpd.sh stop** pour arrêter le serveur
- **resin-2.0.2Lite/lib**  
Il contient les diverses librairies java utilisées par le serveur Resin lui-même, ainsi que celles utilisée par vos propres applications. Ce répertoire contient déjà le pilote JDBC postgresql7.1.1.jar pour Postgresql.

## **GESTION DU SERVEUR**

Le démarrage du serveur :

**httpd.sh start**

Arrêt du serveur :

**httpd.sh stop**

**N'oubliez pas d'éteindre votre serveur avant de quitter la station de travail sinon vous laissez dormir un processus qui monopolise un port !**

## **APPEL DES PAGES DU SERVEUR**

Une fois le serveur lancé sur votre station de travail **votre-machine.essi.fr**, vous pouvez appeler vos applications situées dans **doc**, depuis n'importe où, à l'aide d'un navigateur WEB :

<http://localhost:8091>

Si une page JSP doit être appelée avec des paramètres (récupérés dans cette page grâce à:

`request.getParameter("NOM_DU_CHAMP")`),

on peut alors fournir ces paramètres, soit à travers un formulaire de saisie, soit directement, comme pae exemple:

[http://localhost:8091/marques\\_result.jsp?NOM=Shalimar](http://localhost:8091/marques_result.jsp?NOM=Shalimar)

## **RESUME DE QUELQUES COMMANDES JSP (JavaServerPages)**

Ce sont des pages HTML contenant des balises java.

- Les importations de classes et librairies java sont balisées par :  
**<%@ page language="java" import="java.sql.\*, java.math.\* %>**
- Les déclaration java qui définissent les champs du Servlet qui représentera la page JSP, sont balisées par **<%! ....%>**. Ne les utiliser qu'en tête de page. Exemple:  
**<%!       String chaine2;       .....       %>**
- Le code java à déclencher est balisé par **<% ... %>**. Exemple:  
**<%   ... user= "toto";   out.println ("<BR>" + user);   ....   %>**
- Le code Java à évaluer est balisé par **<%= ....%>**. Exemple:

```
<INPUT>... <OPTION value="<%=codePays%>" <%=nomPays%>...  
</INPUT>
```

- Pour récupérer les valeurs des champs envoyés par un formulaire HTML :  
`<% ... request.getParameter("nom du champ"); ... %>`
- Pour écrire dans la page HTML en cours :  
`<% ... out.println (".....");... %>`
- Pour créer/récupérer un objet persistant appartenant à une classe **MaClasse** et nommé **MonObj**:  
`<jsp:useBean id="MonObj" scope="session" class=MaClasse /jsp>`
- Pour utiliser cet objet pour le modifier, le lire et/ou déclencher ses méthodes:  
`<% ... MonObj.nom= "toto" ... MonObj.decrire() ... %>`

<http://java.sun.com/products/jsp/?frontpage-javaplatform>