Testing A Directive: AHK: **[?tad]** (**T**esting An **A**ngular **D**irective)  
http://thecodebarbarian.com/2015/06/12/testing-angularjs-directives

|  |
| --- |
| Directive that displays the number of times its been clicked using an inline template: |
| var app = angular.module('myApp', ['ng']);  app.directive(**'counterDirective'**, function() {  return {  controller: 'MyController',  template: '<div ng-controller="MyController"' +  ' ng-click="counter = counter + 1">' +  'You\'ve clicked this div {{counter}} times' +  '</div>'  }  });  app.controller('MyController', function($scope) {  $scope.counter = 0;  }); |

|  |
| --- |
| Plain old "Mocha" test. Really? Looks like a Jasmine Tests... |
| describe('counterDirective', function() {  var injector, element, scope;  beforeEach(function() {  // Create a new dependency injector using the 'myApp' module  injector = angular.injector(['myApp']);  injector.invoke(function($rootScope, $compile) {  // Get a new scope  scope = $rootScope.$new();  // Compile some HTML that uses the directive  element = $compile('**<counter-directive></counter-directive>**')(scope);  scope.$apply(); **//What is (scope) for at the end??**  });  });  it('increments counter when you click', function() {  assert.equal(element.text(), 'You\'ve clicked this div 0 times');  element.find('div').click();  assert.equal(element.text(), 'You\'ve clicked this div 1 times');  });  }); |
| 1. AngularJS compiles bootstrap compiles the counterDirective HTML All you need to do is create a new dependency injector that uses your module, get the dependency injector's $compile service, and compile some HTML that uses your directive. Once you compile and attach a scope, you get back a nice jQuery-like element that you can manipulate to your heart's content. |