



EBook Gratis

APRENDIZAJE

Idap

Free unaffiliated eBook created from
Stack Overflow contributors.

#Idap

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con ldap.....	2
Observaciones.....	2
Examples.....	2
Configurando PHP para trabajar con LDAP.....	2
Creditos.....	6

Acerca de

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

Capítulo 1: Empezando con ldap

Observaciones

Esta sección proporciona una descripción general de qué es ldap y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de ldap, y vincular a los temas relacionados. Dado que la Documentación para ldap es nueva, es posible que deba crear versiones iniciales de los temas relacionados.

Examples

Configurando PHP para trabajar con LDAP

Después de configurar su servidor LDAP correctamente, ahora queremos conectarnos. Por ejemplo utilizando PHP.

- DN = nombre distinguido. Esto significa, en qué parte de la base de datos está trabajando. Puede ser un usuario o un grupo (o incluso configuraciones).
- Entrada: una entidad, por ejemplo un usuario.
- Atributo: algo dentro de una entrada, por ejemplo, nombre, número de teléfono y dirección de correo electrónico.

Conectando

Primero, definimos lo siguiente:

```
$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
```

Ahora, estamos conectados. Lo siguiente que queremos es que el servidor sepa que somos una persona confiable. La solución más sencilla es usar el DN raíz O un usuario existente con los permisos adecuados para ver la base de datos (probablemente todos los usuarios de la base de datos pueden hacerlo de manera predeterminada). Nos autenticamos utilizando la función de enlace.

Establecer opciones

Primero, declaramos estas opciones. Dependiendo de la configuración de su servidor, puede omitir esto.

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

Unión

Supongamos que está utilizando admin y la contraseña es 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
```

Eso es muy bueno. Estamos dentro. Ahora podemos realizar muchas operaciones diferentes. Por ejemplo, podemos buscar, leer, escribir y modificar usuarios e incluso grupos.

buscando

Imagina que queremos mostrar a todos los miembros del grupo "usuarios".

```
$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.
```

Puede usar los bucles foreach para mostrar los datos de una manera agradable.

Esta bien entiendes Ahora proceda con un poco de filtrado. Queremos mostrar solo a los usuarios del grupo "propietarios de bicicletas" que hayan registrado una dirección de correo electrónico. Es importante saber que *todos los* usuarios están en cn = usuarios. Junto a esto, también pueden ser miembros de otros grupos.

```
//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
```

Y ahora proceda de nuevo con `ldap_get_entries`.

Eficiencia en conseguir entradas

Un ejemplo en el que se aborda la eficiencia, cada vez más importante con grandes bases de datos con muchos atributos por entrada. Especialmente cuando almacena imágenes en el atributo `jpegphoto`, puede reducir significativamente su tiempo de carga cuando extrae selectivamente sus entradas.

Imagina que queremos buscar usuarios que estén en el grupo de "propietarios de bicicletas". Hay una gran cantidad de información almacenada dentro de estas entradas, digamos que tienen los siguientes atributos: `cn`, `uid`, `nombre`, `nombre para mostrar`, `correo`, `iniciales`, `móvil`, `número de teléfono`, `calle`, `correo postal`, `código postal` y `jpegphoto`. Ahora solo necesitamos su `uid`, `nombre` e `iniciales`.

Para esto, usamos un 4to parámetro opcional de `ldap_search`:

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

Y, por `ldap_get_entries`, ejecutar `ldap_get_entries` le proporcionará estos datos.

Filtrado avanzado

Por supuesto, puede extraer una base de datos completa con LDAP y procesarla en PHP. Sin embargo, al igual que con MySQL, es mucho más eficiente hacer el procesamiento del lado del servidor. Para demostrar esto, podemos utilizar un filtro avanzado.

Tome uno de los ejemplos anteriores como punto de partida, pero cambie el filtro \$ según la línea a continuación. En nuestro ejemplo, queremos mostrar los datos de los usuarios que están activos. Normalmente, el atributo `shadowexpire` se utiliza para almacenar esta información. Sin embargo, esto puede diferir entre varios sistemas LDAP. No solo queremos mostrar los usuarios que están activos, sino que *también* su nombre debe comenzar con una "a" y deben vivir en *Ámsterdam*.

Básicamente queremos hacer tres cosas:

1. Más fácil: hay que vivir en *amsterdam*. En este ejemplo, el lugar de residencia se almacena en el atributo 'postaladdress'

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

2. El nombre debe comenzar con una "a". En este ejemplo, UID se compone del nombre, por lo que:

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

3. El usuario debe estar activo. Este valor se almacena en el atributo predeterminado `shadowexpire`, con el valor `-1`. Dependiendo de la configuración de su servidor, `shadowexpire` puede contener una gran cantidad de valores, incluso las fechas son posibles. Si un usuario está inactivo, `shadowexpire` será `1`. Para asegurarnos de que obtengamos todos los usuarios, excepto aquellos que están realmente inactivos, no elegimos filtrar en `shadowexpire = -1`. En cambio, decimos que *no* queremos que estén inactivos.

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

Ahora la parte más interesante: combina los tres ejemplos. Podemos hacer esto con corchetes, Y, O y NO expresiones

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

Finalmente, podríamos construir una declaración OR usando "|", por ejemplo, si queremos que todos los usuarios comiencen con "a" o "b"

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

Puedes combinar infinitamente y construir filtros bastante impresionantes, ¡disfruta probando!

Lea Empezando con ldap en línea: <https://riptutorial.com/es/ldap/topic/5576/empezando-con-ldap>

Creditos

S. No	Capítulos	Contributors
1	Empezando con Idap	Community , SJDS