SwiftRiver Test Suite - v1.0

All code contributed to the repository should have corresponding tests. As a matter of fact, This section outlines the tools required to run the test suite, how to install them and a guideline on how to write unit tests for the platform.

The SwiftRiver test suite runs on top PHPUnit which can be installed using the PEAR installer. Depending on your OS distribution and/or your PHP environment, you may need to install PEAR or update your existing PEAR installation.

On this page

Installing PEAR

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

OS-X (Snow Leopard, Lion)

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

Debian/Ubuntu

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

CentOS/Fedora

```
$ sudo yum -y install php-pear
```

Required PHP Tools

Once you have PHP PEAR installed, the following PHP tools should be installed using the PEAR installer:

- PHPUnit 3.7+
- phpunit/DbUnit extension

Setting up Your Test Environment

To install PHPUnit and the required extensions, run the following:

```
$ sudo pear config-set auto_discover 1
$ sudo pear install pear.phpunit.de/PHPUnit
$ sudo pear install phpunit/DbUnit
```

Running the Test Suite

Before running the test suite, you must have performed the following:

- Create a test database and provide the connection details in the `unittest` section of the database configuration file (`application/config/database.php`). The following is an example configuration of this section

```
// Test database
'unittest' => array
(
 'type'       => 'MySQL',
 'connection' => array(
  'hostname'   => 'localhost',
  'database'   => 'swiftweb_tests',
  'username'   => 'swiftwebtests',
  'password'   => 'swiftwebtests',
  'persistent' => FALSE,
 ),
 'table_prefix' => '',
 'charset'      => 'utf8',
 'caching'      => TRUE,
 'profiling'    => TRUE,
)
```

- Have a local installation of SwiftRiver using your test database
- Enable the `kohana/unittest` module in your local SwiftRiver installation. This means that the unittest submodule must also have been cloned in your development environment.

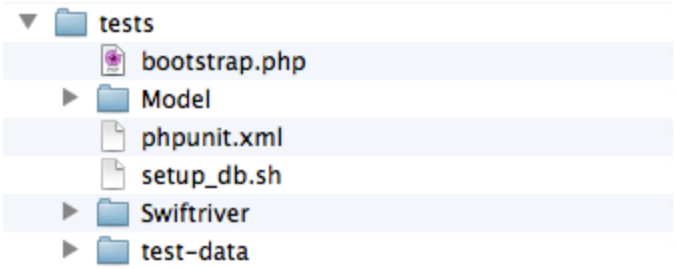
NOTE: As of this writing, the `kohana/unittest` module does not lowercase the PDO driver type

To run the test suite, run the following command:

```
$ phpunit --colors --bootstrap=/path/to/application/tests/bootstrap.php --exclude-group=kohana
/path/to/modules/unittest/tests.php
```

Writing Tests

All unit tests (PHPUnit) are in the `application/tests` directory. The structure of this (tests) directory mirrors that of the `application/classes` directory. See the illustration below:



General Guidelines

- Unless otherwise specified, tests for all other classes (helpers, driver classes etc) must extend the `Unittest_TestCase` class
- The file names for the test classes must be suffixed with `Test`. For example, the file name for the places model test case is `PlaceTest.php`

Database Testing

Database testing is for the model classes i.e. classes located in `application/classes/Model`. The following guidelines apply to unit tests for models:

- All model tests must extend the `Unittest_Database_TestCase` class
- Test datasets for a model's unit test are located in `application/tests/test-data`. SwiftRiver's database tests use Flat XML datasets - an XML format where a tag inside the root node `<dataset>` represents exactly one row in the database. The tags name (and the file name of the Flat XML dataset) corresponds to the table to insert the row into and an attribute represents the (table) column. An example for the `places` table could look like this:

```
<?xml version="1.0" ?>
<dataset>
 <places id="1" hash="867da1cf4e6bd9fc5512a19a90e0141f" place_name="Wales"
place_name_canonical="wales" longitude="146" latitude="-33" />
 <places id="2" hash="5f1823c378ecb68558282f7462e3fd87" place_name="England"
place_name_canonical="england" longitude="-4" latitude="54" />
 <places id="3" hash="c01ceb95cdaf55355a89e7749a3f6c4d" place_name="China"
place_name_canonical="china" longitude="105" latitude="35" />
</dataset>
```