

TP JEE 1

web & servlet

Daniel Hagimont
Daniel.Hagimont@enseeiht.fr

L'objectif de ce TP est de démarrer avec Jboss et avec les applications web dynamiques.

Je suppose que nous utilisons (soit sur votre laptop, soit sur les machines de l'N7) comme en Intergiciels :

- JDK 8 (jdk-8u202-linux-x64.tar.gz)
- Eclipse 2019 (eclipse-jee-2019-09-R-linux-gtk-x86_64.tar.gz)

1. Installation et démarrage de Jboss

- exécuter l'installateur jboss-eap-7.0.0-installer.jar
java -jar jboss-eap-7.0.0-installer.jar
en plaçant installation dans votre home/nosave
- allez dans EAP-7.0.0/bin et lancer standalone.sh

Les dernières lignes à s'afficher devraient être :

```
18:46:11,265 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: JBoss EAP
7.0.0.GA (WildFly Core 2.1.2.Final-redhat-1) a démarré en 3491ms - 267 sur 553 services
ont démarré (371 services sont passifs ou à la demande)
```

Vérifiez que le serveur Jboss est opérationnel (<http://localhost:8080/>) :

2. Créez une servlet retournant HelloWorld

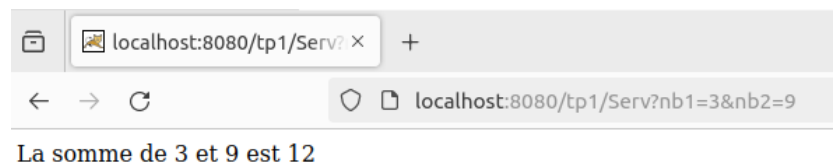
Les TP se feront à l'aide d'Eclipse. Nous n'utiliserons pas les fonctionnalités java-ee propres à Eclipse excepté les exports de projet war. Nous utiliserons les spécifications EJB 3.1 qui permettent d'encapsuler tous les EJB dans un même fichier war.

- créer un Dynamic Web Projet (tp1)
 - on ne lui associe aucun Runtime (on n'exécute pas dans eclipse)
 - on lui ajoute les jars nécessaires
dans EAP-7.0.0/modules/system/layers/base/javax/
persistence/api/main/hibernate-jpa-2.1-api-1.0.0.Final-redhat-1.jar
ejb/api/main/jboss-ejb-api_3.2_spec-1.0.0.Final-redhat-1.jar
servlet/api/main/jboss-servlet-api_3.1_spec-1.0.0.Final-redhat-1.jar
- créer un package (pack)
- créez une servlet (Serv)
dans doGet() , remplacez le corps de la méthode par
`response.getWriter().println("<html><body>Hello World !</body></html>");`
- exportez une archive WAR (tp1.war) dans EAP-7.0.0/standalone/deployments

Si vous avez respecté ce manuel, vous pouvez tester votre servlet à l'adresse suivante :
<http://localhost:8080/tp1/Serv>

3. Calculatrice « manuelle »

Transformez la servlet précédente pour lui faire exécuter une addition à partir de deux opérandes passés dans l'adresse (URL) comme montré sur la capture ci-dessous :



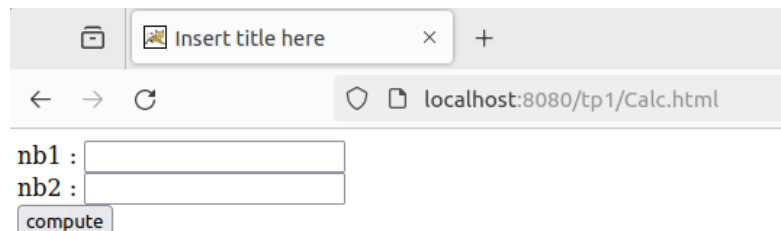
Rappel :

dans la servlet, on accède aux paramètres passés dans l'URL avec :

```
String s1 = request.getParameter("nb1");
```

4. Calculatrice avec formulaire

Sans modifier la servlet de la calculatrice manuelle, permettez à un utilisateur de saisir les opérandes depuis un formulaire HTML ressemblant à la copie d'écran ci-dessous.



5. Calculatrice avec JSP

Créez une page JSP et modifiez la servlet pour permettre de complètement dé-corréler la vue du contrôleur et afficher le résultat du calcul grâce à la page JSP. La même page JSP est appelée depuis le navigateur pour saisir les paramètres et elle est appelée depuis la servlet pour afficher le résultat (et saisir de nouveaux paramètres).