

SASS unit testing with Sassaby



Presented by [@ricardocasares](#)

Previous art

- True
- Node-sass-unit

Drawbacks

- New syntax and functions to learn
- Specific testing library
- Test based on exact output

Writing tests in SCSS syntax?

```
// Test mixin font-size
@import "true";
@include test-module('Utilities') {
  @include test('Font Size [mixin]') {
    @include assert('Has font size & line height.') {
      @include output {
        @include font-size(large);
      }
      @include expect {
        font-size: 2rem;
        line-height: 3rem;
      }
    }
  }
}
```

Hell no! 💩

Why Sassaby?

- Same old JavaScript syntax we all know
- Easy integration with our build system
- Fast, uses `node-sass` for compilation
- Use it with `jasmine`, `mocha`, `you-name-it`
- Use it with any `node.js` assertion library

Hands on

Let's create a `gulp` task for running our tests

- `git clone https://github.com/ricardocasares/sass-unit.git`
- `cd sass-unit`
- `npm install`
- Run `npm test` or `gulp test`

Uh oh! 😬

Seems some tests are not implemented!

...

alert component

 alert mixin

 "â€œ" should have default radius

 – should have custom radius

label component

 – should have a default padding

 – should have a custom padding

3 passing (17ms)

10 pending

Look for the `src/**/*.spec.js` files and complete the tests!

Thank you

You are now a **SASS** test pro.

Not really, but enjoy writing your unit tests! 👍

Created with ❤️ by @ricardocasares