
ccnmtldjango Documentation

Release 2.0.0

Columbia University CTL

Aug 14, 2018

Contents

1	JavaScript Unit Testing	3
1.1	Importing a RequireJS module for testing	3
1.2	Testing an Ajax interaction	4

ccnmtldjango is a Paste Template that we use to replace the standard *django-admin.py startproject* command.

Contents:

CHAPTER 1

JavaScript Unit Testing

ccnmtldjango includes everything necessary to write unit tests for the surrounding Django application. You can run the JavaScript tests like this:

```
make mocha
```

That will run each file in `/media/js/tests/` that ends in `.js`, except for the test runner itself (`test-runner.js`).

Some JavaScript code isn't essential to the functionality of a Django app. But if you have JavaScript that's, for example, making an Ajax POST request to update a user attribute, it's probably essential and should be tested.

This document only describes unit tests in JavaScript: testing the code independent of any surrounding pieces. For more all-encompassing testing, look at [Behave](#), which uses [Selenium](#).

1.1 Importing a RequireJS module for testing

Say you have an `isEven` function in `/media/js/src/utils.js` that returns true if the input is an even number. This is easily unit tested. Here's how to test it from Mocha:

```
// in media/js/tests/utils-test.js

var assert = require('assert');
var requirejs = require('requirejs');

describe('utils', function() {
  // Load the utils module with requirejs
  var utils = requirejs('../src/utils');
  describe('isEven()', function() {
    it('should return true when given even numbers', function() {
      assert.equal(utils.isEven(2), true);
      assert.equal(utils.isEven(10), true);
    });
  });
});
```

(continues on next page)

(continued from previous page)

```
it('should return false when given odd numbers', function() {  
    assert.equal(utils.isEven(1), false);  
    assert.equal(utils.isEven(5), false);  
});  
});  
});
```

1.2 Testing an Ajax interaction

You can test the JavaScript that makes an Ajax request independently of the Django server by using [sinon.js](#).