VITELS - HOWTO

Using the VITELS PHP API

Thomas Jampen
University of Bern
Neubrückstrasse 10
3012 Bern
+41 31 631 86 92
jampen@iam.unibe.ch

1 Using the API for Accessing a Module's Current User

In order to send cookies with slot information to the browser when students visit a module's website, the module's current user and the corresponding slot settings have to be accessible. The file `vitelsldap.inc.php` offers an easy to use interface in order to check whether a user is the current user of a module, whether the supplied password is correct and what the starting time and the ending time of the current slot is.

The needed functionality is provided by the class `VitelsLDAP`. There are several functions that can be called in this API, but the most convenient way to verify a current user and to get the slot boundaries is to use the function `get_slot_if_current_user($dn, $pwd, $mid)`. This function returns an array containing the starting and ending time of the slot only if the user specified by `$dn` is the current user of the module `$mid` and if his password stored in `$pwd` is correct. Otherwise, `false` is returned.

file: **APIexample1.php**

```
<?php

require('vitelsldap.inc.php');

// when composing the dn make sure you know the user's institute!
// choose one of:
// $unibe_base, $unifr_base, $unige_base, $unine_base, $ingfr_base
$dn = "uid=ausername,$unibe_base";

$pwd = "some-password";

// for example module 6
$mid = "6"

// access to the LDAP directory
$ldap = new VitelsLDAP();
$result = $ldap->get_slot_if_current_user($dn, $pwd, $mid);

if ($result) {
  // the date is specified as seconds since 1.1.1970
  $starting_time = $result["starttime"];
  $ending_time   = $result["endtime"];

  // get printable date, e.g. 20020426 1830
  $start = date("Ymd Hi", $starting_time);
  $end   = date("Ymd Hi", $ending_time);

  // ...
}
else {
  print "wrong password or not current user!";
}

?>
```

The above code shows a little example on how to use the class `VitelsLDAP`. The distinguished names for the different educational institutes are saved in the variables `$unibe_base`, `$unifr_base`. `$unige_base`, `$unine_base` and `$ingfr_base`. In order to create a valid dn, the string `uid=ausername` has to be prepended to one of the base dns. After that, an new instance of the class `VitelsLDAP` has to be created in order to access the above described function. The starting and ending times returned within the array are represented as the seconds past since January, 1^{st} 1970. They can be easily converted to the preferred format using the PHP `date($format, $time)` function. The `$format` variable contains a string specifying the desired time format. In the above example `Ymd Hi` is used in order to get a date formatted as follows: `20020426 1830`. Other possible formats are described on the PHP homepage.

A useful function is `authenticate_user($dn, $pwd)` which only returns `true` if the user specified by `$dn` is registered and if the password given by `$pwd` is valid. In order to authenticate a user, the connection to the LDAP server has to be established already. This can be done using the function `connect()` before authentication. After successful authentication, the user has to be authorized for accessing the desired module. In order to do that, the function `authorize_user_for_module($mid)` can be called. This function returns `true` if the authenticated user is, in fact, the current user of the module specified by `$mid`. After the authorization, the LDAP connection can be closed using the function `cleanup()`. The code below shows a little example.

file: **APIexample2.php**

```
<?php

require('vitelsldap.inc.php');

$dn = "uid=ausername,$unibe_base";
$pwd = "some-password";
$mid = "6"

$ldap = new VitelsLDAP();
$ldap->connect();

if ($ldap->authenticate_user($dn, $pwd)) {
  if ($ldap->authorize_user_for_module($mid)) {
        // authorization successful
        // ...
  }
  else {
        print "not current user!";
  }
}
else {
  print "wrong password!";
}

$ldap->cleanup();

?>
```

The second file provided with this little API is `ldaperror.php`. This file contains the HTML code for an error message to display if the LDAP server is not reachable at the moment.