

Plateforme de formation -
Doc Admin LemonLdap



PROHACKTIVE

Sommaire

Paramètres d'authentification	3
Préparation de la bdd	3
Entrée des informations	4
Variables exportées	5
Hôtes virtuels	6
Ajout d'application	6
Configuration du fichier de configuration	6
Portail d'accès	8

- Paramètres d'authentification

Nous traiterons dans cette partie la méthode d'authentification via une base de données sql. Cependant, lemonldap accepte des dizaines de méthode d'authentification, pour celles-ci, consultez la [documentation](#).

1. Préparation de la bdd

Le système de gestion de base de données peut être interne ou externe au lemonldap, il faut simplement indiquer au système toutes les informations.

La table pour le stockage des données peut être simple ou double, comme décrit dans la [documentation](#).

A titre d'exemple, on peut utiliser une table simple comme celle-ci (table: auth):

Field	Type	Null	Key	Extra
id	int(11)	NO	PRIMARY	auto_increment
user	varchar(255)	NO		
password	varchar(255)	NO		
name	varchar(255)	NO		
mail	varchar(255)	YES		
groups	varchar(255)	YES		

PS: En cas d'utilisation de mariadb comme sgbd, ne pas oublier d'installer le paquet "libdbd-mysql-perl".

2. Entrée des informations

Les paramètres d'authentification se trouve sous :
Paramètres généraux → *Paramètres d'authentification*

Pour accéder aux paramètres DBI indiquer les paramètres comme ci-dessous:

Paramètres d'authentification	
Module d'authentification	Database (DBI)
Module d'utilisateurs	Database (DBI)
Module de mot de passe	Database (DBI)
Module d'auto-enregistrement	None

Sous *Paramètres généraux* → *Paramètres d'authentification* → *Paramètres DBI* → *Connexion* se trouve les paramètres à renseigner pour la connexion à la base de données.

Si deux tables ont été créées dans deux base de données distinctes, il faudra renseigner les paramètres adéquats dans *Authentification* et *Utilisateur*, sinon la catégorie *Authentification* suffit.

Il faut donc renseigné les information de connexion sous ce format:

Authentification	
Chaîne	dbi:mysql:database=<dbname>;host=localhost
Utilisateur	<dbUser>
Mot de passe	●●●●●●●●

Une fois la base de données connectée, il faut indiquer à ll::ng comment est structuré notre table. Pour cela, on se rend sous *Paramètres généraux* → *Paramètres d'authentification* → *Paramètres DBI* → *Schéma* et on renseigne les valeurs.



Exemple à partir de la table présenté précédemment:

Schéma	
Table authentification	auth
Table des utilisateurs	auth
Champ identifiant	user
Champ mot de passe	password
Champ mail	mail
Champ identifiant dans la table des utilisateurs	user

3. Variables exportées

Pour faciliter certaines opérations, il peut être utile de stocker des informations dans la table des utilisateurs. A cette fin, il est possible de renseigner des entités supplémentaires de la table, qui seront enregistrés dans les variables de session de l'utilisateur.

Pour ce faire, on se rend sous *Paramètres généraux* → *Paramètres d'authentification* → *Paramètres DBI* → *Variables exportées* et on ajoute une entrée. A titre d'exemple, on utilisera le champ groups présent dans la table précédente.

Variables exportées	
Clefs	Valeurs
<input type="text" value="group"/>	<input type="text" value="groups"/>  

Clefs → Nom de la variable de session qui contiendra la valeur contenu dans la table.

Valeurs → Nom du champ de la table.

On peut retrouver la variable dans le profil d'un utilisateur connecté sous l'onglet Sessions du manager.

Parmi les information de l'utilisateur admin:

Attributs et Macros	
UA	Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
_appsListOrder	sort_3,sort_4,sort_1,sort_2
_language	fr
_session_kind	SSO
_whatToTrace	admin
group	admin

On retrouve la variable group et sa valeur.

- Hôtes virtuels

1. Ajout d'application

Afin d'ajouter des applications à accès contrôlé par ll::ng, il faut les renseigner et leur attribuer des règles d'accès.

Pour ce faire, on se rend sous *Hôtes virtuels*, on ajoute un hôte virtuel en le nommant d'après un sous-domaine.

On peut ensuite ajouter des règles d'accès à l'application sous *Hôtes virtuels* → *<FQDN>* → *Règles d'accès*.

Syntaxes :

Commentaires → Les règles seront lues dans l'ordre alphabétique des commentaires.

Expressions régulières → “^/<path>” désigne un sous répertoire du site web auquel appliquer la règle d'accès. “^/” pour la racine du site web.

Règles → Peut utiliser des expressions booléennes ou autres ([voir documentation](#)).
exemple : ‘ \$group eq "admin" ’ ne donne accès qu'aux utilisateurs dont la variable de session “\$group” contient la valeur “admin”.

2. Configuration du fichier de configuration

Chaque application doit faire l'objet d'un fichier de configuration virtualhost nginx ou apache selon l'installation. Les exemples ci-après concernent nginx. Pour la configuration d'apache, se référer à la [documentation](#).

Pour ce faire, on crée un fichier dans */etc/nginx/site-available/* écoutant sur le même nom de serveur que l'hôte virtuel configuré dans l'interface de ll::ng.

Exemple de configuration en https:

```
# Redirection https
server {
    listen      80;
    server_name app.example.com;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl;
    server_name app.example.com;

    # Configuration HTTPS
```

```

ssl_certificate /etc/ssl/certs/certificat.crt;
ssl_certificate_key /etc/ssl/private/certificat.key;
ssl_protocols SSLv3 TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers RC4:HIGH:!aNULL:!MD5;
ssl_prefer_server_ciphers on;
ssl_session_cache shared:SSL:10m;
ssl_session_timeout 10m;

# Internal authentication request
location = /lmauth {
    internal;

    # FastCGI configuration
    include /etc/nginx/fastcgi_params;
    fastcgi_pass unix:/var/run/llng-fastcgi-server/llng-fastcgi.sock;
    # Drop post datas
    fastcgi_pass_request_body off;
    fastcgi_param CONTENT_LENGTH "";
    # Keep original hostname
    fastcgi_param HOST $http_host;
    # Keep original request (LLNG server will receive /lmauth)
    fastcgi_param X_ORIGINAL_URI $original_uri;
    # Improve performances
    #fastcgi_buffer_size 32k;
    #fastcgi_buffers 32 32k;
}

# Client requests
location / {

    #####
    # CALLING AUTHENTICATION #
    #####
    set $original_uri $uri$is_args$args;
    auth_request /lmauth;
    auth_request_set $lmremote_user $upstream_http_lm_remote_user;
    auth_request_set $lmremote_custom $upstream_http_lm_remote_custom;
    auth_request_set $lmlocation $upstream_http_location;
    # If CDA is used, uncomment this
    auth_request_set $cookie_value $upstream_http_set_cookie;
    add_header Set-Cookie $cookie_value;
    # Remove this for AuthBasic and OAuth2 handlers
    error_page 401 $lmlocation;


    # Reverse proxy
    proxy_pass https://<IP/FQDN>/;
    include /etc/nginx/proxy_params;
}
}

```

3. Portail d'accès

Pour être accessible depuis le portail d'accès, les applications doivent être renseigné sous *Paramètres généraux* → *Portail* → *Menu* → *Catégories et applications* → <Catégorie>

Pour que l'application ne soit visible qu'aux seuls personnes ayant accès, paramétrez comme suit :

Application	
Nom	App
Description	New app
URI	https://app.example.com/
Info-bulle	New app
Logo	 tux.png
Affichage de l'application	<input type="radio"/> Activé <input type="radio"/> Désactivé <input checked="" type="radio"/> Automatique <input type="radio"/> Règle spécifique