Unit Testing Controllers with ngMock

Bradley Braithwaite

@bradoncode | www.bradoncode.com

How do we use Angular’s **$controller** service in our unit tests?

AngularJS Controller Code

```
var myApp = angular.module('myApp',[]);

myApp.controller('GreetingController', function($scope) {
  $scope.greeting = 'Hola!';
});
```

AngularJS Controller Code

```
<div ng-controller="GreetingController">
  {{ greeting }}
</div>
```

Getting a Controller Instance

```
angular.mock.inject(function(_GreetingController_) {

    controllerInstance = _GreetingController_;
});
```

Getting a Controller Instance

```
angular.mock.inject(function(~~_GreetingController_~~) {
    // this won't work!!
    ~~controllerInstance = _GreetingController_;~~
});
```

$controller

A decorator for $controller with an additional bindings parameter.

ngMock $controller Service

```
$controller(constructor, locals, [bindings]);
```

ngMock $controller Service

```
$controller(constructor, locals, [bindings]);


$controller('GreetingController', locals, [bindings]);
```

constructor

This can be either the name (string) of a controller, or a function that creates a new controller.

ngMock $controller Service

```
$controller(constructor, locals, [bindings]);


$controller('GreetingController', locals, [bindings]);

$controller(getInstance(), locals, [bindings]);
```

constructor

This can be either the name (string) of a controller, or a function that creates a new controller.

ngMock $controller Service

```
$controller(constructor, locals, [bindings]);


$controller(constructor, { $scope: myScope }, [bindings]);
```

locals

Accepts an object that maps by key names to the arguments of the controller constructor function e.g. $scope, $http, someService.

AngularJS Controller Code

```
var myApp = angular.module('myApp',[]);


myApp.controller('GreetingController', function($scope) {
  $scope.greeting = 'Hola!';
});
```

AngularJS Controller Code

```
var myApp = angular.module('myApp',[]);


myApp.controller('GreetingController', function($scope) {
  $scope.greeting = 'Hola!';
});
```

AngularJS Controller Code

```
var myApp = angular.module('myApp',[]);


myApp.controller('GreetingController', function($scope) {
  $scope.greeting = 'Hola!';
});
```

ngMock $controller Service

```
$controller(constructor, locals, [bindings]);


$controller(constructor, locals, { name: 'Brad' });
```

bindings

This is optional, accepts an object, where the values of the object e.g. properties or functions, will be bound to the controller's this binding.

AngularJS Controller Code

```
var myApp = angular.module('myApp',[]);

myApp.controller('GreetingController', function() {
  this.greeting = 'Hola!' + this.name;
});
```

ngMock $controller Service

```
$controller(
    constructor,
    locals,
    [bindings]
  );
```

ngMock $controller Service

```
$controller(
    constructor, // string or function
    locals,
    [bindings]
  );
```

ngMock $controller Service

```
$controller(
    constructor, // string or function
    locals,      // $scope object, other controllers args
    [bindings]
  );
```

ngMock $controller Service

```
$controller(
    constructor, // string or function
    locals,      // $scope object, other controllers args
    [bindings]   // object for controllerAs this binding
  );
```