This testing guide is based on Mocha BDD interface (describe / it).

**Main principles**

**Tests must be always start with a clean state**

This means prefer beforeEach to before. Re-instantiate objects before running each it blocks. Create every files required by a test in a beforeEach (or commit them in fixtures/). Reset any side effets done on the test environment after each test.

**Tests must be runnable in isolation**

Each test must pass if they're ran alone. You can run a single test by using mocha test.js --grep 'test name'.

**Stub most performance heavy operation**

When possible, always stub networks or other long operations.

We use [sinon.js](http://sinonjs.org/) for most stubbing needs.

**Naming convention**

describe blocks should cover three types of information: Object to be tested, method/property, circumstantial group (basically, "when *this*").

it blocks cover assertions. They should use as few lines of code as possible. There should be as many it block as there is assertion on a method effect.

Instances methods and property should be prefixed by a bang sign (#find()). Static methods and property should be prefixed by a dot (.exclude()).

*// Given this object*

function Class() {

this.args = nopt();

};

Class.exclude = function () {};

Class.name = 'Yeoman';

Class.prototype.find = function () {};

*// We'd have this test structure*

describe('Class', function () {

describe('.exclude()', function () {});

describe('.name', function () {});

describe('#find()', function () {});

describe('#args', function () {});

});

Methods should end using parentheses, but should not include expected parameters (parameters should be cover in documentation comments).

it blocks should be declarative.

*// BAD*

it('should find generators');

*// GOOD*

it('find generators');

**Assertion**

Don't add message to assertions unless the error thrown makes it unclear what failed.

If you must add a message, then describe the expected outcome and why it failed. For example:

*// BAD*

assert(Generator.appname, 'Generator have an `appname` property');

*// GOOD*

assert(Generator.appname, 'Expected Generator to have an `appname` property');

Remember that these message are the error message thrown with the failure. Let those be useful in these occasions.