

# Guide de paramétrage CAS/SSO LemonLDAP

## WebGFC

## Manuel de configuration

## Document

<b>Auteur</b>	Arnaud AUZOLAT	<b>Date de diffusion</b>	01/08/2016
<b>Responsable de Pôle</b>	Arnaud AUZOLAT	<b>N° de version [révision]</b>	1.0 [112]

## Évolution du document

Version	Auteur	Nature des changements	Date
1.0	Arnaud AUZOLAT	Création du document	27/06/2016
1.1	Arnaud AUZOLAT	Modification du point 4.3.1 Librairie phpcas	01/08/2016
1.2	Arnaud AUZOLAT	Ajout de la partie 4.3.2 pour les patchs en BDD	04/08/2016

## Licence

Ce document n'est pas libre de droits.

Ce manuel est publié sous la licence Creative Commons avec les particularités "Paternité – Partage à l'identique" (également connue sous l'acronyme CC BY-SA).

Détails de cette licence : <http://creativecommons.org/licenses/by-sa/2.0/fr/>




# Manuel de configuration

## Table des matières

---

1.PRÉAMBULE.....	4
2.PRÉ-REQUIS LOGICIELS.....	4
3.LEMONLDAP::NG - INSTALLATION ET CONFIGURATION.....	4
3.1.PRÉ-REQUIS.....	5
3.2. INSTALLATION.....	6
3.3.CONFIGURATION APACHE2.....	6
3.4. CONFIGURATIONS DE LEMONLDAP::NG.....	7
3.4.1.Configuration DNS.....	8
3.5.UTILISATION DE L'ANNUAIRE LDAP.....	9
3.5.1.Configuration LDAP.....	10
3.6.AJOUT D'UNE NOUVELLE APPLICATION.....	11
3.6.1.Sur LemonLDAP::NG.....	12
3.6.2.Sur le serveur hébergeant les machines.....	14
4.CONFIGURATION DU SERVEUR CAS.....	16
4.1.CÔTÉ APACHE 2.....	16
4.1.1.Certificat serveur wildcard.....	16
4.1.2.Passage des applications en HTTPS.....	16
4.2.CÔTÉ LEMONLDAP::NG.....	17
4.3.CÔTÉ WEB-GFC.....	18
4.3.1.Librairie phpCAS.....	18
4.3.2.Mise à jour des bases de données.....	19
4.3.3.Formulaire de connexion CAS dans web-GFC.....	19
4.3.4.Connexion à web-GFC.....	21

Vous pouvez accéder à chaque chapitre en cliquant directement sur la ligne. L'icône  en bas de chaque page permet de revenir à cette table des matières.

# 1. PRÉAMBULE

Le but est de proposer une solution d'authentification unique et centralisée (SSO (Single Sign On)) pour web-GFC.

C'est-à-dire déléguer la phase d'authentification à une application tierce. Cela apporte deux avantages :

- Pour l'utilisateur, il s'authentifie une seule fois et est ré-authentifié automatiquement sur d'autres applications connectées elles-mêmes au serveur permettant l'authentification unique
- Les applications n'ont plus le mot de passe de l'utilisateur à gérer car il n'y a plus qu'une seule application qui peut interagir le cas échéant avec les annuaires, d'où une sécurité accrue.

Cette solution va s'appuyer sur l'utilisation d'un serveur CAS (Central Authentication Service) et du logiciel LemonLDAP::NG<sup>1</sup>.

## 2. PRÉ-REQUIS LOGICIELS

Les distributions Linux de référence sont :

Pour le serveur virtuel qui compose la plate-forme, Ubuntu server 12.04 LTS (plate-forme supportée et support long terme par la société Canonical Ltd.).

La configuration citée dans cette documentation doit être modifiée et adaptée à vos choix.

Pendant l'installation, le serveur doit être connecté sur un réseau relié à Internet afin de récupérer et installer les dernières mises-à-jour de composants logiciels disponibles.

Les pré requis pour web-GFC sont listés sur la FAQ :

<http://faq.adullact.org/pre-requis/101-pre-requis-pour-une-installation-de-web-gfc-v1-7-0>

Les pré-requis pour LemonLDAP::NG sont listés ici :

<http://lemonldap-ng.org/documentation/1.9/prereq>

## 3. LEMONLDAP::NG - INSTALLATION ET CONFIGURATION

Ce document décrit l'installation pour la dernière version stable de LemonLDAP::NG : la **version 1.9**.

Pour tous renseignements, ne pas hésiter à se référer à la documentation fournie par LemonLDAP::NG :

<http://lemonldap-ng.org/documentation/1.9/start>

LemonLDAP::NG est une infrastructure d'authentification unique distribuée avec gestion centralisée des droits. Il se présente sous la forme d'une suite logicielle reposant notamment sur le serveur web Apache2<sup>2</sup>

Il implémente à la fois :

- les services de fournisseur d'identité et de service SAML1, OpenID, CAS, OpenID Connect2 ;
- l'authentification basée sur LDAP, Kerberos, SQL, Twitter et d'autres protocoles ;
- un système d'authentification unique basé sur des cookies sécurisés ;
- un dispositif de notification ;
- un explorateur de sessions.

Pour plus de détails voir le site dédié à LemonLDAP::NG (<http://lemonldap-ng.org>)

NB : Il existe de nombreux paramétrages pour le portail LemonLDAP::NG qui offre de nombreuses fonctionnalités mais tous ne seront pas traités dans ce document.

---

1 Source : <http://lemonldap-ng.org/welcome/>

2 Source : <https://httpd.apache.org/docs/2.4/fr/getting-started.html>

### 3.1. PRÉ-REQUIS

Installer les paquets nécessaires au fonctionnement du serveur web :

```
# apt-get install libapache-session-perl libnet-ldap-perl libcache-cache-perl libdbi-perl perl-modules
libwww-perl libcache-cache-perl libxml-simple-perl libsoap-lite-perl libhtml-template-perl libregexp-
assemble-perl libregexp-common-perl libjs-jquery libxml-libxml-perl libcrypt-rijndael-perl libio-string-
perl libxml-libxslt-perl libconfig-inifiles-perl libjson-perl libstring-random-perl libemail-date-format-
perl libmime-lite-perl libcrypt-openssl-rsa-perl libdigest-hmac-perl libdigest-sha-perl libclone-perl
libauthen-sasl-perl libnet-cidr-lite-perl libcrypt-openssl-x509-perl libauthcas-perl libtest-pod-perl
libtest-mockobject-perl libauthen-captcha-perl libnet-openid-consumer-perl libnet-openid-server-perl
libunicode-string-perl libconvert-pem-perl libmouse-perl libplack-perl apache2 libapache2-mod-perl2
libapache2-mod-fcgid
```

(Voir la page <http://lemonldap-ng.org/documentation/1.9/installdeb>)

Ajout dans les sources (en fin de fichier) :

```
# vi /etc/apt/sources.list
```

```
# LemonLDAP::NG
deb http://us.archive.ubuntu.com/ubuntu precise main universe
```

Recherche des packages LemonLDAP sous Ubuntu 12.04 :

```
# sudo apt-cache search lemonldap-ng
```

```
lemonldap-ng - OpenID-Connect, CAS and SAML compatible Web-SSO system
lemonldap-ng-doc - Lemonldap::NG Web-SSO system documentation
lemonldap-ng-fastcgi-server - Lemonldap::NG FastCGI server
lemonldap-ng-fr-doc - French documentation of Lemonldap::NG Web-SSO system
liblemonldap-ng-common-perl - Lemonldap::NG common files
liblemonldap-ng-conf-perl - transitional dummy package
liblemonldap-ng-handler-perl - Lemonldap::NG Apache handler part
liblemonldap-ng-manager-perl - Lemonldap::NG manager part
liblemonldap-ng-portal-perl - Lemonldap::NG authentication portal part
```

Mise à jour des sources de LemonLDAP::NG

```
# vi /etc/apt/sources.list.d/lemonldap-ng.list
```

Ajouter les lignes suivantes en fin de fichier :

```
# LemonLDAP::NG repository
deb http://lemonldap-ng.org/deb stable main
deb-src http://lemonldap-ng.org/deb stable main
```

Récupérer la GPG key depuis le site : [http://lemonldap-ng.org/\\_media/rpm-gpg-key-ow2](http://lemonldap-ng.org/_media/rpm-gpg-key-ow2)

```
# cd /tmp
# wget http://lemonldap-ng.org/_media/rpm-gpg-key-ow2
# sudo apt-key add rpm-gpg-key-ow2
```

Mise à jour des sources de la machine suite à l'ajout des nouvelles sources:

```
# sudo apt-get update
```

## 3.2. INSTALLATION

Pour l'installation de LemonLDAP::NG, une simple commande est à utiliser :

```
sudo aptitude install lemondldap-ng lemondldap-ng-fr-doc
```

Vous devriez avoir comme fichiers installés :

```
# cd /etc/lemondldap-ng
# ls -l
total 80
drwxr-xr-x  2 root root   4096 juin  22 11:06 ./
drwxr-xr-x 149 root root  12288 juin  27 08:53 ../
-rw-r--r--  1 root root    93 mai  16 10:17 for_etc_hosts
-rw-r--r--  1 root root   2281 mai  16 10:20 handler-apache2.conf
-rw-r--r--  1 root root   1385 mai  16 10:17 handler-nginx.conf
-rw-r----- 1 root www-data 10600 juin  22 10:58 lemondldap-ng.ini
-rw-r--r--  1 root root   4138 mai  16 10:21 manager-apache2.conf
-rw-r--r--  1 root root    972 mai  16 10:17 manager-nginx.conf
-rw-r--r--  1 root root   4595 juin  22 09:05 portal-apache2.conf
-rw-r--r--  1 root root   4619 juin  22 09:09 portal-apache2.ssl.conf
-rw-r--r--  1 root root   1634 mai  16 10:17 portal-nginx.conf
-rw-r--r--  1 root root   1322 mai  16 10:22 test-apache2.conf
-rw-r--r--  1 root root   2270 mai  16 10:17 test-nginx.conf
```

NB : web-GFC étant compatible avec *Apache2* mais pas *Nginx*<sup>3</sup>, nous nous occuperons uniquement des fichiers *\*\*\*\*-apache2.conf*

Mais également :

```
# cd /var/lib/lemondldap-ng
# ls -l
drwxr-xr-x 10 root  root   4096 mai  16 14:36 ./
drwxr-xr-x 70 root  root   4096 juin  15 13:56 ../
drwxrwx--- 2 www-data www-data 4096 mai   1 21:33 captcha/
drwxr-x--- 2 www-data www-data 4096 juin  24 10:12 conf/
drwxr-xr-x 2 root  root   4096 mai  16 10:16 manager/
drwxrwx--- 2 www-data www-data 4096 mai   1 21:33 notifications/
drwxr-xr-x 3 root  root   4096 mai  16 10:16 portal/
drwxrwx--- 3 www-data www-data 4096 mai  17 17:03 psessions/
drwxrwx--- 3 www-data www-data 4096 juin  27 09:01 sessions/
drwxr-xr-x 2 root  root   4096 mai  16 11:54 test/
```

## 3.3. CONFIGURATION APACHE2

Une fois l'installation de LemonLDAP::NG réalisée, il faut paramétrer le service Apache2. Il faut tout d'abord ajouter les lignes suivantes dans le fichier de configuration d'Apache2 :

```
# vi /etc/apache2/apache2.conf
```

On ajoute les informations suivantes en fin de fichier :

```
# Lemonldap::NG
include /etc/lemondldap-ng/portal-apache2.conf
include /etc/lemondldap-ng/handler-apache2.conf
include /etc/lemondldap-ng/manager-apache2.conf
include /etc/lemondldap-ng/test-apache2.conf
```

3 Source : <https://www.nginx.com/>

Une fois ces nouvelles configurations ajoutées, il faut activer les sites définis :

```
# a2ensite manager-apache2.conf
# a2ensite portal-apache2.conf
# a2ensite handler-apache2.conf
# a2ensite test-apache2.conf
```

Il reste un petit peu de modification « à la main » dans les fichiers de configuration LemonLDAP en lien avec Apache2. Il s'agit de mettre en commentaire tous les blocs **<IfVersion..>** présents dans les fichiers suivants :

```
# vi /etc/apache2/sites-enabled/handler-apache2.conf
# vi /etc/apache2/sites-enabled/manager-apache2.conf
# vi /etc/apache2/sites-enabled/portal-apache2.conf
# vi /etc/apache2/sites-enabled/test-apache2.conf
```

Les blocs **<IfVersion>** ressemblent à ceux présents en gras ci-dessous :

```
<VirtualHost "*:80">
  ServerName reload.webgfc.fr

  # Configuration reload mechanism (only 1 per physical server is
  # needed): choose your URL to avoid restarting Apache when
  # configuration change
  <Location /reload>
    # <IfVersion >= 2.3>
    # Require ip 127 ::1
    # </IfVersion>
    # <IfVersion < 2.3>
    Order Deny,Allow
    Deny from all
    Allow from 127.0.0.0/8 ::1
    # </IfVersion>
    PerlHeaderParserHandler Lemonldap::NG::Handler->reload
  </Location>
```

Une fois fait, on active les modules suivants :

```
# a2enmod fcgid perl alias rewrite ssl proxy proxy_ajp authnz_idap
```

On relance Apache2 :

```
# service apache2 restart
```

### 3.4. CONFIGURATIONS DE LEMONLDAP::NG

Dans les exemples qui suivent, nous allons nous positionner dans un domaine DNS qui sera nommé **webgfc.fr**.

Par défaut, le nom de domaine DNS est *example.com*.

Nous allons donc utiliser une commande *sed* afin de remplacer tous les intitulés *example.com* en *webgfc.fr*.

```
# sed -i 's/example\.com/webgfc.fr/g' /etc/lemonldap-ng/* /var/lib/lemonldap-ng/conf/lmConf-1.js /var/lib/lemonldap-ng/test/index.pl
```

NB : il vous faudra adapter le nom de domaine à votre instance.

### 3.4.1. CONFIGURATION DNS

Une fois ces informations renseignées, et avant de pouvoir profiter de cette nouvelle installation de LemonLDAP::NG, il reste une étape qui consiste à créer certaines entrées DNS.

Pour fonctionner, LemonLDAP::NG a besoin d'une entrée **reload.<votre-domaine>** qui pointe vers **127.0.0.1**. Cette entrée n'a pas besoin d'être publiée pour tout le monde. Il suffit de modifier le fichier `/etc/hosts` du serveur en y ajoutant une ligne comme ceci :

```
# echo "127.0.0.1 reload.webgfc.fr" >> /etc/hosts
```

Note : il faudra bien entendu remplacer `webgfc.fr` par votre propre domaine.

De plus, afin de pouvoir configurer le serveur pour qu'ils connaissent toutes les URLs, exécuter la ligne suivante :

```
# cat /etc/lemonldap-ng/for_etc_hosts >> /etc/hosts
```

#### 3.4.1.1. ENTRÉES PUBLIQUES

Par défaut, deux entrées sont nécessaires pour utiliser LemonLDAP::NG. Si c'est la première fois que vous installez LemonLDAP::NG, et surtout s'il s'agit pour vous d'un laboratoire, alors deux entrées supplémentaires sont recommandées, afin de tester correctement votre installation.

- **auth.webgfc.fr** (obligatoire) : URL d'accès au portail LemonLDAP::NG. C'est le point d'entrée pour tous vos utilisateurs.
- **manager.webgfc.fr** (obligatoire) : URL d'accès au gestionnaire de configuration LemonLDAP::NG. C'est via cette application que l'administrateur peut mettre à jour la configuration LemonLDAP::NG.
- **test1.webgfc.fr** (facultative) : par défaut, LemonLDAP::NG installe deux applications destinées aux tests. Cette URL permet d'accéder à la première application.
- **test2.webgfc.fr** (facultative) : par défaut, LemonLDAP::NG installe deux applications destinées aux tests. Cette URL permet d'accéder à la deuxième application.

Note : il faudra bien entendu remplacer `webgfc.fr` par votre propre domaine.

À ce stade, tout est prêt pour accéder au portail LemonLDAP::NG.

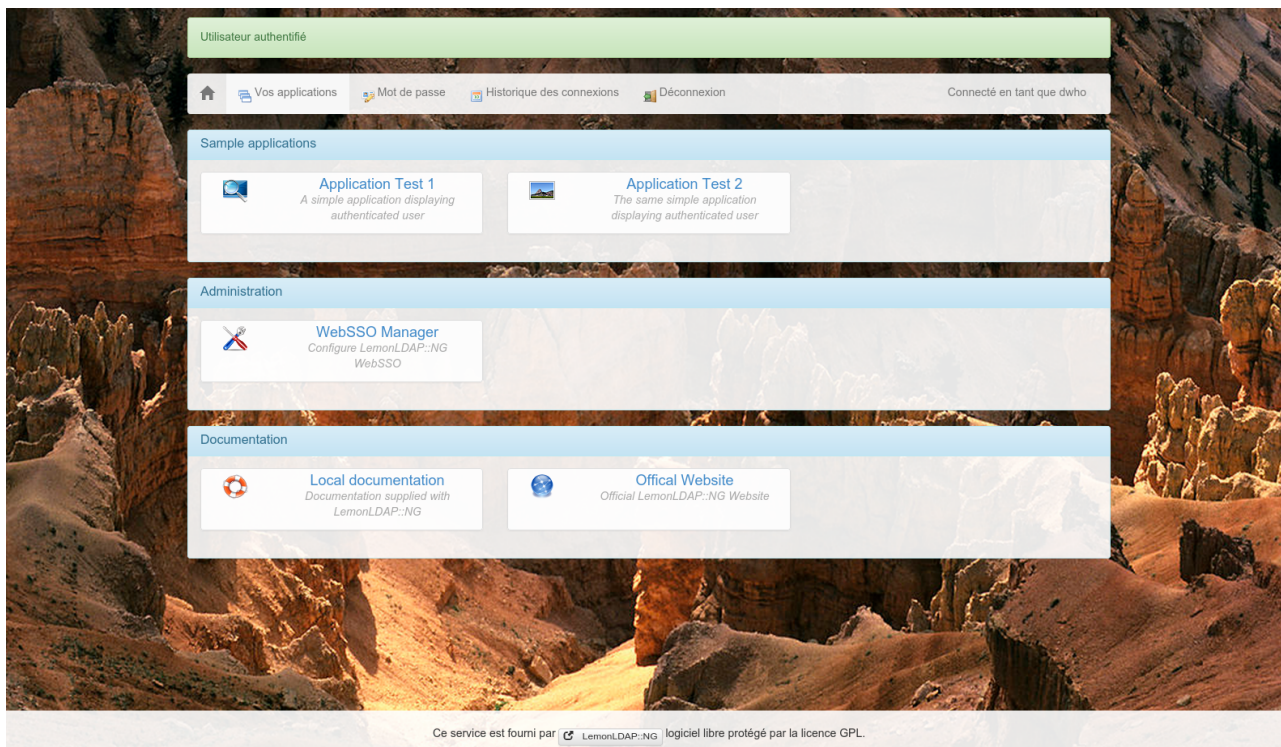
Pour cela, il suffit de se rendre à l'adresse [http://auth.<votre-domaine>/](http://auth.<votre-domaine>) (<http://auth.webgfc.fr> dans notre exemple).

Par défaut, LemonLDAP::NG utilise une base de données locale pour authentifier les utilisateurs.

Cette base est composée, entre autres, de l'utilisateur **dwho**, qui possède les autorisations nécessaires pour configurer LemonLDAP::NG.

Après avoir saisi **dwho** dans les champs *Identifiant* et *Mot de passe*, vous devriez alors avoir accès au portail, à l'image de la capture d'écran ci-dessous :





*Premier accès au portail LemonLDAP:NG*

Sur ce portail, toutes les applications accessibles (pour lesquelles l'utilisateur courant possède les autorisations nécessaires) sont visibles.

Dans notre cas, nous voyons :

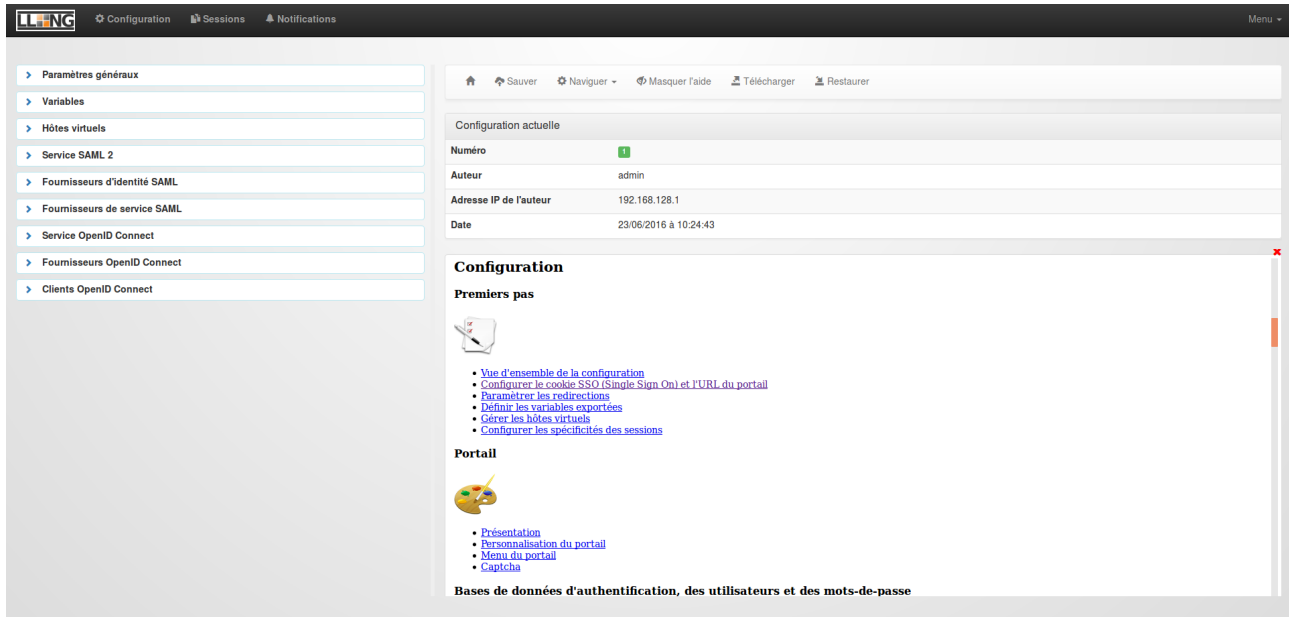
- les deux applications destinées aux tests :
  - Application Test 1
  - et Application Test 2
- l'application d'administration de LemonLDAP:NG
  - WebSSO Manager
- ainsi que des liens vers la documentation LemonLDAP:NG.

### 3.5. UTILISATION DE L'ANNUAIRE LDAP

Pour le moment, LemonLDAP:NG utilise sa base de données locale pour authentifier les utilisateurs. Nous allons donc dans ce chapitre relier LemonLDAP:NG à notre annuaire LDAP.

Il est possible également d'authentifier des utilisateurs via d'autres moyens, notamment via des comptes Google, Facebook ou Twitter. Mais ces aspects ne seront pas traités dans ce document.

Dans un premier temps, il suffit de cliquer sur le lien WebSSO Manager, et d'obtenir le résultat ci-dessous :



Page de configuration du portail LemonLDAP (<http://manager.webgfc.fr> ici)

C'est via cette application que toute la configuration de LemonLDAP::NG sera gérée.

Nous nous intéresserons dans un premier temps à la rubrique *Paramètres généraux* pour utiliser l'annuaire LDAP comme source d'authentification.

Puis dans un autre chapitre, nous définirons nos applications Web via la catégorie *Hôtes virtuels*.

Note : afin de suivre correctement la procédure et ne pas bloquer vos accès, veuillez à ne pas cliquer sur le bouton Sauver avant la fin de la procédure. Sachez qu'il sera toujours possible d'annuler une éventuelle fausse manipulation, en supprimant simplement un fichier de configuration. Ce fichier de configuration se trouve dans `/var/lib/lemonldap-ng/conf`. Tous les fichiers sont de la forme `lmConf-<n°_de_version>.js`

### 3.5.1. CONFIGURATION LDAP

Pour utiliser l'annuaire LDAP, il suffit de suivre la procédure ci-dessous :

**Paramètres généraux** → **Modules d'authentification**, sélectionner **LDAP** pour les trois modules (Module d'authentification, Module d'utilisateurs et Module de mot de passe).

Une fois les paramètres d'authentification sélectionnés sur LDAP, un nouveau menu apparaît : **Paramètres généraux** → **Modules d'authentification** → **Paramètres LDAP**.

Il suffit de cliquer sur **Paramètres LDAP** → **Connexion** afin de définir les différents paramètres :

- Hôte : `ldaps://<adresse_de_votre_serveur_ldap>` (ou `ldap://<adresse_de_votre_serveur_ldap>` si nous n'utilisons pas de connexion chiffrée)
- Port : 636 (ou 389 si vous n'utilisez pas de connexion chiffrée)
- Base de recherche des utilisateurs : `dc=webgfc,dc=fr`
- Compte de connexion LDAP : `<vide>` (chez moi les connexions anonymes à l'annuaire LDAP sont autorisées et ces connexions permettent de récupérer la liste des utilisateurs)
- Mot de passe LDAP : `<vide>` (à saisir si vous avez défini un compte de connexion LDAP)

Il faut ensuite définir les **Groupes** qui sont utilisés pour le serveur LDAP.

Paramètres généraux → Modules d'authentification → Paramètres LDAP → **Groupes**

- Base de recherche : `dc=webgfc,dc=fr`
- Classe d'objet : `inetOrgPerson`
- Attribut cible : `memberUid`
- Attribut source utilisateur : `uid`

- Attribut recherché : <vide>
- Récursif : Désactivé
- Attribut source groupe : <vide>

Enfin, via le menu Paramètres généraux → Modules d'authentification → Paramètres LDAP → **Mot de passe**, on va pouvoir définir notre politique d'accès pour les utilisateurs

- Contrôle password policy : désactivé
- Opération étendue password modify : activé
- Changement en tant qu'utilisateur : activé
- Encodage des mots de passe LDAP : utf-8
- Utiliser l'attribut de réinitialisation : désactivé

À ce stade, LemonLDAP::NG est prêt à utiliser l'annuaire LDAP pour la gestion des utilisateurs. Mais, dans ce cas, l'utilisateur dwho n'existera plus, et c'est le seul utilisateur autorisé à utiliser le manager.

Il va donc falloir modifier la configuration de l'hôte virtuel **manager.<votre-domaine>** pour autoriser l'utilisateur admin et (optionnellement) autoriser tous les utilisateurs qui font partie du groupe administrateurs.

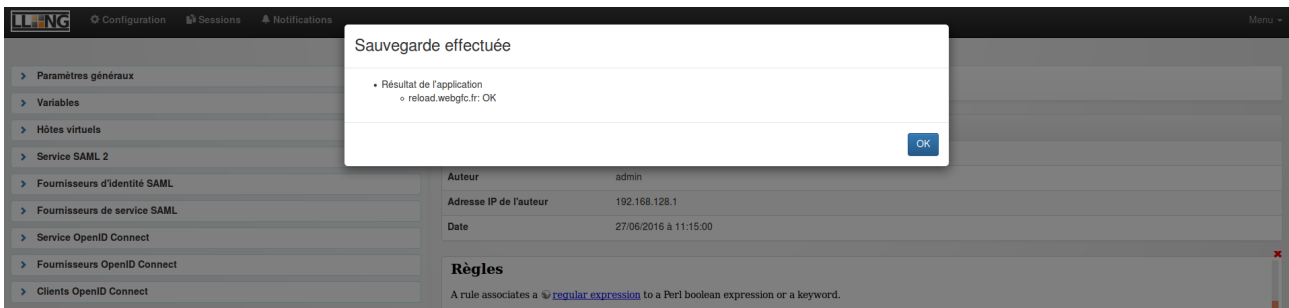
Pour cela, il suffit d'accéder au paramètre **Hôtes virtuels** → **manager.<votre-domaine>** → **Règles d'accès** → **default** et de remplacer :

```
$uid eq "dwho"
```

par :

```
$uid eq "dwho" or $uid eq "admin" or $groups =~ /administrateurs\b/
```

Une fois ces opérations effectuées, il suffit alors de cliquer sur le bouton Sauver, pour indiquer à LemonLDAP::NG de mettre à jour sa configuration :



*Sauvegarde et message de confirmation de mise à jour de la configuration*

Vous pouvez alors tester de vous déconnecter, puis de vous connecter à nouveau à l'aide de l'utilisateur admin ou de tout autre utilisateur faisant partie du groupe administrateurs de votre annuaire LDAP.

Note : si vous ne parvenez pas à vous connecter, vérifiez l'existence de l'utilisateur dans votre annuaire LDAP. Si vraiment vous êtes bloqués, il est possible d'annuler la dernière configuration en supprimant le dernier fichier du répertoire **/var/lib/lemonldap-ng/conf**.

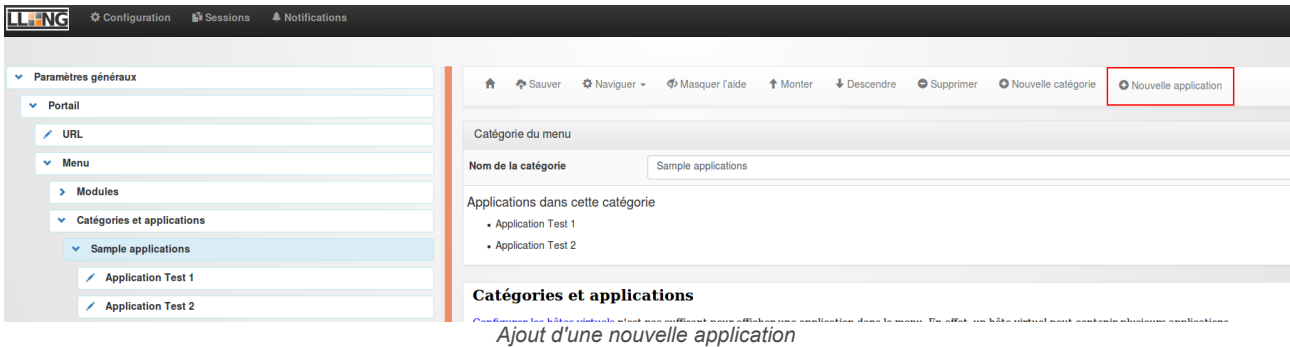
### 3.6. AJOUT D'UNE NOUVELLE APPLICATION

Dans l'exemple ci-dessous, nous allons ajouter l'application **webgfc17.webgfc.fr** correspondant à la version 1.7.0 de web-GFC.

### 3.6.1. SUR LEMONLDAP::NG

Afin de pouvoir ajouter une nouvelle application, il faut se connecter au gestionnaire en tant qu'administrateur (<http://manager.webgfc.fr>), via le bouton *WebSSO Manager*.

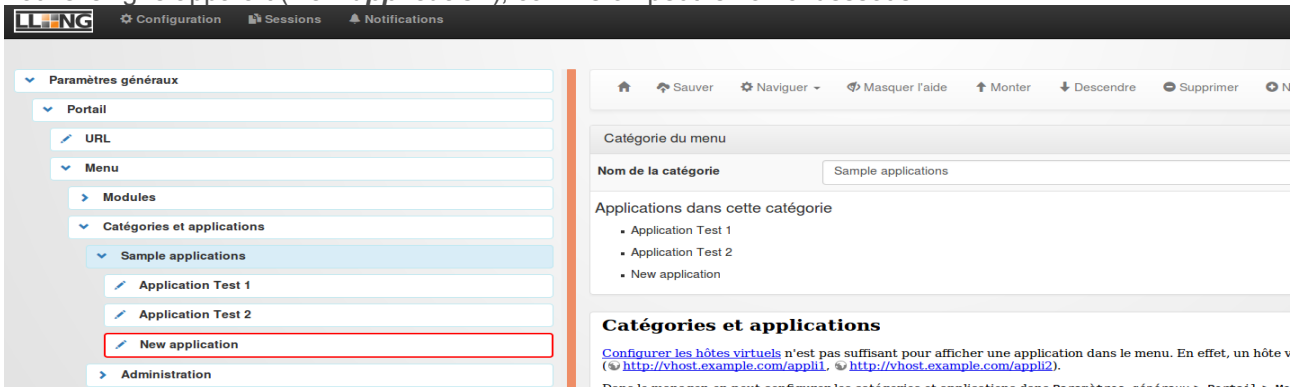
Une fois fait, il faut aller dans **Paramètres Généraux** → **Portail** → **Menu** → **Catégories et applications** → **Sample applications**.



Ajout d'une nouvelle application

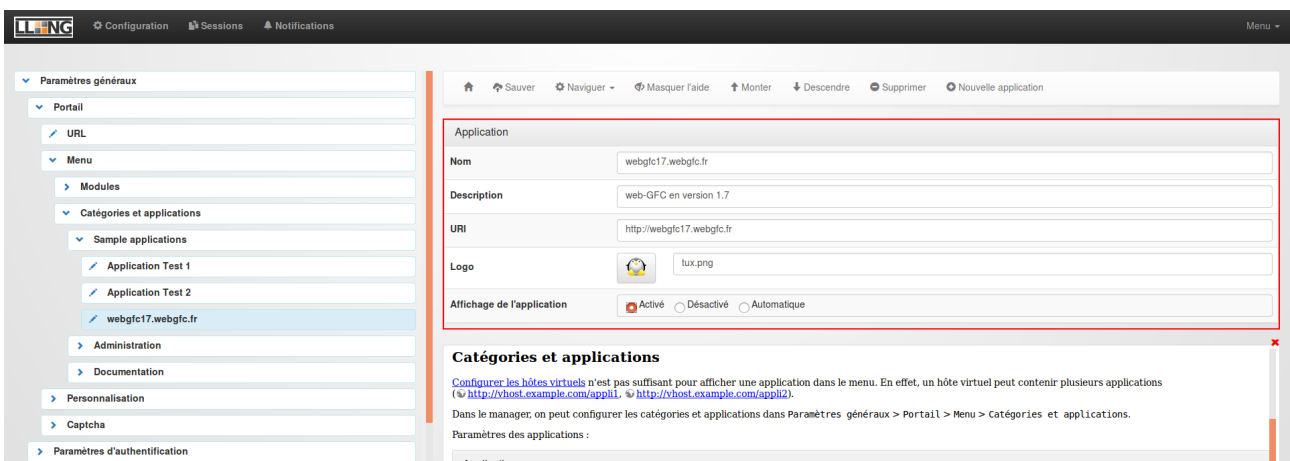
Dans le panneau latéral de droite, un bouton *Nouvelle application* est disponible permettant d'ajouter une nouvelle application.

Une fois que l'administrateur a cliqué sur *Nouvelle application*, dans le panneau latéral de gauche, une nouvelle ligne apparaît (*New application*), comme on peut le voir ci-dessous.



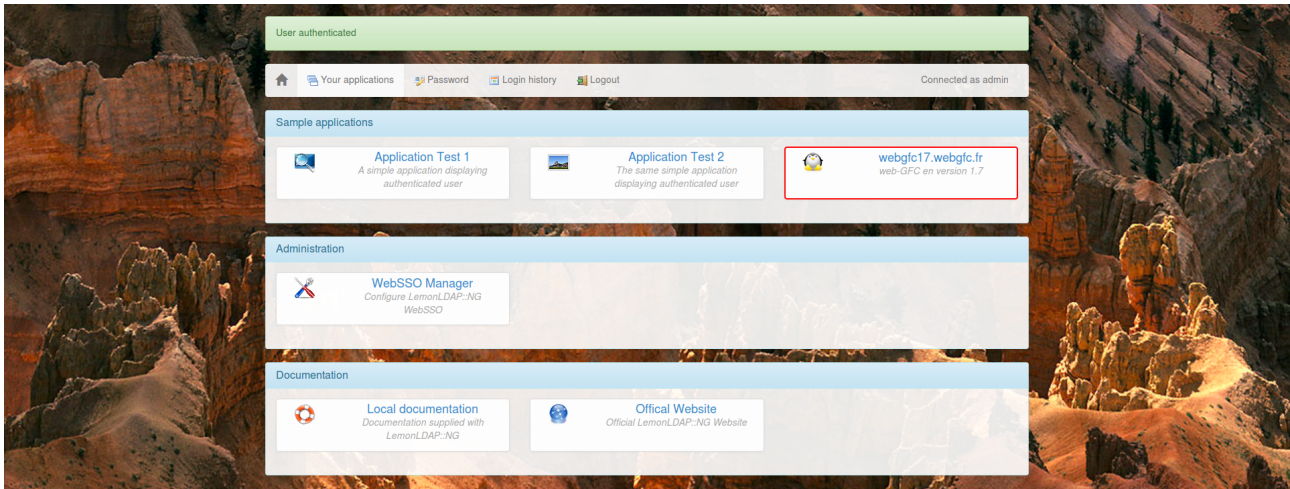
Configurer les hôtes virtuels n'est pas suffisant pour afficher une application dans le menu. En effet, un hôte v (<http://vhost.example.com/appl1>, <http://vhost.example.com/appl2>).

Dès lors, il suffit de cliquer sur cette nouvelle application (*New application*) et de renseigner le formulaire affiché sur la partie droite puis de cliquer sur *Sauver*.



Formulaire d'ajout d'une nouvelle application

NB : A noter qu'il est important de bien cocher **Activé** dans la partie *Affichage de l'application* si on souhaite voir apparaître cette dernière sur l'écran d'accueil du portail.



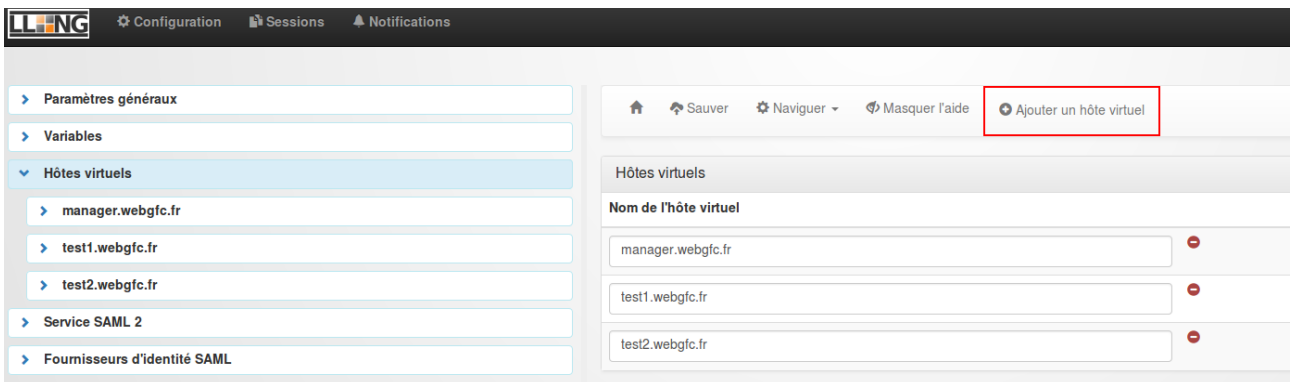
Affichage de la nouvelle application

Ce n'est pas encore totalement finalisé.

Une fois la nouvelle application ajoutée, il faut désormais la renseigner dans les Hôtes virtuels de LemonLDAP::NG.

Pour cela, il faut se rendre sur le gestionnaire en tant qu'administrateur (<http://manager.webgfc.fr>), via le bouton *WebSSO Manager*.

Là, on accède au menu **Hôtes virtuels** et on ajoute un nouvel Hôte virtuel, comme sur la copie d'écran ci-dessous.



Ajout d'un nouvel hôte virtuel

Une fois fait, l'hôte apparaît parmi la liste de ceux présents.



Il ne reste plus qu'à configurer cet hôte virtuel

Pour cela, il suffit de cliquer sur le nouvel hôte virtuel et de renseigner les paramètres suivants :

- Règles d'accès :
  - Commentaires = Règle par défaut
  - Expressions régulières = default
  - Règles = accept
- En-têtes exportées. On clique sur *Nouvelle entrée* et on renseigne :
  - Clefs = Auth-User
  - Valeurs = \$uid

Puis finalement de cliquer sur *Sauver*.

### 3.6.2. SUR LE SERVEUR HÉBERGEANT LES MACHINES

Une fois fait, il faut configurer les Virtual Hosts de la machine hébergeant les applications. En effet, la nouvelle application (dans notre cas <http://webgfc17.webgfc.fr>) doit être connue d'Apache 2.

Nous allons donc ajouter l'application dans les Virtual hosts.

```
# cd /etc/apache2/sites-available
# vi webgfc17.webgfc.fr
```

```
<VirtualHost *:80>
    ServerName    webgfc17.webgfc.fr
    DocumentRoot  /var/www/webgfc

    PerlHeaderParserHandler Lemonldap::NG::Handler # Pour la connexion SSO avec LemonLDAP

    <Directory "/var/www/webgfc/1.7/">
        Options Indexes MultiViews FollowSymlinks
        AllowOverride all
        Order allow,deny
        Allow from all
        ExpiresActive On
        ExpiresByType image/jpg "access plus 2592000 seconds"
        ExpiresByType image/jpeg "access plus 2592000 seconds"
        ExpiresByType image/png "access plus 2592000 seconds"
        ExpiresByType image/gif "access plus 2592000 seconds"
        ExpiresByType text/css "access plus 2592000 seconds"
        ExpiresByType text/javascript "access plus 2592000 seconds"
        ExpiresByType application/javascript A259200
        ExpiresByType application/x-javascript "access plus 2592000 seconds"
    </Directory>
</VirtualHost>
```

```
ExpiresByType application/x-shockwave-flash "access plus 2592000 seconds"
</Directory>

<Directory "/var/www/webgfc/1.7/app/webroot/files/webdav">
    DAV On
    order allow,deny
    allow from all
    #SetHandler default-handler
</Directory>
<Directory "/var/www/webgfc/1.7/app/webroot/files/repository">
    DAV On
    order allow,deny
    allow from all
    #SetHandler default-handler
    AuthType Basic
    AuthName "web-GFC repository"
    AuthUserFile /etc/apache2/webgfc_davlogin/repository
<Limit PUT POST DELETE PROPFIND PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
    Require valid-user
</Limit>
    RewriteEngine off
</Directory>

Alias /webdav /var/www/webgfc/1.7/app/webroot/files/webdav
Alias /repository /var/www/webgfc/1.7/app/webroot/files/repository
</VirtualHost>
```

Il faut ensuite :

- ajouter le site au niveau des vhosts du serveur :

○ **# vi /etc/hosts**

```
#LemonLDAP::NG
127.0.0.1 manager.webgfc.fr auth.webgfc.fr test1.webgfc.fr test2.webgfc.fr reload.webgfc.fr webgfc17.webgfc.fr
```

- activer le site

○ **# a2ensite webgfc17.webgfc.fr**

- et relancer Apache 2

○ **# service apache2 restart**

Une fois fait, un simple clic sur le lien **webgfc17.webgfc.fr** depuis le portail (<http://auth.webgfc.fr>) redirigera l'utilisateur vers l'application web-GFC en version 1.7.0.

## 4. CONFIGURATION DU SERVEUR CAS

### 4.1. CÔTÉ APACHE 2

Avant de pouvoir configurer LemonLDAP::NG et web-GFC pour une connexion CAS, il faut configurer le serveur Apache 2 afin que les différentes briques soient sur une connexion en HTTPS.

#### 4.1.1. CERTIFICAT SERVEUR WILDCARD

Pour cela, tout d'abord, il nous faut un certificat serveur.

Dans l'exemple ci-dessous, nous utilisons la PKI d'ADULLACT (<https://pki.adullact.org>) afin de générer un certificat couvrant le nom de domaine **\*.webgfc.fr**

Le certificat se présente sous la forme de 2 fichiers :

- wildcard.webgfc.fr.key
- wildcard.webgfc.fr.pem

Nous allons déposer ces 2 fichiers dans le répertoire **/etc/apache2/ssl/**

Nous déposons également l'AC (Autorité de Certification) de nos certificats dans **/etc/apache2/ssl/**.

Nous avons donc désormais 3 fichiers dans le répertoire **/etc/apache2/ssl/**

- /etc/apache2/ssl/wildcard.webgfc.fr.pem
- /etc/apache2/ssl/wildcard.webgfc.fr.key
- /etc/apache2/ssl/ACADULLACTProjetg3

#### 4.1.2. PASSAGE DES APPLICATIONS EN HTTPS

Une fois les certificats disponibles, nous allons tout d'abord passer le portail LemonLDAP (<http://auth.webgfc.fr>) en HTTPS.

Pour cela, nous allons dans **/etc/apache2/sites-available/** où se trouve le Virtual Host de notre portail.

On crée une copie du vhost utilisé pour l'accès au portail (<http://auth.webgfc.fr>) comme ceci :

```
# cd /etc/apache2/sites-available
# cp portal-apache2.conf portal-apache2.ssl.conf
```

On édite le nouveau *vhost* en remplaçant le port 80 par le port **443** et en ajoutant les directives **SSL** comme ceci :

```
# vi portal-apache2.ssl.conf
```

```
# Portal Virtual Host (auth.webgfc.fr)
<VirtualHost "*:443">
    ServerName auth.webgfc.fr

    # CAS/LemonLDAP
    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/wildcard.webgfc.fr.pem
    SSLCertificateKeyFile /etc/apache2/ssl/wildcard.webgfc.fr.key
    SSLCACertificateFile /etc/apache2/ssl/ACADULLACTProjetg3

    # DocumentRoot
    DocumentRoot /var/lib/lemonldap-ng/portal/
    .....

```



La même démarche doit être réalisée à chaque ajout de nouvelles applications.

Pour notre application webgfc17.webgfc.fr, nous allons donc réaliser une copie du vhost mis en place.

```
# cd /etc/apache2/sites-available
# cp webgfc17.webgfc.fr webgfc17.webgfc.fr.ssl.conf
```

On édite ce nouveau vhost :

```
<VirtualHost *:443>
    ServerName webgfc17.webgfc.fr
    DocumentRoot /var/www/webgfc

    PerlHeaderParserHandler Lemonldap::NG::Handler
    # CAS LemonLDAP
    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/wildcard.webgfc.fr.pem
    SSLCertificateKeyFile /etc/apache2/ssl/wildcard.webgfc.fr.key
    SSLCACertificateFile /etc/apache2/ssl/ACADULLACTProjetg3

    <Directory "/var/www/webgfc/1.7/">
        Options Indexes MultiViews FollowSymlinks
    </Directory>
</VirtualHost>
```

Il ne nous reste plus qu'à activer ces nouveaux vhosts :

```
# a2ensite portal-apache2.ssl.conf
# a2ensite webgfc17.webgfc.fr.ssl.conf
```

Et enfin de relancer apache2 :

```
# service apache2 restart
```

Désormais:

- le portail sera accessible via l'URL <https://auth.webgfc.fr>
- et la nouvelle application via <https://webgfc17.webgfc.fr>

Néanmoins, malgré le passage des applications en HTTPS, cela ne suffit pas.

Il faut déclarer ces modifications dans la configuration du LemonLDAP mais également activer le module CAS.

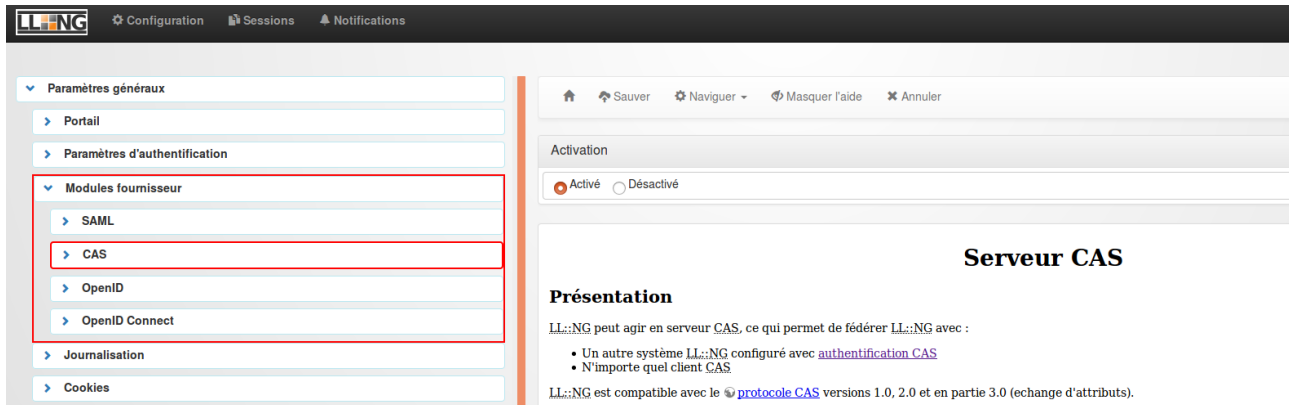
## 4.2. CÔTÉ LEMONLDAP::NG

La première chose, vu que l'application est désormais accessible en HTTPS est de modifier l'URL d'accès de l'application qui auparavant était en http.

Via le menu **Paramètres Généraux** → **Portail** → **Menu** → **Catégories et applications** → **Sample applications** → **webgfc17.webgfc.fr**, on modifie l'URI en mettant <https://webgfc17.webgfc.fr> et on clique sur *Sauver*.

Ensuite, nous allons activer le module CAS de LemonLDAP::NG afin que ce dernier se comporte comme un serveur CAS.

Pour cela, il faut aller dans **Paramètres Généraux** → **Modules fournisseur** → **CAS**.



Dans ce menu, il faut simplement aller dans le sous-menu **Activation** et mettre **Activé**.

NB : si le sous-menu *Règle d'utilisation* n'est pas positionné sur **Activé**, le faire.

On clique sur *Sauver*, et à partir de ce moment, le portail LemonLDAP se comportera comme un serveur CAS.

RQ : comme indiqué précédemment, si on souhaite revenir en arrière dans sa configuration du portail LemonLDAP::NG, il suffit d'aller supprimer le dernier fichier de configuration, généré à chaque sauvegarde, dans le répertoire `/var/lib/lemonldap-ng/conf/`

## 4.3. CÔTÉ WEB-GFC

Du côté de l'application, nous avons développé un connecteur CAS afin de pouvoir interfacier l'application web-GFC avec le portail LemonLDAP::NG en mode serveur CAS.

### 4.3.1. LIBRAIRIE PHPCAS

**La version 1.7.0 est livrée avec la librairie phpCAS en version 1.3.4 mais si ce n'est pas le cas, veuillez suivre les informations suivantes, sinon vous pouvez passer au point suivant.**

Pour cela, il est nécessaire de récupérer la librairie phpCAS via [jasig](https://github.com/Jasig/phpCAS)<sup>4</sup>.

Il faut donc récupérer les sources de phpCAS en version **1.3.4** via <https://github.com/Jasig/phpCAS>.

Un dossier **jasig** doit être créé dans le répertoire `/var/www/webgfc/vendors` de web-GFC pour que les paramètres CAS puissent être pris en compte.

Ci-dessous le détail des commandes à réaliser :

```
# cd /tmp
# git clone https://github.com/Jasig/phpCAS.git
# cd phpCAS.git
# git checkout tags/1.3.4
# cd ..
# mv -R phpCAS phpcas

# mkdir /var/www/webgfc/vendors/jasig
# cp -R phpcas /var/www/webgfc/vendors/jasig/
```

Une fois le dossier `phpcas` créé dans `/var/www/webgfc/vendors/jasig`, 2 fichiers doivent être modifiés :

- `/var/www/webgfc/vendors/jasig/phpcas/source/CAS/Request/CurlRequest.php`
  - il faut remplacer les variables présentes entre les lignes 121 et 129 comme ceci :
    - remplacement de `$this->caCertPath` par `$this->caCertFile`

4 Source : <https://wiki.jasig.org/display/CAS/Home>

```

.....
/*****
* Set SSL configuration
*****/
if ($this->caCertFile) {
  if ($this->validateCN) {
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);
  } else {
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
  }
  curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 1);
  curl_setopt($ch, CURLOPT_CAINFO, $this->caCertFile);
  phpCAS::trace('CURL: Set CURLOPT_CAINFO ' . $this->caCertFile);
} else {
.....

```

- /var/www/webgfc/vendors/jasig/phpcas/source/CAS/Request/AbstractRequest.php
  - il faut ajouter la variable suivante à la ligne 50 :
    - **protected \$caCertFile = null;**

```

protected $isPost = false;
protected $postBody = null;
protected $caCertPath = null;
protected $caCertFile = null;
protected $validateCN = true;
private $_sent = false;

```

### 4.3.2. MISE À JOUR DES BASES DE DONNÉES

Il faut également mettre à jour la base de données en version 1.7.0.

Si l'application est en version 1.6.x, il faut passer le patch **226\_patch\_1.7.0.sql** dans la (les) base(s) de données de sa (ou ses) collectivité(s) :

```

# cd /var/www/webgfc
# psql -U webgfc -p 5432 webgfc_collectivite -f 226_patch_1.7.0.sql

```

Une fois ce patch exécuté, il reste un dernier patch à passer en base de données, le patch **227\_patch\_admin\_schema.sql**. Mais cette fois-ci, le patch doit être passé dans la base de données gérant l'administration de l'application, généralement nommée **webgfc\_admin** :

```

# cd /var/www/webgfc
# psql -U webgfc -p 5432 webgfc_admin -f 227_patch_admin_schema.sql

```

### 4.3.3. FORMULAIRE DE CONNEXION CAS DANS WEB-GFC

Un formulaire a donc été mis en place dans web-GFC, uniquement sur l'environnement du *super-administrateur*.

**IMPORTANT** : la connexion CAS dans web-GFC se fait pour l'intégralité des comptes utilisateurs liés à l'application. Quelque soit la (ou les) collectivité(s) définie(s) dans l'application, la connexion ne se fera plus qu'à travers le portail CAS si la connexion CAS est configurée à *Oui* dans web-GFC.

Une fois connecté en tant que super-administrateur dans web-GFC, un nouveau menu est disponible : *Authentification CAS*



Version : 1.7.0-alpha (rev: 3750)

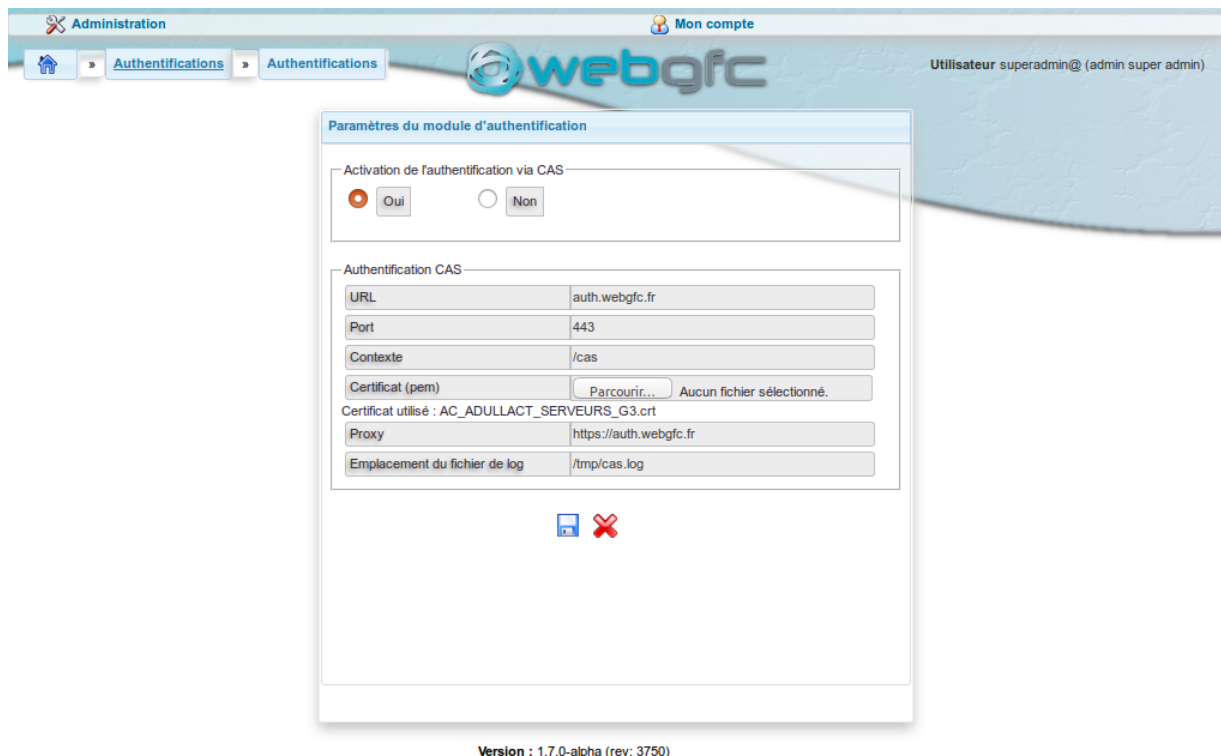
*Nouveau menu permettant d'activer ou non l'authentification CAS dans web-GFC*

Par défaut, la connexion CAS est désactivée.

Afin de pouvoir accéder au formulaire de configuration du serveur CAS, il suffit de cliquer sur *Authentification (CAS)* et de sélectionner *Oui* au niveau du bouton radio.

Dès lors, un formulaire apparaît où il faut renseigner :

- l'URL du serveur CAS (sans le http://) → `auth.webgfc.fr`
- le Port (généralement 443) → 443
- le « Contexte » (généralement /cas) → /cas
- le certificat (au format pem) : la clé publique du certificat reconnu par le certificat serveur préalablement installé (cf paragraphe 4.1.1 *Certificat serveur wildcard*)
- le proxy → <https://auth.webgfc.fr>
- l'emplacement du fichier de log => /tmp/cas.log



Version : 1.7.0-alpha (rev: 3750)

*Formulaire de paramétrage du connecteur CAS dans web-GFC*

**RQ :** Ces données, à l'enregistrement, sont stockées dans le fichier *webgfc.inc* (dans `/var/www/webgfc/app/Config/webgfc.inc`).

De nouvelles valeurs ont été ajoutées :

- Configure::write('AuthManager.Authentification.use', false);  
→ si la valeur est à **true**, le connecteur CAS est actif, sinon la valeur est **false**
  - Configure::write('AuthManager.Authentification.type', "");  
→ si la valeur est à **CAS**, le connecteur est actif et de type CAS, sinon c'est <vide> (")
  - Configure::write('AuthManager.Cas.host', "");  
→ l'URL du CAS (ex : auth.webgfc.fr), sinon <vide>
  - Configure::write('AuthManager.Cas.port', "");  
→ le port utilisé pour se connecter au CAS (ex : 443), sinon c'est <vide>
  - Configure::write('AuthManager.Cas.uri', "");  
→ le contexte défini (ex : /cas), sinon c'est <vide>
  - Configure::write('AuthManager.Cas.cert\_path', APP . 'Config/cert\_cas/client.pem');  
→ le chemin où se stocke les informations du certificat (ex : AC\_ADULLACT\_SERVEURS\_G3.crt), (cette donnée n'est pas vouée à être modifiée)
  - Configure::write('AuthManager.Cas.proxy', "");  
→ l'URL du proxy (ex : <https://auth.webgfc.fr>), sinon <vide>
- // Suffixe renseigné ici afin que les utilisateurs n'aient pas à le saisir  
// @info: NE PAS RENSEIGNER SI web-GFC EST EN MULTI-COLLECTIVITES (laisser vide)
- Configure::write('Suffixe.connexion', "");  
→ le suffixe de connexion pouvant être défini afin que les utilisateurs n'aient pas à saisir leur suffixe dans l'identifiant (ex : cogitis). L'agent saisira auzolat au lieu de [auzolat@cogitis](mailto:auzolat@cogitis) pour s'identifier

#### 4.3.4. CONNEXION À WEB-GFC

Depuis le portail auth.webgfc.fr, nous avons vu qu'un lien webgfc17.webgfc.fr est disponible (cf point 3.6.1).

Lors du clic sur ce lien, et si la connexion CAS est active sur web-GFC, si l'application est en mode multi-collectivités (donc la variable `Configure::write('Suffixe.connexion', '')`; laissée à vide), l'agent ne sera pas directement dirigé vers son environnement de travail mais devra choisir sa collectivité avant de pouvoir accéder à son environnement.

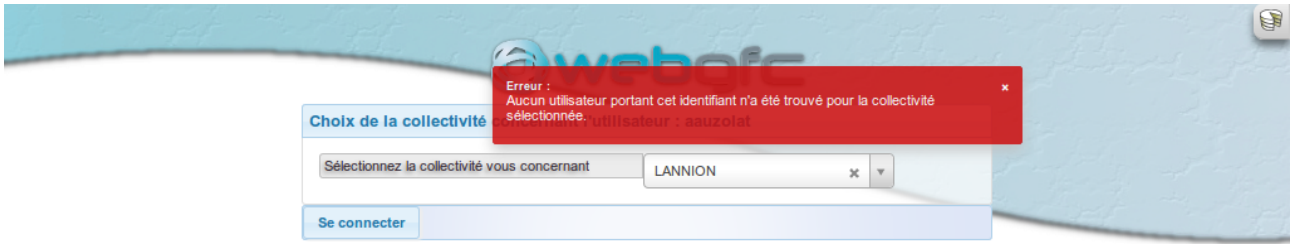


The screenshot shows the webgfc login interface. At the top, there is a logo for 'webgfc'. Below it, a form titled 'Choix de la collectivité concernant l'utilisateur : auzolat' is displayed. The form contains a text input field with the placeholder 'Sélectionnez la collectivité vous concernant' and a dropdown menu with the placeholder 'Sélectionner une collectivité'. Below these fields is a blue button labeled 'Se connecter'. At the bottom of the page, the version information 'Version : 1.7.0-alpha (rev: 3809)' is shown.

*Formulaire de sélection de collectivité (uniquement si CAS actif + multi-collectivités)*

Une fois la collectivité sélectionnée, si l'agent existe bien dans cette collectivité, alors il sera redirigé vers son environnement de travail.

Sinon, un message d'erreur indique qu'aucun agent n'existe pour cette collectivité



Version : 1.7.0-alpha (rev: 3809)

*Message d'erreur si l'identifiant n'est pas trouvé pour la collectivité sélectionnée*