

14.4 Documentación de los tests

Wiki Unit Tests documentation

The purpose of unit testing is to test the individual parts of the program (functions, and methods of classes) to make sure that each individually does the right thing. Once unit testing has given you the confidence that each part works, then you should use other forms of testing to ensure that the different parts work together properly.

The unit testing framework is based on the SimpleTest framework.

Why SimpleTest?

Basically because Moodle has an interface to run this kind of tests and is simple implement them.

Running the wiki unit tests in Moodle

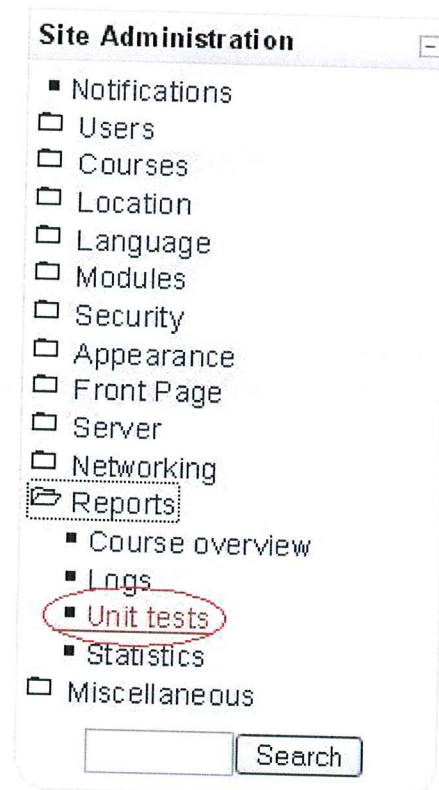
This data is based on manuals from moodle.org.

Running the basic tests

1. Log in with an admin account.
2. Click on the **Reports** link near the bottom of the site administration window.



3. Click on the **Unit Tests** link.
4. Click on the **Run the unit tests** link.



5. Write in the box **Only run tests in the path of the wiki test.**
1.mod/wiki/lib/simpletest/testwiki.php

2. `mod/wiki/db/simpletest/testwiki_persistor.php`

Site Administration

- Notifications
- Users
- Courses
- Location
- Language
- Modules
- Security
- Appearance
- Front Page
- Server
- Networking
- Reports
 - Course overview
 - Logs
 - Unit tests
 - Statistics
- Miscellaneous

Run the unit tests

Show passes as well as fails.

Show the search for test files.

Run a thorough test (may be slow).

Only run tests in:

6. Click on **Run tests**.

Site Administration

- Notifications
- Users
- Courses
- Location
- Language
- Modules
- Security
- Appearance
- Front Page
- Server
- Networking
- Reports
 - Course overview
 - Logs
 - Unit tests
 - Statistics
- Miscellaneous

Run the unit tests

Show passes as well as fails.

Show the search for test files.

Run a thorough test (may be slow).

Only run tests in:

Options for running the tests

At the bottom of the tests page, there is form that lets you adjust the options used when running the tests.

Show passes as well as fails

Normally, only details of the tests that have failed are printed. Turning on this option shows details of all the passes too.

Show the search for test files

The tests to run are found automatically by searching the codebase for files whose names match **test*.php** in directories called **simpletest**. Turning on this option will print a list of the folders searched and the test files found. This

is sometimes useful for debugging.

This option is particularly useful when one of your test files has a syntax error. When this happens, you sometimes just get a blank page with no error message. Turning on the show search option lets you see which test file it was that gave the error. If necessary, you can enable this option manually by adding "showsearch=1" to the end of the URL.

Run a thorough test (may be slow)

If you turn on this option, then as well as looking for files called **test*.php**, the search also looks for files called **slowtest*.php**.

To be useful, the full test run should find most bugs, but not take too long to complete. So if you have very, very detailed tests of an area of the code, it may be better to select a subset for everyday testing, and only use the more detailed tests when a bug is reported, or you are doing new development in that area of the code.

This option is most useful when combined with the next option.

Only run tests in

This is the option that we choose. Normally, tests from all parts of the codebase are run. However, when you are just doing development of one part of the code that is a waste of time. You can type the name of a folder (for example **mod/quiz**) or a particular test file (for example **lib/simpletest/testdatalib.php**) and then only those tests will be run.

```
Pass: mod/wiki/lib/simpletest/testwiki.php > wiki_test_wiki > test_wiki >
[GET] Whether wiki identifiers are equals at [C:\Documents and Settings\User\Mis
documentos\FIB\PFPC\moodle\moodle\st\testwiki.php line 112]
```

Instead of typing a path into this box, there is an easier way. Whenever a pass or fail is displayed, the name of the test file is printed. Each section of the path

name is a link to run only the tests in that folder or file.

Modifying API tests for the Wiki

Where are the tests for the API.

Tests are in two directories:

- **mod/wiki/lib/simpletest**
There is the **testwiki.php** code.
- **mod/wiki/db/simpletest**
There is the **testwiki_persistor.php** code.

How to extend test code

When code is being tested, it gets included from inside one of the simpletest library function. If the code is expected to be run directly (for example a view.php or index.php function), you are likely to get errors because that expectation is no longer true.

To avoid errors it's needed to include these instructions.

```
require_once(dirname(__FILE__) . '/../../../config.php');  
global $CFG; // This global variable is needed  
require_once($CFG->libdir . '/simpletestlib.php');
```

Comments in functions from testwiki.php and testwiki_persistor.php specify how to extend these tests.

Further information

The simpletest documentation is at <http://simpletest.sourceforge.net/>.

