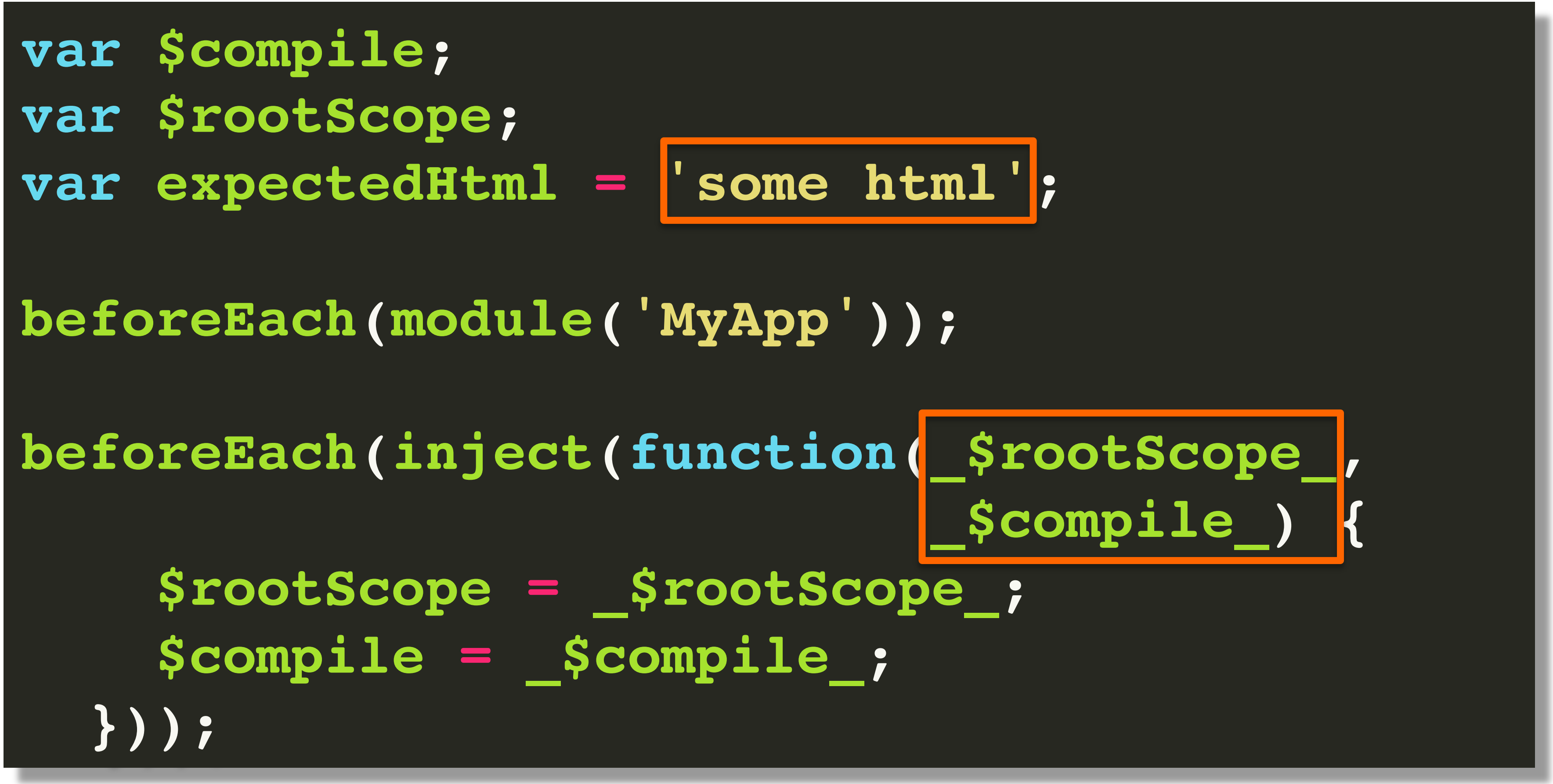
Testing AngularJS Directives

beforeEach Setup

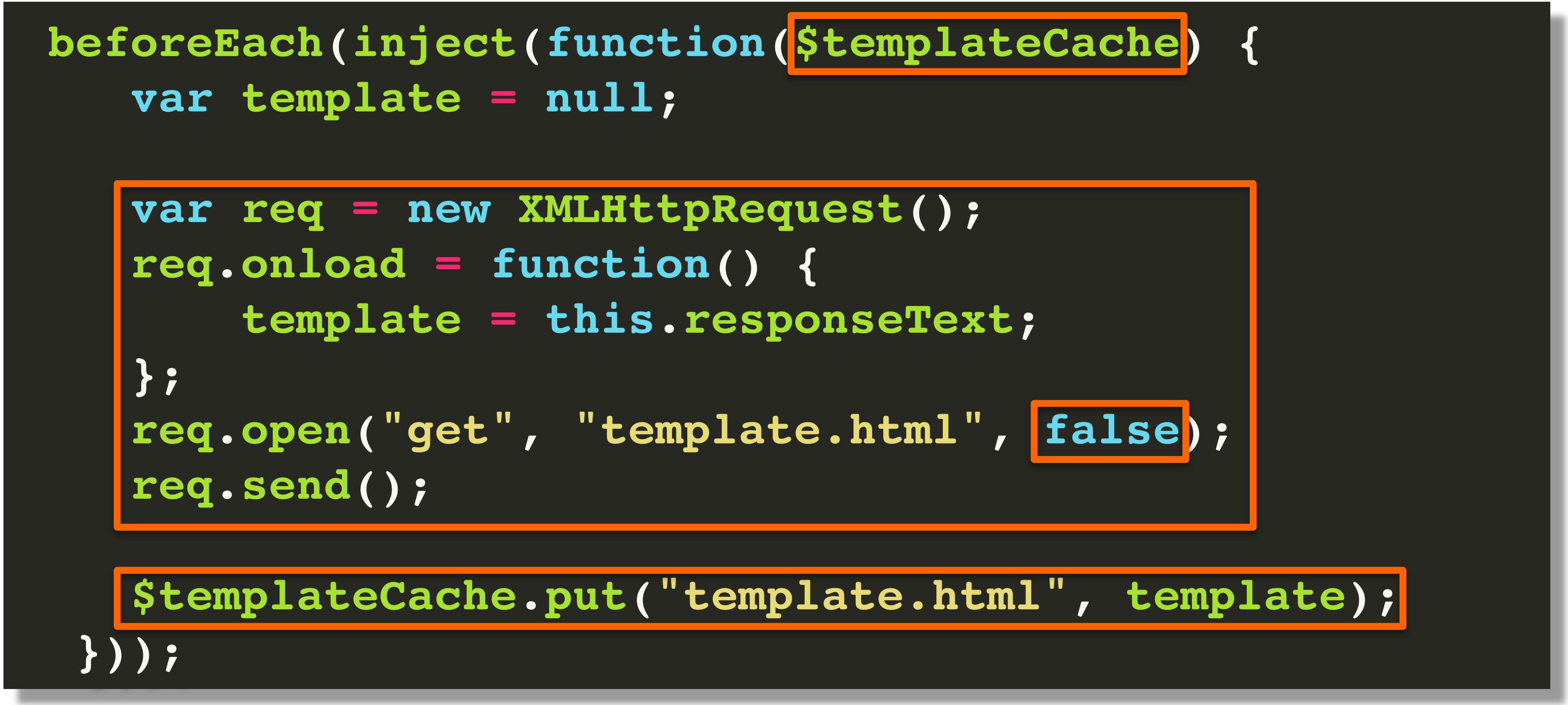


```
var $compile;
var $rootScope;
var expectedHtml = 'some html';

beforeEach(module('MyApp'));

beforeEach(inject(function(_$rootScope_,
                           _$compile_) {
    $rootScope = _$rootScope_;
    $compile = _$compile_;
  }));
```

beforeEach Setup For Using templateUrl

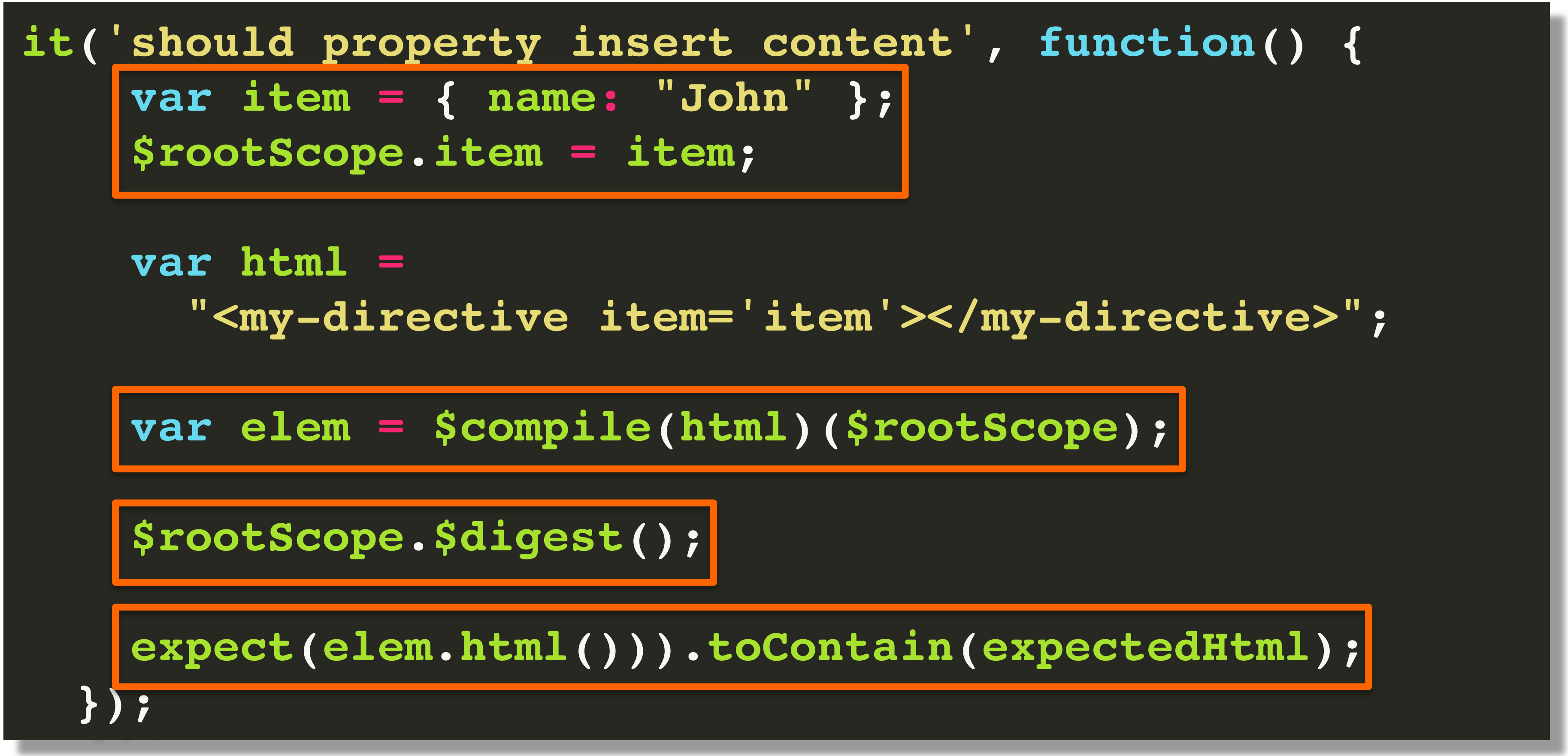


```
beforeEach(inject(function($templateCache) {
   var template = null;

   var req = new XMLHttpRequest();
   req.onload = function() {
       template = this.responseText;
   };
   req.open("get", "template.html", false);
   req.send();

   $templateCache.put("template.html", template);
 }));
```

Test Method



```
it('should property insert content', function() {
    var item = { name: "John" };
    $rootScope.item = item;

    var html =
      "<my-directive item='item'></my-directive>";

    var elem = $compile(html)($rootScope);

    $rootScope.$digest();

    expect(elem.html()).toContain(expectedHtml);
  });
```

Summary

- Use mock inject to inject $rootScope and $compile services in beforeEach
- If directive uses templateUrl, use some method to place the template contents into $templateCache
- In the test method, place whatever props are needed for the directive on the $rootScope
- Use $compile service to compile directive string and then using the returned function to bind with data by passing in $rootScope
- Call $rootScope.digest() to update the compiled HTML
- Retrieve produced HTML with .html() method and compare