Classe Ldap

Table des matières

1. Contenu

Ce document à pour but d'expliquer et de définir l'usage de la classe **Ldap** de l'outil **UniversalWeb**.

2. Définition

La classe **Ldap** permet d'accéder à un annuaire externe pour la gestion des utilisateurs. Ce peut être une alternative à la gestion des utilisateurs implémentée en standard dans UniversalWeb qui utilise sa base de données interne (constante **_ANNUAIRE_** positionnée à **_ANNUAIRE_INTERNE_** dans le fichier de configuration **config.inc.php** de UniversalWeb). Cette classe est disponible dans le script **Ldap.class.php**. Veuillez vous reporter au chapitre « A propos des annuaires » de la documentation UniversalWeb pour plus d'informations sur les possibilités offertes par la mise en œuvre de cette classe.

Si vous avez besoin d'interroger un annuaire LDAP il est conseillé de créer votre propre classe LDAP qui héritera de cette classe plutôt que de la modifier.

```
class monLdap extends Ldap {
}
```

L'utilisation de la classe implique d'avoir une bonne connaissance de la structure de l'annuaire interrogé et des résultats qu'il renvoie.

3. Constantes

Liste des constantes utilisables par le développeur et proposées par la classe.

Nom de la constante	Valeur
MLDAP_TICKET_KERBEROS_KO	Ticket Kerberos non valide
MLDAP_CANT_CONTACT_SERVER	Impossible de se connecter au serveur LDAP
MLDAP_SUCCESS	Aucune erreur
MLDAP_OPERATIONS_ERROR	Erreur LDAP
MLDAP_PROTOCOL_ERROR	Erreur de protocole
MLDAP_TIMELIMIT_EXCEEDED	Time Out
MLDAP_SIZELIMIT_EXCEEDED	Taille maximale de retour dépassée. Résultats incomplets
LDAP_NO_SUCH_ATTRIBUTE	Attribut inexistant
MLDAP_CONSTRAINT_VIOLATION	Opération impossible
MLDAP_TYPE_OR_VALUE_EXISTS	Valeur existe déjà
MLDAP_NOT_CONNECTED	Annuaire LDAP non connecté

MLDAP_PROTOCOL_3	Erreur de modification du protocole 3
MLDAP_NO_SUCH_OBJECT	Cible non trouvée
MLDAP_INVALID_DN_SYNTAX	Syntaxe DN incorrecte
MLDAP_INVALID_CREDENTIALS	Compte / mode passe erroné
MLDAP_INSUFFICIENT_ACCESS	Droits insuffisants
MLDAP_BUSY	Serveur LDAP trop occupé
MLDAP_UNAVAILABLE	Serveur LDAP non disponible
MLDAP_UNWILLING_TO_PERFORM	Requête impossible à accomplir (du par exemple à des restrictions du serveur LDAP ou AD)
MLDAP_LOOP_DETECT	Détection d'une boucle
MLDAP_NAMING_VIOLATION	Violation des règles de structure du LDAP
MLDAP_OBJECT_CLASS_VIOLATION	Violation des règles d'objet LDAP
MLDAP_ALREADY_EXISTS	Attribut existe déjà
MLDAP_OTHER	Erreur non référencée

4. Méthodes

Getters

getHost

Renvoie le nom du serveur LDAP interrogé.

Description
```
string getHost();
```

Liste des paramètres
Aucun.

Valeurs de retour
Chaine de caractère représentant le serveur LDAP interrogé.

Exemple
```
echo $this->getHost();
```

getConnexionId

Renvoie l'identificateur (ressource) de la connexion LDAP en cours.

Description

```
resource getConnexionId();
```

Liste des paramètres

Aucun.

Valeurs de retour

La méthode renvoie :

- **NULL** si aucune demande de connexion n'a encore été réalisée (c'est par exemple le cas si l'interrogation a lieu juste après la création de la classe).
- Un identificateur de liaison LDAP (ressource) si l'interrogation du serveur a été possible (tentative de connexion)
- **false** si l'interrogation du serveur n'a pas été possible à la suite d'une tentative de connexion.

Exemple

```
$myldap = new Ldap('serveur', 389);
$myldap->connect();
$connexion = $myldap->getConnexionId();
```

getErreur

Retourne le numéro d'erreur LDAP de la dernière commande exécutée.

Description

```
integer getErreur();
```

Liste des paramètres

Aucun.

Valeurs de retour

La méthode renvoie un entier correspondant au code de retour de la dernière commande exécutée. Cet entier peut être comparé à l'une des constantes définie au chapitre 0.

Exemple

```
$myldap = new Ldap('serveur', 389);
$myldap->connect();
if ($myldap->getErreur() != Ldap::MLDAP_SUCCESS) {
}
```

Note

La méthode **getErreurNum** est un clone de **getErreur** et a exactement la même fonction.

getErreurNum

Retourne le numéro d'erreur LDAP de la dernière commande exécutée.

Description

```
integer getErreurNum();
```

Liste des paramètres

Aucun.

Valeurs de retour
La méthode renvoie un entier correspondant au code de retour de la dernière commande exécutée. Cet entier peut être comparé à l'une des constantes définie au chapitre 0.

Exemple
```
$myldap = new Ldap('serveur', 389);
$myldap->connect();
if ($myldap->getErreurNum() != Ldap::MLDAP_SUCCESS) {
}
```

Note
La méthode **getErreurNum** est un clone de **getErreur** et a exactement la même fonction.

getErreurMessage

Retourne le message explicatif en clair (en anglais) correspondant au numéro de l'erreur LDAP de la dernière commande exécutée.

Description
```
string getErreurMessage();
```

Liste des paramètres
Aucun.

Valeurs de retour
Chaine de caractère en anglais d'explication correspondant au code de retour de la dernière commande exécutée.

Exemple
```
$myldap = new Ldap('serveur', 389);
$myldap->connect();
if ($myldap->getErreurNum() != Ldap::MLDAP_SUCCESS) {
  echo 'Erreur rencontrée : '.$myldap->getErreurMessage();
}
```

Note
Bien que les numéros d'erreur LDAP soient standardisés, différentes bibliothèques retournent des messages différents ou même des textes d'erreur localisés. N'utilisez jamais les messages d'erreur pour identifier une erreur, mais bien les numéros (méthodes **getErreurNum** et **getErreur**).

getErreurMessageFr

Retourne le message explicatif en clair (en français) correspondant au numéro de l'erreur LDAP de la dernière commande exécutée.

Description
```
string getErreurMessageFr();
```

Liste des paramètres
Aucun.

Valeurs de retour
Chaine de caractère en français d'explication correspondant au code de retour de la dernière commande exécutée.

Exemple

```
$myldap = new Ldap('serveur', 389);
$myldap->connect();
if ($myldap->getErreurNum() != Ldap::MLDAP_SUCCESS) {
  echo 'Erreur rencontrée : '.$myldap->getErreurMessageFr();
}
```

Note
Bien que les numéros d'erreur LDAP soient standardisés, différentes bibliothèques retournent des messages différents ou même des textes d'erreur localisés. N'utilisez jamais les messages d'erreur pour identifier une erreur, mais bien les numéros (méthodes **`getErreurNum`** et **`getErreur`**).

getDn

Retourne la chaine du dossier DN (*Distinguished Name*) actuellement interrogé.

Description
```
string getDn();
```

Liste des paramètres
Aucun.

Valeurs de retour
Dossier DN interrogé sur le serveur LDAP.

Exemple
```
echo $this->getDn();
```

Setters

setDn

Positionne la chaine du dossier DN (*Distinguished Name*) à interroger.

Description
```
getDn(string $dn);
```

Liste des paramètres
dn : dossier à interroger sur le serveur LDAP sous forme de *Distinguished Name* (collection d'attributs séparés par une virgule).

Valeurs de retour
Aucune.

Exemple
```
echo $this->setDn('ou=people,dc=sun.com');
```

setHost

Positionne le nom du serveur LDAP à interroger.

Description
```
setHost(string $host);
```

Liste des paramètres
host : nom du serveur LDAP à interroger.

Valeurs de retour
Aucune.

Exemple
```
echo $this->setHost('ldap.sun.com');
```

Méthodes publiques

_construct

Constructeur de la classe

Description
```
__construct(string $host, integer $port [, string $compte = null] [,
string $mdp = null])
```

Liste des paramètres
host : nom du serveur LDAP à consulter.
port : port d'interrogation (par défaut, port 389)
compte : (optionnel) compte de service permettant d'accéder à la base (les connexions anonymes ne sont pas acceptées)
mdp : (optionnel) mot de passe du compte de service d'accès à la base

Valeurs de retour
Un objet LDAP.

Exemple
Dans l'exemple ci-dessous, on se connecte au serveur LDAP **ldap21** par l'intermédiaire du port **389** et avec le compte de service **invite** et le mot de passe **monmdp**.

```
$objAnnuaire = new Ldap('ldap21', 389, 'invite', 'monmdp');
```

Note
Attention, le constructeur ne connecte pas la base annuaire, il ne fait que préparer l'objet à l'utilisation de la base en procédant à diverses initialisations. Ainsi – entre autres - le développeur devra ensuite procéder à la connexion de la base via la méthode **connect()**.

connect

Réalise la connexion au serveur LDAP.

Description
```
boolean connect();
```

Liste des paramètres
Aucun.

Valeurs de retour
Un booléen qui rend compte de l'état de la connexion réalisé (**`true`** : la classe est connectée au serveur / **`false`** : la connexion a échoué).

Exemple
```
$myldap = new Ldap('serveur', 389);
$connected = $myldap->connect();
if (!$connected) {
  echo 'Erreur rencontrée : '.$myldap->getErreurMessageFr();
}
```

connectUsingKerberos

Réalise la connexion au serveur LDAP en utilisant l'authentification Kerberos.

Description
```
boolean connectUsingKerberos(string $binduser);
```

Liste des paramètres
`Binduser` : serveur nommé qui héberge kerberos.

Valeurs de retour
Un booléen qui rend compte de l'état de la connexion réalisé (**`true`** : la classe est connectée au serveur / **`false`** : la connexion a échoué).

isConnected

Retourne l'état de connexion actuel au serveur LDAP..

Description
```
boolean isConnected();
```

Liste des paramètres
Aucun.

Valeurs de retour
Booléen rendant compte de l'état connecter ou déconnecté de l'annuaire LDAP.
`true` : connecté à l'annuaire
`false` : non connecté à l'annuaire

Exemple
```
echo $this->getDn();
```

close

Fermeture de la connexion LDAP.

Description
```
close();
```

Liste des paramètres
Aucun.

Valeurs de retour
Aucune valeur en retour. Cependant le développeur pourra contrôler l'état de la connexion en faisant appel aux getters **`getConnexionId`**, **`getError`**, **`getErrorMessage`**, **`getErrorMessageFr`** et à la méthode **`isConnected`**.

decodeSID

Décode un SID Microsoft Active Directory.

Description
```
string decodeSID(string $value);
```

Liste des paramètres
`value` : la valeur du SID codée.

Valeurs de retour
Chaine de caractère du SID décodé.

Note
Un SID est un identifiant de sécurité unique mis en place par Microsoft qui identifie chaque système. L'annuaire Active Directory utilise ces identifiants.
Cette méthode est ici fournie en tant qu'utilitaire que le développeur pourra utiliser si il est nécessaire de décoder cet identifiant Microsoft (connexion à Active Directory en particulier).

search

Effectue une recherche **`$valeur`** sur l'annuaire LDAP et charge les données trouvées dans **`$lesInfos`**.

Description
```
integer|boolean search(string $valeur, array &$lesInfos);
```

Liste des paramètres
`valeur` : filtre de recherche (ce que l'on recherche)
`lesInfos` : tableau d'informations trouvées en retour (structure d'informations renvoyée par l'annuaire)

Valeurs de retour
Nombre d'entrées trouvées pour la valeur recherchée (entier).
`false` (booléen) si la recherche n'a rien donné ou bien si une erreur s'est produite.

Exemple
```
$myldap = new Ldap('serveur', 389);
$connected = $myldap->connect();
if ($connected) {
  $myldap->setDn('ou=people,dc=sun.com');
  $myldap->search('cn=Barbara Jensen', $lesInfos);     //ou 'cn=*' pour chercher tout le monde
  if ($myldap->getErreurNum() != Ldap::MLDAP_SUCCESS) {
    echo 'Erreur rencontrée : '.$myldap->getErreurMessageFr();
  }
}
```

Note
La recherche est effectuée sur le dossier DN (*Distinguished Name*) à interroger. Il est donc nécessaire d'avoir appelé la méthode **`setDn`** avant **`search`** pour spécifier sur quel dossier lancer la recherche.

authentifieUser

Tente d'authentifier un utilisateur **$id** avec son mot de passe **$mdp**.

Description

```
boolean authentifieUser(string $id, string $mdp);
```

Liste des paramètres

id : nom recherché sous la forme DN (*Distinguished Name*)
mdp : mot de passe proposé pour l'authentification

Valeurs de retour

true : l'authentification est correcte.
false : l'authentification a échoué.

Exemple

```
$myldap = new Ldap('serveur', 389);
$connected = $myldap->connect();
if ($connected) {
  $myldap->setDn('ou=people,dc=sun.com');
  $myldap->authentifieUser('id=barbara.jensen', 'mdpdebarbara');
  if ($myldap->getErreurNum() != Ldap::MLDAP_SUCCESS) {
    echo 'Erreur rencontrée : '.$myldap->getErreurMessageFr();
  }
  else {
    echo 'Nom d\'utilisateur et mot de passe correct pour Barbara Jensen';
  }
}
```

Note

Cette méthode permet d'authentifier un utilisateur mais en aucun cas ne le logue à l'application. Il s'agit simplement d'une interrogation de l'annuaire LDAP.

readAttrib

Lit une entrée d'objet LDAP.

Description

```
array readAttrib(array $data, integer $num, string $entry);
```

Liste des paramètres

data : jeu de données à lire renvoyé par une recherche LDAP
num : numéro de l'entrée à lire
entry : indice de l'entrée à lire pour le numéro

Valeurs de retour

Tableau contenant l'information de l'entrée **$entry** pour le numéro **$num** de **$data**.

Exemple

```
$myldap = new Ldap('serveur', 389);
$connected = $myldap->connect();
if ($connected) {
  $myldap->setDn('ou=people,dc=sun.com');
  $myldap->search('id=barbara.*', $lesInfos);
  if ($myldap->getErreurNum() != Ldap::MLDAP_SUCCESS) {
    echo 'Erreur rencontrée : '.$myldap->getErreurMessageFr();
  }
  else {
    //récupère l'adresse email du premier 'jérôme' trouvé dans l'annuaire
```

```
    $info = $myldap->readAttrib($lesInfos, 0, 'mail');
  }
}
```

Note
Lit une entrée LDAP récupérée après une opération de recherche. Il est nécessaire de bien connaitre la structure de votre annuaire pour accéder correctement à l'information recherchée.

addAttrib

Ajoute une ou plusieurs entrées à l'annuaire LDAP.

Description
```
boolean addAttrib(array $valeur);
```

Liste des paramètres
valeur : tableau associatif représentant l'entrée à ajouter

Valeurs de retour
true : succès
false : erreur

Exemple
```
$myldap = new Ldap('serveur', 389);
$connected = $myldap->connect();
if ($connected) {
  $myldap->setDn('ou=people,dc=sun.com');
  $entry['memberuid'] = 'username';
  $myldap->addAttrib($entry);
  if ($myldap->getErreurNum() != Ldap::MLDAP_SUCCESS) {
    echo 'Erreur rencontrée : '.$myldap->getErreurMessageFr();
  }
}
```

Note
Tableau associatif répertoriant les valeurs des attributs à ajouter. Si un attribut n'était pas encore existant, il sera ajouté. Si un attribut est existant, vous pouvez uniquement y ajouter des valeurs s'il prend en charge plusieurs valeurs.
L'ajout se fait dans l'entrée DN préconfigurée.

deleteAttrib

Supprimer une ou plusieurs entrées de l'annuaire LDAP.

Description
```
boolean deleteAttrib(array $valeur);
```

Liste des paramètres
valeur : tableau associatif représentant l'entrée à supprimer

Valeurs de retour
true : succès
false : erreur

Exemple
```
$myldap = new Ldap('serveur', 389);
$connected = $myldap->connect();
if ($connected) {
  $myldap->setDn('ou=people,dc=sun.com');
```

```
  $entry['memberuid'] = 'username';
  $myldap->deleteAttrib($entry);
  if ($myldap->getErreurNum() != Ldap::MLDAP_SUCCESS) {
    echo 'Erreur rencontrée : '.$myldap->getErreurMessageFr();
  }
}
```

Note
L'ajout se fait dans l'entrée DN préconfigurée.