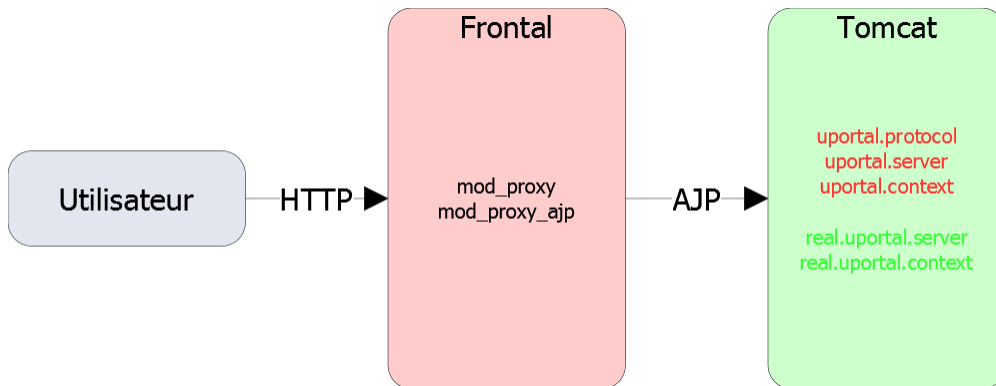


Apache frontal (esup 4)

Mise en frontal d'un serveur Apache

L'usage d'un Frontal (notamment Apache dans notre communauté) devant Tomcat est assez usuel et permet de mettre en oeuvre plus facilement certaines fonctionnalités techniques comme le load-balancing (ou/et le failover), la mise en place du https (SSL), l'[optimisation des paramètres caches HTTP](#), l'envoi de page de "maintenance" lors d'un redémarrage du serveur applicatif, etc.)

Voici le schéma décrivant une telle architecture :



Quelques explications à propos du schéma ci-dessus

- Les flèches représentent les liens et le protocole utilisé
- Le texte en couleur représente l'adresse de l'entité de la même couleur
- L'emplacement des "mod_..." et des propriétés définies dans le filtre (`uportal.protocol`, `uportal.server`, ...) représentent l'emplacement de stockage de l'information
- Le [fichier source du schéma est disponible en PJ](#) (créé avec le logiciel DIA)

Les instructions ci-dessous permettent de connecter un serveur Apache frontal sur une instance uPortal.

⚠ Deux cas seront distingués :

- Un mode simple où le serveur Apache n'est connecté qu'à un serveur applicatif hébergeant uPortal,
- Un mode de load balancing, où le serveur Apache redistribue les requêtes entre plusieurs serveurs en fonction de la charge.

⚠ Sécurité

Pour l'utilisation du protocole HTTPS avec le serveur Apache, il conviendra de définir un virtualhost sur le port 443.
http://httpd.apache.org/docs/2.4/fr/ssl/ssl_howto.html

Pour des raisons de sécurité, nous recommandons de diffuser votre application ENT via HTTPS.

Mode Simple

On est dans le cas de figure d'une architecture composée d'un serveur Apache et d'un serveur applicatif (Tomcat).

Configuration de l'application

On va commencer par configurer l'instance de l'application en modifiant le fichier `filters/esup.properties` qui regroupe les paramètres principaux (généraux).

filters/esup.properties

```
...
#####
## uPortal server configuration properties                                ##
#####
environment.build.uportal.server=ent.univ.fr
environment.build.real.uportal.server=ent.univ.fr
environment.build.uportal.protocol=https
environment.build.uportal.context=/uPortal
environment.build.real.uportal.context=/uPortal
...
```

La ligne *environment.build.uportal.server* désigne l'adresse (et le port si nécessaire) du virtualhost du serveur frontal Apache par lequel l'utilisateur va se connecter à son ENT. Dans le cadre d'un ENT en load-balancing (avec plusieurs serveurs d'applications) cette adresse reste la même quel que soit l'instance (tomcat) de l'ENT.

La ligne *environment.build.real.uportal.server* désigne l'adresse par lequel CAS pourra se connecter en HTTPS à un ENT précis. Dans le cadre d'un ENT en load-balancing (avec plusieurs serveurs d'applications) cette adresse sera spécifique à l'instance considérée. Elle pourra correspondre à un virtualhost Apache spécifique et dédié à un ProxyPass sur uniquement cette instance ENT (tomcat).

Les lignes *environment.build.uportal.context* et *environment.build.real.uportal.context* permettent de définir la racine du contexte de l'application (on laissera comme donné par défaut /uPortal).

Enfin, le paramètre *environment.build.uportal.protocol* permet de définir le protocole utilisé pour se connecter à l'application (http/https).

Bonne pratique

La propriété *environment.build.real.uportal.server* est dépendant de l'instance Tomcat (== souvent la machine) sur lequel l'application tournera ; pour avoir tout de même les mêmes configurations (webapp) sur les différentes machines, on conseille ici d'utiliser les system properties Java :

exemple : *environment.build.real.uportal.server=\${runnEntServerHostNum}*
... puis lors de l'exécution on pourra ajouter un
export JAVA_OPTS="\$JAVA_OPTS -DrunnEntServerHostNum=ent1.univ.fr"
dans le fichier *env.sh* de la machine considérée.

Configuration du serveur Tomcat

On va ensuite modifier le fichier *server.xml* configurant le serveur Tomcat.

Dans un premier temps, on peut commenter le connecteur HTTP par défaut (8080) si celui-ci ne sert plus :

server.xml

```
...
<!-- <Connector port="8080" protocol="HTTP/1.1"
      connectionTimeout="20000"
      redirectPort="8443" emptySessionPath="true" /> -->
...
```

On peut ensuite éditer le connecteur AJP pour y modifier quelques attributs de manière usuelle :

server.xml

```
...
<Connector port="8009"
      enableLookups="false" redirectPort="8443" tomcatAuthentication="false" protocol="AJP/1.3"
      emptySessionPath="true" URIEncoding="UTF-8" />
...
```

Pour rappel, le paramètre *port* définit celui qui sera utilisé pour la connexion entre les deux serveurs avec Apache (8009 par défaut pour AJP).

Configuration du serveur Apache



Fichiers de configuration

Les fichiers de configuration impactés étant assez différents selon les distributions, il sera nécessaire de définir les différents éléments spécifiés en fonction de la distribution (httpd.conf , fichier inclus...).

Maintenant que le serveur Tomcat est configuré pour recevoir les requêtes du serveur Apache, il faut configurer ce dernier pour les transmettre au bon port, via le protocole AJP. La configuration va se faire par le fichier de configuration d'Apache.

Pour permettre la communication entre les deux serveurs, on utilisera 2 modules :

- le mod_proxy
- le mod_proxy_ajp, qui autorisera l'utilisation du protocole AJP.

L'activation de ces modules se fait via les deux lignes suivantes (à ajouter, décommenter, activer ... suivant votre distribution : sous distribution type debian l'utilitaire **a2enmod** est très efficace) :

Fichier de configuration Apache

```
...
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
...
```

Une fois ces deux modules activés, on va rajouter dans le virtualhost cible la configuration nécessaire au bon fonctionnement du mod proxy_ajp :

Fichier de configuration Apache

```
#Configuration du mod_proxy_ajp pour le frontal
<VirtualHost *:443>
...
ProxyRequests Off
ProxyPass          / ajp://tomcat1.univ.fr:8009/
...
</VirtualHost>
```

`<VirtualHost>` permet de définir un hôte virtuel pour lequel les paramètres de balancing seront appliqués. Plusieurs hôtes peuvent être configurés avec des stratégies différentes.

La ligne ProxyPass se décompose en deux paramètres importants :

- `/` qui désigne l'URL à rediriger de l'hôte virtuel du serveur Apache vers Tomcat. Ici, `/` redirigera toutes les requêtes.
- `ajp://localhost:8009/` qui désigne l'adresse vers laquelle rediriger la requête. `localhost` définit l'adresse du serveur Tomcat, et le port (ici 8009) doit être identique à celui configuré dans le connecteur AJP de la configuration du serveur Tomcat.

Au redémarrage des deux serveurs, la configuration sera mise en place. En accédant à l'url <https://ent.univ.fr/uPortal/> on devrait donc arriver à la page d'accueil du portail.

Pour que <https://ent.univ.fr> redirige directement sur /uPortal on pourra également ajouter un rewrite du type :

Fichier de configuration Apache

```
<VirtualHost *:443>
...
RewriteEngine on
RewriteRule ^/$ /uPortal [L,R]
...
</VirtualHost>
```

Load Balancing



La configuration du load balancing via un serveur Apache est détaillée sur [cette page](#).



Références

<https://wiki.jasig.org/display/UPM40/Fronting+Tomcat+with+Apache+HTTP+Server>

http://httpd.apache.org/docs/2.2/mod/mod_proxy.html

http://httpd.apache.org/docs/2.2/mod/mod_proxy_ajp.html

[Exemple de fichier de configuration](#)