



Бесплатная электронная книга

УЧУСЬ

Idap

Free unaffiliated eBook created from
Stack Overflow contributors.

#Idap

.....	1
1: ldap	2
.....	2
Examples.....	2
PHP LDAP.....	2
.....	6

Около

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

глава 1: Начало работы с Idap

замечания

В этом разделе представлен обзор того, что такое Idap, и почему разработчик может захотеть его использовать.

В нем также должны быть указаны любые большие темы в Idap и ссылки на связанные темы. Поскольку документация для Idap является новой, вам может потребоваться создать начальные версии этих связанных тем.

Examples

Настройка PHP для работы с LDAP

После того, как вы правильно настроили свой LDAP-сервер, мы хотим подключиться. Например, используя PHP.

- DN = отличительное имя. Это означает, в какой части базы данных вы работаете. Может быть пользователем или группой (или даже настройками конфигурации).
- Запись: объект, например пользователь.
- Атрибут: что-то внутри записи, например имя, номер телефона и адрес электронной почты.

соединительный

Сначала мы определяем следующее:

```
$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
```

Теперь мы связаны. Следующее, что мы хотим, это дать серверу знать, что мы заслуживающий доверия человек. Самым простым решением является использование корневого DN или существующего пользователя с надлежащими разрешениями для просмотра базы данных (возможно, каждый пользователь в базе данных может это сделать по умолчанию). Мы выполняем аутентификацию с помощью функции bind.

Настройка параметров

Во-первых, мы объявляем эти параметры. В зависимости от конфигурации вашего сервера вы можете оставить это.

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

переплет

Предположим, вы используете admin и пароль 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
```

Это очень мило. Мы находимся. Теперь мы можем выполнять много разных операций. Например, мы можем искать, читать, писать и изменять пользователей и даже группы.

поиск

Представьте, что мы хотим отобразить всех членов группы «пользователи».

```
$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.
```

Вы можете использовать петли foreach для отображения данных в хорошем виде.

Хорошо, понял? Теперь приступим к некоторой фильтрации. Мы хотим отображать только пользователей в группе «bikeowners», которые зарегистрировали адрес электронной почты. Важно знать, что *все* пользователи находятся в cn = users. Кроме того, они также могут быть членами других групп.

```
//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
```

И теперь переходите к ldap_get_entries снова.

Эффективность в получении записей

Пример, в котором эффективность рассматривается, становится все более важной с большими базами данных с большим количеством атрибутов на запись. Особенно, когда вы храните изображения в атрибуте jpegphoto, это может значительно сократить время загрузки, когда вы выборочно извлекаете свои записи.

Представьте, что мы хотим искать пользователей, которые находятся в группе «bikeowners». Внутри этих записей хранится много информации, скажем, они имеют следующие атрибуты: cn, uid, name, displayname, mail, initials, mobile, telephonenumber,

street, postaladdress, postalcode и jpegphoto. Теперь нам нужны только их uid, имя и инициалы.

Для этого мы используем необязательный 4-й параметр `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
```

И вуаля, запуская `ldap_get_entries`, даст вам только эти данные.

Расширенная фильтрация

Конечно, вы можете вытащить всю базу данных с помощью LDAP и обработать это в PHP. Однако, как и в случае с MySQL, гораздо эффективнее выполнять обработку на стороне сервера. Чтобы продемонстрировать это, мы можем использовать расширенный фильтр.

В качестве отправной точки возьмите один из приведенных выше примеров, но измените фильтр `$` в соответствии с приведенной ниже строкой. В нашем примере мы хотим отображать данные активных пользователей. Обычно для хранения этой информации используется атрибут `shadowexpire`. Однако это может различаться между различными системами LDAP. Мы не только хотим отображать активных пользователей, но и их имя должно начинаться с «а», и они должны жить в Амстердаме.

В принципе, мы хотим сделать три вещи:

1. Самый простой: должен жить в Амстердаме. В этом примере место жительства хранится в атрибуте 'postaladdress'

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

2. Имя должно начинаться с символа «а». В этом примере UID состоит из имени, поэтому:

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

3. Пользователь должен быть активным. Это значение сохраняется в атрибуте по умолчанию `shadowexpire` со значением `-1`. В зависимости от конфигурации вашего сервера `shadowexpire` может содержать множество значений, даже даты возможны. Если пользователь неактивен, `shadowexpire` будет равен `1`. Чтобы убедиться, что мы получаем всех пользователей, кроме тех, кто действительно неактивен, мы не выбираем фильтр на `shadowexpire = -1`. Вместо этого мы говорим, что *не* хотим, чтобы они были неактивными.

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

Теперь самая интересная часть: объедините все три примера. Мы можем сделать это с помощью скобок, AND, OR и NOT выражений

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

Наконец, мы могли бы построить инструкцию OR с помощью «|», например, если мы хотим, чтобы все пользователи, начинающие с «a» или «b»,

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

Вы можете комбинировать бесконечно и создавать довольно впечатляющие фильтры, попробовать!

Прочитайте Начало работы с ldap онлайн: <https://riptutorial.com/ru/ldap/topic/5576/начало-работы-с-ldap>

кредиты

S. No	Главы	Contributors
1	Начало работы с Idap	Community , SJDS