



A Guide To
Drupal SimpleTests

What is unit testing?

- Definition

- “...unit testing is a procedure used to validate that individual units of source code are working properly.” —Wikipedia
- “A unit test is a method of testing the correctness of a particular module of source code.” —JavaWorkshop

Benefits of unit testing

- Ensure that modules are tested
- Provide an easy method for validating a patch
 - Tests can be run on the patch to confirm that everything still works
 - Prevent catchable mistakes from being committed
- Facilitate change
 - By allowing a developer to make sure that changes to core functionality do not have detrimental effects on dependent modules

What is SimpleTest?

- SimpleTest is an open source PHP framework
- The framework provides
 - Abstract class that automatically calls test functions
 - Common testing functions
 - Internal web browser to imitate user requests
 - Feedback system that displays the test results

Drupal SimpleTesting

- Integrated SimpleTest environment
 - SimpleTest module that fits into Drupal system
 - Administration page to run and review test results
 - Convenience functions for common Drupal testing tasks
 - Specialized internal browser

Interface

- Streamlined interface
 - Select tests
 - Run
- Categorized



Blog API Tests

Select all tests in this group
Select all tests in group Blog API Tests

—▶ Tests

Run tests

Run all tests (WARNING, this may take a long time)

Run selected tests

Interface

- Easy to read
 - Green—passed
 - Red—failed

▼ Path alias functionality

82 passes, 0 fails and 14 exceptions.

Path alias functionality: Add, edit, delete, and change alias and verify its consistency in the database.

[module] path already enabled at
[/home/jimmy/public_html/drupal/modules/simpletest/drupal_test_case.php line 188] OK

[role] created name: simpletest_S_nK, id: 628 at
[/home/jimmy/public_html/drupal/modules/simpletest/drupal_test_case.php line 275] OK

[role] created permissions: edit own page content, create page content, administer url aliases,
create url aliases at [/home/jimmy/public_html/drupal/modules/simpletest/drupal_test_case.php
line 279] OK

[user] name: simpletest_UQIE pass: AegzXhciEi created at
OK

Code



```
<?php
// $Id:

class PresentationModuleTestCase extends DrupalTestCase {
  function get_info() {
    return array(
      'name' => t('Presentation abilities'),
      'desc' => t('Does a complete test of presentation abilities'),
      'group' => t('Presentation Tests'),
    );
  }

  function test_presentation() {
    $this->drupalModuleEnable('presentation');
    $admin_user = $this->drupalCreateUserRolePerm(array('administer content types'));

    $this->drupalLoginUser($admin_user);

    $edit = array();
    $edit['name'] = 'John Doe';
    $edit['foo'] = 'bar';
    $this->drupalPostRequest('presentation/test', $edit, 'Save');

    $this->assertText(t('Saved Changes.'));
  }
}
```



Code

```
<?php
// $Id:

class PresentationModuleTestCase extends DrupalTestCase {
  function get_info() {
    return array(
      'name' => t('Presentation abilities'),
      'desc' => t('Does a complete test of presentation abilities'),
      'group' => t('Presentation Tests'),
    );
  }

  function test_presentation() {
    $this->drupalModuleEnable('presentation');
    $admin_user = $this->drupalCreateUserRolePerm(array('administer content types'));

    $this->drupalLoginUser($admin_user);

    $edit = array();
    $edit['name'] = 'John Doe';
    $edit['foo'] = 'bar';
    $this->drupalPostRequest('presentation/test', $edit, 'Save');

    $this->assertText(t('Saved Changes. '));
  }
}
```

Information to be displayed in administration area.

Test function

Create user with set of permissions

Simulate post request using internal browser.

Check for text resulting from post request.

References

- [Wikipedia.org](https://www.wikipedia.org)
- [JavaWorkshop.sourceforge.net](https://www.javaworkshop.org)
- [Drupal.org](https://www.drupal.org)



Author

