



EBook Gratuito

APPENDIMENTO

Idap

Free unaffiliated eBook created from
Stack Overflow contributors.

#Idap

Sommario

Di.....	1
Capitolo 1: Iniziare con ldap.....	2
Osservazioni.....	2
Examples.....	2
Configurare PHP per lavorare con LDAP.....	2
Titoli di coda.....	6

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [ldap](#)

It is an unofficial and free ldap ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official ldap.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con ldap

Osservazioni

Questa sezione fornisce una panoramica di cosa è ldap e perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare eventuali soggetti di grandi dimensioni all'interno di LDAP e collegarsi agli argomenti correlati. Poiché la Documentazione per ldap è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Examples

Configurare PHP per lavorare con LDAP

Dopo aver configurato correttamente il server LDAP, ora vogliamo connetterci. Ad esempio usando PHP.

- DN = nome distinto. Ciò significa, in quale parte del database stai lavorando. Può essere un utente o un gruppo (o anche impostazioni di configurazione).
- Entrata: un'entità, ad esempio un utente.
- Attributo: qualcosa all'interno di una voce, ad esempio nome, numero di telefono e indirizzo email.

Collegamento

Innanzitutto, definiamo quanto segue:

```
$server = "server.example.com"; //this is the LDAP server you're connecting with
$ds = ldap_connect("ldaps://$server", 636); //always connect securely via LDAPS when possible
```

Ora, siamo connessi. La prossima cosa che vogliamo è far sapere al server che siamo una persona affidabile. La soluzione più semplice consiste nell'utilizzare il DN radice O un utente esistente con le autorizzazioni appropriate per visualizzare il database (probabilmente ogni utente nel database può farlo per impostazione predefinita). Autenticiamo usando la funzione bind.

Imposta le opzioni

Innanzitutto, dichiariamo queste opzioni. A seconda della configurazione del tuo server, puoi lasciarlo fuori.

```
ldap_set_option($ds, LDAP_OPT_PROTOCOL_VERSION, 3);
ldap_set_option($ds, LDAP_OPT_REFERRALS, 0);
```

Rilegatura

Supponiamo che tu stia usando admin e la password è pass123notsafe

```
$dn = "uid=admin,cn=users,dc=server,dc=example,dc=com";  
$pass = "pass123notsafe";  
$ldapbind = ldap_bind($ds, $dn, $pass); //this is the point we are authenticating
```

È molto carino Siamo dentro. Ora possiamo eseguire molte diverse operazioni. Ad esempio, possiamo cercare, leggere, scrivere e modificare utenti e persino gruppi.

Ricerca

Immagina di voler mostrare tutti i membri del gruppo "utenti".

```
$dn = "cn=users,dc=server,dc=example,dc=com"; //very important: in which part of your database  
are you looking  
$filter = "uid=*"; //don't filter anyone out (every user has a uid)  
$sr = ldap_search($ds, $dn, $filter) or die ("bummer"); //define your search scope  
  
$results = ldap_get_entries($ds, $sr); //here we are pulling the actual entries from the  
search we just defined  
var_dump($results); //will give you all results in array form.
```

È possibile utilizzare foreach loop per visualizzare i dati in un modo piacevole.

Ok, capito? Ora procedi con alcuni filtri. Vogliamo mostrare solo gli utenti del gruppo "ciclisti" che hanno registrato un indirizzo email. È importante sapere che *tutti gli* utenti sono in cn = utenti. Accanto a questo, possono anche essere membri di altri gruppi.

```
//did the connecting and binding  
  
$dn = "cn=bikeowners,cn=groups,dc=server,dc=example,dc=com"; //note the extra "cn=groups" for  
looking in a group that is not "users"  
$filter = "email=*"; //email address must be set but can be anything  
$sr = ldap_search($ds, $dn, $filter) or die ("bummer"); //define your search scope
```

E ora procedi di nuovo con `ldap_get_entries` .

Efficienza nell'ottenere le voci

Un esempio in cui viene affrontata l'efficienza, sempre più importante con grandi database con molti attributi per voce. Soprattutto quando si memorizzano le immagini nell'attributo jpegphoto, si può ridurre in modo significativo il tempo di caricamento quando si selezionano le voci in modo selettivo.

Immagina di voler cercare utenti appartenenti al gruppo "ciclisti". Ci sono molte informazioni memorizzate all'interno di queste voci, diciamo che hanno i seguenti attributi: cn, uid, nome, displayname, mail, iniziali, cellulare, numero di telefono, via, indirizzo postale, codice postale e jpegphoto. Ora abbiamo solo bisogno del loro uid, nome e iniziali.

Per questo, usiamo un 4 ° parametro opzionale di `ldap_search`:

```
$dn = "cn=bikeowners,cn=groups,dc=server,dc=example,dc=com";
$filter = "uid=*"; //
$justthese = array("uid", "name", "initials");
$sr = ldap_search($ds, $dn, $filter, $justthese) or die ("bummer"); //define your search scope
```

E voilà, eseguire `ldap_get_entries` ti darà solo questi dati.

Filtro avanzato

Ovviamente, è possibile estrarre un intero database con LDAP e elaborarlo ulteriormente in PHP. Tuttavia, proprio come con MySQL, è molto più efficiente eseguire l'elaborazione lato server. Per dimostrarlo, possiamo usare un filtro avanzato.

Prendi uno degli esempi sopra come punto di partenza, ma cambia `$filter` in base alla riga sottostante. Nel nostro esempio, vogliamo visualizzare i dati degli utenti che sono attivi.

Normalmente, l'attributo `shadowexpire` viene utilizzato per memorizzare queste informazioni.

Tuttavia, questo può differire tra i vari sistemi LDAP. Non solo vogliamo mostrare gli utenti attivi, ma *anche il* loro nome deve iniziare con una "a" e devono vivere ad Amsterdam.

Fondamentalmente vogliamo fare tre cose:

1. Il più semplice: deve vivere ad Amsterdam. In questo esempio il luogo di residenza è memorizzato nell'attributo 'postaladdress'

```
$filter= "postaladdress=Amsterdam";
```

2. Il nome deve iniziare con una "a". In questo esempio, l'UID è composto dal nome, quindi:

```
$filter= "uid=a*";
```

3. L'utente deve essere attivo. Questo valore è memorizzato nell'attributo predefinito `shadowexpire`, con valore `-1`. A seconda della configurazione del server, `shadowexpire` può contenere una miriade di valori, anche le date sono possibili. Se un utente è inattivo, `shadowexpire` sarà `1`. Per essere sicuri di ottenere tutti gli utenti tranne quelli che sono davvero inattivi, non scegliamo di filtrare su `shadowexpire = -1`. Invece diciamo che *non* vogliamo che siano inattivi.

```
$filter= "(!(shadowexpire=1))"; //NOT is represented with "!"
```

Ora la parte più interessante: combina tutti e tre gli esempi. Possiamo farlo con parentesi, AND, OR e NON espressioni

```
$filter= "(&(postaladdress=Amsterdam)(uid=a*)(!(shadowexpire=1)))";
```

Infine, potremmo creare un'istruzione OR usando "|", ad esempio se vogliamo che tutti gli utenti inizino con "a" o "b"

```
$filter= "(|(uid=a*)(uid=n*))";
```

Puoi combinare all'infinito e costruire filtri piuttosto impressionanti, divertiti a provare!

Leggi Iniziare con ldap online: <https://riptutorial.com/it/ldap/topic/5576/iniziare-con-ldap>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con ldap	Community , SJDS