

Startup Engineering

More AngularJS and JS Testing

Sung Kim

AngularJS



O'REILLY®

Brad Green & Shyam Seshadri

Bookmark homework revisit

- RestAPI

◀ ▶ ⌂ jdkim.ddns.net/thumbnail.php?url=www

```
{ "thumbnail" : "http://jdkim.ddns.net-thumbnails/247b520f1fdcc871acfe9e02ea88d960.jpg" }
```

- Our logic

```
$scope.getImgAndAdd = function () {
  var imgURL = null;

  // 1. get img set the img URL
  $http.get('http://jdkim.ddns.net/thumbnail.php',
    {params: {url: url}}).
    success(function(data, status, headers, config) {
      imgURL = data.thumbnail;
    }).error(function(data, status, headers, config) {});

  // 2. add to the firebse
  $scope.urlArr.$add({
    url: $scope.url,
    img: imgURL
  });
}
```

Why this does not work?

```
$scope.getImgAndAdd = function () {
  var imgURL = null;

  // 1. get img set the img URL
  $http.get('http://jdkim.ddns.net/thumbail.php',
  {params: {url: url}}).
  success(function(data, status, headers, config) {
    imgURL = data.thumbnail;
  }).error(function(data, status, headers, config) {});

  // 2. add to the firebse
  $scope.urlArr.$add({
    url: $scope.url,
    img: imgURL
  });
}
```

asynchronization: promise

- `$promise = doJobToTakeLongTime();`
- `$promise.then(function() {...});`
- `...`

logic2: synchronization (wait!)

```
$scope.getImgAndWaitAdd = function () {
  var imgURL = null;

  // 1. get img set the img URL
  $p = $http.get('http://jdkim.ddns.net/thumbail.php',
    {params: {url: url}});

  $p.success(function(data, status, headers, config) {
    imgURL = data.thumbnail;

    // 2. wait to get the resut and add to the firebse
    $scope.urlArr.$add({
      url: $scope.url,
      img: imgURL
    });
  }).error(function(data, status, headers, config) {});
}
```

Why this is bad (for users)?

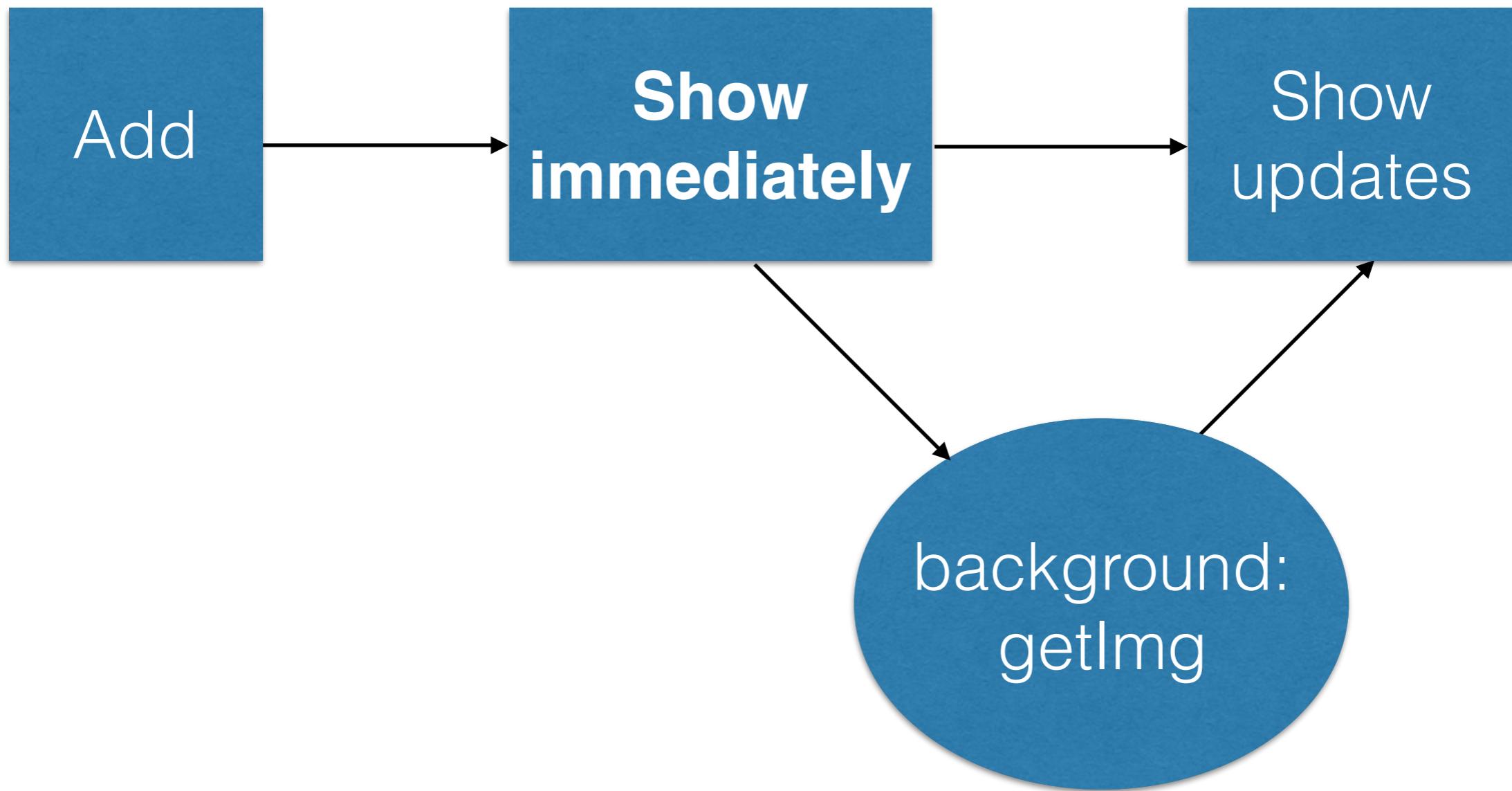
```
$scope.getImgAndWaitAdd = function () {
  var imgURL = null;

  // 1. get img set the img URL
  $p = $http.get('http://jdkim.ddns.net/thumbail.php',
    {params: {url: url}});

  $p.success(function(data, status, headers, config) {
    imgURL = data.thumbnail;

    // 2. wait to get the resut and add to the firebse
    $scope.urlArr.$add({
      url: $scope.url,
      img: imgURL
    });
  }).error(function(data, status, headers, config) {});
}
```

Add something first and update later?



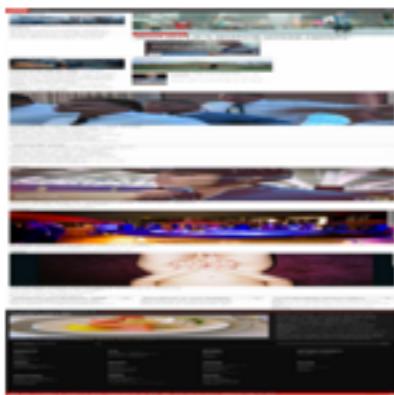
How about add and update?

```
$scope.addAndUpdate = function() {
  var $p = $scope.urlArr.$add({
    img: 'img/loading.gif', url: $scope.url,
  }).then(function(ref) {
    $scope.updateImg(ref, $scope.url);
    $scope.url = "";
  });
  return $p;
};

$scope.updateImg = function (ref, url) {
  $http.get('http://jdkim.ddns.net/thumbnail.php',
  {params: {url: url}}).
  success(function(data, status, headers, config) {
    console.log(data);

    // update firebase
    if (ref!=null) {
      ref.update({ img: data.thumbnail});
    }
  }).
  error(function(data, status, headers, config) {});
};
```

Loading first and img up later!



1. cnn.com [X]



2. cse.ust.hk [X]



3. mit.edu [X]

How about the title?

```
$scope.addAndUpdate = function() {
  var $p = $scope.urlArr.$add({
    img: 'img/loading.gif',
    title: "Reading...",
    url: $scope.url,
  }).then(function(ref) {
    $scope.updateImg(ref, $scope.url);
    $scope.updateTitle(ref, $scope.url);

    $scope.url = "";
  });
  return $p;
};
```

AngularJS Features

- Client template
- Date binding
- Dependency Injection
- Directives

Which feature do you like? Why?

- Client template
- Date binding
- Dependency Injection
- Directives

AngularJS Native Directives

- ng-app,
- ng-click, ng-change, ng-dblclick, ng-submit
- ng-hide, ng-show, ng-disabled
- ng-href, ng-src
- ng-repeat
- ng-model
- ng-style, ng-class

A loooooong controller!

```
9  var app = angular.module("Bookmark", ['firebase']);
10 app.controller("Ctrl", function ($scope, $firebaseArray, $http) {
11   $scope.url;
12
13   var firebaseURL = "https://sbookmark.firebaseio.com/";
14
15   $scope.getList = function() {
16     var echoRef = new Firebase(firebaseURL);
17     var query = echoRef.orderByChild("url");
18     $scope.urlArr = $firebaseArray(query);
19   };
20
21   $scope.add = function() {
22     $scope.urlArr.$add({
23       url: $scope.url,
24       title: "TBA"
25     });
26   };
27
28   $scope.remove = function (url) {
29     $scope.urlArr.$remove(url);
30   };
31
32
33   $scope.FBLogin = function () {
34     var ref = new Firebase(firebaseURL);
35     ref.authWithOAuthPopup("facebook", function(error, authData) {
36       if (error) {
37         console.log("Login Failed!", error);
38       } else {
39         $scope.$apply(function() {
40           $scope.$authData = authData;
41         });
42         console.log("Authenticated successfully with payload:", authData);
43
44         // do something with the login info
45       }
46     });
47   };
48
49   $scope.FBLogout = function () {
50     var ref = new Firebase(firebaseURL);
51     ref.unauth();
52     delete $scope.$authData;
53
54     // do something after logout
55   };

```

More organized app.js

- multiple controllers
- use filters/create filters
- my own directives
- factory/service/provider

Multiple Controllers - script.js

```
1 var firstController = function ($scope)
2 {
3     // Initialize the model variables
4     $scope.firstName = "John";
5     $scope.lastName = "Doe";
6
7     // Define utility functions
8     $scope.getFullName = function ()
9     {
10         return $scope.firstName + " " + $scope.lastName;
11     };
12 };
13
14 var secondController = function ($scope)
15 {
16     // Initialize the model variables
17     $scope.firstName = "Bob";
18     $scope.middleName = "Al";
19     $scope.lastName = "Smith";
20
21     // Define utility functions
22     $scope.getFullName = function ()
23     {
24         return $scope.firstName + " " + $scope.middleName + " " + $scope.lastName;
25     };
26 };
```

Multiple Controllers - index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <script src="angular.js"></script>
5   <script src="script.js"></script>
6 </head>
7
8 <body ng-app>
9   <div ng-controller="firstController">
10    <h2>Model managed by the first controller</h2>
11    <strong>First name:</strong> {{firstName}}<br />
12    <strong>Last name:</strong> <span ng-bind="lastName"></span><br />
13    <strong>Full name:</strong> {{getFullName()}}<br />
14    <br />
15    <label>Set the first name: <input type="text" ng-model="firstName"/></label><br />
16    <label>Set the last name: <input type="text" ng-model="lastName"/></label>
17 </div>
18 <div ng-controller="secondController">
19    <h2>Model managed by the second controller</h2>
20    <strong>First name:</strong> {{firstName}}<br />
21    <strong>Middle name:</strong> {{middleName}}<br />
22    <strong>Last name:</strong> <span ng-bind="lastName"></span><br />
23    <strong>Full name:</strong> {{getFullName()}}<br />
24    <br />
25    <label>Set the first name: <input type="text" ng-model="firstName"/></label><br />
26    <label>Set the middle name: <input type="text" ng-model="middleName"/></label><br />
27    <label>Set the last name: <input type="text" ng-model="lastName"/></label>
28 </div>
29 </body>
```

Filters

```
{{ expression | filterName : parameter1 : ...parameterN }}
```

```
{{12.9 | currency}}
```

This bit of code will display the following:

\$12.90

```
{{12.9 | currency | number:0 }}
```

displays:

\$13

Filters |

- **currency:** Format a number to a currency format.
- **date:** Format a date to a specified format.
- **filter:** Select a subset of items from an array.
- **json:** Format an object to a JSON string.

Filters 2

- **limit:** To Limits an array/string, into a specified number of elements/characters.
- **lowercase:** Format a string to lower case.
- **number:** Format a number to a string.
- **orderBy:** Orders an array by an expression.
- **uppercase:** Format a string to upper case.

Filters - OrderBy

Example

The `orderBy` filter sorts an array:

```
<div ng-app="myApp" ng-controller="namesCtrl">

<ul>
  <li ng-repeat="x in names | orderBy:'country'">
    {{ x.name + ', ' + x.country }}
  </li>
</ul>

</div>
```

uppercase/lowercase

```
<div ng-app="myApp" ng-controller="personCtrl">  
  
<p>The name is {{ lastName | uppercase }}</p>  
  
</div>
```

```
<div ng-app="myApp" ng-controller="personCtrl">  
  
<p>The name is {{ lastName | lowercase }}</p>  
  
</div>
```

Filters on Array

- Simple search with ‘filter’ filter.
- Filter on array
- {{arr | filter:key}}
 - return array elements only match to the key

```
<li ng-repeat="u in urlArr|filter:url"> ·
```

Search on array

```
<li ng-repeat="u in urlArr|filter:url"> <
```

mit Submit



1. MIT Compu... csail.mit.edu [\[X\]](#)



2. MIT - Mas... mit.edu [\[X\]](#)

Creating my own filters

```
var homeModule = angular.module('HomeModule', []);
homeModule.filter('titleCase', function() {
  var titleCaseFilter = function(input) {
    var words = input.split(' ');
    for (var i = 0; i < words.length; i++) {
      words[i] = words[i].charAt(0).toUpperCase() + words[i].slice(1);
    }
    return words.join(' ');
  };
  return titleCaseFilter;
});
```

Bookmark title? too long??



2.  CNN - Breaking News, U.S., World, Weather, Entertainment & Video News cnn.com [X]



3.  MIT Computer Science and Artificial Intelligence Laboratory | MIT CSAIL csail.mit.edu [X]



‘shorttitle’ filter

```
app.filter('shorttitle', function() {
  return function(t) {
    var n = 10;
    return (t.length > n) ? t.substr(0,n-1)+'...' : t;
  };
});
```

```
<ol>
  <li ng-repeat="u in urlArr"> <img ng-
    | {{u.title|shorttitle}} {{u.url}} <
  </ol>
  <
```

‘shorttitle’ filter

- A screenshot of the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL) website. The header features the CSAIL logo and navigation links for Home, About, News, Research, People, Events, and Contact. A large banner image shows a night view of a city skyline with a bridge. Below the banner, a section titled "CSAIL REACHES 1 MILLION DOWNLOADS OF RESEARCH SOFTWARE & TOOLS" is displayed. The main content area contains several sections: "RECENT PAPERS", "RECENT PRESENTATIONS", "RECENT SOFTWARE RELEASES", and "RECENT NEWS". Each section includes a thumbnail image, a title, and a brief description. At the bottom of the page, there is a footer with links to various MIT websites and social media icons.



4. Google google.com [X]



The screenshot shows the MIT homepage. The header includes the MIT logo, the name "Massachusetts Institute of Technology", the date "Wednesday, January 11, 2012", and the time "11:00 AM EST". A search bar is at the top right. The main content features a large portrait of a Black man wearing glasses and a white shirt. To the left is a sidebar with links like "about", "admissions", "current students", "alumni", "research", "news", "community", "shop MIT", "admissions media", "donations", and "report". The right sidebar contains sections for "self-study programs", "graduation requirements", "class components", "Academic Year", "completing a degree", "MIT User Services", "new undergraduate majors", "support", "info", and "events". At the bottom, there's a footer with links for "MIT News", "Contact MIT", "Visit MIT", "Give to MIT", "Employment", "Alumni", "Facebook", "Twitter", "YouTube", "RSS", and "Search".

5. MIT - Mas... mit.edu [X]

More organized app.js

- multiple controllers
- use filters/create filters
- my own directives
- factory/service/provider

AngularJS Native Directives

- `ng-app`,
- `ng-click`, `ng-change`, `ng-dblclick`, `ng-submit`
- `ng-hide`, `ng-show`, `ng-disabled`
- `ng-href`, `ng-src`
- `ng-repeat`
- `ng-model`
- `ng-style`, `ng-class`

Creating my own Directives

```
app.directive('urlLink', function () {
  return {
    restrict: 'EA',
    template: '<A href="{{u.url}}>Click {{u.url}}</A>'
  };
});
```

```
    {{u.title|shorttitle}} <url-link/>
```



2.  CNN - Bre... [Click cnn.com](#)



3.  MIT Compu... [Click csail.mit.edu](#)

More organized app.js

- multiple controllers
- use filters/create filters
- my own directives
- **factory/service/provider**

factory

```
app.factory("$bookmarkList", ["$firebaseArray",
  function($firebaseArray) {
    var firebaseURL = "https://sbookmark.firebaseio.com/101";
    var ref = new Firebase(firebaseURL);
    var query = ref.orderByChild("url");
    return $firebaseArray(query);
  }
]);
```

```
app.controller("Ctrl", function ($scope, $firebaseArray, $http, $bookmarkList) {

  $scope.urlArr = $bookmarkList;

  /*
  var firebaseURL = "https://sbookmark.firebaseio.com/101";
  $scope.getList = function() {
    var echoRef = new Firebase(firebaseURL);
    var query = echoRef.orderByChild("url");
    $scope.urlArr = $firebaseArray(query);
  }; */
```

factories vs services vs providers



From the AngularJS mailing list I got [an amazing thread](#) that explains service vs factory vs provider and their injection usage. Compiling the answers:

2022



Services



Syntax: `module.service('serviceName', function);`

Result: When declaring serviceName as an injectable argument **you will be provided with an instance of the function. In other words** `new FunctionYouPassedToService()`.

Factories

Syntax: `module.factory('factoryName', function);`

Result: When declaring factoryName as an injectable argument you will be provided with **the value that is returned by invoking the function reference passed to module.factory.**

Providers

Syntax: `module.provider('providerName', function);`

Result: When declaring providerName as an injectable argument **you will be provided with** `(new ProviderFunction()).$get()`. The constructor function is instantiated before the \$get method is called - `ProviderFunction` is the function reference passed to module.provider.

Structure for the angular-phonecat app

- app
 - index.html
 - css
 - img
 - lib: angular.js, firebase.js, etc.
 - js
 - app.js, controllers.js, directives.js, filters.js
- test
 - e2e
 - unit

More organized app.js

- multiple controllers
- use filters/create filters
- my own directives
- factory/service/provider

AngularJS Testing!

- Code and fix?
- What testing?
 - We test (run browser) and fix!
- Unit (spec) testing
- e2e testing

“CODE NOT TESTED IS
BROKEN BY DESIGN”

JACOB KAPLAN-MOSS

Karma and Jasmine

- **Karma** is a JavaScript command line tool
 - spawn a web server which loads your application's source code
 - executes your tests
- **Jasmine** is a behavior-driven development framework for testing JavaScript code
 - the most popular choice for testing Angular applications.

<https://docs.angularjs.org/guide/unit-testing>

Installation (onetime hack)

- download all js (angularjs, angular-mock.js, firebase, etc.) in app/lib
- install karma and jasmine
 - npm install karma --save-dev
 - npm install karma-jasmine —save-dev
 - sudo npm install -g karma-cli
- mkdir -p tests/unit-tests; cd tests
- karma init karm.conf.js
- karma start karma.conf.js

karma.conf

```
// list of files / patterns to load in the browser
files: [
  'app/lib/angular.js',
  'app/lib/angular-mocks.js',

  'app/lib.firebaseio.js',
  'app/lib/angularfire.min.js',

  'app/js/*.js',
  'test/*.js'
],
```

Or using npm

- `git clone https://github.com/hunkim/bookmark.git`
- `cd bookmark/`
- `sudo apt-get install npm`
- `sudo ln -sf /usr/bin/nodejs /usr/bin/node`
- `sudo apt-get install libfontconfig (#for ubuntu)`
- **npm install**
- **npm test**

Rhythm of testing (AAA)

- Arrange
- Action
- Assertion



Jasmine

- Suites: **describe** your tests.
- Spec (unit): **it** is ...

```
describe("A suite", function() {  
  it("contains spec with an expectation", function() {  
    expect(true).toBe(true);  
  });  
});
```

Jasmine: It's Just Function

```
describe("A suite is just a function", function() {  
  var a;  
  
  it("and so is a spec", function() {  
    a = true;  
  
    expect(a).toBe(true);  
  });  
});
```

Jasmine: Expectations

```
it("and has a positive case", function() {  
  expect(true).toBe(true);  
});
```

```
it("and can have a negative case", function() {  
  expect(false).not.toBe(true);  
});
```

Jasmine: Expectations

- `expect(foo).toEqual(bar)`
- `expect(foo).not.toEqual(bar)`
- `expect(foo).toMatch(/bar/)`
- `expect(foo).toBe(bar)`
- ...

jasmine matchers

to(Not)Be(*null | true | false*)

to(Not)Equal(*value*)

to(Not)Match(*regex | string*)

toBeDefined()

toBeUndefined()

toBeNull()

toBeTruthy()

toBeFalsy()

to(Not)Contain(*string*)

toBeLessThan(*number*)

toBeGreaterThanOrEqual(*number*)

toBeNaN()

toBeCloseTo(*number, precision*)

toThrow()

Matchers can be chained with a **not** between
the actual and assertion.

Jasmine: Setup & Teardown

```
describe("A spec using beforeEach and afterEach", function() {  
  var foo = 0;  
  
  beforeEach(function() {  
    foo += 1;  
  });  
  
  afterEach(function() {  
    foo = 0;  
  });  
  
  it("is just a function, so it can contain any code", function() {  
    expect(foo).toEqual(1);  
  });  
  
  it("can have more than one expectation", function() {  
    expect(foo).toEqual(1);  
    expect(true).toEqual(true);  
  });  
});
```

AngularJS unit testing

- "Did I think about the logic correctly?"
- "Does the sort function order the list in the right order?"

Dependency Injection

- Angular comes with dependency injection built-in
- makes testing components much easier
 - You can pass in a component's dependencies and stub or mock.
 - add/include angular-mocks.js (in local and karma.conf)

Testing controllers

```
'use strict';
describe('PasswordController', function() {
  var $controller, $firebaseArray, $httpBackend, $http, $scope;

  // Injecting variables
  // http://stackoverflow.com/questions/13664144/how-to-unit-test-angularjs-controller-with-location
  beforeEach(function() {
    module('Bookmark');

    inject(function(_$rootScope_, _$http_, _$httpBackend_, _$controller_, _$firebaseArray_) {
      // The injector unwraps the underscores (_) from around the parameter names when matching
      $controller = _$controller_;
      $firebaseArray = _$firebaseArray_;
      $httpBackend = _$httpBackend_;
      $http = _$http_;

      // need to get a new scope
      $scope = _$rootScope_.$$new();
    });
  });

  describe('$scope.add', function() {
    it('add testing', function() {
      var $scope = {};
      var controller = $controller('Ctrl', { $scope: $scope });
      expect($scope.add(1,1)).toEqual(2);
    });
  });
});
```



mock injection!

Testing filters

```
// http://stackoverflow.com/questions/21094569/how-to-unit-test-a-filter-in-angularjs
'use strict';
describe('The test filter', function () {
  var $filter;

  beforeEach(function () {
    module('Bookmark');

    inject(function (_$filter_) {
      $filter = _$filter_;
    });
  });

  it('should shorten a long title', function () {
    // Arrange.
    var title = 'hello world';

    // Act.
    var result = $filter('shorttitle')(title, 'shorttitle');

    // Assert.
    expect(result).toEqual('hello wor...');
  });
});
```

Testing directives

```
// http://www.sitepoint.com/angular-testing-tips-testing-directives/
describe('The test directives', function () {
  'use strict';
  var compile, scope, directiveElem;

  beforeEach(function(){
    module('Bookmark');
    inject(function($compile, $rootScope){
      compile = $compile;
      scope = $rootScope.$new();
    });
  });

  it('should shorten a long title', function () {
    // Arrange.
    var url = 'http://www';
    scope.u = {url:url};

    // Act.
    var element = compile("<url-link></url-link>")(scope);
    scope.$digest();

    // Assert.
    expect(element.html()).toContain('<a href="' + url);
  });
});
```

Testing factories

```
//http://stackoverflow.com/questions/16565531/unit-testing-angularjs
describe('app: myApp', function() {
  beforeEach(module('Bookmark'));
  var $controller;
  beforeEach(inject(function(_$controller_) {
    $controller = _$controller_;
  }));
  // Factory of interest is called MyFactory
  describe('factory: $bookmarkList', function() {
    var factory = null;
    beforeEach(inject(function($bookmarkList) {
      factory = $bookmarkList;
    }));
    it('Should define methods', function() {
      expect(factory).toBeDefined();
      expect(factory.length).toBe(10);
    });
  });
});
```

Rhythm of testing (AAA)

- Arrange
- Action
- Assertion

Continuous Testing

- **karma start**
- **karma.conf.js**
 - singleRun: false,

Summary

- Organizing Angular App
 - multiple controllers
 - use filters/create filters
 - my own directives
 - factory
- Add AngularJS Testing (AAA)

Todo

- Fork and read examples in github
 - <https://github.com/hunkim/bookmark>
 - setup and run tests
- Use filters, directives for your project
- Setup and Add **Jasmine tests** in your project
 - run karma before launching your browser
 - Setup may takes sometime/effort
 - But you will be very productive

Project presentation

- Show
 - specification: trello, git issues
 - code: pull requests (in PM repos)
 - test results and CI results (tomorrow's lecture)
- Demonstrate
 - your web app and android

Bookmark mobile trello demo

The screenshot shows a Trello board titled "Bookmark Mobile". The board has four main columns: "Main Features", "Error handling", "Design", and "Backend".

- Main Features:**
 - As a <> I want to <> so that <>
 - As a user I want to ADD bookmark so that I can have list
 - Show visited count
 - List book mark
 - Automatic thumbnail for better UI
 - Automatic title for better UX (No need to type title)
 - List order: url/title/added date/visited count
 - Remove bookmark
- Error handling:**
 - URL validation 1: syntactic during input
 - URL validation 2: when get title/img, get HTTPD code
 - RestAPI connection error handling
 - Firebase connection error handling (Firebase can handle itself?)
- Design:**
 - Mobile (IOS/Android) using ionic.
 - imglist
 - swipe to delete
 - scroll-down to get more
- Backend:**
 - get title: url in, code, title out.
 - get img: url in, http code, img url out
 - Add a card...

At the bottom of the board, there is a button labeled "Add a card...". The top navigation bar includes "Boards", a search icon, a "New stuff!" button, the "Trello" logo, and a user profile icon.

github issues/commit

hunkim / bookmark-mobile

Unwatch 1 Star 0 Fork 0

Code Issues 4 Pull requests 0 Wiki Pulse Graphs Settings

Filters is:issue is:open Labels Milestones New issue

4 Open 0 Closed Author Labels Milestones Assignee Sort

- httpBacked mocking** #4 opened 2 days ago by hunkim 0
- Uncaught ReferenceError: require is not defined at /Users/sunghunkim/gitworkspace/bookmark-mobile/www/lib/angularfire/index.js:1** #3 opened 3 days ago by hunkim 0
- Add test framework using karma** #2 opened 3 days ago by hunkim 4
- Quick prototype** #1 opened 3 days ago by hunkim 0

hunkim / bookmark-mobile

Unwatch 1 Star 0 Fork 0

Code Issues 4 Pull requests 0 Wiki Pulse Graphs Settings

Initial prototype commit/import. Issue #1 Browse files

master

Your Name committed 3 days ago 0 parents commit 6ecb8884ac1413cde906e3e3f31551b33be7ad7e

Showing 123 changed files with 150,743 additions and 0 deletions. Unified Split

Project presentation

- Show
 - specification: trello, git issues
 - code: pull requests (in PM repos)
 - test results and CI results (tomorrow's lecture)
- Demonstrate
 - your web app and android