

# Writing Unit and Functional Tests

Developers (and contributors) of the platform should use a [test-driven development](#) approach.

The platform's test suite comprises the following:

- [Functional Tests](#) - Using Selenium
- [Unit Tests](#) - Using PHPUnit

Ushahidi platform tests are located in the Ushahidi\_Web/tests directory.

## Functional Tests

Functional tests mainly target the controllers. Their main purpose is to verify that the controllers respond in a certain fashion when supplied with a specific set of parameters via HTTP POST or HTTP GET. The expected behavior could be: displaying a specific message or HTML element, redirecting the user to another page, show/hide a DOM node etc. More information on what Selenium can be used for can be found [here](#).

## Unit Tests

As already mentioned above, the unit testing component of the test suite is based on PHPUnit. Therefore, in order to write and run unit tests, PHPUnit (v3.5.4 and above) be installed and added to the include path of your PHP development environment. Conventionally, PHPUnit is installed via PHP [PEAR](#).

### Installing PHP PEAR

NOTE: If you already have PHP PEAR installed, you may skip this section.

OS-X (Snow Leopard, Lion)

Assumption: PHP is already installed (via Macports, Homebrew etc) and the executable has been added to the system \$PATH

```
curl http://pear.php.net/go-pear.phar > pear.php
sudo php -q pear.php
```

Change the installation base (item 1) to /usr/local. Use the default values for the other options.

In /etc/php.ini change

```
;include_path = "./php/includes"
```

to

```
include_path = "./usr/local/share/pear"
```

Then restart apache.

```
sudo apachectl restart
```

Debian/Ubuntu

```
sudo apt-get install php-pear
```

## CentOS/Fedora

```
yum -y install php-pear
```

## Installing PHPUnit

```
sudo pear channel-discover pear.phpunit.de
sudo pear channel-discover components.ez.no
sudo pear channel-discover pear.symfony-project.com
sudo pear install --alldeps phpunit/PHPUnit
sudo pear install --alldeps phpunit/dbUnit
```

## Running Unit Tests

Unit tests are located in the `Ushahidi_Web/tests/phpunit` directory.

This directory is organized as follows:

```
Ushahidi_Web/
|-- tests
|   |--phpunit/
|       |--bootstrap.php
|       |--classes/
|           |--helpers
|           |--hooks
|           |--libraries
|           |--models
|       |--data
|       |--phpunit.xml.template
|       |--testbootstrap.php
```

To run the current battery of tests bundled with the platform, perform the following:

- Create `phpunit.xml` from `phpunit.xml.template` and save it in `Ushahidi_Web/tests/phpunit`. This XML file specifies the following:
  - The location of the bootstrap file; The bootstrap file is a PHP file that has to be run before the tests; `bootstrap.php` in this case. The bootstrap file ensures that the test cases have access to the Kohana API
  - The location of the test cases to be executed
  - Server environment variables necessary for the test environment
- Ensure that `application/logs` is writable by the PHPUnit executable (`phpunit`) otherwise the tests will fail
- Run the following command from the unit tests directory (`tests/phpunit`)

```
phpunit --stderr
```

More information on the various parameters for the XML configuration file can be found [here](#)

## Writing Unit Tests

Unit tests are mainly target helpers, hooks, libraries and models; pieces of code that are re-used/referenced within the platform.

All unit tests must sub-class `PHPUnit_Framework_TestCase` (in one way or another). Also, the tests do not have a naming convention per se except for the fact that all the testcase files are in upper case and the suffix of the file is always `Test`.