

Unit Testing Directives with \$rootScope



Bradley Braithwaite

@bradoncode | www.bradoncode.com

How do we unit test a Directive?

When should we use `$rootScope` in our
Unit Tests?

`$compile`

Compiles an HTML string or DOM into a template and produces a template function, which can then be used to link the scope and the template together.

Using \$rootScope in Controllers

```
$rootScope.$apply();
```

Using \$rootScope in Controllers

```
$rootScope.$apply();
```

```
$rootScope.$digest();
```

`$rootScope` in AngularJS

Every application has a single root scope. All other scopes are descendant scopes of the root scope.

`$rootScope`
in ngMock

This is a scope type decorated with helper methods useful for testing.

ngMock \$rootScope Helper Methods

```
// counts all the direct and indirect child scopes of the  
current scope.  
$rootScope.$countChildScopes();
```

ngMock \$rootScope Helper Methods

```
// counts all the watchers of direct and indirect child  
scopes of the current scope.  
$rootScope.$countWatchers();
```

Using \$rootScope in Controllers

```
beforeEach( inject( function( $controller, $rootScope ) {  
    var $scope = $rootScope.$new();  
    $controller( 'GreetingController', { $scope: $scope } );  
} ) );
```

Using \$rootScope in Controllers

```
beforeEach( inject( function( $controller ) {  
    var $scope = {};  
    $controller( 'GreetingController', { $scope: $scope } );  
} ) );
```

\$rootScope

\$scope

\$scope

\$scope

\$scope