Using University Accounts

Prerequisites

For information about installing the epiCAS gem or using it for authentication please refer to the Guide - Authenticating University Users.

Please note that the University's LDAP servers can only be accessed when connected to the VPN. If you use the following functionality when you are not connected it will error and/or time out.

Looking up a University account

You can obtain a University account's details via LDAP using the following code:

```
SheffieldLdapLookup::LdapFinder.new(identifier).lookup
```

Where **`identifier`** is either a University username or an @sheffield.ac.uk email address. This will return a structure containing all of the published details. You might want to start a Rails console (**`bundle exec rails c`**) and run the above code with your username to see the available details.

You can look up specific details from structure as if it were a hash:

```
# Using a username

SheffieldLdapLookup::LdapFinder.new('aca03cem').lookup[:mail]
=> ["Chris.Murray@sheffield.ac.uk"]
# Note that email addresses are sometimes returned in mixed case.

SheffieldLdapLookup::LdapFinder.new('aca03cem').lookup[:givenName]
=> ["Christopher"]

SheffieldLdapLookup::LdapFinder.new('aca03cem').lookup[:sn]
=> ["Murray"]

SheffieldLdapLookup::LdapFinder.new('aca03cem').lookup[:ou]
=> ["EPG"]
# This is the department code associated with the account.

SheffieldLdapLookup::LdapFinder.new('aca03cem').lookup[:dn]
=> ["uid=aca03cem,ou=Staff,ou=Users,dc=sheffield,dc=ac,dc=uk"]
# This indicates the part of the tree in which the user's account is located.

# Using an email address

SheffieldLdapLookup::LdapFinder.new('chris.murray@sheffield.ac.uk').lookup[:uid]
=> ["aca03cem"]
# Note that the email address is not case sensitive.
```

Department codes and names

LDAP can only provide the three letter department code associated with an account. If you need to map department codes onto department names (e.g. to display a user's department on a user management page) you will need to store and process the code-name pairs in your application. You can download **`DepartmentCodesAndNames.csv`** from Canvas.

User type (e.g. staff or student)

There is no dedicated attribute for user type. However, the **dn** attribute indicates the part of the tree in which the user's account is located, and this can be used to determine a user type. The mappings used by epiCAS within **epi_cas_settings.yml** are:

ou=staff,ou=users	:staff
ou=retired,ou=users	:staff_retired
ou=honorary,ou=staff,ou=users	:staff_honorary
ou=visiting,ou=staff,ou=users	:staff_visiting
ou=external,ou=users	:external
ou=research,ou=postgraduates,ou=students,ou=users	:student_pg_research
ou=taught,ou=postgraduates,ou=students,ou=users	:student_pg_taught
ou=undergraduates,ou=students,ou=users	:student_undergraduate
ou=roles,ou=users	:role

Adding users using a University username or email address

It is possible to create a fully populated user record from only a username or an email address. Your **User** table will need string columns for **username** and **email**, plus each of the details obtained via LDAP (**dn**, **uid**, **mail**, **ou**, **givenname**, and **sn**). If you create an instance of your **User** class with either a **username** or an **email** the remaining details can then be populated via LDAP using the **get_info_from_ldap** method provided by epiCAS:

```
user = User.new(username: 'aca03cem')
user.get_info_from_ldap
user.save
```

In order to create a **User** in your application you could do something like the following:

```
# app/controllers/users_controller.rb

def create
  user = User.new(username: params[:user][:username])
  User.get_info_from_ldap
  if user.save
    # ...
  end
end
```

If you wish to also store the details differently to how they appear in LDAP you can add a **generate_attributes_from_ldap_info** method to your **User** model, and this will be called automatically by **get_info_from_ldap**, for example:

```
# app/models/user.rb

def generate_attributes_from_ldap_info
  self.first_name = self.givenname
  self.last_name  = self.sn
  super           # This needs to be left in so the default fields are also set.
end
```