


|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

### Document présentation


|                    |   |
|--------------------|---|
| <b>Description</b> | Manuel développeur pour réaliser des mesures de performances pour son application Web Java. |
|--------------------|---|

### Document certification

|                | <i>Name</i>   | <i>Fonction</i>       | <i>Date livraison</i> |
|----------------|---------------|-----------------------|-----------------------|
| <i>Author</i>  | DJEBIEN Tarik | Etudiant Miage IPI NT |                       |
| <i>Decidor</i> | ROUVOY Romain | Enseignant QL         |                       |

### Document Version history

| <i>Version</i> | <i>Date</i>    | <i>Change Summary</i>    | <i>Change Author</i> |
|----------------|----------------|--------------------------|----------------------|
| 1.0            | 05 / 03 / 2013 | Initialisation du manuel | Djebien Tarik        |
| 2.0            | 11 / 03 / 2013 | Finalisation du manuel   | Djebien Tarik        |

|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

### Pré-requis

Installation d'un outil de test de charge Apache Jakarta JMeter

Utilisation de Jmeter : faire un test de charge

Préparer son scénario fonctionnel

« Jmeteriser » son scénario fonctionnel

Affinage du scénario de tests

Exécuter son scénario de tir de charge

Faire un tir de charge par paliers et exploiter les résultats


Identification scénario et préparation des paliers

Personnalisation, Enregistrement du scénario et mise en place des paliers

Exécution du tir de charge par paliers


Génération de graphique

Fin du tutoriel.

|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

## Pré-requis

- Pour faire fonctionner JMeter, la seule contrainte est de disposer d'une JVM sur le serveur d'installation (Java 6 ou supérieur est requis). N'oubliez pas de configurer votre variable d'environnement JAVA\_HOME.
- Disposer d'un projet webApp Java sur lequel on souhaite effectuer des tests de montée en charge avec JMeter.
- Installer un serveur d'application comme tomcat : <http://tomcat.apache.org/> pour déployer votre application web.  
N'oubliez pas de configurer votre variable d'environnement TOMCAT\_HOME, CATALINA\_HOME.

|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

## Installation d'un outil de test de charge Apache Jakarta JMeter

Cet outil de test est fournie par la fondation Apache et peut être téléchargé ici :

[http://jmeter.apache.org/download\\_jmeter.cgi](http://jmeter.apache.org/download_jmeter.cgi)


JMeter est un outil de test de charge qui peut donc « tester » les performances d'une application en simulant des utilisateurs (qualifiés de 'virtuels') et en enregistrant les temps de réponses pour permettre leur analyse.

Concrètement voici quelques exemples de tir de charge possibles :

- Tester les temps de réponses d'une application web, en fonction du nombre d'utilisateurs virtuels
- Tester les écarts de temps de réponse en fonction de paliers d'utilisateurs
- Tester les différences de comportements d'une même application sur deux (ou plus) environnements d'exécution différents (par exemple cluster ou non, configuration 1 / configuration 2, etc)
- Tester la robustesse d'une application (à partir de quand la solution s'interrompt brutalement (crash) ?...)
- Tester fonctionnellement une application de manière automatique (tests de non régression, de validité fonctionnelle, etc)
- Superviser une application (soit via une sorte de 'ping', soit par des éléments plus complexes).

D'autres possibilités existent, et ne demandent qu'à être pensées.

Pour conclure, JMeter est un outil qui permettra de valider le bon fonctionnement d'une application dans son contexte.

|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

## Utilisation de Jmeter : faire un test de charge

### Préparer son scénario fonctionnel

(Pour ce test de charges nous allons utiliser Tomcat 6, et le projet wordcount en exemple.)

#### Identifier le scénario fonctionnel

Avant de travailler avec JMeter, commençons par identifier le scénario fonctionnel que l'on souhaite tester. Le scénario correspond à l'enchaînement des écrans que devra effectuer le robot JMeter de test.

Pour cela, on lance Tomcat et on va sur l'URL : <http://localhost:8080/wordcount-webapp/>

**Qualité du logiciel – TP Test d'intégration et déploiement**

**Service WordCount**

Le texte cible :

Tarik Djebien Tarik Tarik Tarik Djebien Djebien Tarik

Map<String, Integer> countAllWordInText()

countAllWordInText()

int countOneWordInText(String word)


boolean isWordInText(String word)

List<String> getAllFrequentWordsInText(List<String> words, int N)

Call All Service

© Copyright DJEBIEN Tarik Version : 2.0 FR

Ensuite, nous allons cliquer sur l'onglet **int countOneWordInText(String word)** et remplir le champ **countOneWordInText** par la valeur **Tarik**.

|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

De même, on cliquera ensuite sur l'onglet **boolean isWordInText(String word)** et remplirons le champ **isWordInText** par la valeur **Tarik**.

### Qualité du logiciel – TP Test d'intégration et déploiement

#### Service WordCount

Le texte cible :

Tarik Djebien Tarik Tarik Tarik Djebien Djebien Tarik

```

Map<String, Integer> countAllWordInText()
int countOneWordInText(String word)
boolean isWordInText(String word)
isWordInText(  )
List<String> getAllFrequentWordsInText(List<String> words, int N)

```

© Copyright DJEBIEN Tarik Version : 2.0 FR

Finalement, nous terminerons sur l'onglet :

**List<String> getAllFrequentWordsInText(List<String> words, int N)**

avec les deux champs remplis par les valeurs **Tarik** et **1**.

### Qualité du logiciel – TP Test d'intégration et déploiement

#### Service WordCount

Le texte cible :

Tarik Djebien Tarik Tarik Tarik Djebien Djebien Tarik


```

Map<String, Integer> countAllWordInText()
int countOneWordInText(String word)
boolean isWordInText(String word)
List<String> getAllFrequentWordsInText(List<String> words, int N)
getAllFrequentWordsInText(  ,  )

```

© Copyright DJEBIEN Tarik Version : 2.0 FR

On termine en cliquant sur le bouton **Call All Service** et voilà, on a un scénario fonctionnel. Il

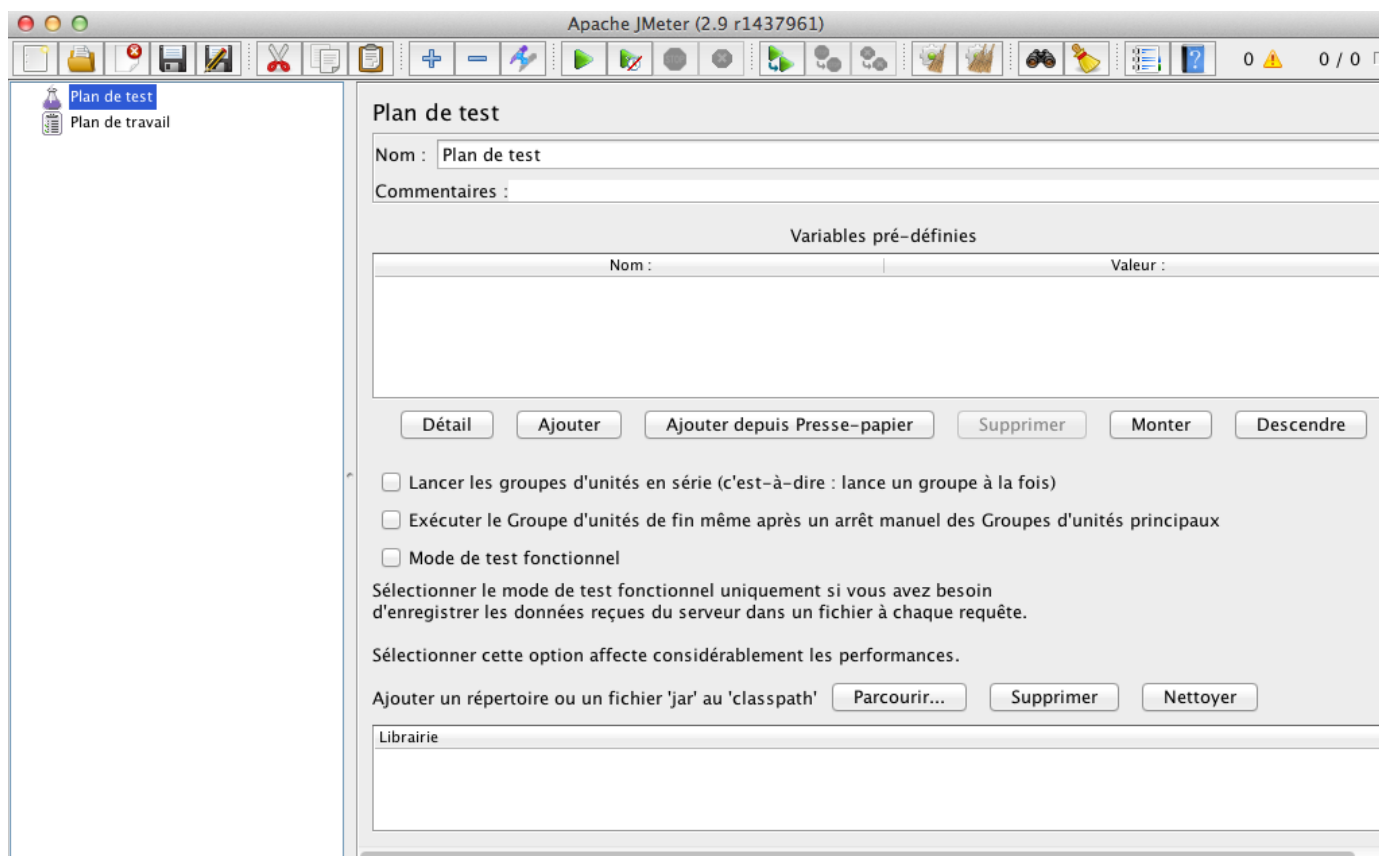
|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |


faut maintenant passer à l'étape suivante, afin de l'implémenter dans JMeter.

## « Jmeteriser » son scénario fonctionnel

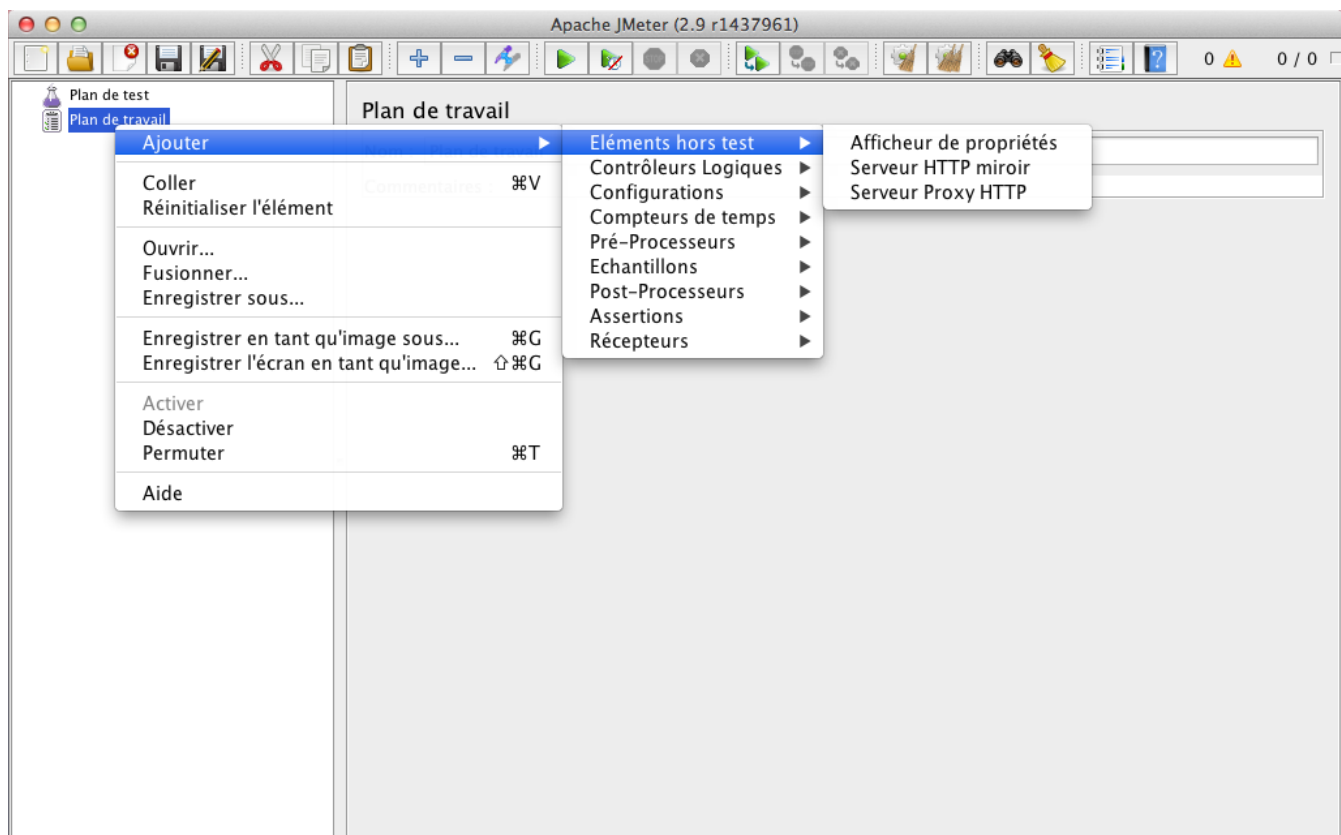
Lancer JMeter, (sous Mac, on utilisera le fichier `JMETER_HOME/bin/jmeter.sh`)

```
MacBookPro:bin tarik$ pwd
/Users/tarik/Documents/Informatique/Projet/lib/apache-jmeter-2.9/bin
MacBookPro:bin tarik$ ./jmeter.sh
Mar  5 14:52:15 MacBookPro.local java[6648] <Error>: CGContextGetCTM: invalid context 0x0
Mar  5 14:52:15 MacBookPro.local java[6648] <Error>: CGContextSetBaseCTM: invalid context 0x0
Mar  5 14:52:15 MacBookPro.local java[6648] <Error>: CGContextGetCTM: invalid context 0x0
Mar  5 14:52:15 MacBookPro.local java[6648] <Error>: CGContextSetBaseCTM: invalid context 0x0
```




|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

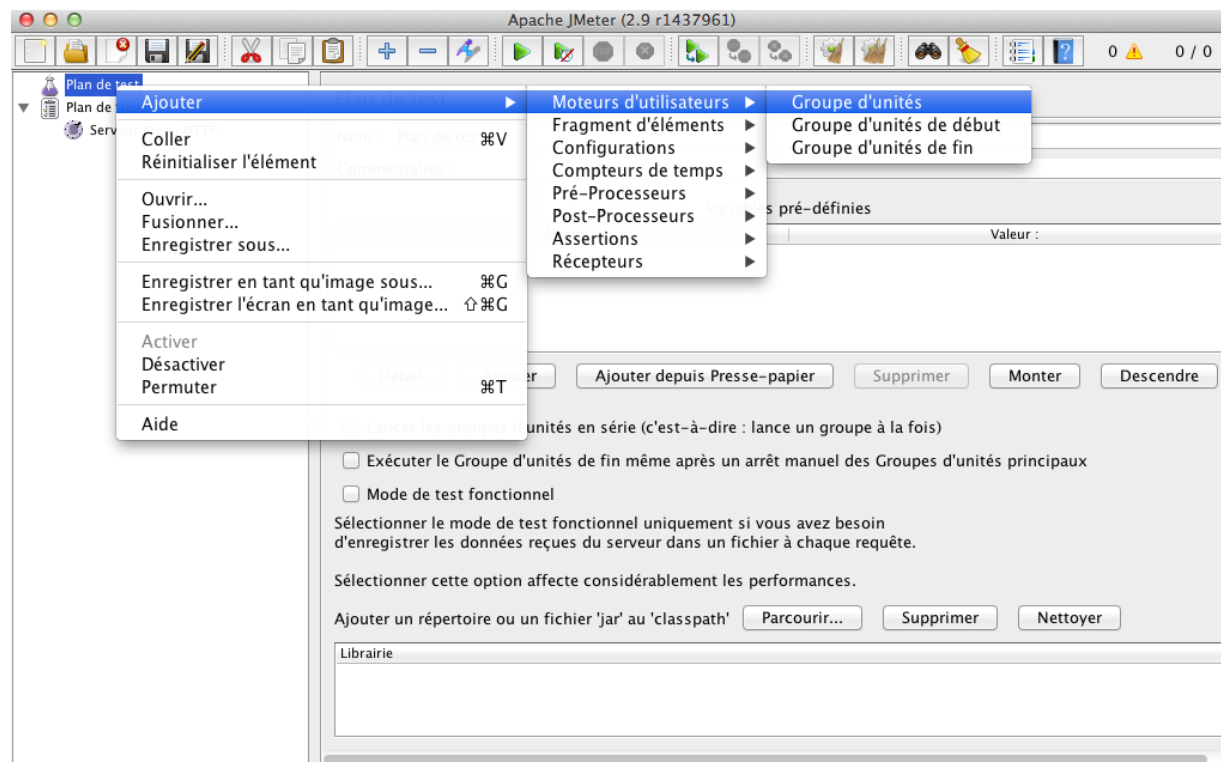
Sélectionner l'item **Plan de travail**, puis clic droit sur la souris, et ajouter un élément hors test > **Serveur Proxy HTTP**.






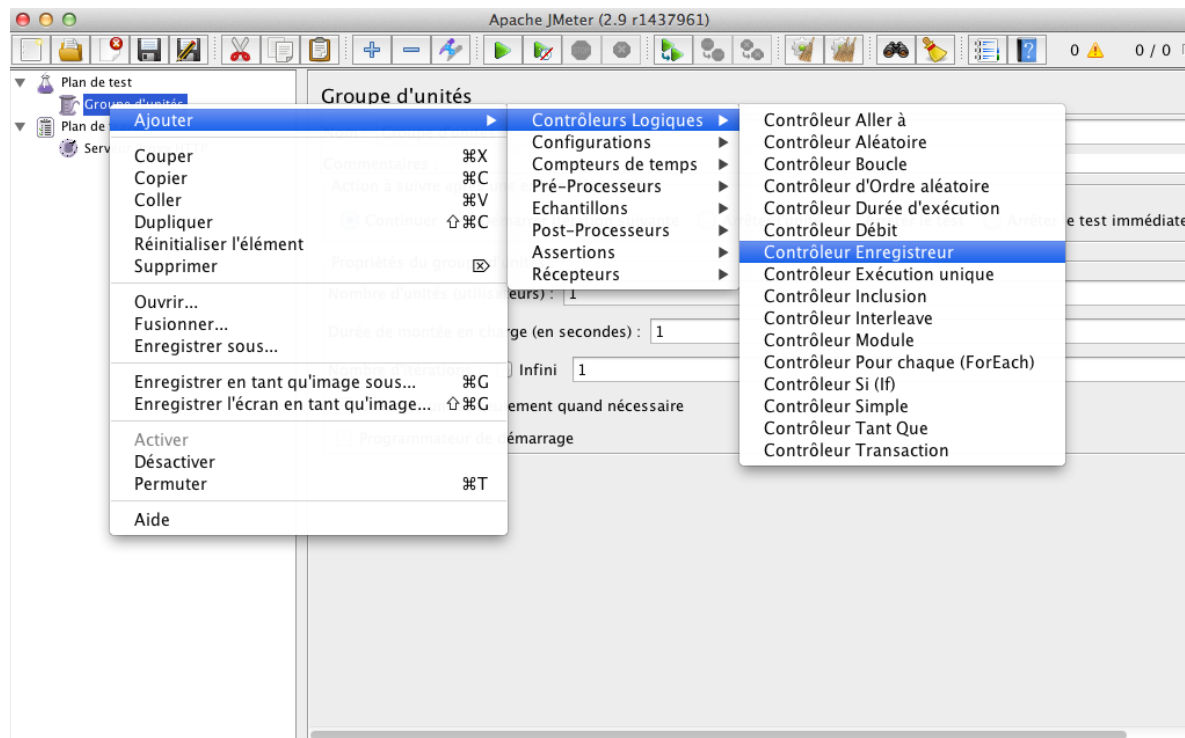
|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

Puis aller sur l'item **Plan de test**, un clic droit sur la souris, et ajouter un élément **Groupe d'unités**.




|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

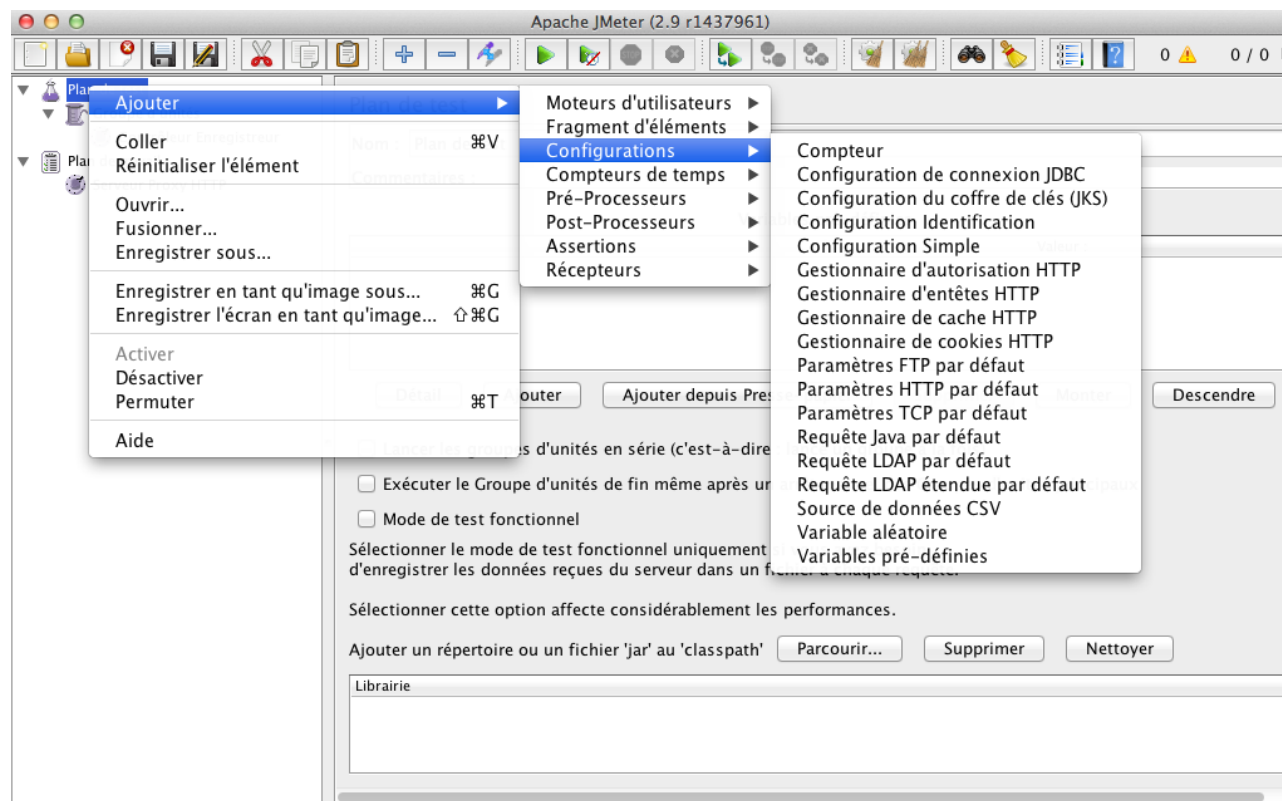
Ensuite, sélectionner le nouvel élément **Groupe d'unités**, un clic droit sur la souris, et ajouter un élément Contrôleurs Logiques > **Contrôleur Enregistreur**.




Le **Serveur Proxy HTTP** et le **Contrôleur Enregistreur** sont les deux seuls éléments nécessaires pour l'enregistrement d'une session de navigation. Cependant nous allons y ajouter quelques autres éléments pour répondre à quelques contraintes et/ou pour nous faciliter le travail.

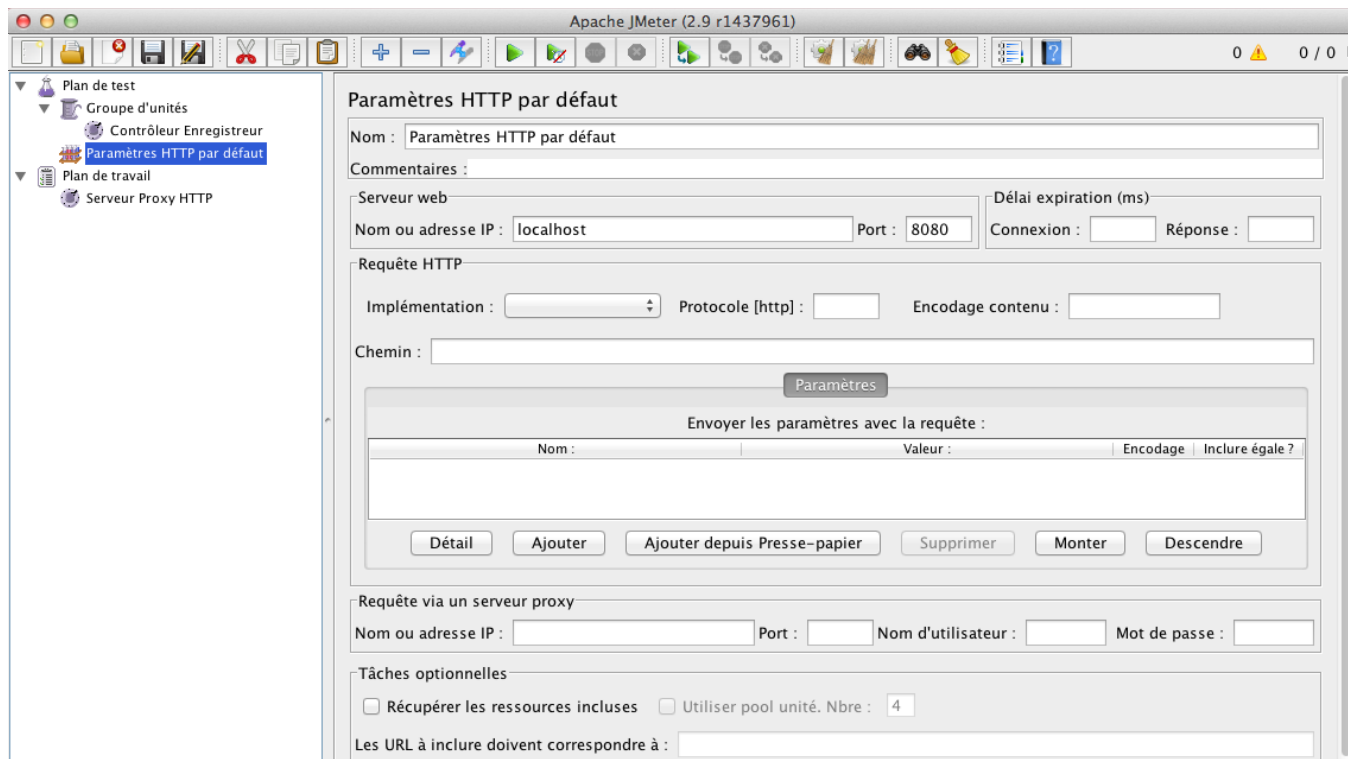
|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

L'élément de configuration **Paramètres HTTP par défaut** va nous permettre de 'variabiliser' le nom du serveur web, son port, etc., c'est-à-dire que les requêtes HTTP (qui seront créées automatiquement lors de la session de navigation par le serveur Proxy HTTP), verront leur champ nom du serveur Web, Port, etc. vides.



|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

Ensuite, dans l'élément de configuration, nous paramètrerons les champs suivants **Nom ou adresse IP** par la valeur du nom du serveur à tester, et le champ **Port** par la valeur du port du serveur (ici 8080) :



Apache JMeter (2.9 r1437961)

0 0 / 0

Plan de test

- Groupe d'unités
  - Contrôleur Enregistreur
  - Paramètres HTTP par défaut**
  - Plan de travail
  - Serveur Proxy HTTP

### Paramètres HTTP par défaut

Nom : Paramètres HTTP par défaut

Commentaires :

Serveur web

Nom ou adresse IP : localhost Port : 8080

Délai expiration (ms)

Connexion : Réponse :

Requête HTTP

Implémentation : Protocole [http] : Encodage contenu :

Chemin :

Paramètres

Envoyer les paramètres avec la requête :

| Nom | Valeur | Encodage | Inclure égale ? |
|-----|--------|----------|-----------------|
|     |        |          |                 |

Détail Ajouter Ajouter depuis Presse-papier Supprimer Monter Descendre


Requête via un serveur proxy

Nom ou adresse IP : Port : Nom d'utilisateur : Mot de passe :

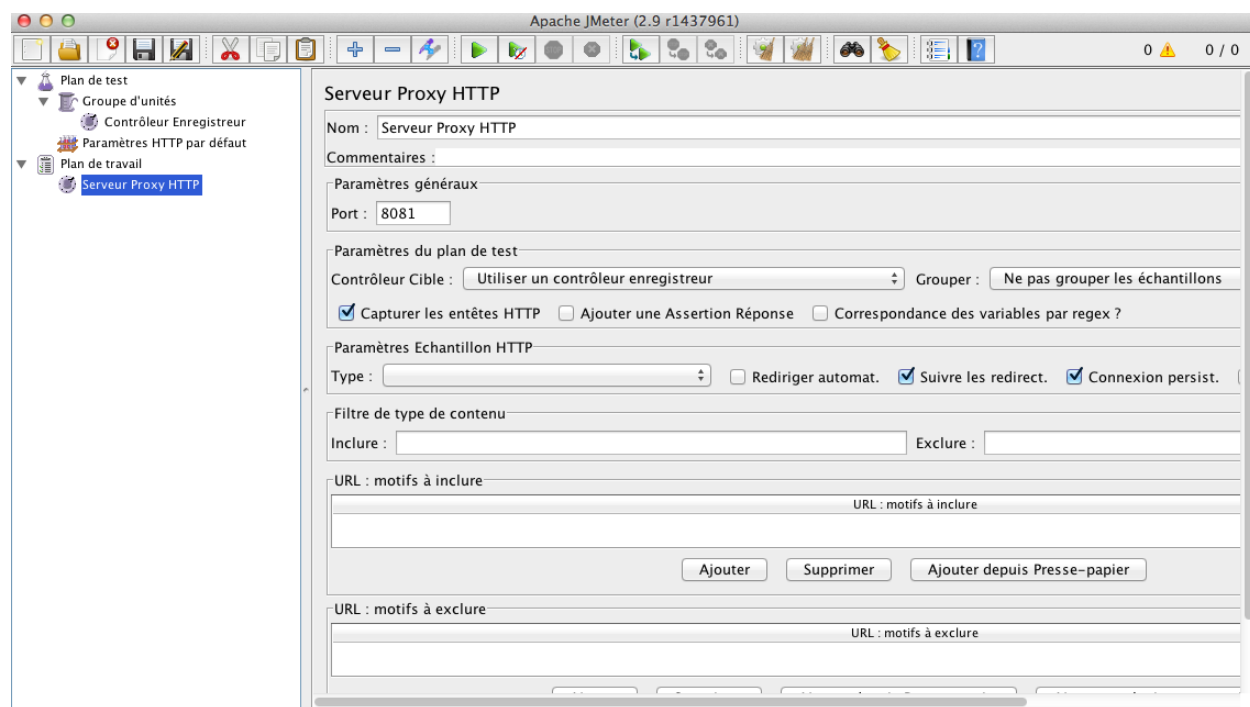
Tâches optionnelles


☐ Récupérer les ressources incluses ☐ Utiliser pool unité. Nbre : 4

Les URL à inclure doivent correspondre à :

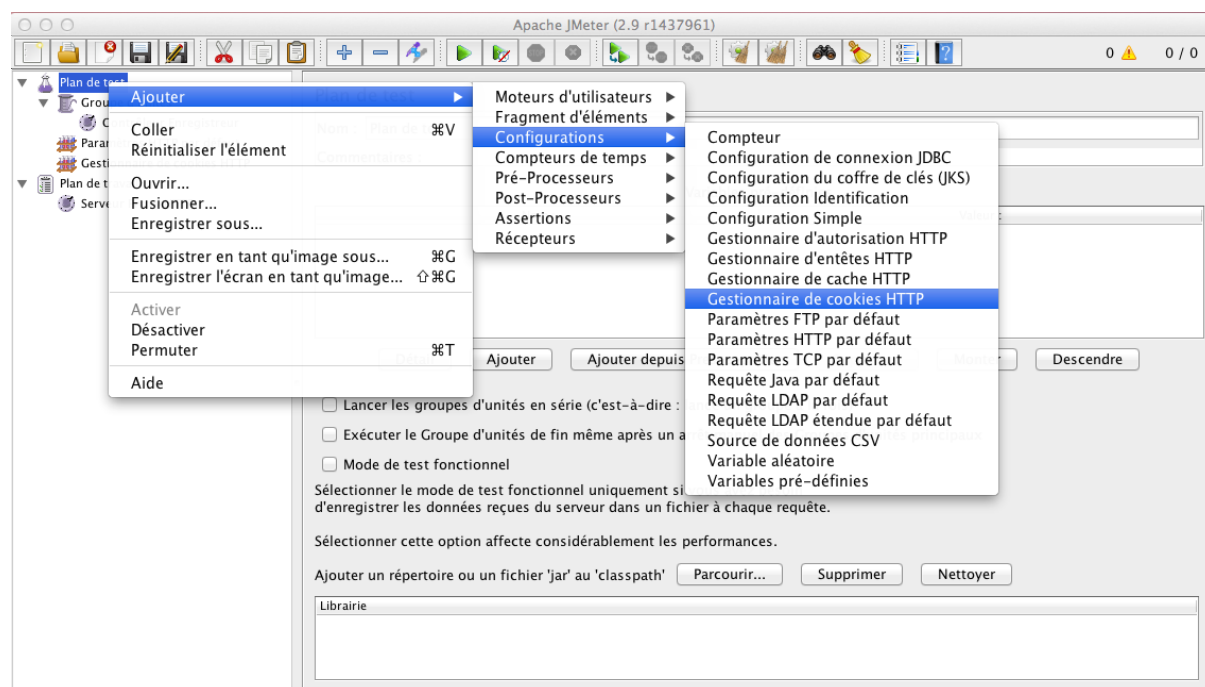
|  |                            |                           |
|--|----------------------------|---------------------------|
|  | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013   | Manuel développeur         | Current doc version : 2.0 |

**ATTENTION :** si vous utilisez un Tomcat sur la même machine que votre JMeter, il y aura certainement un conflit sur le port 8080 (car Tomcat, par défaut utilise le 8080 et l'élément **Serveur Proxy HTTP** de JMeter utilise également le port 8080). Il convient donc de mettre des ports différents. Par exemple mettre 8081 dans le champ **Port** sur l'élément **Serveur Proxy HTTP** de JMeter (et dans la configuration du navigateur au niveau du proxy, cf. plus bas).

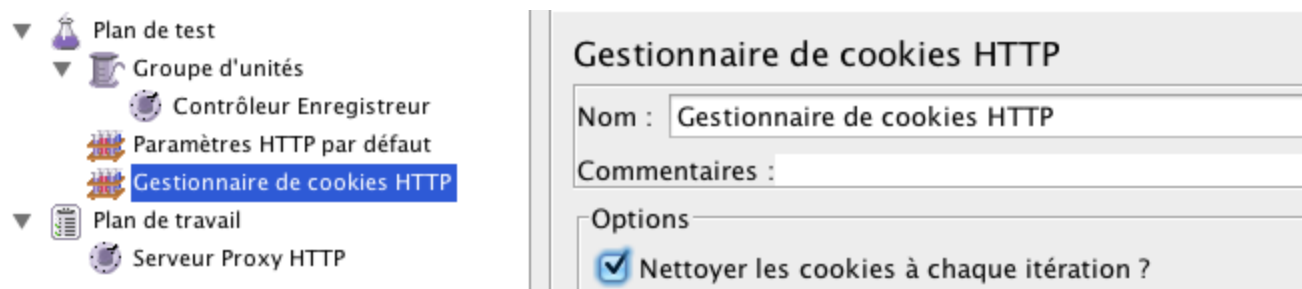


|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

L'élément **Gestionnaire de cookies HTTP** va nous permettre de faire du suivi de cookie (en particulier le suivi de session). C'est-à-dire que JMeter va se comporter comme un navigateur Internet au niveau des cookies, en envoyant la valeur du cookies quand les conditions sont remplies.




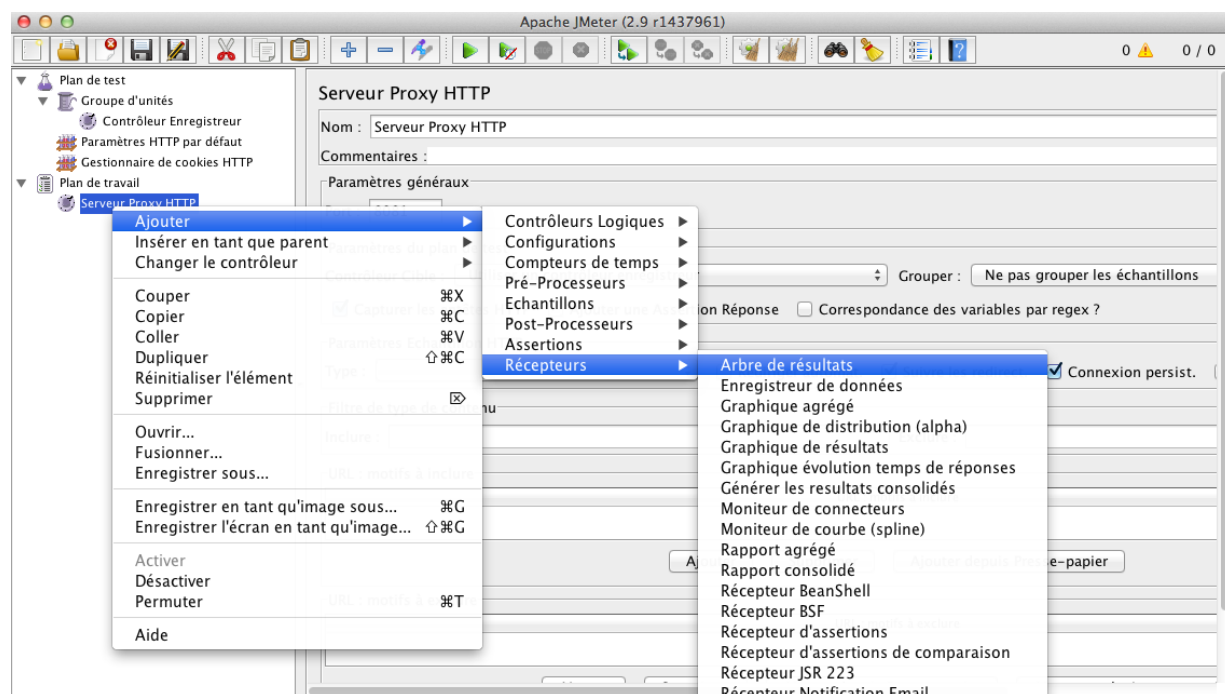
On va cocher l'option *Nettoyer les cookies à chaque itération* ? afin d'avoir un effacement des cookies (et autre session) à chaque itération lors du test de charges.



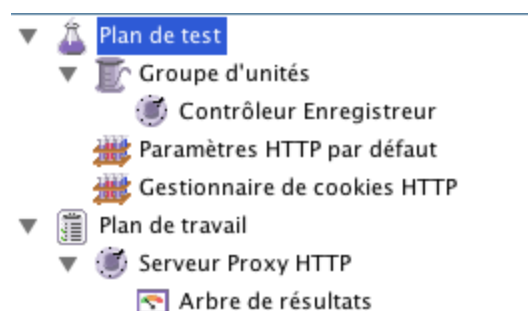
L'ajout du récepteur **Arbre de résultats** va permettre de voir les échanges HTTP, et si besoin, nous permettre de les consulter plus tard (à froid) lors de l'affinage du scénario de tir de charges.


|                   |                |
|-------------------|----------------|
| Titre du document | Page 14/<br>27 |
|-------------------|----------------|

|   |                            |                           |
|---|----------------------------|---------------------------|
|  Université<br>Lille 1 | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |



Au final, on a donc les éléments suivants :

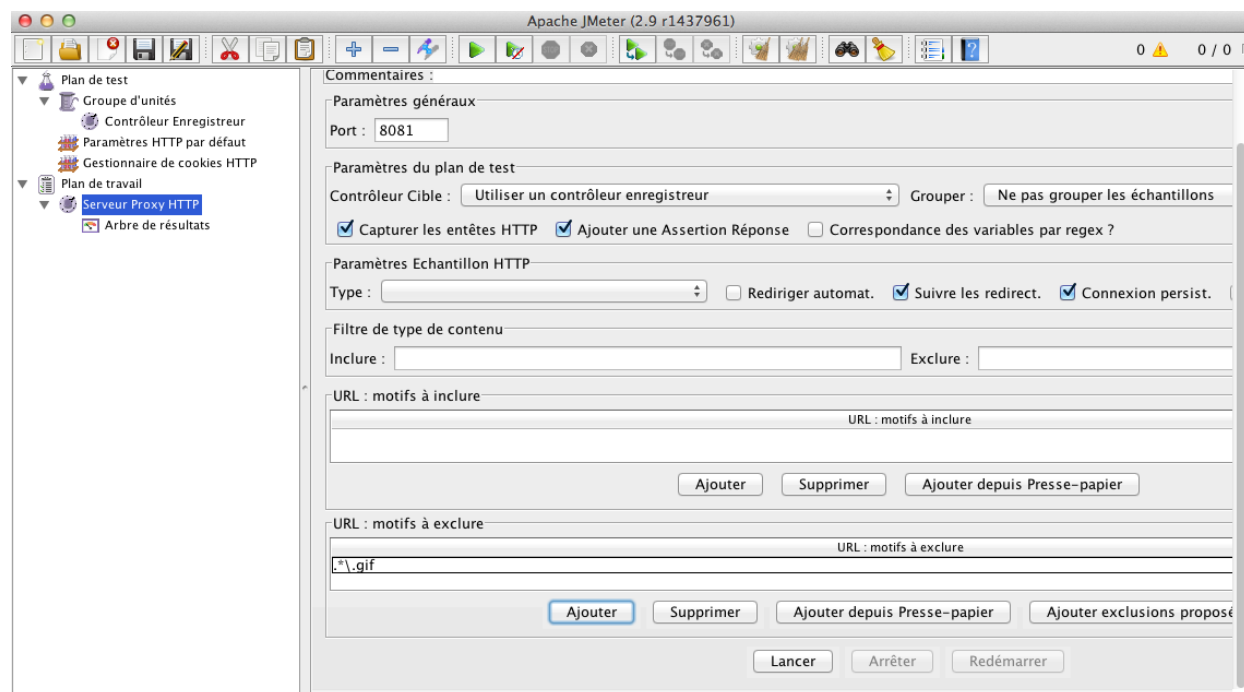


|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |


Avant de faire le démarrage du serveur Proxy de JMeter, on ira cocher le paramètre **Ajouter une Assertion Réponse** dans l'élément **Serveur Proxy HTTP**. Ce paramètre ajoutera pour chaque **Requête HTTP** un sous-élément assertion qui nous servira pour vérifier la validité de la réponse reçue lors du test de charges.

On va ensuite exclure les images GIF de l'enregistrement. Pour cela, dans la partie **URL : motifs à exclure**, on va ajouter la valeur suivante « **.\*\.gif** » qui est l'expression régulière pour les URL qui contiennent un nom de fichier GIF.

**A noter :** l'exclusion des éléments associés à une page, c'est-à-dire les images, feuilles de style, etc. est un choix à faire en fonction du type de test de charge voulu. Par exemple pour un test de charges dont l'objectif est la validation de la tenue de performance du serveur d'applications et des temps de réponse, il est préférable de ne pas inclure les images dans le scénario de tir. Par contre, si l'objectif est la validation de la bande passante, les images et autres éléments sont à inclure.

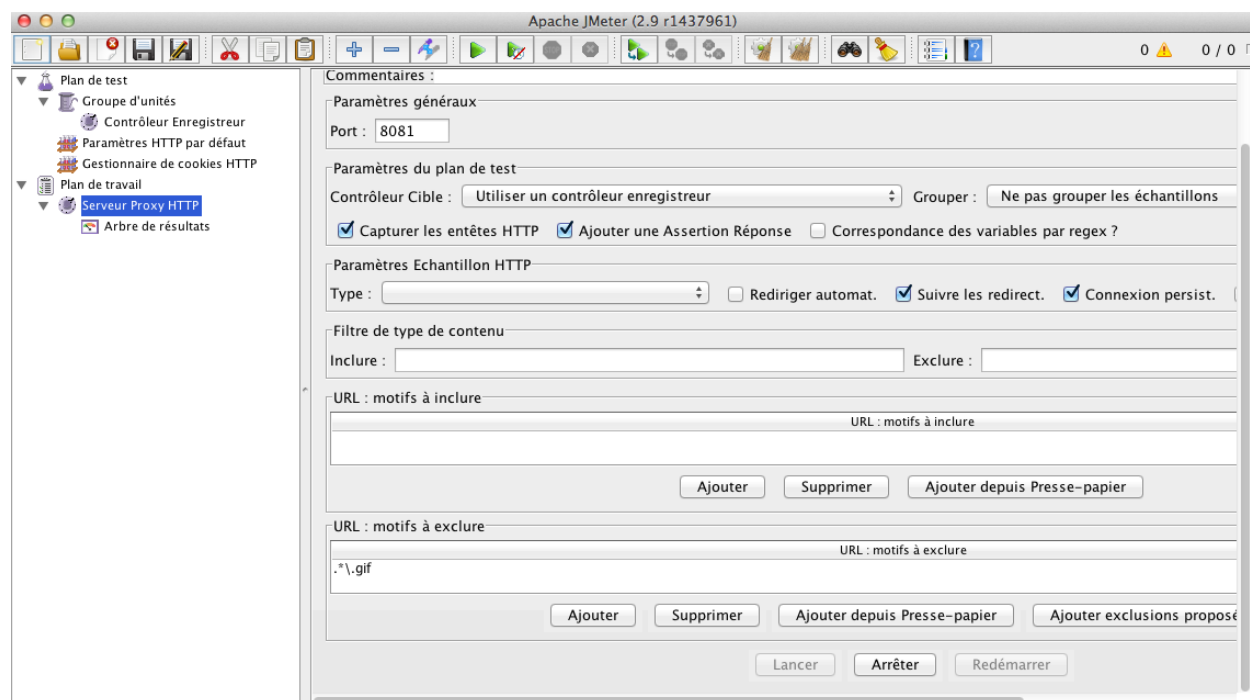





|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

### Démarrage du serveur Proxy JMeter

Par défaut le serveur Proxy n'est pas lancé, il faut donc le démarrer pour qu'il puisse répondre aux demandes (requêtes) du navigateur qui seront faites pendant l'enregistrement de la session de navigation. Pour cela, on se place sur l'élément **Serveur Proxy HTTP** et on clique sur le bouton **Lancer** situé en bas.



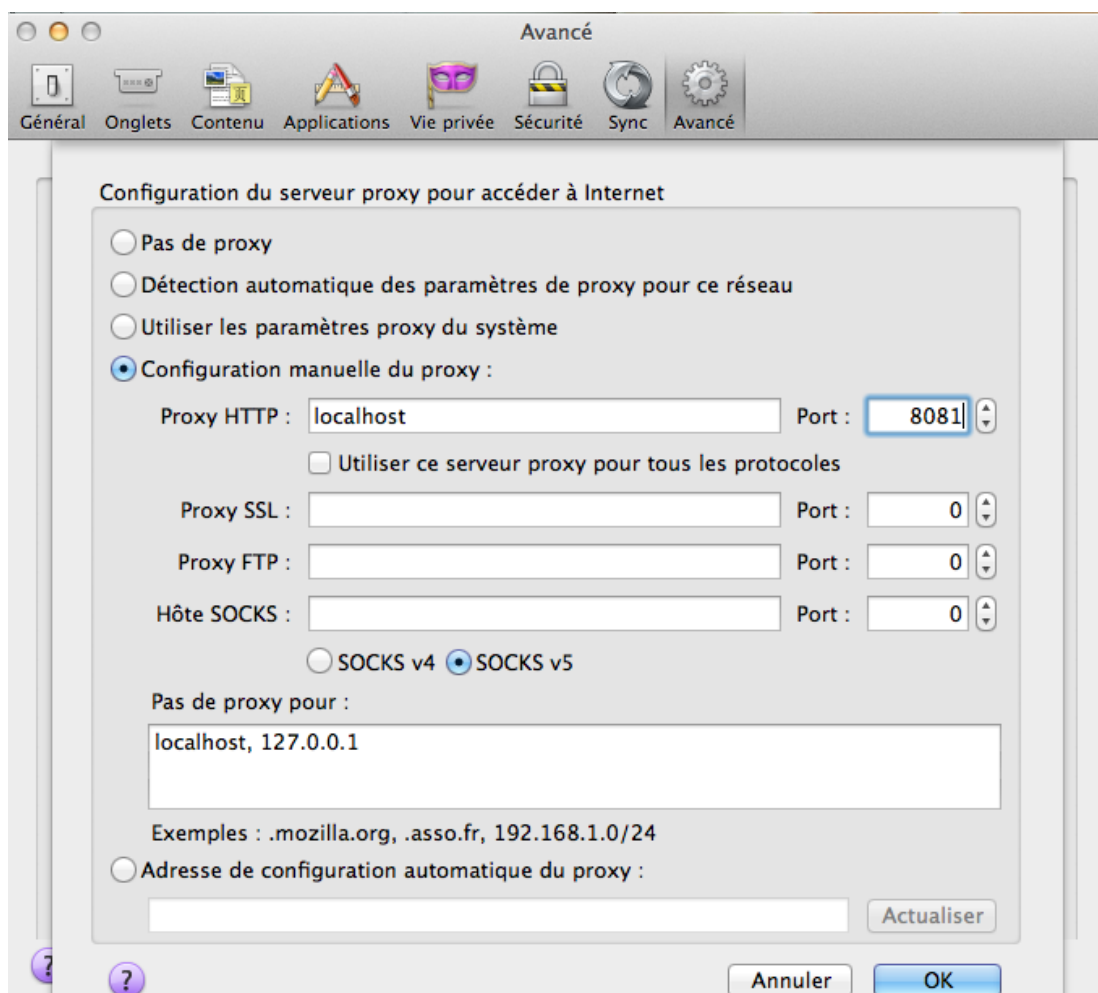
|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |


### Configuration du navigateur

Il faut ensuite configurer le navigateur pour qu'il utilise JMeter comme proxy. Par exemple avec le navigateur Firefox, menu **Edition > Préférences**.

On renseigne les champs suivants :

- Configuration manuelle du proxy choisi
- Proxy : localhost (correspondant à l'adresse IP loopback 127.0.0.1)
- Port : 8080 (ce numéro de port doit être le même que la valeur du champ **Port** dans l'élément **Serveur Proxy HTTP** de JMeter, si par exemple, vous avez mis 8081 car le Tomcat est aussi le poste JMeter, alors, il faut mettre 8081 sur la configuration proxy de votre navigateur)



|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

Également on configurera le navigateur pour ne pas avoir de cache, ainsi on videra son cache de navigateur avant la session d'enregistrement. Si on ne fait pas cela, on aura pour certains éléments des codes de retours HTTP 304 (i.e. Déjà dans le cache du navigateur) lors de leur premier appel. Et donc le test de charges sera faussé, car le robot ne récupéra pas « réellement » les éléments.


**Contenu web en cache**

Le contenu web en cache utilise actuellement 0 octets d'espace disque

Vider maintenant

☐ Modifier la gestion automatique du cache

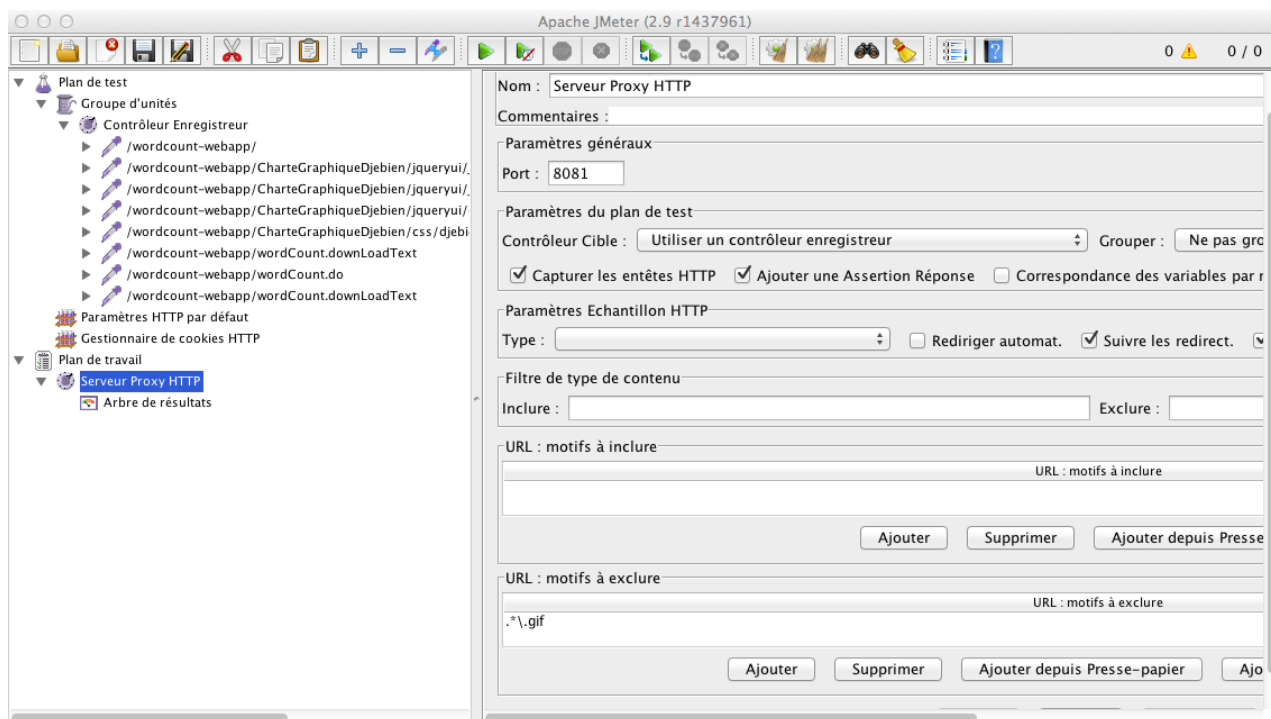
Limiter le cache à  Mo d'espace disque


|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

Et hop, tout est maintenant prêt pour démarrer sa navigation en suivant le scénario fonctionnel établi.

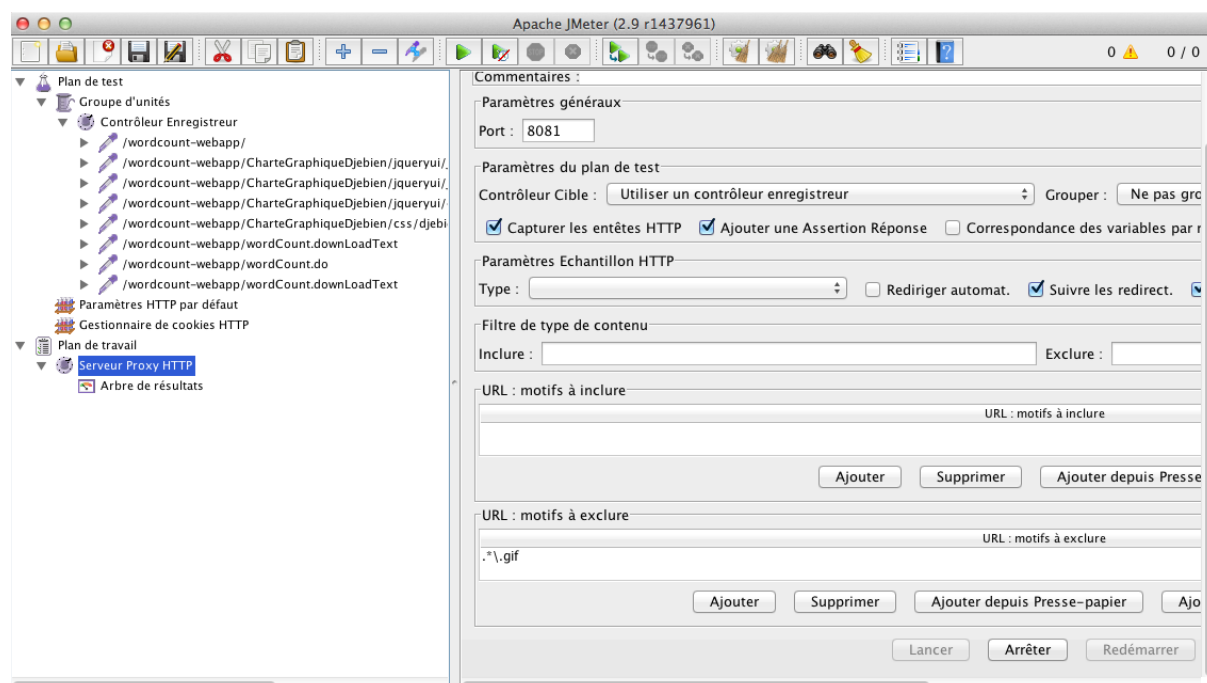
**On prend donc son navigateur Internet (ici Firefox) et on reproduit le scénario fonctionnel étape par étape**, en étant calme et concentré, car tout ce l'on fait est enregistré par JMeter.

Pendant cette navigation, on remarquera dans la fenêtre JMeter que de nouveaux éléments de type **Requête HTTP** viennent s'ajouter en sous-éléments du **Contrôleur Enregistreur**.




|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

Une fois sa session de navigation finie, on clique sur le bouton **Arrêter** de l'élément **Serveur Proxy HTTP**.



Et voilà le résultat de l'enregistrement au niveau de JMeter : 8 requêtes HTTP dans le **Contrôleur Enregistreur**.

|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |



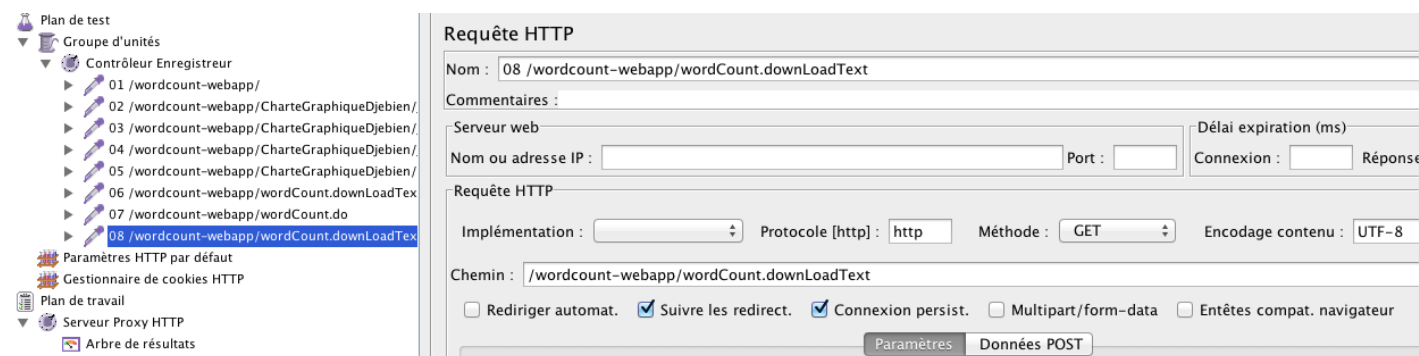
## Affinage du scénario de tests


### Affinage du scénario de tests


JMeter avec sa fonction de proxy nous a permis de faire automatiquement un gros travail d'écriture du scénario de tir. Cependant il reste un travail de vérification et d'affinage à faire.

### Ajout des identifiants des Requêtes HTTP

Afin de mieux retrouver les différents éléments testés dans les récepteurs de résultats, nous allons ajouter un identifiant dans le nom des éléments **Requêtes HTTP**. Ainsi avant le champ **Nom** correspondait à l'URI de la page, maintenant ce sera un identifiant + l'URI de la page. Par exemple, avant le nom était : « /wordcount-webapp/wordCount.do », ensuite le nom deviens : « 07 /wordcount-webapp/wordCount.do ». Le 07 pour la septième requête.

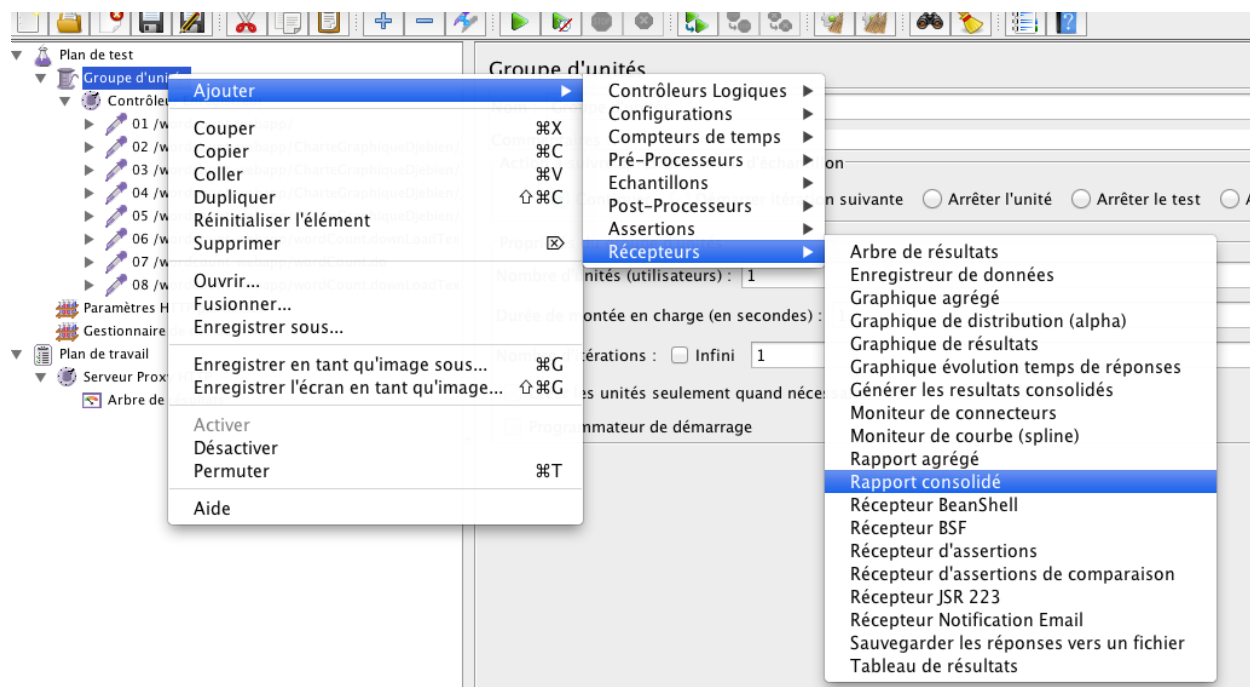


|  |                            |                           |
|--|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br><small>Sciences et Technologies</small> | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013   | Manuel développeur         | Current doc version : 2.0 |

|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

### Ajout d'élément de type Récepteur


Maintenant nous allons ajouter à notre scénario de tir des nouveaux éléments afin de pouvoir « voir » les résultats d'un test de charge. Pour cela, on se place sur l'item **Groupe d'unités**, et avec un clic droit avec la souris on ajoute un **Rapport Consolidé**.

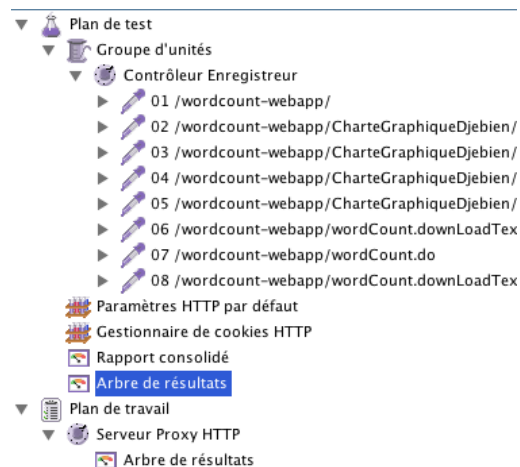


Et on termine avec un élément **Récepteurs > Arbre de résultats**.

On a donc pour finir, deux éléments *récepteurs* différents pour notre scénario de tir.



|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |




## Exécuter son scénario de tir de charge


Bonjour Mr Rouvoy,

étant provisoirement dans l'incapacité d'écrire normalement, je n'ai pas pu terminer mon tutoriel suite à mon accident de la main. Je vous remet donc mon tutoriel dans l'état et je vous transmettrai par mail une version finalisée une fois guéri. Merci de votre compréhension.

Cordialement,

|  |                            |                           |
|--|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br><small>Sciences et Technologies</small> | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013   | Manuel développeur         | Current doc version : 2.0 |

Tarik.

|   |                            |                           |
|---|----------------------------|---------------------------|
|  <b>Université<br/>Lille1</b><br>Sciences et Technologies | <b>Qualité du logiciel</b> | Mesure de performances    |
| Latest update : 11/03/2013  | Manuel développeur         | Current doc version : 2.0 |

## Faire un tir de charge par paliers et exploiter les résultats

### Identification scénario et préparation des paliers

### Personnalisation, Enregistrement du scénario et mise en place des paliers

### Exécution du tir de charge par paliers

### Génération de graphique

Fin du tutoriel.