# 3.x Testing

We use PHPUnit for unit tests, and Behat and Mink for functional testing.

The PHPUnit tests don’t have full coverage but should cover most libraries and models. Controllers are not unit tested YET.

Behat tests cover anything that happens on the api end: All API calls, OAuth flow, Login/logout/registration.

The frontend is NOT tested yet, but should be. We’re still figuring out how to handle this.

## When developing new features or new code:

- Tests can be written before or after code.
- Tests MUST exist and pass before code is merged

## Running the tests

We use PHPUnit for unit tests, PHPSpec for spec tests, and Behat and Mink for functional testing.

You can install the Behat, Mink and other required packages using Composer, Just run

```
composer install
```

Composer will drop the the runtimes into `bin/.`

Before running Behat tests, it will be necessary to create a `behat.yml` file:

```
cp application/tests/behat.template application/tests/behat.yml
```

Then edit the `behat.yml` file and ensure that both `base_url` variables matches your installation.

To run the tests themselves

```
# Run ALL behat tests
bin/behat -c application/tests/behat.yml

# Run just posts test
bin/behat -c application/tests/behat.yml features/api.posts.feature

# Run ALL spec tests
bin/phpspec run

# Run just the config entity spec
bin/phpspec run spec/Ushahidi/Entity/ConfigSpec.php

# Run ALL phpunit tests
bin/phpunit -c application/tests/phpunit.xml

# Run just posts model test
bin/phpunit -c application/tests/phpunit.xml classes/models/PostModelTest.php

# You could also run these from application/tests without specifying config files
cd application/tests
../../bin/behat
../../bin/phpunit
```

## Travis-CI

Unit and functional tests are run automatically by Travis-CI.
See .travis.yml for config details.