ldap-carddav server

- Installation via Docker

Installation via Docker

Beschreibung:

Ein ldap Server für Adressbücher mit carddav sync.

Installation:

Docker installieren

```
apt install docker.io docker-compose curl
```

Nun Projektverzeichnisse erstellen

```
ldap-carddav-stack/
├── docker-compose.yml
├── .env
├── ldap-carddav/
│   └── conf.php      # deine Konfiguration
├── Dockerfile         # für ldap-carddav
├── ldap-carddav-data
```

```
mkdir -p /root/ldap-carddav-stack/ldap-carddav
mkdir -p /root/ldap-carddav-stack/ldap-carddav-data
```

Dockerfile erstellen zum image bauen

```
nano /root/ldap-carddav-stack/Dockerfile
```

Inhalt

```
FROM debian:bullseye

ENV DEBIAN_FRONTEND=noninteractive

# Abhängigkeiten installieren
RUN apt-get update && apt-get install -y \
    apache2 \
    php \
    php-ldap \
    php-xml \
    php-mbstring \
    php-sqlite3 \
    sqlite3 \
    libapache2-mod-php \
    nano \
    curl \
    ldap-utils \
    composer \
    && apt-get clean

# ldap-carddav klonen
RUN git clone https://github.com/isubsoft/ldap-carddav.git /var/www/html/ldap-carddav

# Composer-Abhängigkeiten installieren
WORKDIR /var/www/html/ldap-carddav
RUN composer install

# Rewrite-Modul aktivieren
RUN a2enmod rewrite

# 000-default.conf ersetzen
RUN rm /etc/apache2/sites-enabled/000-default.conf && \
    echo '<VirtualHost *:80>\n\
    ServerAdmin admin@example.org\n\
    DocumentRoot /var/www/html/ldap-carddav\n\
\n\
    <Directory /var/www/html/ldap-carddav>\n\
        Options Indexes FollowSymLinks\n\
        AllowOverride All\n\
        Require all granted\n\
```

```
        DirectoryIndex server.php\n\
        RewriteEngine On\n\
        RewriteCond %{REQUEST_FILENAME} !-f\n\
        RewriteCond %{REQUEST_FILENAME} !-d\n\
        RewriteRule ^(.*)$ server.php [QSA,L]\n\
    </Directory>\n\
\n\
    Redirect 301 /.well-known/carddav /server.php\n\
</VirtualHost>' > /etc/apache2/sites-enabled/000-default.conf

# Apache starten
CMD ["apachectl", "-D", "FOREGROUND"]

EXPOSE 80
```

Die .env Datei

```
nano /root/ldap-carddav-stack/.env
```

Inhalt

```
LDAP_ORGANISATION=ExampleCorp
LDAP_DOMAIN=example
LDAP_TOP_DOMAIN=local
LDAP_ADMIN_PASSWORD=admin
```

Die compose Datei

```
nano /root/ldap-carddav-stack/docker-compose.yml
```

Inhalt

```
version: '3.8'

services:
  ldap:
```

```
  image: osixia/openldap:1.5.0
  container_name: ldap
  environment:
    LDAP_ORGANISATION: ${LDAP_ORGANISATION}
    LDAP_DOMAIN: ${LDAP_DOMAIN}.${LDAP_TOP_DOMAIN}
    LDAP_ADMIN_PASSWORD: ${LDAP_ADMIN_PASSWORD}
  volumes:
    - ./ldap_data:/var/lib/ldap
    - ./ldap_config:/etc/ldap/slapd.d
  ports:
    - "389:389"

phpldapadmin:
  image: osixia/phpldapadmin:0.9.0
  container_name: phpldapadmin
  environment:
    PHPLDAPADMIN_LDAP_HOSTS: ldap
  ports:
    - "6443:443"

carddav:
  build:
    context: .
    dockerfile: Dockerfile
  container_name: ldap-carddav
  ports:
    - "80:80"
  volumes:
    - ./ldap-carddav/conf.php:/var/www/html/ldap-carddav/conf/conf.php:ro
    - ./ldap-carddav-data:/var/www/html/ldap-carddav/data
  depends_on:
    - ldap
  environment:
    - LDAP_HOST=ldap
    - LDAP_BASE_DN=dc=${LDAP_DOMAIN},dc=${LDAP_TOP_DOMAIN}
    - LDAP_BIND_DN=cn=admin,dc=${LDAP_DOMAIN},dc=${LDAP_TOP_DOMAIN}
    - LDAP_BIND_PASSWORD=${LDAP_ADMIN_PASSWORD}
   # - LDAP_BASE_DN_SYNC=dc=${LDAP_DOMAIN},dc=${LDAP_TOP_DOMAIN}
   # - LDAP_BIND_DN_SYNC=cn=shacker,dc=${LDAP_DOMAIN},dc=${LDAP_TOP_DOMAIN}
   # - LDAP_BIND_PASSWORD_SYNC=1234
```

Container starten:

```
docker-compose up -d
```

PHP File

```
nano /root/ldap-carddav-stack/ldap-carddav/conf.php
```

Version 1: Nur ein Adressbuch, für alle schreibend:

Inhalt

```
<?php

$config = [];

// === TEMP / DATA ===
$config['tmpdir'] = '%systempdir';
$config['datadir'] = '/var/www/html/ldap-carddav/data';

// === DATABASE ===
$config['sync_database'] = [
    'dsn' => 'sqlite:/var/www/html/ldap-carddav/data/cards.db',
    'username' => '',
    'password' => '',
    'options' => [],
    'init_commands' => []
];

// === LDAP SERVER CONFIG ===
$config['server']['ldap'] = [
    'host' => getenv('LDAP_HOST') ?: 'localhost',
    'network_timeout' => 10,
    'connection_security' => 'none'
];

// === LDAP AUTH ===
$config['auth']['ldap'] = [
    'base_dn' => getenv('LDAP_BASE_DN') ?: 'dc=example,dc=local',
    'bind_dn' => '%dn',
```

```
    'bind_pass' => '%p',
    'search_base_dn' => '',
    'search_filter' => '(&(objectclass=inetOrgPerson)(uid=%u))',
    'search_bind_dn' => getenv('LDAP_BIND_DN') ?: 'cn=admin,dc=example,dc=local',
    'search_bind_pw' => getenv('LDAP_BIND_PASSWORD') ?: 'admin',
    'scope' => 'list'
];

// === PRINCIPAL SEARCH ===
$config['principal']['ldap'] = [
    'base_dn' => getenv('LDAP_BASE_DN') ?: 'dc=example,dc=local',
    'search_base_dn' => '',
    'search_filter' => '(&(objectclass=inetOrgPerson)(uid=*))',
    'search_bind_dn' => getenv('LDAP_BIND_DN') ?: 'cn=admin,dc=example,dc=local',
    'search_bind_pw' => getenv('LDAP_BIND_PASSWORD') ?: 'admin',
    'scope' => 'list',
    'fieldmap' => [
        'id' => 'uid',
        'displayname' => 'cn',
        'mail' => 'mail'
    ]
];

// Hinweis: Die folgenden Einträge sind stark gekürzt. Siehe Original für volle Struktur.
// Du kannst z. B. $config['card']['addressbook']['ldap']['me'], ['global'], ['personal'] wie oben mit getenv()
einbinden.

// Beispiel für ein Adressbuch-Eintrag mit bind_dn über ENV
$config['card']['addressbook']['ldap']['personal'] = [
    'name' => 'starface',
    'description' => 'Starface Kontakte',
    'user_specific' => true,
    'writable' => true,

    'group_LDAP_Object_Classes' => ['groupOfNames'],
    'group_required_fields' => ['cn', 'member'],
    'group_LDAP_rdn' => 'cn',
    'group_member_map' => [ 'MEMBER' => [ 'field_name' => 'member' ] ],

    'base_dn' => getenv('LDAP_BASE_DN') ?: 'dc=example,dc=local',
```

```
    'filter' => '(objectClass=inetOrgPerson)',
    'bind_dn' => getenv('LDAP_BIND_DN') ?: 'cn=admin,dc=example,dc=local',
    'bind_pass' => getenv('LDAP_BIND_PASSWORD') ?: 'admin',
    'scope' => 'sub',

    'LDAP_Object_Classes' => ['inetOrgPerson'],
    'required_fields' => ['cn','sn'],
    'LDAP_rdn' => 'cn',

    // Schreibrechte aktivieren
    //'field_acl' => [
   //     'eval' => 'w'
   //     'list' => ['displayName', 'homePhone', 'telephoneNumber', 'facsimileTelephoneNumber', 'pager', 'mobile',
'homePostalAddress', 'preferredLanguage']
□□□□□□□□□□  // leere Liste = alles erlaubt
 //   ],

    // Vollständige Feldzuordnung für Outlook / CalDAV
    'fieldmap' => [
        'FN' => ['field_name' => 'cn'],
        'N' => ['field_name' => [
            'last_name' => 'sn',
            'first_name' => 'givenName',
            'prefix' => 'personalTitle'
        ]],
        'EMAIL' => ['field_name' => 'mail'],
        'ORG' => ['field_name' => [
            'org_name' => 'o',
            'org_unit_name' => 'ou'
        ]],
        'TITLE' => ['field_name' => 'title'],
        'ROLE' => ['field_name' => 'employeeType'],
        'NICKNAME' => ['field_name' => 'displayName'],
        'PHOTO' => [[
            'field_name' => 'jpegphoto',
            'parameters' => [],
            'reverse_map_parameter_index' => 0,
            'decode_file' => true
        ]],
        'NOTE' => ['field_name' => 'description'],
```

```
        'TEL' => [
    [   // Fax number
    'field_name' => 'facsimileTelephoneNumber',
    'parameters' => [
        ['TYPE' => ['fax']],
        ['TYPE' => ['fax', 'work']],
        ['TYPE' => ['work', 'fax']],
        ['TYPE' => 'facsimile'],
        ['TYPE' => ['voice', 'fax']],
        ['TYPE' => ['fax', 'voice']],
        null
    ],
    'reverse_map_parameter_index' => 0
],
    [   // Work number
        'field_name' => 'telephoneNumber',
        'parameters' => [
            ['TYPE' => ['work']],
            ['TYPE' => ['voice', 'work']],
            ['TYPE' => 'work'],
            ['TYPE' => 'voice'],
            null
        ],
        'reverse_map_parameter_index' => 0
    ],
    [   // Home number
        'field_name' => 'homePhone',
        'parameters' => [
            ['TYPE' => ['home']],
            ['TYPE' => ['voice', 'home']],
            ['TYPE' => 'home'],
            null
        ],
        'reverse_map_parameter_index' => 0
    ],
    [   // Mobile number
        'field_name' => 'mobile',
        'parameters' => [
            ['TYPE' => ['cell']],
            ['TYPE' => ['voice', 'cell']],
```

```
            ['TYPE' => 'cell'],
            null
        ],
        'reverse_map_parameter_index' => 0
    ],
    [   // Pager
        'field_name' => 'pager',
        'parameters' => [
            ['TYPE' => ['pager']],
            null
        ],
        'reverse_map_parameter_index' => 0
    ]
],
   'NOTE' => [
    'field_name' => 'description',
    'parameters' => [],
    'reverse_map_parameter_index' => 0
],
     'ADR' => [
    [
        'field_name' => [
            'po_box'      => 'postOfficeBox',
            'street'      => 'street',
            'locality'    => 'l',
            'province'    => 'st',
            'postal_code' => 'postalCode'
        ],
        'parameters' => ['TYPE' => 'work'],
        'map_component_separator' => ';',
        'reverse_map_parameter_index' => 0
    ],
    // Privatadresse
    [
        'field_name' => 'homePostalAddress',
        'parameters' => ['TYPE' => 'home'],
        'map_component_separator' => '$',
        'reverse_map_parameter_index' => 0
    ]
],
```

```
        'LANG' => ['field_name' => 'preferredLanguage']
    ]
];
```

Version 2, nur schreibend mit einer Gruppe

```
<?php

$config = [];

// === TEMP / DATA ===
$config['tmpdir'] = '%systempdir';
$config['datadir'] = '/var/www/html/ldap-carddav/data';

// === DATABASE ===
$config['sync_database'] = [
    'dsn' => 'sqlite:/var/www/html/ldap-carddav/data/cards.db',
    'username' => '',
    'password' => '',
    'options' => [],
    'init_commands' => []
];

// === LDAP SERVER CONFIG ===
$config['server']['ldap'] = [
    'host' => getenv('LDAP_HOST') ?: 'localhost',
    'network_timeout' => 10,
    'connection_security' => 'none'
];

// === LDAP AUTH ===
$config['auth']['ldap'] = [
    'base_dn' => getenv('LDAP_BASE_DN') ?: 'dc=example,dc=local',
    'bind_dn' => '%dn',
    'bind_pass' => '%p',
    'search_base_dn' => '',
    'search_filter' => '(&(objectclass=inetOrgPerson)(uid=%u))',
    'search_bind_dn' => getenv('LDAP_BIND_DN') ?: 'cn=admin,dc=example,dc=local',
    'search_bind_pw' => getenv('LDAP_BIND_PASSWORD') ?: 'admin',
    'scope' => 'list'
```

```
];

// === PRINCIPAL SEARCH ===
$config['principal']['ldap'] = [
    'base_dn' => getenv('LDAP_BASE_DN') ?: 'dc=example,dc=local',
    'search_base_dn' => '',
    'search_filter' => '(&(objectclass=inetOrgPerson)(uid=*))',
    'search_bind_dn' => getenv('LDAP_BIND_DN') ?: 'cn=admin,dc=example,dc=local',
    'search_bind_pw' => getenv('LDAP_BIND_PASSWORD') ?: 'admin',
    'scope' => 'list',
    'fieldmap' => [
        'id' => 'uid',
        'displayname' => 'cn',
        'mail' => 'mail'
    ]
];

// Funktion einfügen oder inkludieren
function getWritableUserIdsFromLdapGroup(): array {
    $groupCN = getenv('LDAP_WRITE_GROUP') ?: 'write';
    $ldapDomain = getenv('LDAP_DOMAIN') ?: 'example';
    $ldapTopDomain = getenv('LDAP_TOP_DOMAIN') ?: 'local';
    $groupDN = "cn={$groupCN},dc={$ldapDomain},dc={$ldapTopDomain}";

    $ldap = ldap_connect('localhost');
    ldap_set_option($ldap, LDAP_OPT_PROTOCOL_VERSION, 3);

    $bindDN = getenv('LDAP_BIND_DN') ?: "cn=admin,dc={$ldapDomain},dc={$ldapTopDomain}";
    $bindPW = getenv('LDAP_BIND_PASSWORD') ?: 'admin';

    if (!@ldap_bind($ldap, $bindDN, $bindPW)) {
        error_log("⚠ LDAP Bind fehlgeschlagen mit DN: $bindDN");
        return [];
    }

    $results = ldap_search($ldap, $groupDN, '(objectClass=groupOfNames)', ['member']);
    if (!$results) {
        error_log("⚠ Konnte Mitglieder von Gruppe $groupDN nicht abrufen.");
        return [];
    }
```

```
    $entries = ldap_get_entries($ldap, $results);
    if (!isset($entries[0]['member'])) {
        return [];
    }

    $uids = [];
    foreach ($entries[0]['member'] as $key => $dn) {
        if (is_numeric($key) && preg_match('/uid=([^,]+)/i', $dn, $matches)) {
            $uids[] = $matches[1];
        }
    }

    return $uids;
}

// Benutzer-ID abrufen
$user = $_SERVER['PHP_AUTH_USER'] ?? null;

// Schreibrechte prüfen
$writeUsers = getWritableUserIdsFromLdapGroup();
$isWritable = in_array($user, $writeUsers);



// Hinweis: Die folgenden Einträge sind stark gekürzt. Siehe Original für volle Struktur.
// Du kannst z. B. $config['card']['addressbook']['ldap']['me'], ['global'], ['personal'] wie oben mit getenv()
einbinden.

// Beispiel für ein Adressbuch-Eintrag mit bind_dn über ENV
$config['card']['addressbook']['ldap']['personal'] = [
    'name' => 'starface',
    'description' => 'Starface Kontakte',
    'user_specific' => true,
    'writable' => $isWritable,

    'group_LDAP_Object_Classes' => ['groupOfNames'],
    'group_required_fields' => ['cn', 'member'],
    'group_LDAP_rdn' => 'cn',
    'group_member_map' => [ 'MEMBER' => [ 'field_name' => 'member' ] ],
```

```
    'base_dn' => getenv('LDAP_BASE_DN') ?: 'dc=example,dc=local',
    'filter' => '(objectClass=inetOrgPerson)',
    'bind_dn' => getenv('LDAP_BIND_DN') ?: 'cn=admin,dc=example,dc=local',
    'bind_pass' => getenv('LDAP_BIND_PASSWORD') ?: 'admin',
    'scope' => 'sub',

    'LDAP_Object_Classes' => ['inetOrgPerson'],
    'required_fields' => ['cn','sn'],
    'LDAP_rdn' => 'cn',

    // Schreibrechte aktivieren
    //'field_acl' => [
   //     'eval' => 'w'
   //     'list' => ['displayName', 'homePhone', 'telephoneNumber', 'facsimileTelephoneNumber', 'pager', 'mobile',
'homePostalAddress', 'preferredLanguage']
□□□□□□□□□□□ // leere Liste = alles erlaubt
 //   ],

    // Vollständige Feldzuordnung für Outlook / CalDAV
    'fieldmap' => [
        'FN' => ['field_name' => 'cn'],
        'N' => ['field_name' => [
            'last_name' => 'sn',
            'first_name' => 'givenName',
            'prefix' => 'personalTitle'
        ]],
        'EMAIL' => ['field_name' => 'mail'],
        'ORG' => ['field_name' => [
            'org_name' => 'o',
            'org_unit_name' => 'ou'
        ]],
        'TITLE' => ['field_name' => 'title'],
        'ROLE' => ['field_name' => 'employeeType'],
        'NICKNAME' => ['field_name' => 'displayName'],
        'PHOTO' => [[
            'field_name' => 'jpegphoto',
            'parameters' => [],
            'reverse_map_parameter_index' => 0,
            'decode_file' => true
        ]],
```

```
        'NOTE' => ['field_name' => 'description'],
        'TEL' => [
     [   // Fax number
    'field_name' => 'facsimileTelephoneNumber',
    'parameters' => [
        ['TYPE' => ['fax']],
        ['TYPE' => ['fax', 'work']],
        ['TYPE' => ['work', 'fax']],
        ['TYPE' => 'facsimile'],
        ['TYPE' => ['voice', 'fax']],
        ['TYPE' => ['fax', 'voice']],
        null
    ],
    'reverse_map_parameter_index' => 0
],
    [   // Work number
        'field_name' => 'telephoneNumber',
        'parameters' => [
            ['TYPE' => ['work']],
            ['TYPE' => ['voice', 'work']],
            ['TYPE' => 'work'],
            ['TYPE' => 'voice'],
            null
        ],
        'reverse_map_parameter_index' => 0
    ],
    [   // Home number
        'field_name' => 'homePhone',
        'parameters' => [
            ['TYPE' => ['home']],
            ['TYPE' => ['voice', 'home']],
            ['TYPE' => 'home'],
            null
        ],
        'reverse_map_parameter_index' => 0
    ],
    [   // Mobile number
        'field_name' => 'mobile',
        'parameters' => [
            ['TYPE' => ['cell']],
```

```
            ['TYPE' => ['voice', 'cell']],
            ['TYPE' => 'cell'],
            null
        ],
        'reverse_map_parameter_index' => 0
    ],
    [   // Pager
        'field_name' => 'pager',
        'parameters' => [
            ['TYPE' => ['pager']],
            null
        ],
        'reverse_map_parameter_index' => 0
    ]
],
   'NOTE' => [
    'field_name' => 'description',
    'parameters' => [],
    'reverse_map_parameter_index' => 0
],
     'ADR' => [
    [
        'field_name' => [
            'po_box'      => 'postOfficeBox',
            'street'      => 'street',
            'locality'    => 'l',
            'province'    => 'st',
            'postal_code' => 'postalCode'
        ],
        'parameters' => ['TYPE' => 'work'],
        'map_component_separator' => ';',
        'reverse_map_parameter_index' => 0
    ],
    // Privatadresse
    [
        'field_name' => 'homePostalAddress',
        'parameters' => ['TYPE' => 'home'],
        'map_component_separator' => '$',
        'reverse_map_parameter_index' => 0
    ]
```

```
],
        'LANG' => ['field_name' => 'preferredLanguage']
    ]
];
```

Nun noch die. env erweitern. write ist hier der gruppenname

```
...
LDAP_WRITE_GROUP=write
...
```

Nun noch wieder das Adressbuch mit syncdb.php löschen neu anlegen, siehe hier

Aufrufen LDAP und co:

- **phpLDAPadmin:** https://localhost:6443
  Benutzername aus unserem Beispiel : cn=admin,dc=example,dc=local
  Passwort aus unserem Beispiel : admin

- **ldap-carddav WebDAV/CardDAV:** http://localhost/ldap-carddav/
  Die Benutzer dazu werden im LDAP Webgui angelegt, dazu ein ein eigenes Kaptitel

Datenbank initialiseren

```
docker-compose exec carddav /bin/bash
sqlite3 /var/www/html/ldap-carddav/data/cards.db < /var/www/html/ldap-carddav//sql/sqlite/ddl.sql
php /var/www/html/ldap-carddav/src/App/syncdb.php init
```

Danach vom conatiner wieder abmelden und chmod 777 über die card.db

```
chown www-data:www-data -R /var/www/html/ldap-carddav/data
```

PHP ini Änderungen durchführen und neu mit der Datenbank synchroniesieren:

Wenn die PHP geändert wird um zum Beispiel Felder hinzugefügt werden, muss ide Datenbank neu initialisert werden.

```
docker-compose exec carddav /bin/bash
php /var/www/html/ldap-carddav/src/App/syncdb.php
```

Ausgabe:
Dort 0 Auswählen

```
hoose the entity you want to operate upon. Enter 0 for addressbook and 1 for user:
```

Nun mit 3 bestätigen

```
Enter the operation to perform on address book. Enter 0 to list, 1 to add, 2 to rename and 3 to delete:
```

Nun Adressbuchname eingeben:personal.

```
Enter name of the address book to delete:
```

Nun wurde das Buch gelöscht

```
Address book 'personal' has been deleted.
```

Nun kann ein neuer init stattfinden

```
php /var/www/html/ldap-carddav/src/App/syncdb.php init
```

Nun die Rechte neu setzten

```
chown www-data:www-data -R /var/www/html/ldap-carddav/data
```

Ausgabe:

```
Initializing sync database ...
Address book 'personal' has been successfully added to sync database.
```

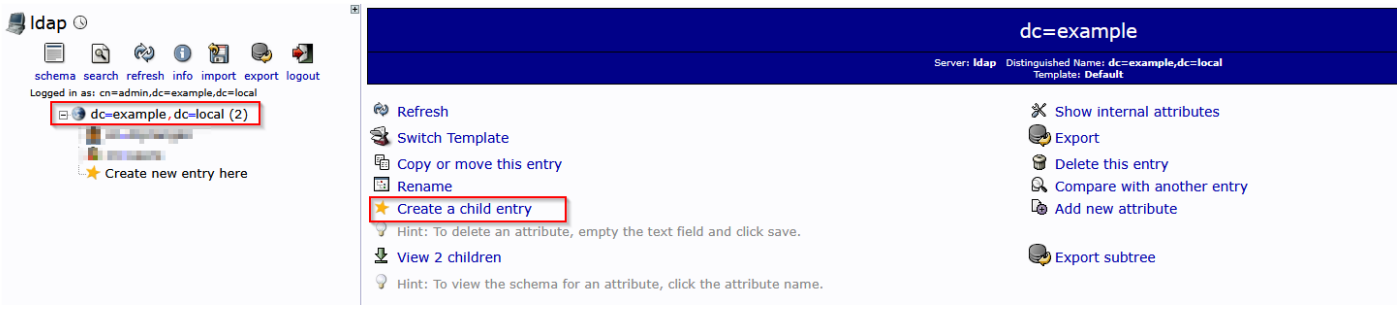
```
Address book(s) successfully imported.
```

Benutzer anlegen:

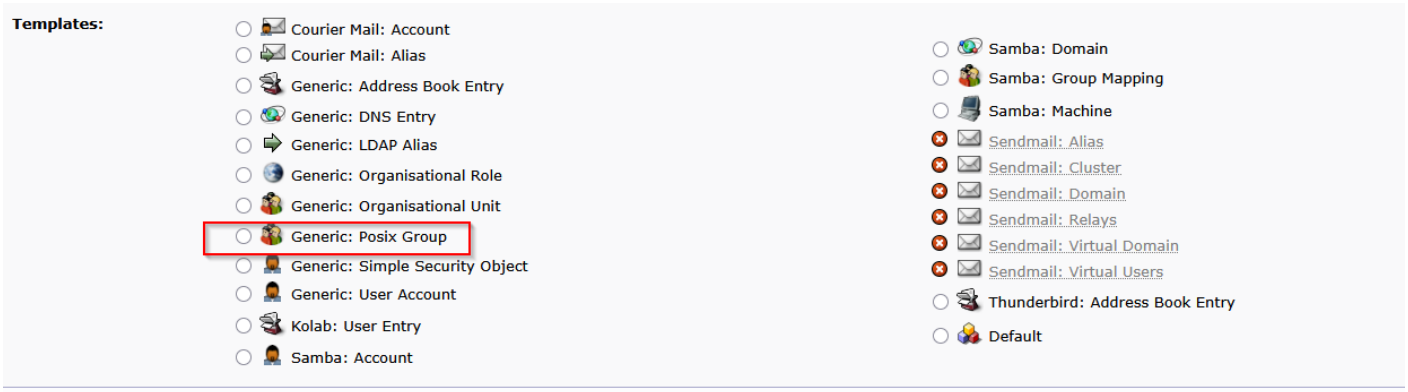
Im LDAP Webgui einloggen unter

```
https://<ip>:6443
```

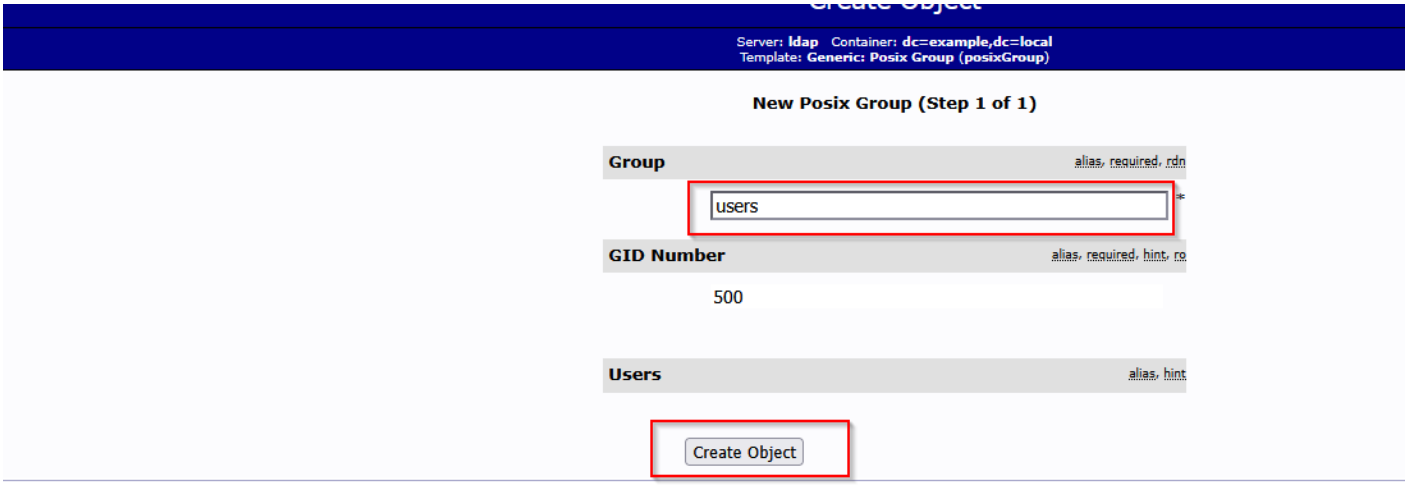
Dort neues child Element anlegen...



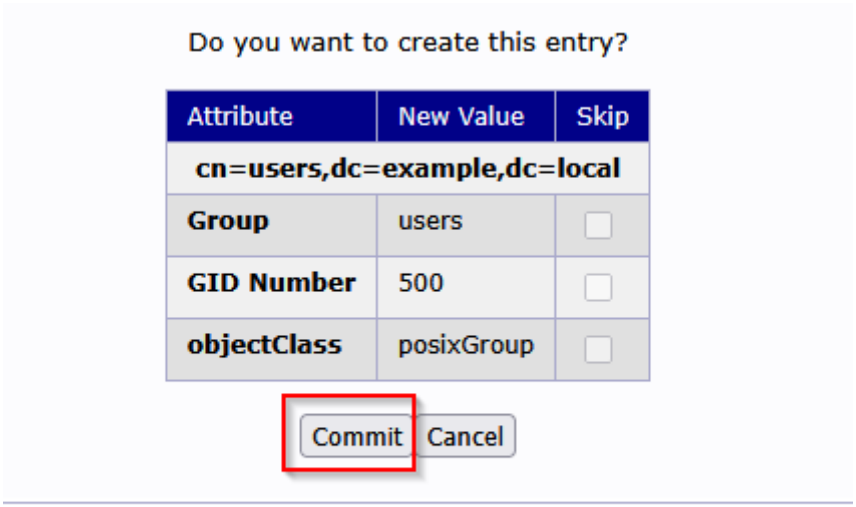
Vom Typ PosixGroup



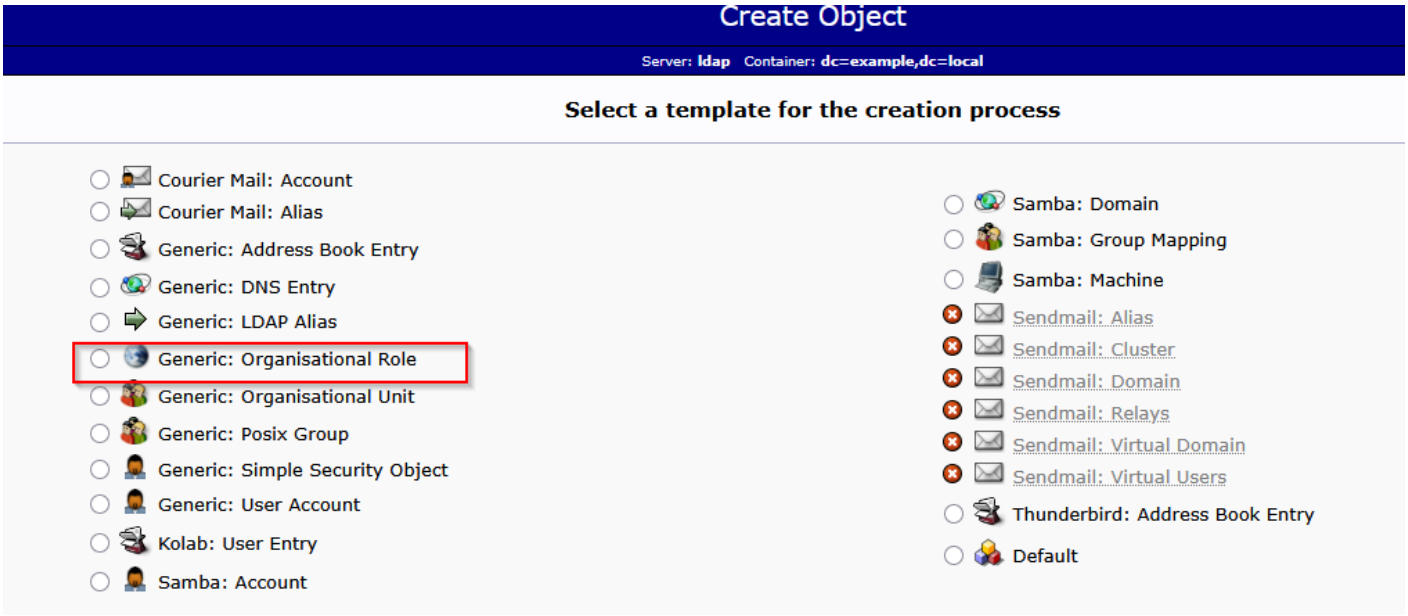
mit dem namen users



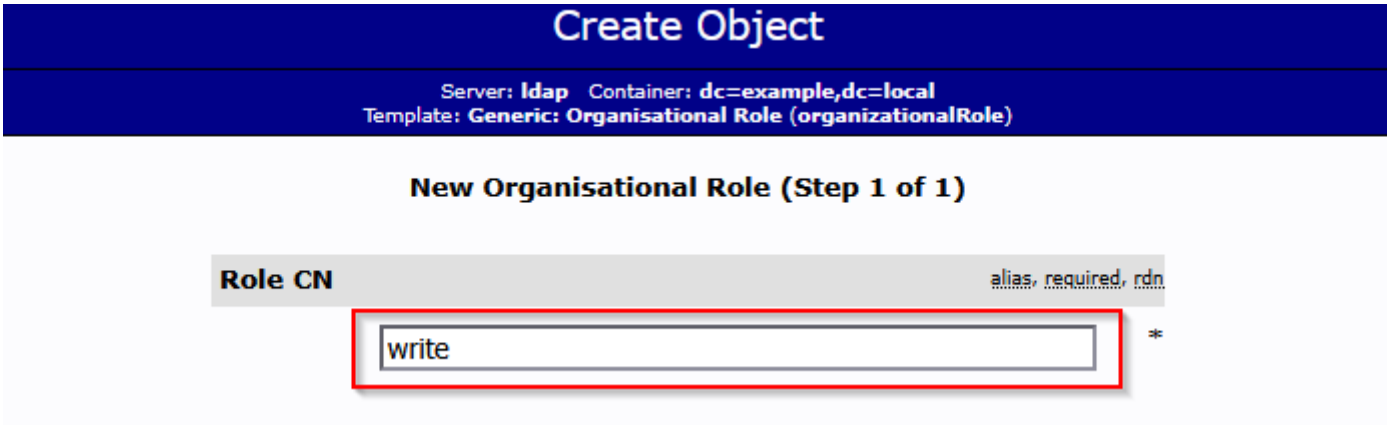
Nun nochmals bestätigen



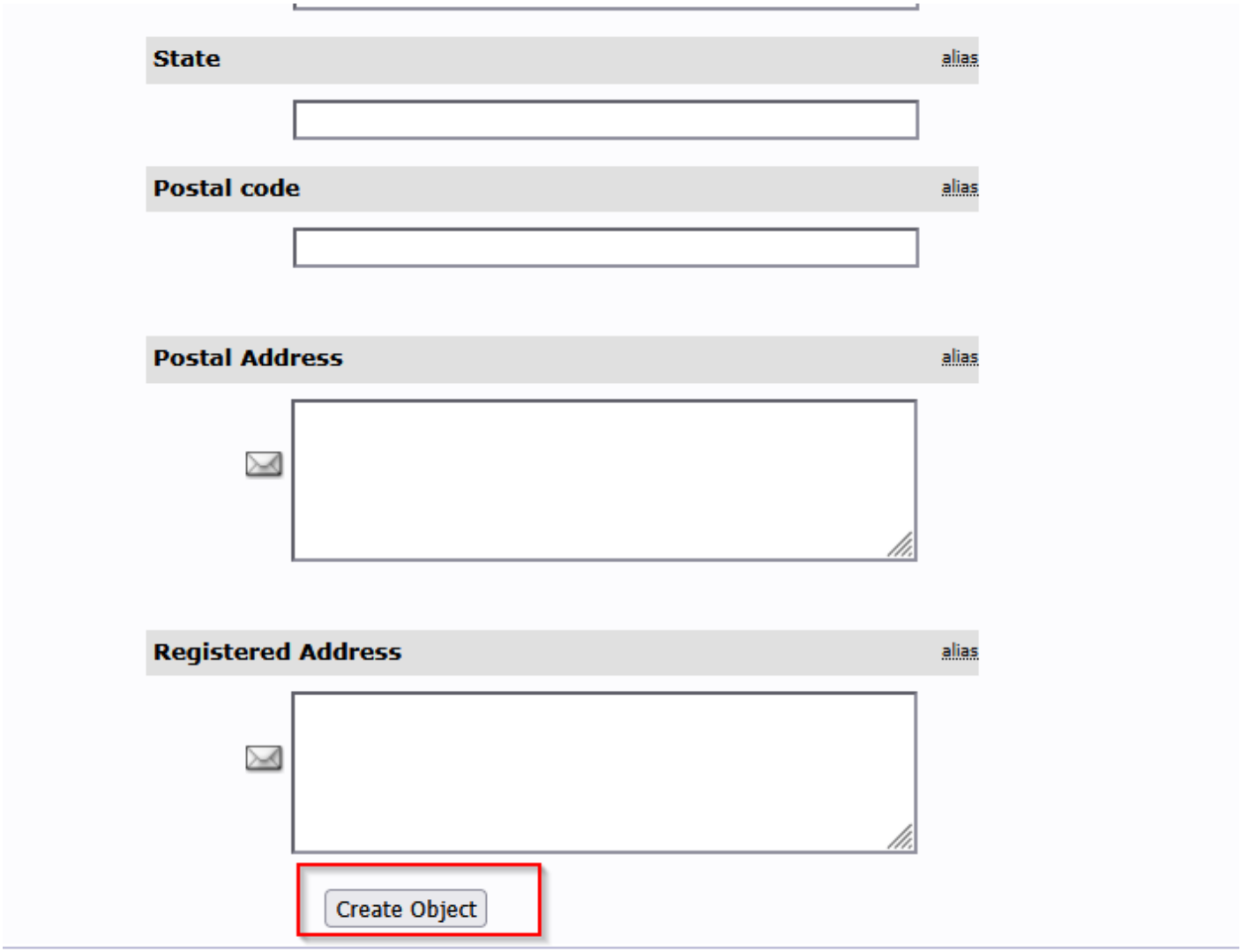
nun eine weitere elemt vom typ organizationRole mit den namen write anlegen



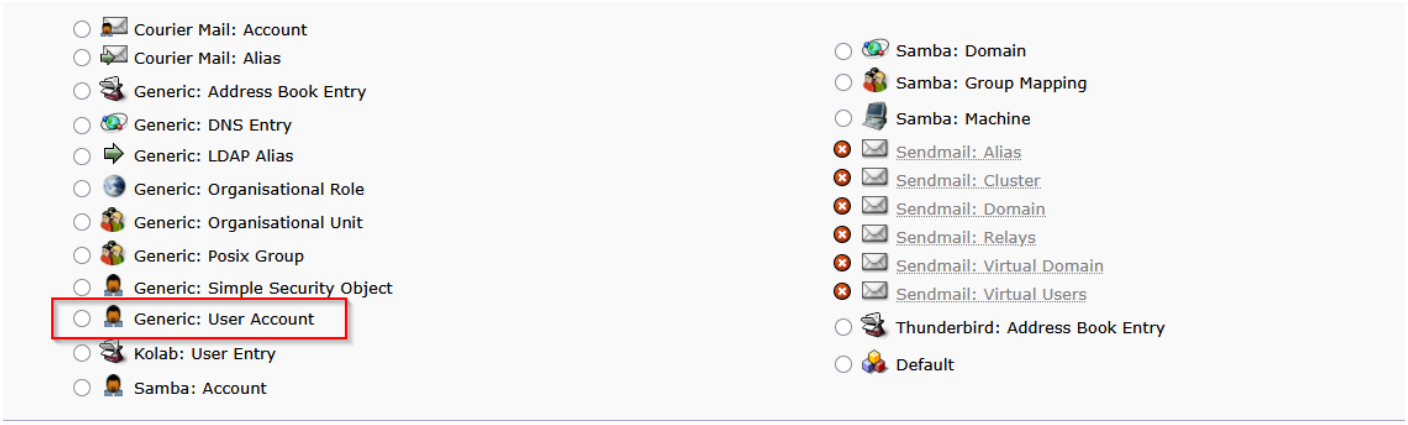
nun den Namen vergeben



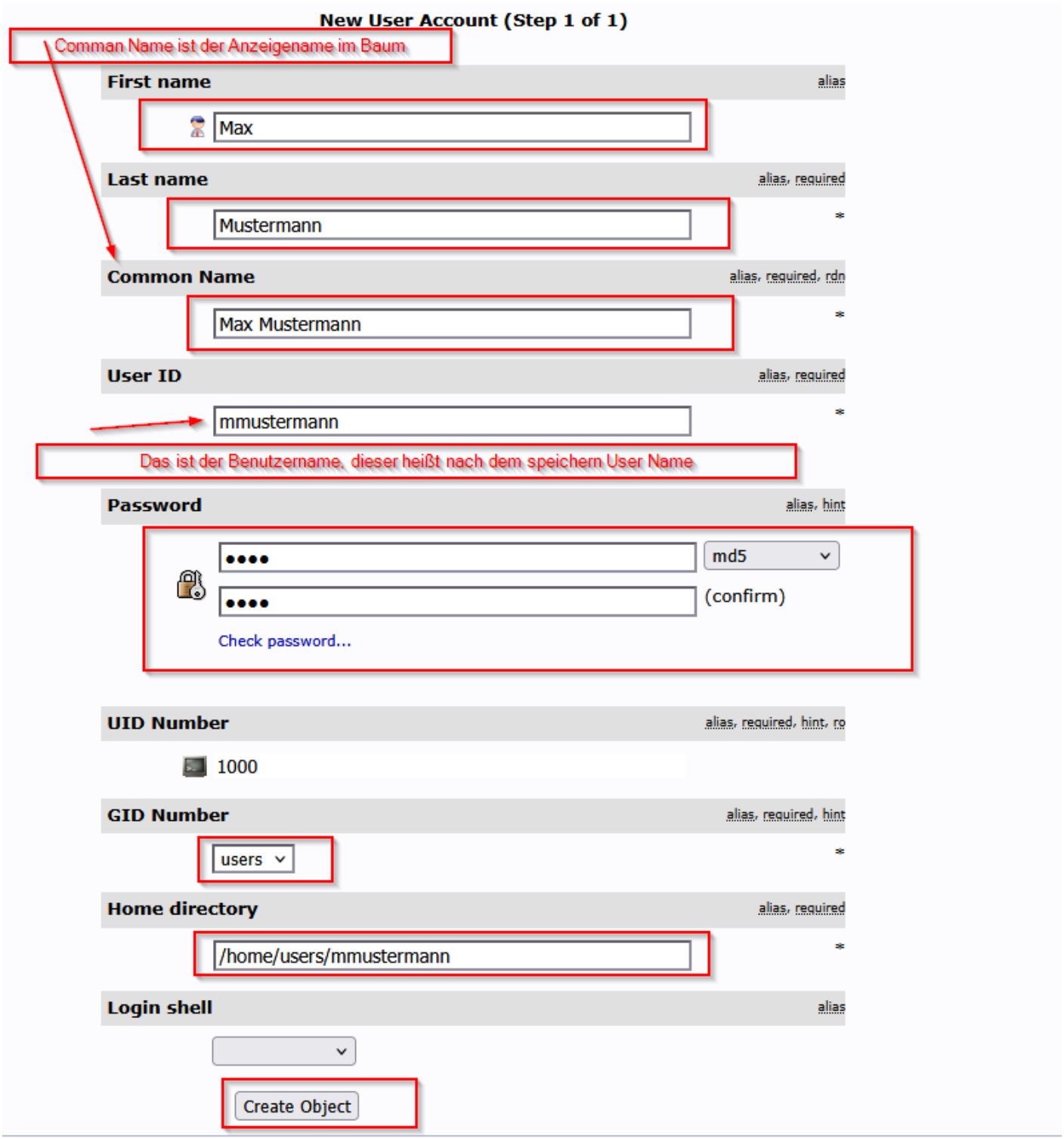
runter scrollen bis create object



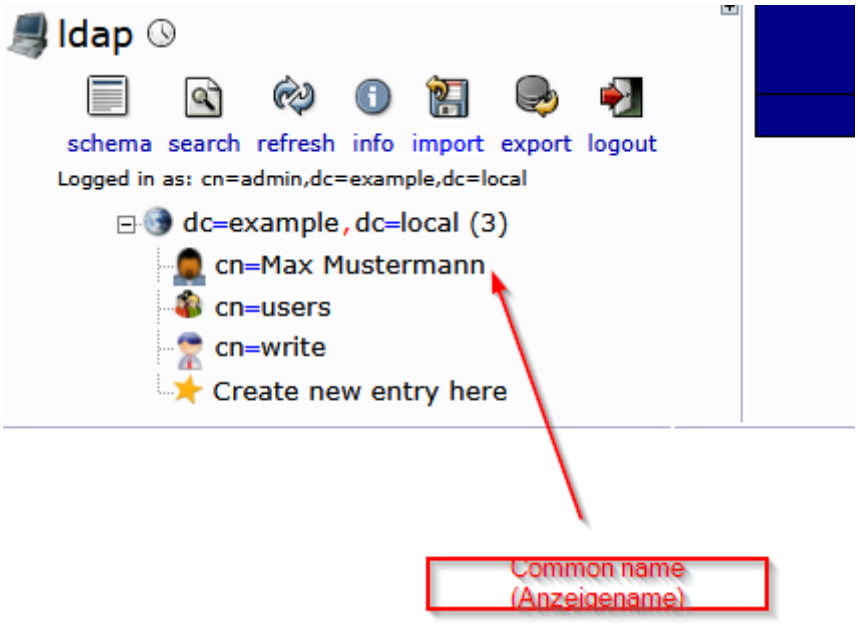
Nun einen Benutzer anlegen vom typ inetOrgPerson



Daten ausfüllen



Nun sieht das ganz so aus:



Nun den benutzer im carddav Backend anlegen, nicht im ldap, das haben wir gerade getan.

```
docker-compose exec carddav /bin/bash
php /var/www/html/ldap-carddav/src/App/syncdb.php
```

Fehlersuche:

Invalid Credentials:

Sollte Verbindung nicht zustande kommen, dann mal in die Apache2 log schauen.
Im container anmelden

```
docker-compose exec carddav /bin/bash
cat /var/log/apache2/error.log
```

Ausgabe:

```
[Sun May 18 09:28:57.236797 2025] [php:warn] [pid 13:tid 13] [client 172.0.0.2:61975] PHP Warning:  ldap_bind(): Unable to bind to server: Invalid credentials in /var/www/h
ml/ldap-carddav/src/DAV/Utility/LDAP.php on line 87
[Sun May 18 09:28:57.237857 2025] [php:notice] [pid 13:tid 13] [client 172.0.0.2:61975] Could not establish bind connection to backend server in ISubsoft\\DAV\\Auth\\Backen
\\LDAP::validateUserPass at line 65
[Sun May 18 09:29:15.836070 2025] [php:warn] [pid 14:tid 14] [client 172.0.0.2:62681] PHP Warning:  ldap_bind(): Unable to bind to server: Invalid credentials in /var/www/h
ml/ldap-carddav/src/DAV/Utility/LDAP.php on line 87
[Sun May 18 09:29:15.836147 2025] [php:notice] [pid 14:tid 14] [client 172.0.0.2:62681] Could not establish bind connection to backend server in ISubsoft\\DAV\\Auth\\Backen
\\LDAP::validateUserPass at line 65
[Sun May 18 09:31:28.821090 2025] [php:warn] [pid 9:tid 9] [client 172.0.0.2:62874] PHP Warning:  ldap_bind(): Unable to bind to server: Invalid credentials in /var/www/htm
/ldap-carddav/src/DAV/Utility/LDAP.php on line 87
[Sun May 18 09:31:28.821189 2025] [php:notice] [pid 9:tid 9] [client 172.0.0.2:62874] Could not establish bind connection to backend server in ISubsoft\\DAV\\Auth\\Backend\
LDAP::validateUserPass at line 65
root@7dcf55307843:/var/www/html/ldap-carddav#
```

Testen der Daten

Hinweis: Paramter -w ist das Passwort aus der .env Datei für ldap

```
ldapwhoami -x -D "cn=admin,dc=example,dc=local" -w admin -H ldap://ldap
```

Ausgabe:

```
root@c44a0bc55808:/var/www/html/ldap-carddav# ldapwhoami -x -D "cn=admin,dc=example,dc=local" -w admin -H ldap://ldap
dn:cn=admin,dc=example,dc=local
```

Testen ob ein Objekt sich ändern lässt

```
ldapmodify -x -D "cn=admin,dc=example,dc=local" -w admin -H ldap://ldap
```

Einfügen und dann strg+d drücken

```
dn: cn=shacker,dc=example,dc=local
changetype: modify
replace: mail
mail: test@example.org
```

Ein komplettes Objekt ausgeben

```
ldapsearch -x \
  -D "cn=admin,dc=example,dc=local" \
  -w admin \
  -b "cn=Wolf\, SDaniel,dc=example,dc=local" \
  -LLL
```

Ausgabe:

```
dn: cn=Wolf\2C SDaniel,dc=example,dc=local
objectClass: inetOrgPerson
cn: Wolf, SDaniel
sn: Wolf
givenName: SDaniel
mail: daniel.wolf@test.de
```

Database readonly:

Fehler aus der /var/log/apache2/error.log

```
 attempt to write a readonly database, referer: http://192.168.0.231/server.php/addressbooks/mprangen/
[Thu Jul 31 19:11:24.152676 2025] [php7:notice] [pid 9:tid 9] [client 192.168.0.26:51450] Database query could not be executed: ISubsoft\\DAV\\CardDAV\\Backend\\LDAP::fullSyncOperation at line no 1856, SQLSTATE[HY000]: General error: 8 attempt to write a readonly database, referer: http://192.168.0.231/server.php/addressbooks/mprangen/
[Thu Jul 31 19:12:16.335653 2025] [php7:notice] [pid 12:tid 12] [client 192.168.0.26:51452] Database query could not be executed: ISubsoft\\DAV\\CardDAV\\Backend\\LDAP::fullSyncOperation at line no 1856, SQLSTATE[HY000]: General error: 8 attempt to write a readonly database, referer: http://192.168.0.231/server.php/addressbooks/mprangen/
[Thu Jul 31 19:12:18.828919 2025] [php7:notice] [pid 13:tid 13] [client 192.168.0.26:51453] Database query could not be executed: ISubsoft\\DAV\\CardDAV\\Backend\\LDAP::fullSyncOperation at line no 1856, SQLSTATE[HY000]: General error: 8 attempt to write a readonly database, referer: http://192.168.0.231/server.php/addressbooks/mprangen/
```

Wenn syncdb.php ohne Probleme ausgeführt werden kann, liegt es nicht an Dateirechten bei root, sondern bei www-datat user.
wir werden das Verzeichnis nochmal neu mit rechten vergeben.

In den Container einloggen

```
 docker-compose exec carddav /bin/bash
```

Dnn rechte vergeben

```
chown www-data:www-data -R /var/www/html/ldap-carddav/data
```

Ansonsten schauen wir uns mal an mit welchen umser der apache2 ausgeführt wird

```
ps aux | grep apache



root          1  0.0  0.0   2480   580 ?        Ss   19:32   0:00 /bin/sh /usr/sbin/apachectl -D FOREGROUND
root          8  0.0  1.4 206336 28308 ?        S    19:32   0:00 /usr/sbin/apache2 -D FOREGROUND
www-data      9  0.0  1.1 208744 23500 ?        S    19:32   0:00 /usr/sbin/apache2 -D FOREGROUND
www-data     10  0.0  1.1 208876 24056 ?        S    19:32   0:00 /usr/sbin/apache2 -D FOREGROUND
www-data     11  0.0  1.1 208876 23536 ?        S    19:32   0:00 /usr/sbin/apache2 -D FOREGROUND
www-data     12  0.0  1.1 208876 23556 ?        S    19:32   0:00 /usr/sbin/apache2 -D FOREGROUND
www-data     13  0.0  0.4 206376  9264 ?        S    19:32   0:00 /usr/sbin/apache2 -D FOREGROUND
root         42  0.0  0.0   3240   648 pts/0    S+   19:38   0:00 grep apache
```