UI Architecture Developer Tool Kit- Whitepaper

(Reusable component for all the projects)

- Author: Varadarajan, Baskaran

VERSION HISTORY

Date	Version	Author	Description
25/9/2015	0.1	Varadarajan, Baskaran baskaran.varadarajan@accenture.com	First draft
8/10/2015	0.2	Soundararajan, D. A. d.a.soundararajan@accenture.com Joseph, Jerish jerish.joseph@accenture.com	Updated Framework and reviewed
24/1/2016	1.1	Varadarajan, Baskaran baskaran.varadarajan@accenture.com	Updated review and added Yslow reported
16/2/2016	2.0	Manoharan, Ramasamy ramasamy.manoharan@accenture.com	Final version

Contents

1. ABSTRACT UI ARCHITECTURE DEVELOPER TOOL KIT 4

1.1 Abstract: 4

2. UI ARCHITECTURE DEVELOPER TOOL KIT 5

2.1 SONAR: 5

2.1.1 Develoment Tool List for Angular JS UI framework 6

2.1.2 Install and Configure sonarqube 7

2.1.3 Install and Configure sonar-runner for angularjs 9

2.1.4 Starting sonar server with client ui project 10

2.2 Karma jasmine unit test case for Angular JS 15

2.2.1 Karma, jasmine and yeoman 16

2.2.2 Folder structure 16

2.2.3 Configuration files 17

2.2.4 Dependency files and bower components 18

2.2.5 writing test file using jasmine 20

2.2.6 running test file using karma 22

2.2.7 R2 code base configuration 25

2.3 HTML Validation for UI 26

2.3.1 Sample configuration for HTML Validation: 27

2.3.2 Sample HTML Validation report: 27

2.4 CSS Validation for UI 27

2.4.1 Sample configuration for CSS validation 28

2.4.2 Sample CSS Validation Report 28

2.5 YSlow report 29

2.5.1 Sample performance report of YSlow application 29

3. REFERENCES LINK IN KX SITE 30

3.1 KX site links: 30

1. ABSTRACT UI ARCHITECTURE DEVELOPER TOOL KIT

1.1 ABSTRACT:

UI ARCHITECTURE DEVELOPER TOOL KIT: For developers using AngularJS, Angular Material is both a UI Component framework and a reference implementation of Google's Material Design Specification. This project provides a set of reusable, well-tested, and accessible UI components based on Material Design.

Angular UI **Bootstrap** is built on top of the front-end framework called **Bootstrap**. The framework contains a set of native AngularJS directives based on **Bootstrap HTML** and CSS components.

Now that you are well-versed in the basics, it is time to get started on building your own web application with AngularJS. AngularJS made building a JavaScript-based app more intuitive using what's called directives, which works hand-in-hand with your HTML mark-ups.

if building a web application from the ground seems overwhelming to you, not to worry. Some very generous developers have adapted a few frontend frameworks to support AngularJS. Like a typical framework, they come with pre-built web components. These make using the framework the perfect tool for anyone who needs to get a web application up and running quick.

- Sonar:
- Karam Jasmine unite test for angular js
- HTMl Validation for UI
- CSS Validation for UI
- Y-slow report
- Page load Performances

2. UI ARCHITECTURE DEVELOPER TOOL KIT

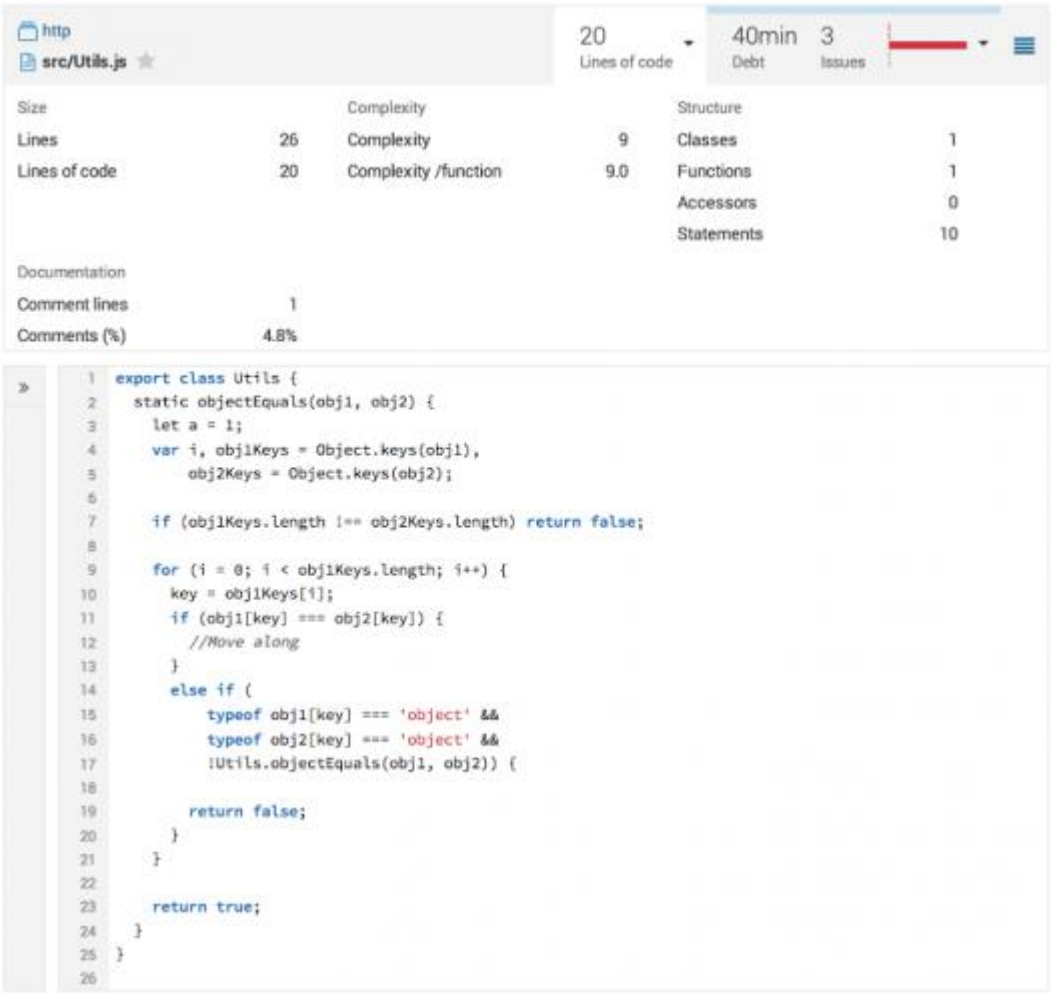
2.1 SONAR:

A new version of the standard also means that each browser needs to provide support for it, at least the major ones. It might take years before it happens, but you don't have to wait for that to take advantage of the innovations in ECMAScript 6!
Thanks to the availability of ES6-to-ES5 transpilers, it is possible to use ES6 features today. A transpiler will translate your ECMAScript 6 code to ECMAScript 5 code so it can be run by today's browsers.

The SonarQube JavaScript Plugin 2.1 fully supports ES6. What does that mean?

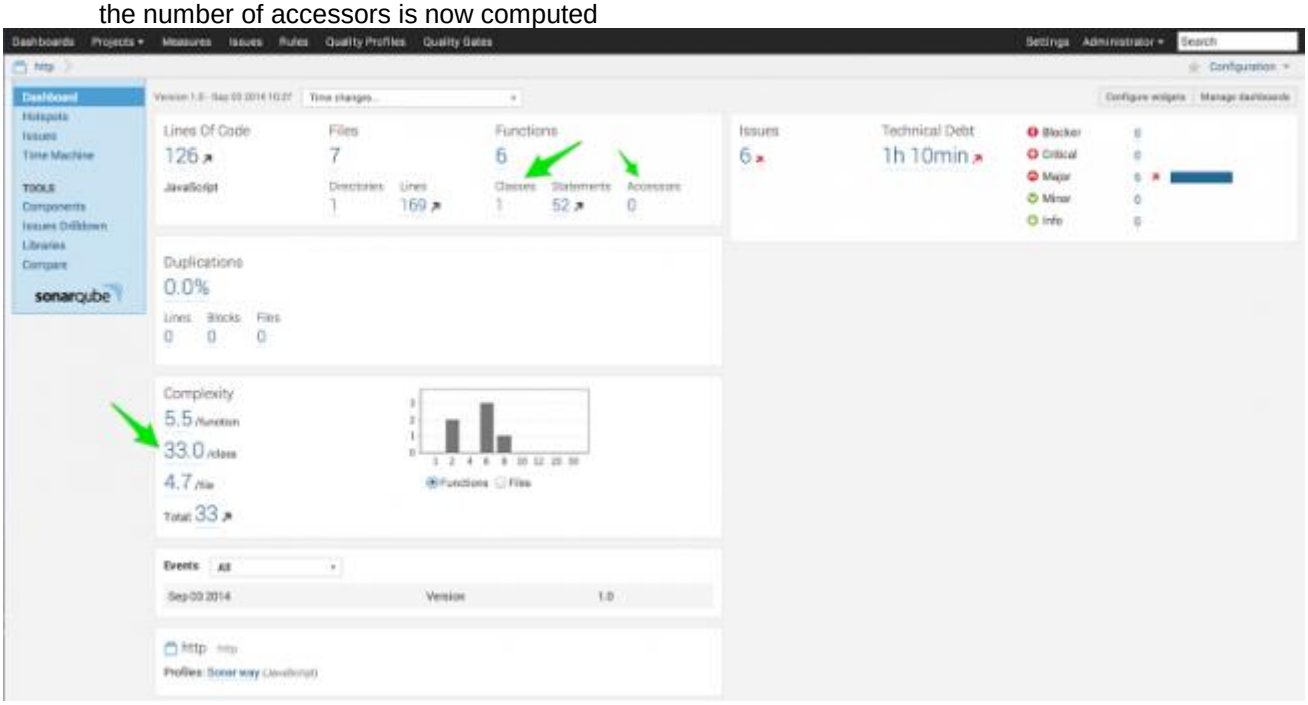
It means that the plugin is able to:

1. parse ES6 source code,



2. Compute all relevant metrics accordingly:
   a classes count is computed when classes are used

class complexity is computed when classes are used
the function count includes generator functions
general complexity metrics take generators into account
the number of accessors is now computed



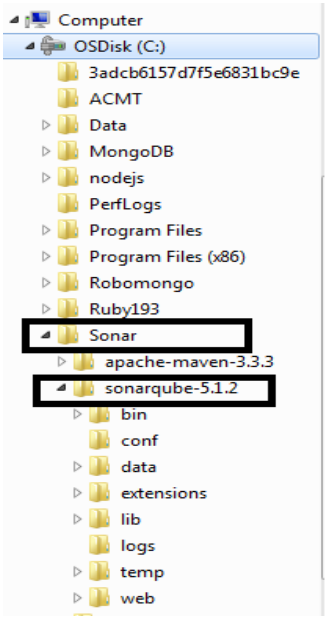
3.Analyse code against rules, all existing coding rules have been updated to cover the new features, e.g: “unused variable” will detect unused variables & constants declared with let and const.

2.1.1 Develoment Tool List for Angular JS UI framework

Tool Name	Version	Download Url
sonarQube	5.1.2	https://sonarsource.bintray.com/Distribution/sonarqube/sonarqube-5.1.2.zip
sonar-runner	2.4	http://repo1.maven.org/maven2/org/codehaus/sonar/runner/sonar-runner-dist/2.4/sonar-runner-dist-2.4.zip

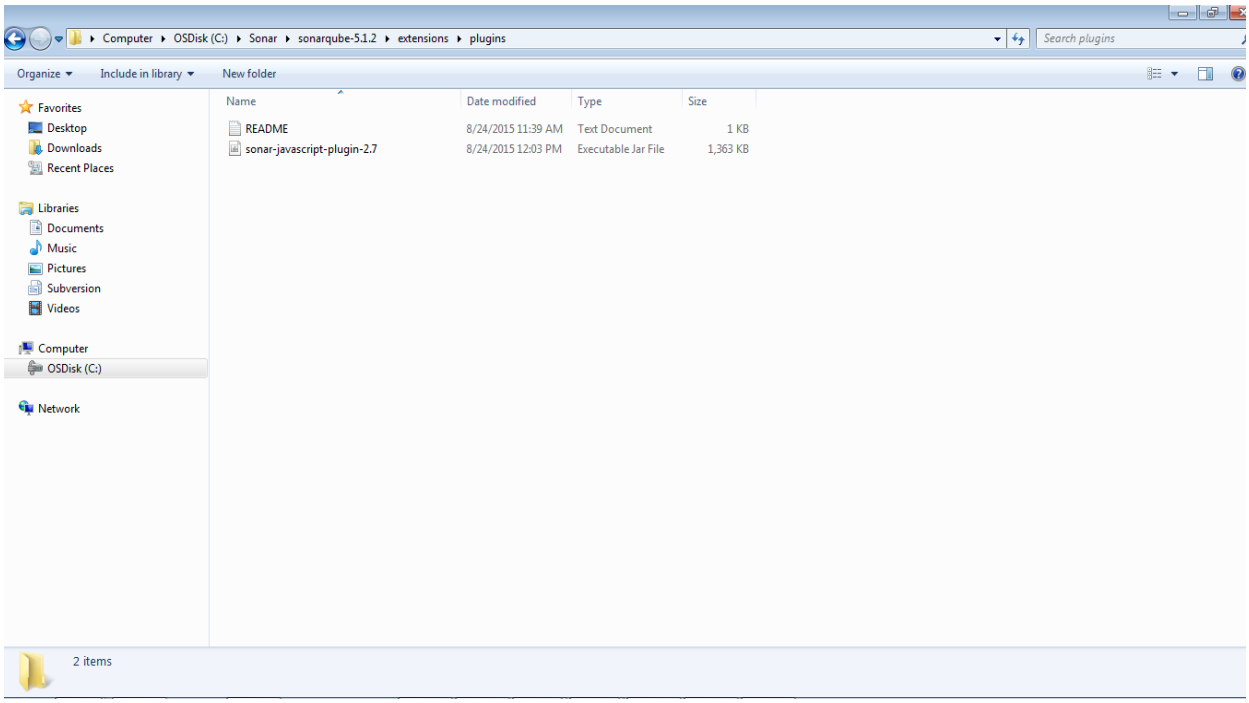
2.1.2 Install and Configure sonarqube

1. Download the correct version from download url for angular UI framework.
2. Unzip the file and move files to **c:\sonar** folder (Please verify the folder structure should be same as mentioned below)



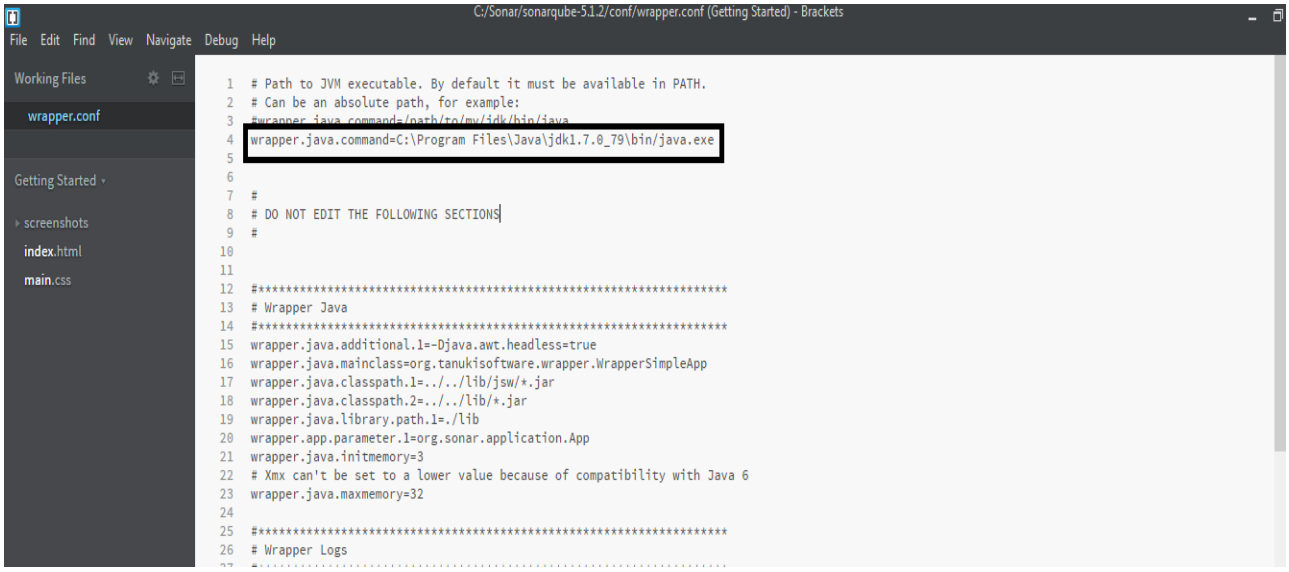
3. Place the below mentioned jar file under **c:\sonar\sonarqube-5.1.2\extensions\plugins**

sonar-javascript-plugin-2.7.jar



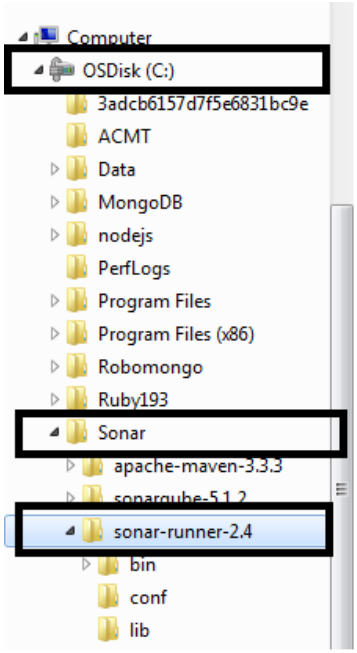
4. Open **C:\Sonar\sonarqube-5.1.2\conf\wrapper.conf** file and make sure **wrapper.java.command** key value is set to your java.exe path.

Example: wrapper.java.command=C:\Program Files\Java\jdk1.7.0_79\bin\java.exe



2.1.3 Install and Configure sonar-runner for angularjs

1. Download the correct version from download url to map with sonarQube for validating angular JS file.
2. Unzip the file and move files to **C:\sonar** (Please verify the file structure should be same as shown below)



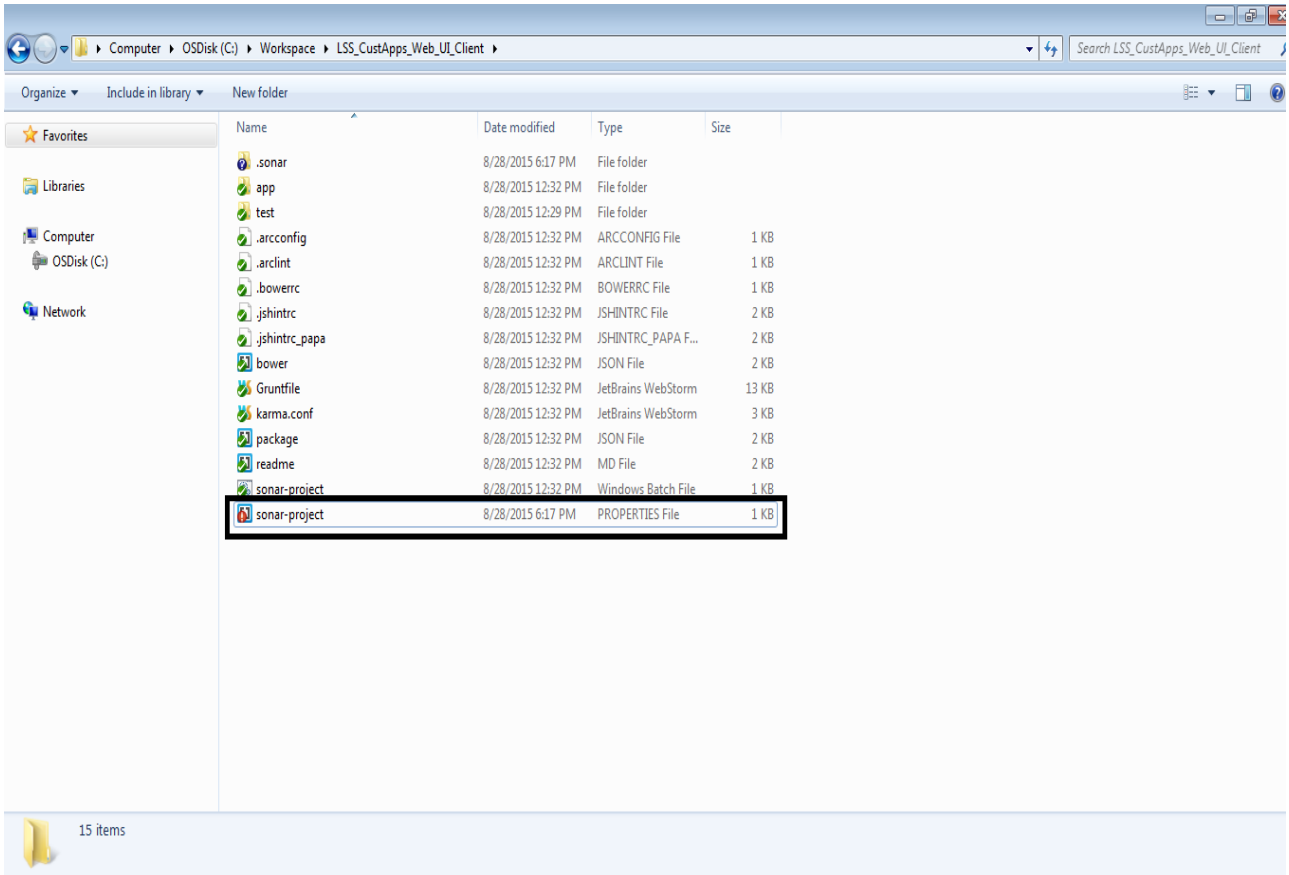
3. Remove the **sonar-runner.properties** from **c:\sonar\sonar-runner-2.4\conf** folder and copy the below file into the same directory.

sonar-runner.properties

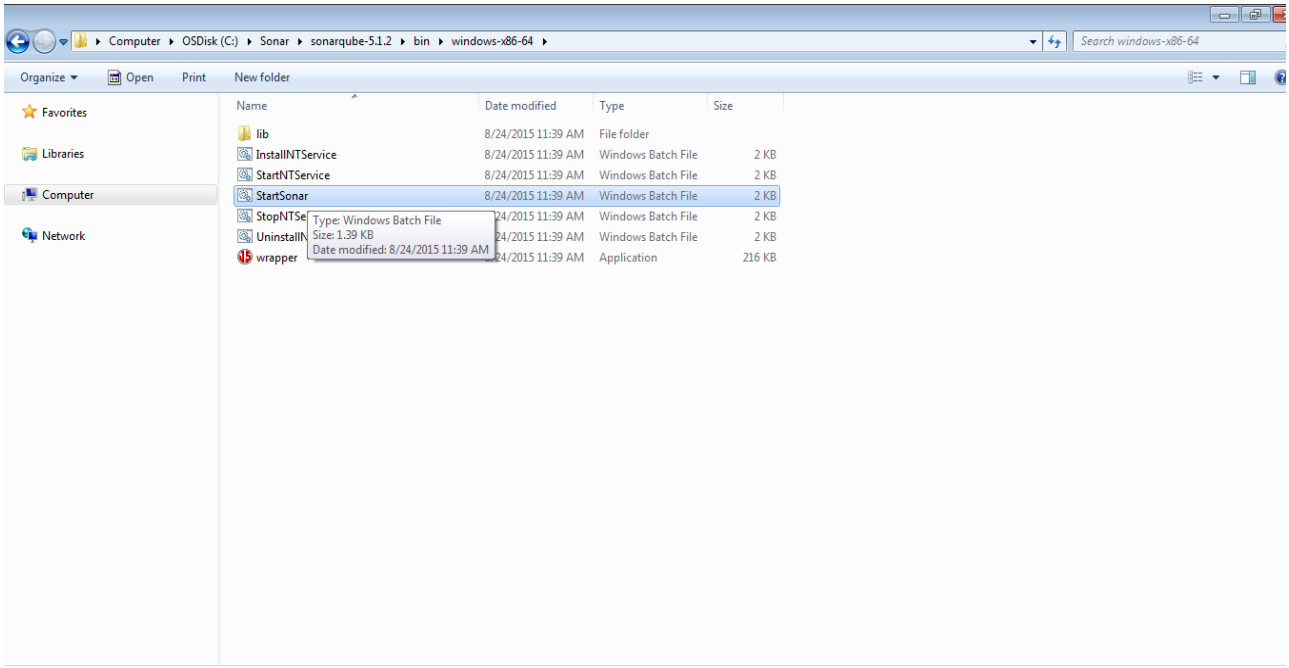
2.1.4 Starting sonar server with client ui project

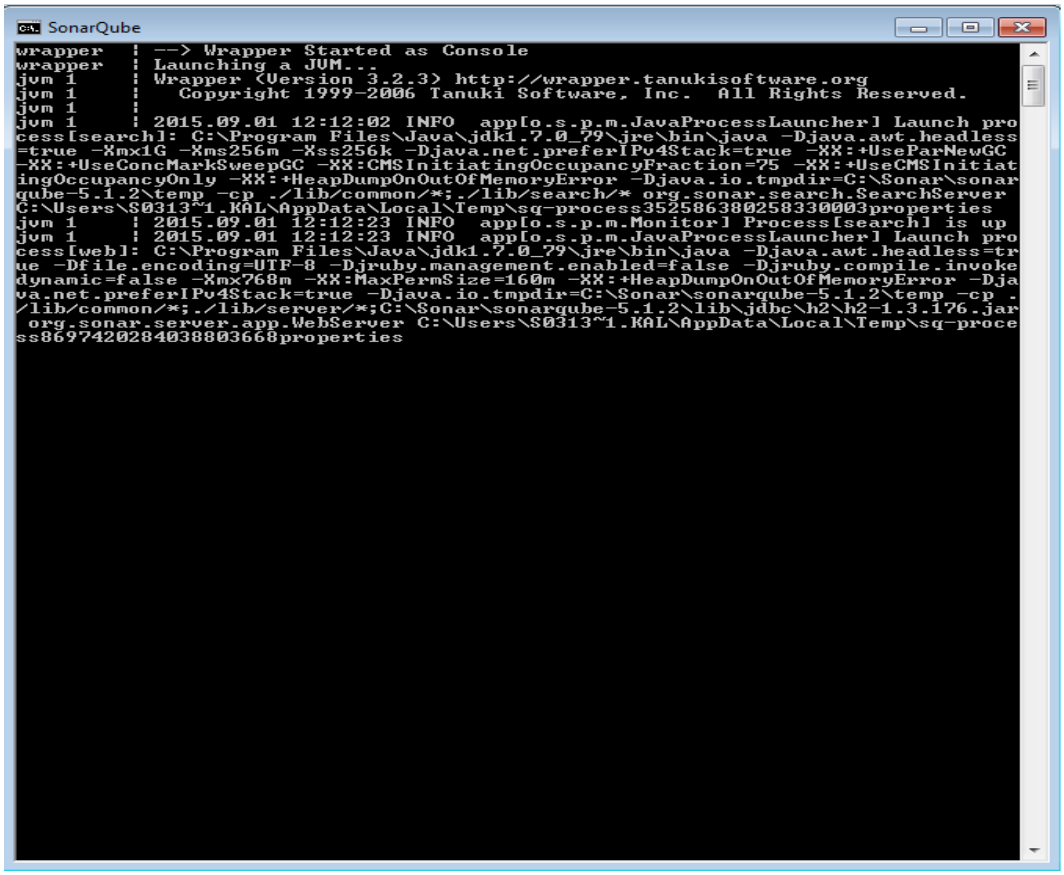
1. Place the below sonar-project.properties file into your client project folder directly.

sonar-project.properties

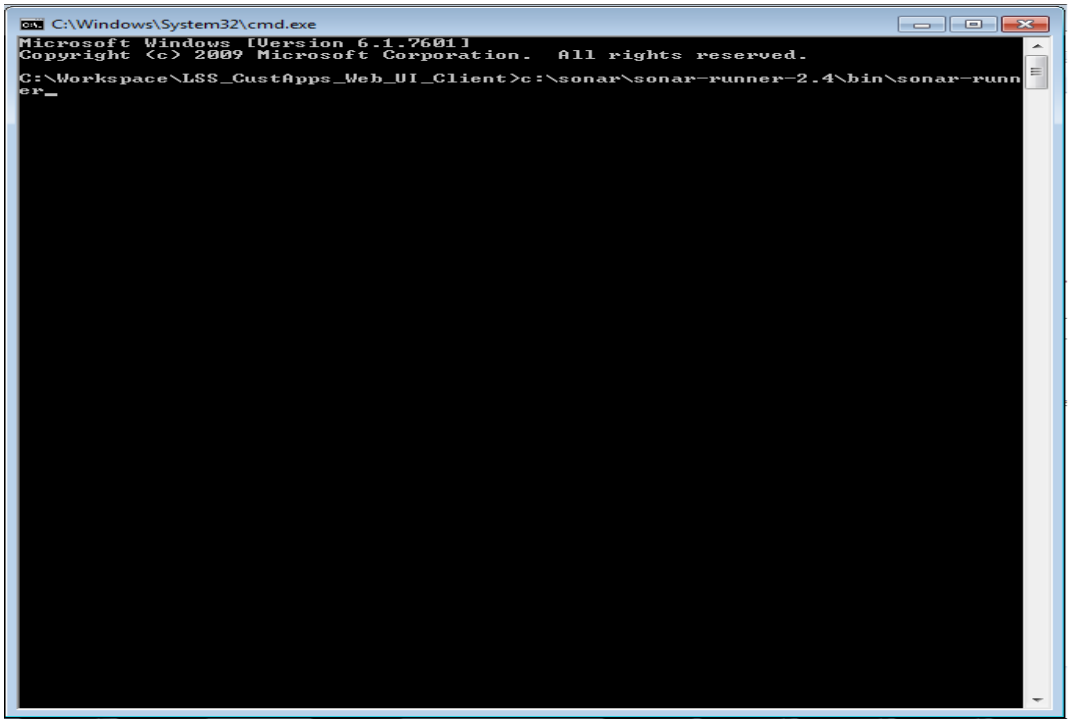


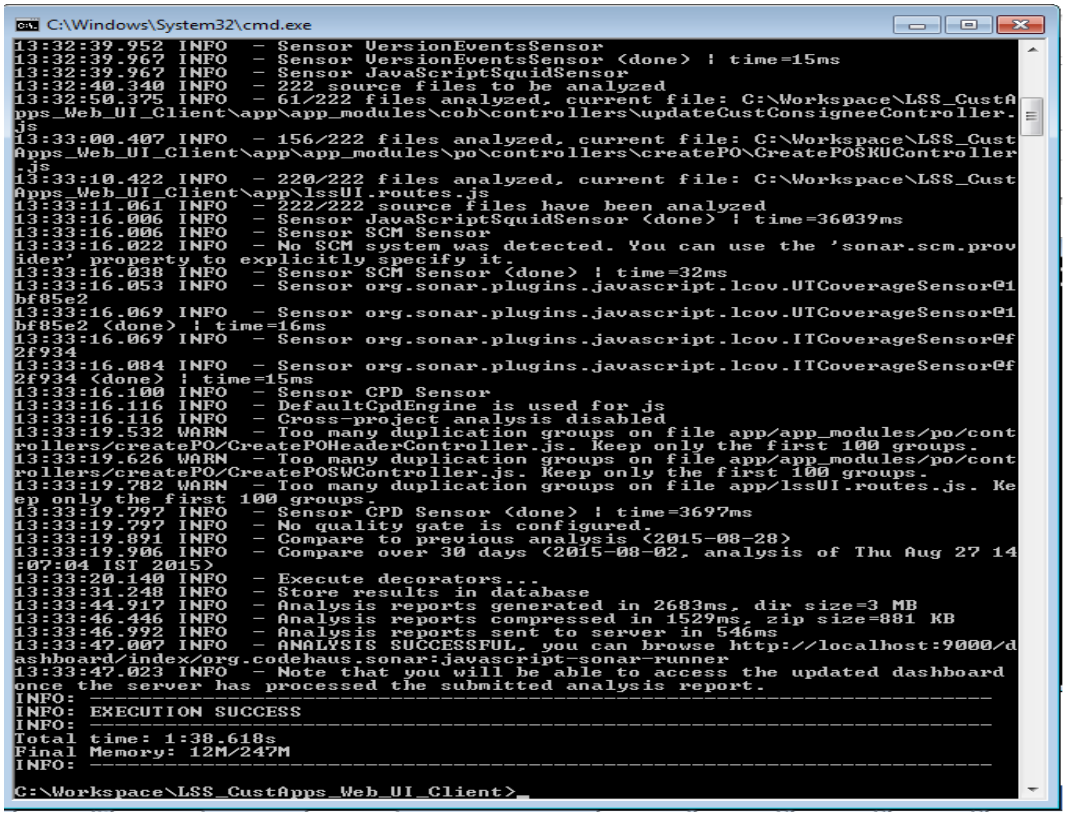
2. Start the sonar server **StartSonar.bat** file from **C:\Sonar\sonarqube-5.1.2\bin\windows-x86-XX** as shown below



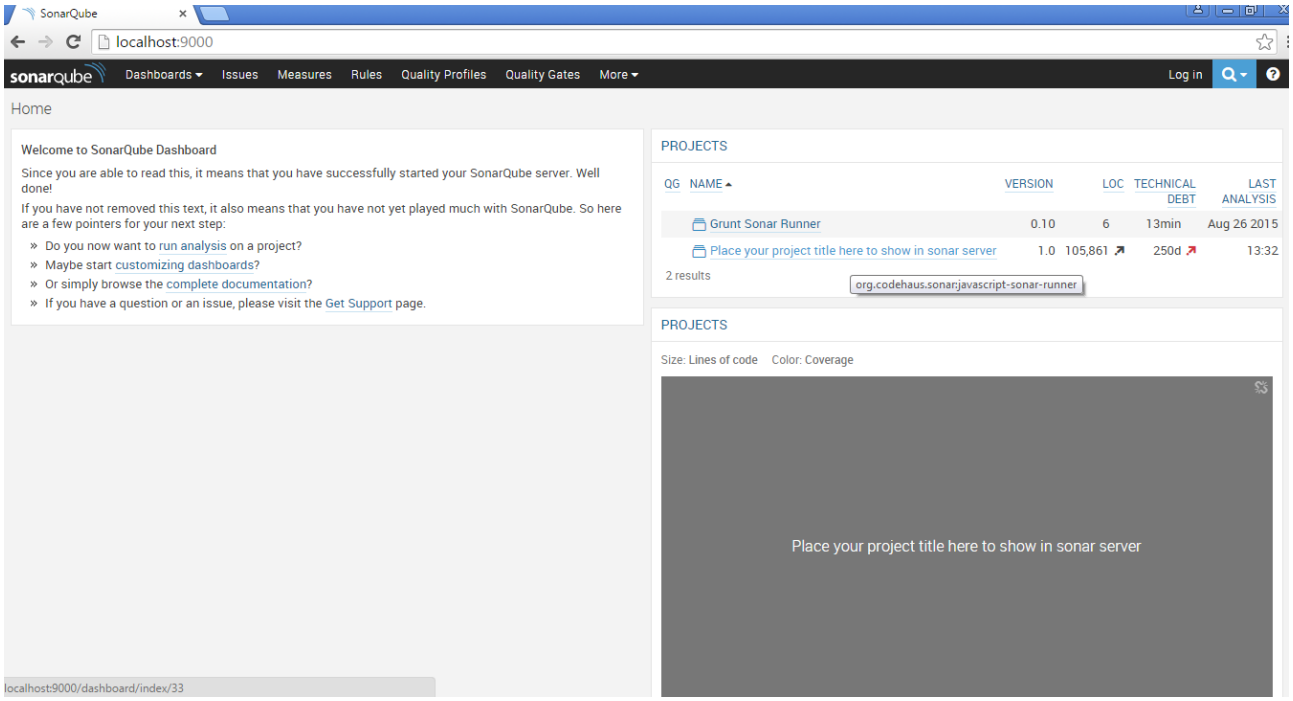


3. Once sonar server started, start the sonar runner from your client path command prompt to validate the UI angular project.

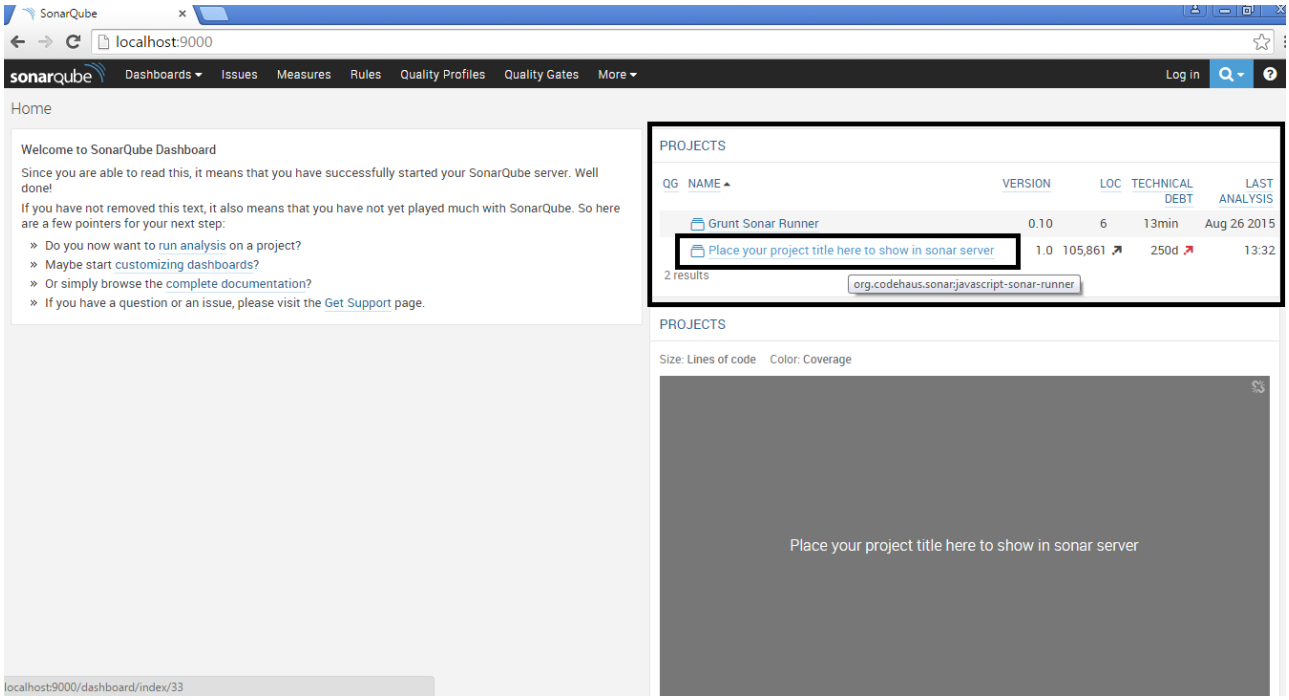


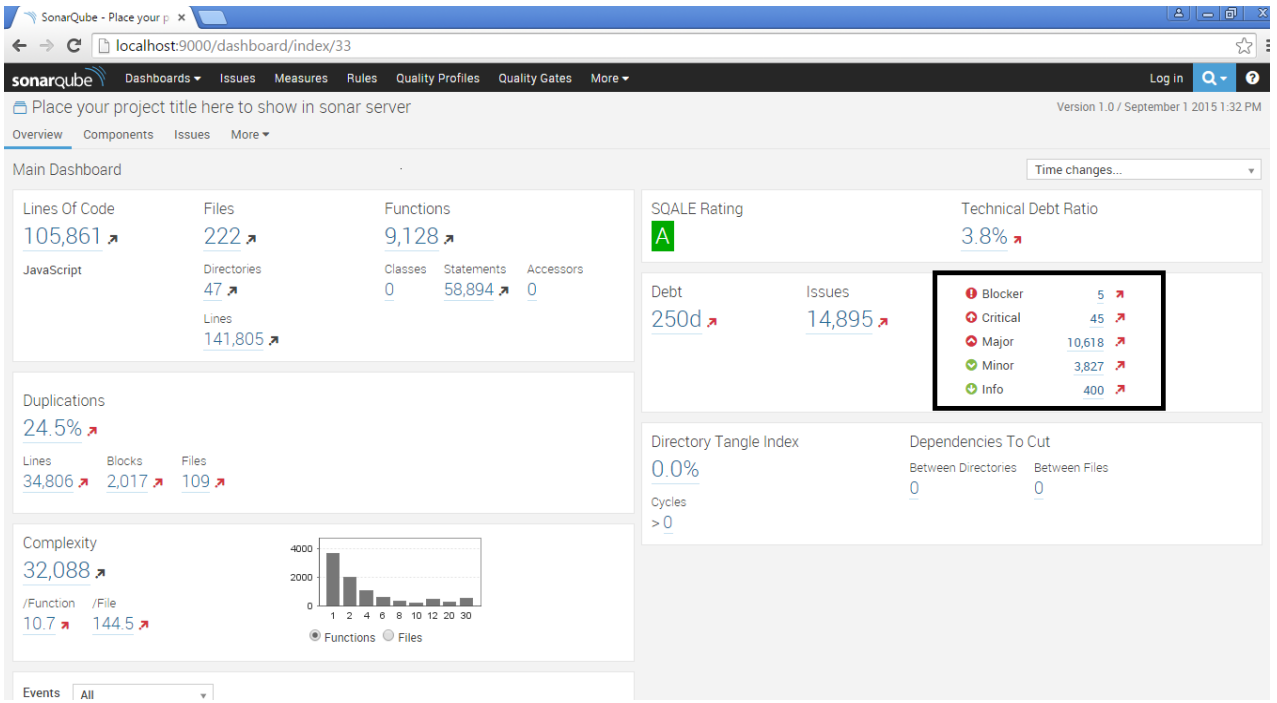


4. Once sonar runner done successfully, open the sonar server in your browser using http://localhost:9000 url.

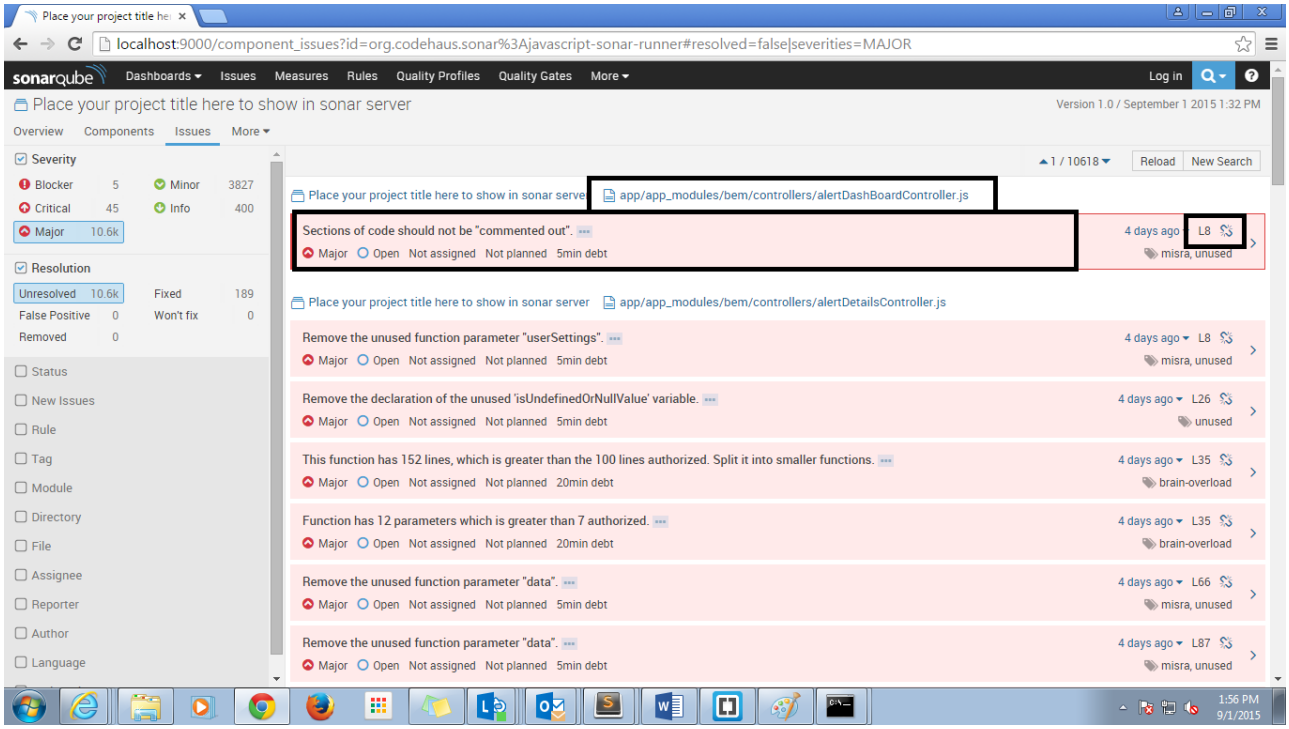


5. Choose your specific project from the PROJECT lists in sonar home page. Once selected specific project, the page will be looks as shown below





6. By selecting the issues on the priority type, sonar will show the error message with js file name and line number with error description.



2.2 KARMA JASMINE UNIT TEST CASE FOR ANGULAR JS

Unit testing involves **breaking your program into pieces**, and subjecting each piece to a series of tests.

Unit testing simply verifies that individual units of code (mostly functions) work as expected. Usually you write the test cases yourself

Some of unit test case frameworks:

1. Junit
2. Mocha
3. UnitTesting
4. JSpec
5. Jasmine.

Jasmine is one of the framework which will be used to write test cases for javascript based files. Once done writing the test case using jasmine framework, that you can run using the karma test runner.

2.2.1 Karma, jasmine and yeoman

Karma:

Karma is a test runner for JavaScript that runs on Node.js. It is very well suited to testing AngularJS or any other JavaScript projects. Using Karma to run tests using one of many popular JavaScript testing suites (Jasmine, Mocha, QUnit, etc.) and have those tests executed not only in the browsers of your choice, but also on the platform of your choice (desktop, phone, tablet.) Karma is highly configurable, integrates with popular continuous integration packages (Jenkins, Travis, and Semaphore) and has excellent plugin support.

Jasmine:

Jasmine is an open source testing framework for JavaScript. It aims to run on any JavaScript-enabled platform, to not intrude on the application nor the IDE, and to have easy-to-read syntax.

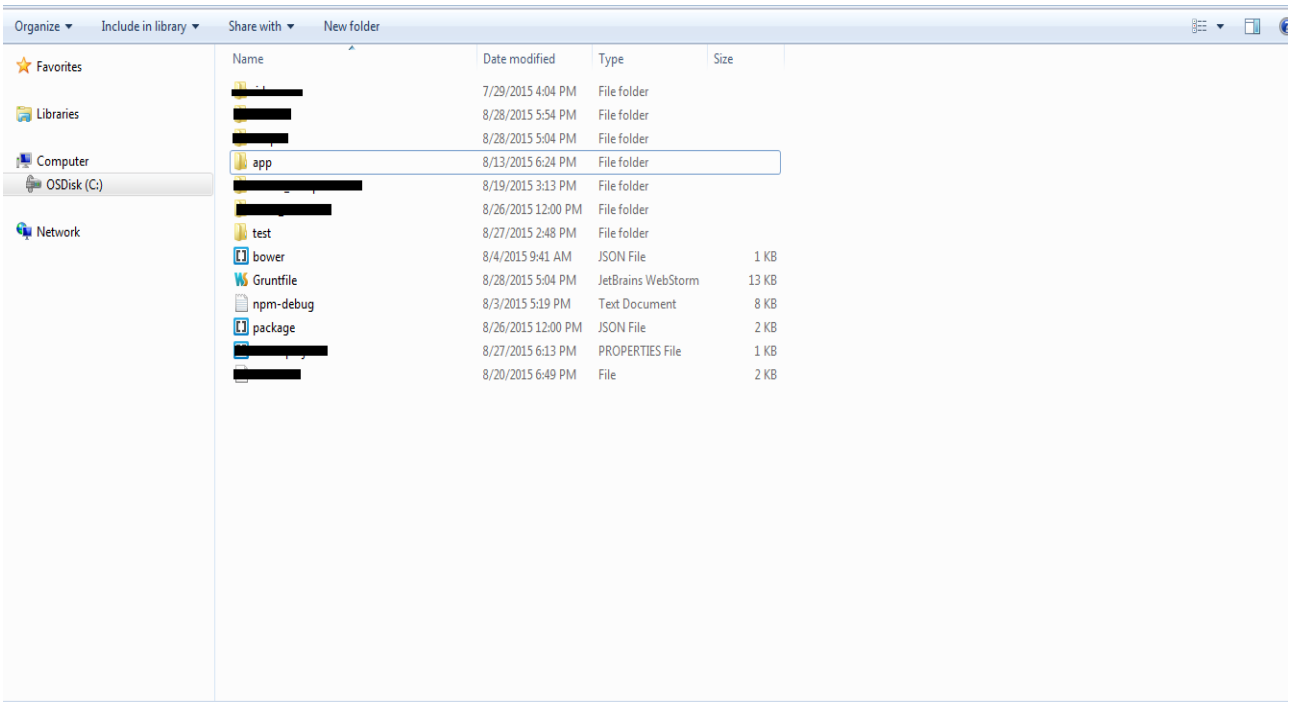
Yeoman:

Yeoman helps you to kick start new projects, prescribing best practices and tools to help you stay productive.

To do so, we provide a generator ecosystem. A generator is basically a plugin that can be run with the `yo` command to scaffold complete projects or useful parts.

2.2.2 Folder structure

Yeoman installation will provide us the default MVC folder structure for our project. Which includes karma and jasmine by default in our configuration. Once conifgured the yeoman, the folder structure will be looks like as shown below.



As presents in screen shot above, **app, test, bower.json, gruntfile.js and package.json** are the important files that yeoman will create while configure it.

Test folder will contains all the unit test case file for respective source file present under app folder.

Note:

Running the **'YO'** command in your command prompt under specific path, will generate the folder structure as shown above.

2.2.3 Configuration files

Karma.conf.js:

- The one important file required for testing jasmine and karma is karma.conf.js

This file contains all the dependencies that are belong to our angular js application module. While running grunt test command, this will loads all the dependencies files presents in karma.conf.js file and cross check with index.html.

Sample Karma.conf.js file

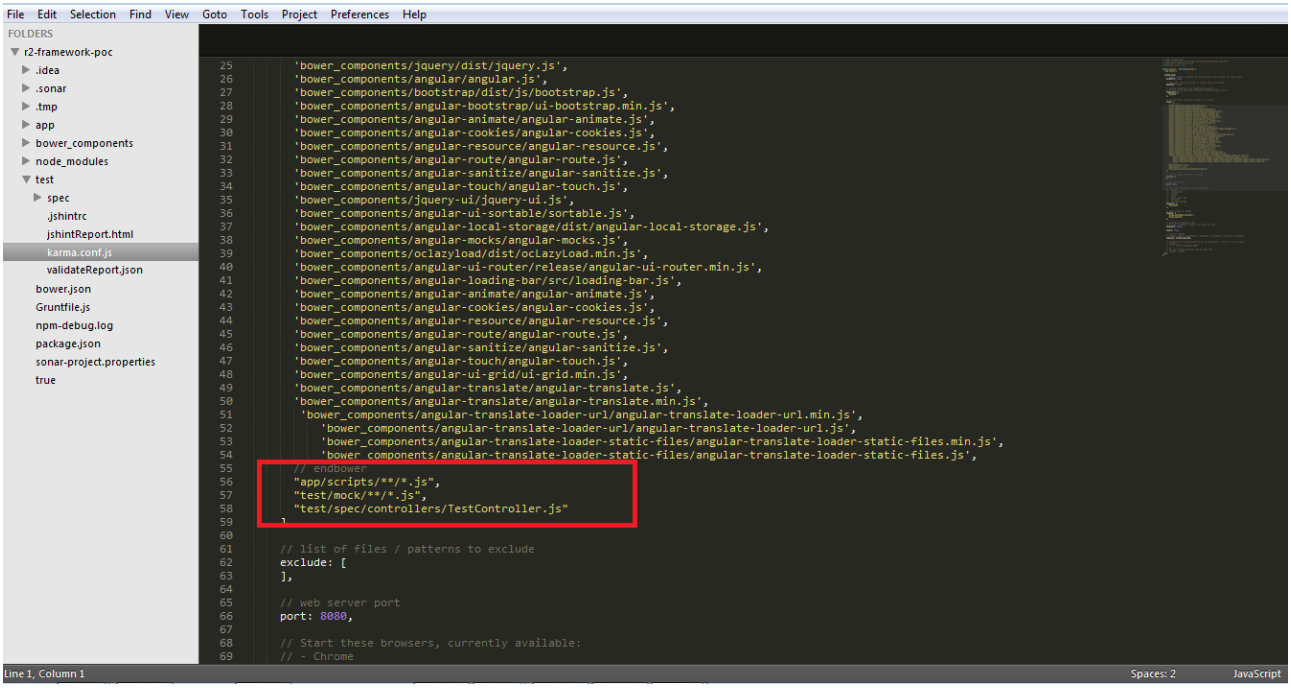
In the above screen shot, last three lines are js files mapped from app and test folder. While running grunt test command, these test files will be tested and the respective source file will be mapped for controller as well as module verification.

And also there are dependency files, which are mapped before we mentioned app and test folder. These dependencies are presents in our angular application modules.

2.2.4 Dependency files and bower components

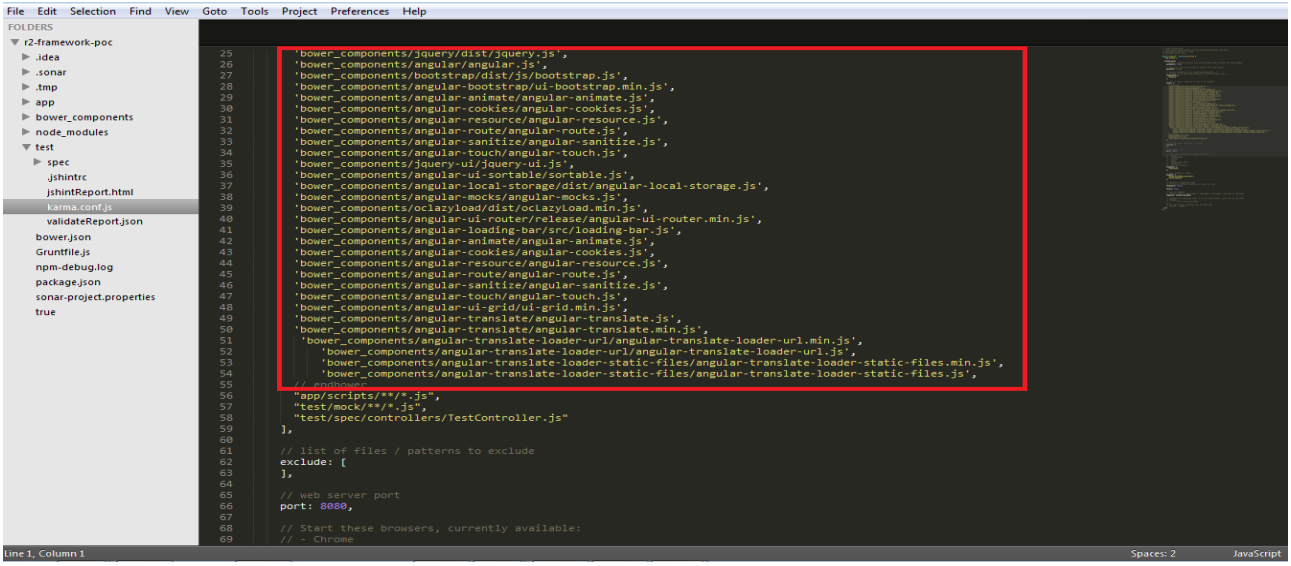
Dependency files and bower components are configured in karma.conf.js file. These dependency and bower files are two types. One is project dependency file and second one is bower components that we used in app modules as sub modules.

Dependency file sample (Type 1):



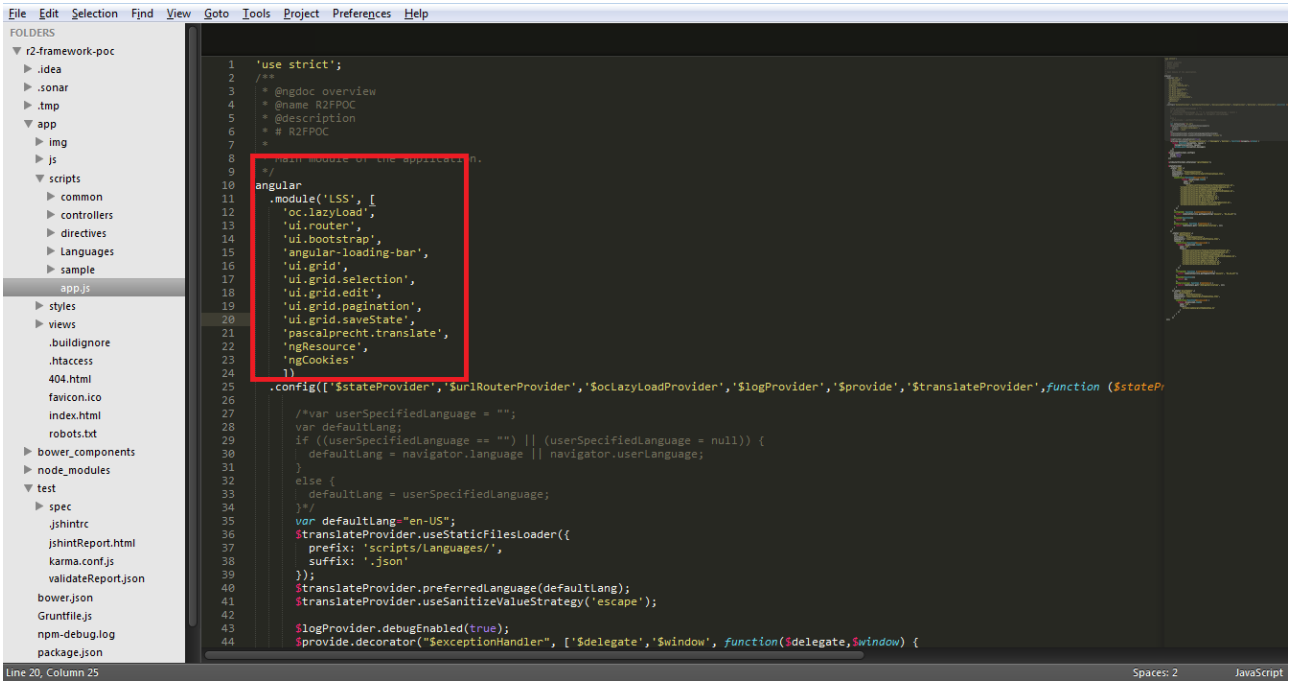
In the above screen shot, you can see that we have mapped our source and test files in karma.conf.js file.

Dependency file sample (Type 2):



In above screen shot, highlighted bower dependencies that we are using in our angular JS module as shown below.

App.js:



Whenever instantiating application module name, loading sub module or dependency files with that as shown above.

For the same dependencies file, you have to inject files in karma.conf.js as shown in type 2 screen shot.

Note:

These dependencies should be added in the proper order in karma.conf.js file.

2.2.5 writing test file using jasmine

Look at the screen shots below for the sample source file and test case file.

```
'use strict';
/**
 * @ngdoc function
 * @name R2FPOC.controller:TestController
 * @description
 * # TestController
 * Controller of the R2FPOC
 */
angular.module('LSS')
    .controller('SimpleController', function($scope) {
        $scope.Title = "Search Finance Engine";
        $scope.formName = "SearchFinanceForm";
        $scope.selectedValue="";
        $scope.data=[];
        $scope.url='';
        $scope.name= "";
        $scope.dataJSON={};
        $scope.refData={};
        $scope.refData.data={};
        $scope.getGridData=function(){
            $scope.url='/COB/warehouse';
        };
    });
```

Sample source file

```
describe('SimpleController', function () {
    beforeEach(module('LSS'));

    var controller, $controller, $scope = {};

    beforeEach(inject(function (_$controller_) {

        $controller = _$controller_;
        controller = $controller('SimpleController', {$scope: $scope});
        $scope.Title = "Search Finance Engine";
        $scope.formName = "SearchFinanceForm";
        $scope.selectedValue = "";
        $scope.data = [];
        $scope.url = '';
        $scope.name = "";
        $scope.dataJSON = {};
        $scope.refData = {};
        $scope.refData.data = {};
        $scope.getGridData();
    }));

    it('Scope title equal to Search Finance Engine: Success scenario', function () {
        expect($scope.Title).toEqual('Search Finance Engine');
    });

    it('Verify the URL value from scope', function () {
        expect($scope.url).toEqual('/COB/warehouse');
    });
/*
    it('Scope title not equal to Search Finance Engine: Failure scenario', function () {
        expect($scope.Title).toEqual('Fail');
    });*/

/*
    it('Verify the URL value from scope', function () {
        expect($scope.url).toEqual('/COB/warehoue');
    });*/


});
```

Sample test file

Key concepts in jasmine framework:

Describe:

```
describe('SimpleController', function () { });
```

Describe is the function that will combine all the unit test case into one section with the name of Source Controller name. One test case file may contain more than one describe, each will contain their own test cases writing inside it.

BeforeEach:

```
beforeEach(module('LSS'));

beforeEach(inject(function (_$controller_) {
    $controller = _$controller_;
    controller = $controller('SimpleController', {$scope: $scope});
    $scope.Title = "Search Finance Engine";
  }));
```

Before each function will load the content inside the function before running each test case. So that the required data and module will be created before running the test case and the populated data will be used by test case block.

IT:

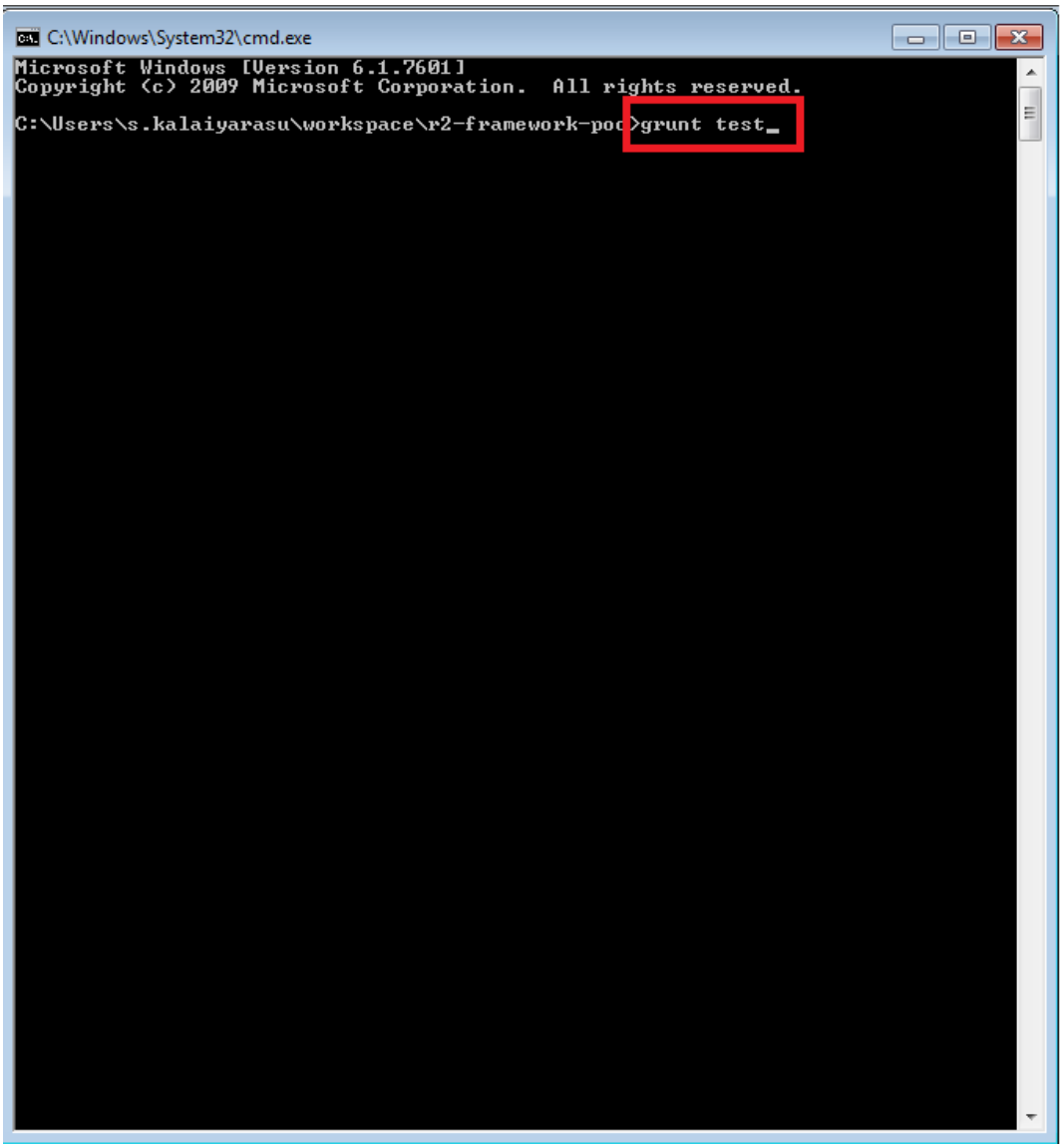
```
it('Scope title equal to Search Finance Engine: Success scenario', function () {
    expect($scope.Title).toEqual('Search Finance Engine');
 });
```

'it' function will act as a test case. Each it function is a separate test case that will be created under describe function. Inside it function we can have some test code as shown above.

Have a look into sample source file and test case file screen shot for better understanding.

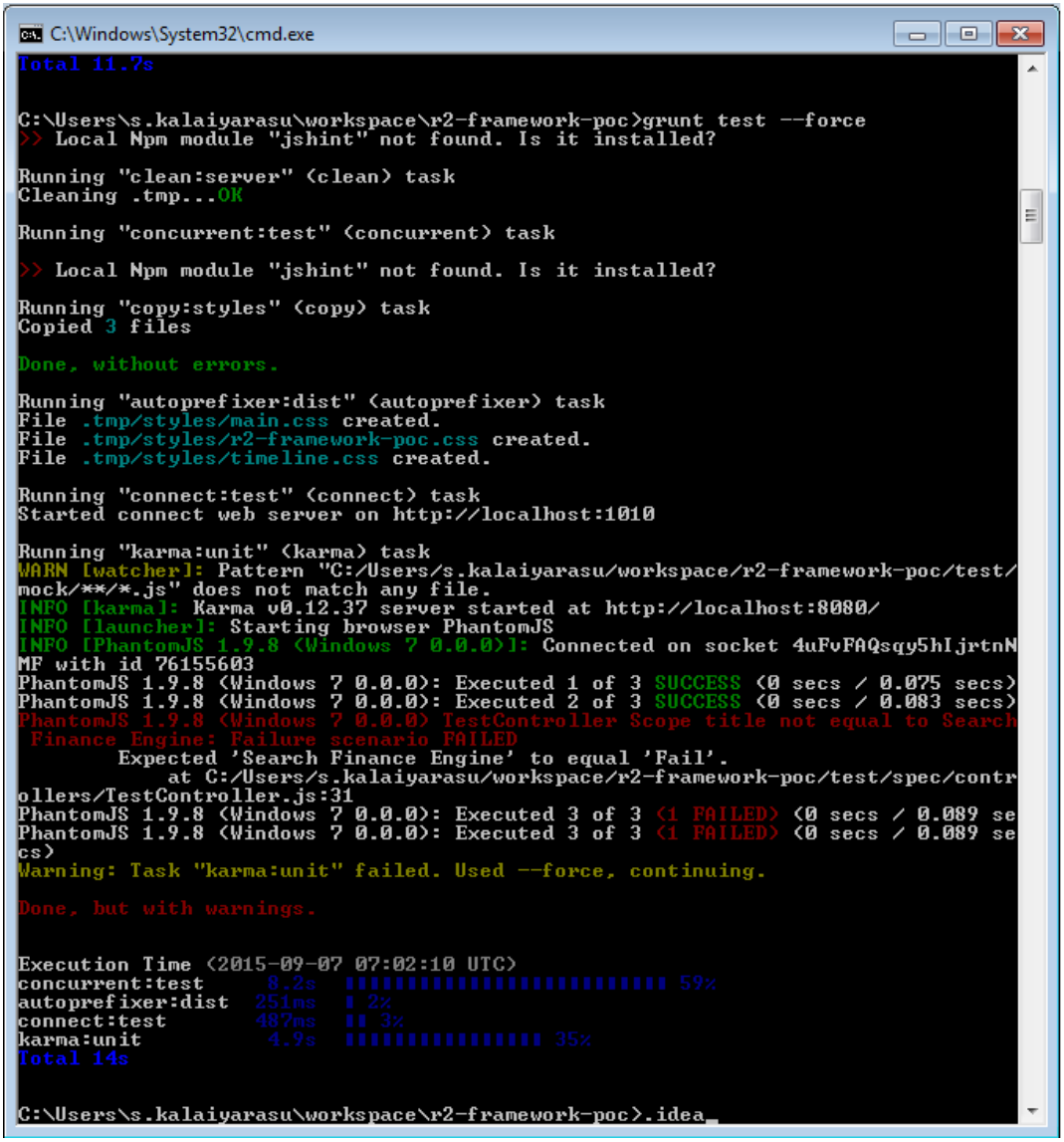
2.2.6 running test file using karma

grunt test is the command to run the test file. This command you have to run from your client path command prompt.

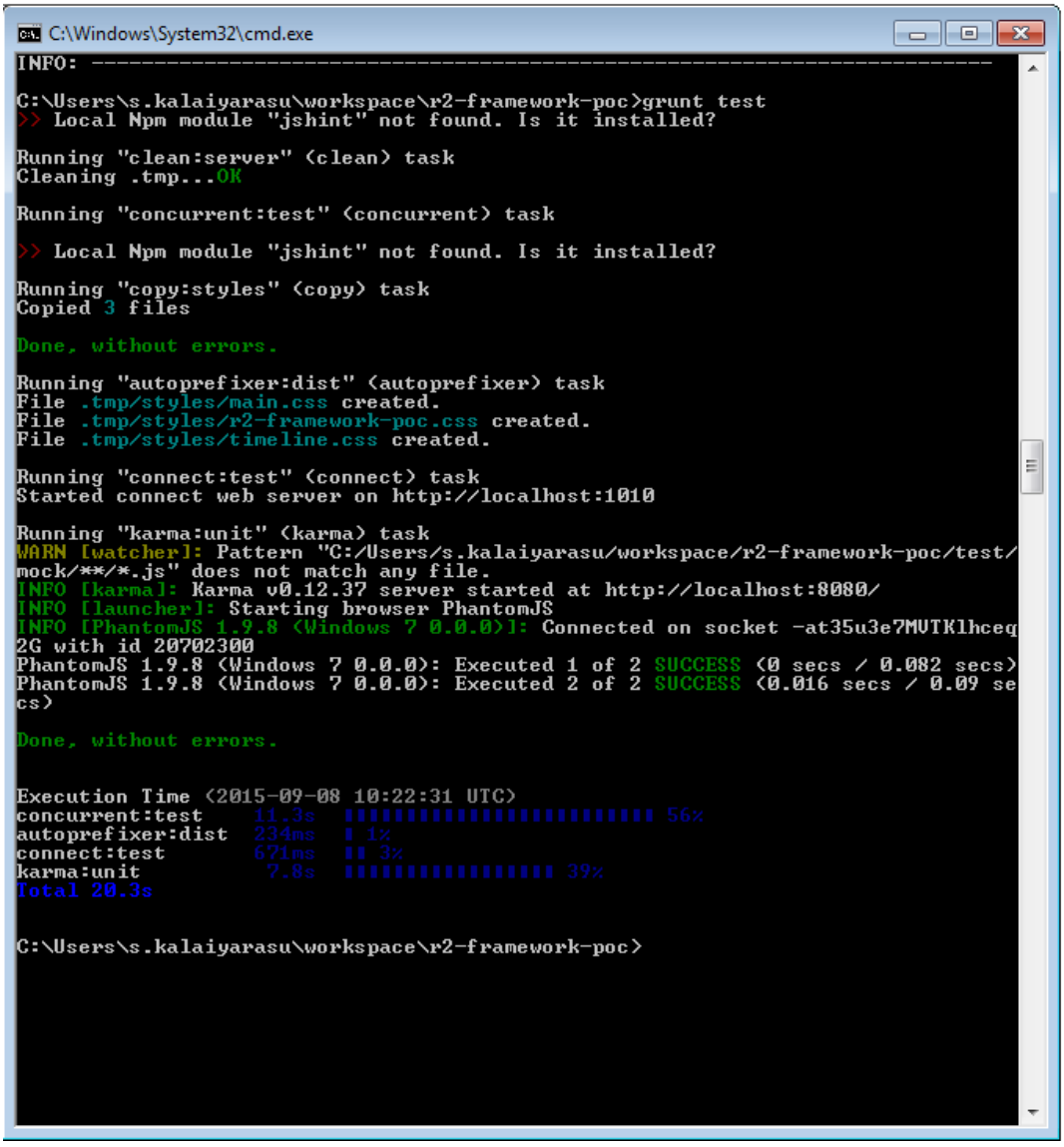


Once the command executed test case successfully, it will show the success and failure count of test case.

Make sure you are running the test command from your client path.



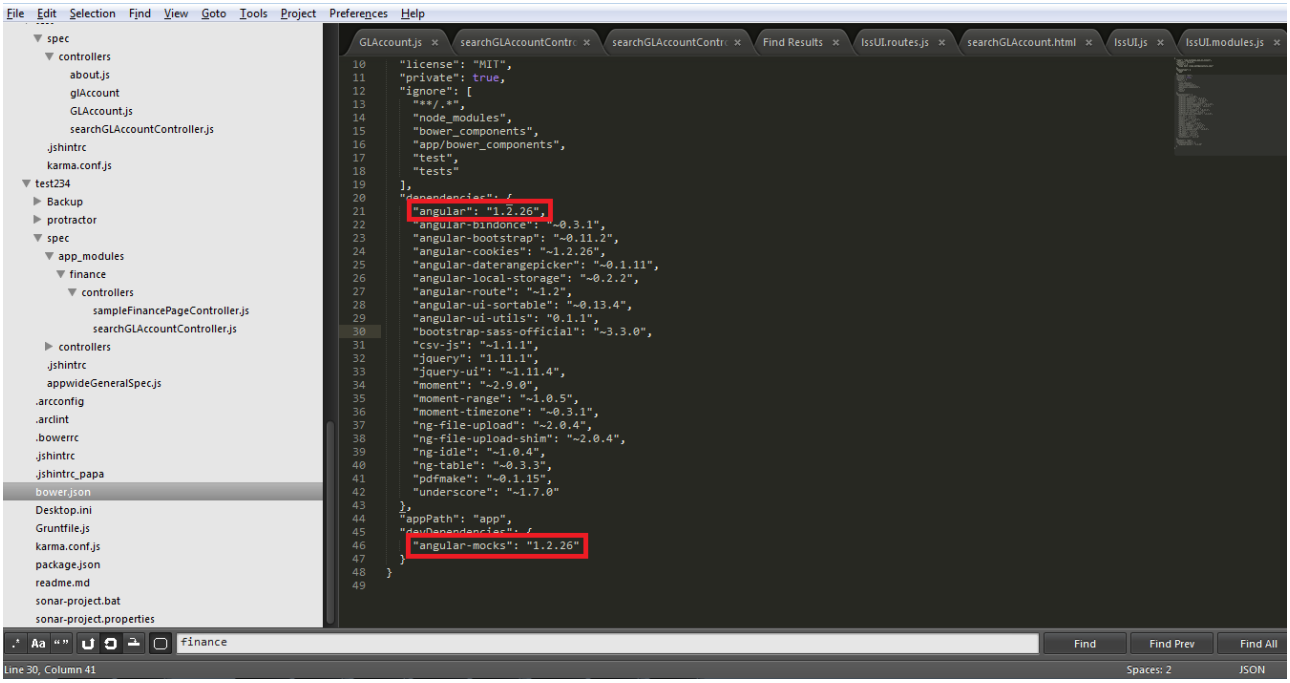
In the above test case result, out of three test cases, two are getting passed and one got failure due to expect function not matching with actual result. The failure error will throw with test case description name which is showing in red.



In the above screen shot, executed all the test cases successfully.

2.2.7 R2 code base configuration

In your bower.json check the version of angular and angular-mocks versions are same.



Note: (Important)

After migrating Angular to 1.3 please verify your bower.json file angular and angular-mocks version. Both versions should be same. Do bower install command in your client path and start configuring the dependency files into karma.conf.js file.

2.3 HTML VALIDATION FOR UI

An HTML validator aimed at AngularJS projects.

While there are other Grunt plugins that will validate HTML files, there are lacking a couple important features:

- Support for AngularJS attributes and tags (both from AngularJS and custom created)
- Support for templated/fragmented HTML files
- Ability to concurrently validate files for greatly increased speed

This plugin looks to solve these problems and provide the value that comes with having HTML validation in the build chain.

2.3.1 Sample configuration for HTML Validation:

```
  htmlangular: {
    options: {
      reportpath: 'html-angular-validate-report.json'
  },
  files: {
    src: ['app/index.html']
    }
  },
  'css-validation': {
  options: {
      reportpath:'cssValidateReportPath.json',
    },
    files: {
      src: ['app/css/grid.css']
    }
},
```

2.3.2 Sample HTML Validation report:

```
Running "htmlangular:files" (htmlangular) task
Validating app/index.html ...ERROR
[L5:C36] Element "title" must not be empty.
[L37:C16] Element "[redacted]-ui-loading" not allowed as child of element "body" in this context. (Suppressing further errors from this subtree.)
[L722:C26] Element "[redacted]-google-analytics" not allowed as child of element "div" in this context. (Suppressing further errors from this subtree.)
Warning: HTML validation failed Use --force to continue.
```

2.4 CSS VALIDATION FOR UI

An CSS validator aimed at CSS projects.

While there are other Grunt plugins that will validate CSS files, there are lacking a couple important features:

- Support for AngularJS and application.
- Support for templated/fragmented CSS files
- Ability to concurrently validate files for greatly increased speed

2.4.1 Sample configuration for CSS validation

```
htmlangular: {
  options: {
    reportpath: 'html-angular-validate-report.json'
},
files: {
  src: ['app/index.html']
  }
},
'css-validation': {
options: {
    reportpath:'cssValidateReportPath.json',
  },
  files: {
    src: ['app/css/grid.css']
  }
},
```

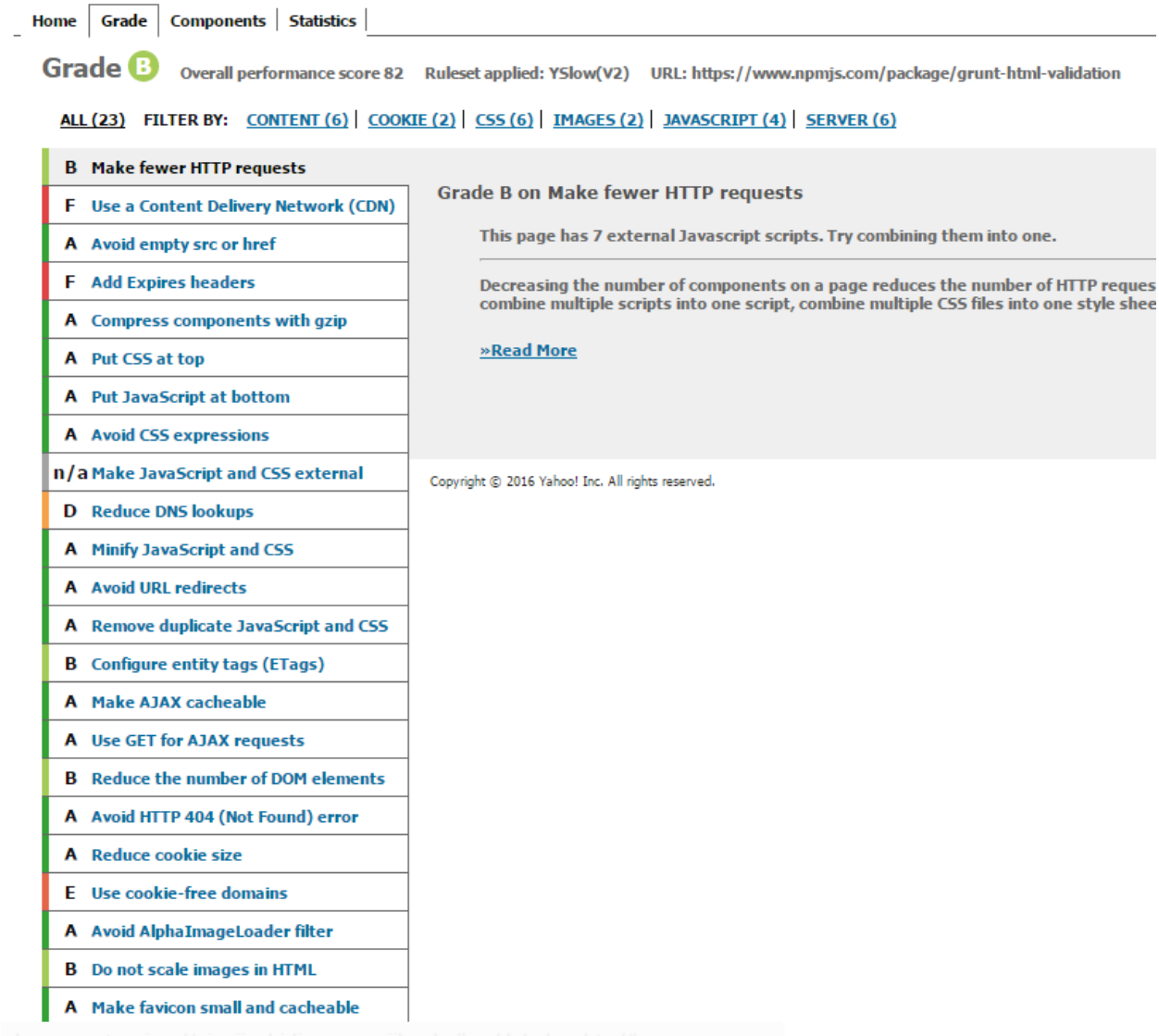
2.4.2 Sample CSS Validation Report

```
Running "css-validation:files" (css-validation) task
Validation started for.. app/css/grid.css
1=> ".dataTables_wrapper: Property zoom doesn't exist : 1 1" Line no: "5"
2=> ".dataTables_scrollBody: Parse Error ×margin-top: -1px;" Line no: "52"
3=> ".dataTables_scrollBody: Parse Error }" Line no: "53"
4=> ".paging_full_numbers span.paginate_button, .paging_full_numbers span.paginate_active: Parse Error ×cursor: hand;" Line no: "94"
5=> ": Parse Error [: left; ).paging_full_numbers span.paginate_button]" Line no: "98"
6=> ".paging_full_numbers span.paginate_active: Parse Error ×cursor: hand;" Line no: "106"
7=> ": Parse Error [: underline;/×background-color: #99B3FF;×/ )#gridFont]" Line no: "111"
8=> ".tag_count: Parse Error [empty string]" Line no: "254"
9=> ".grid_action: Value Error : -moz-inline-box is not a display value : -moz-inline-box -moz-inline-box" Line no: "263"
10=> ".grid_action_clp: Value Error : -moz-inline-box is not a display value : -moz-inline-box -moz-inline-box" Line no: "271"
11=> ".icon_status_active, .icon_status_inactive: Value Error : -moz-inline-box is not a display value : -moz-inline-box -moz-inline-box" Line no: "366"
12=> ".tag_count: Parse Error [empty string]" Line no: "501"
13=> ".grid_action: Value Error : -moz-inline-box is not a display value : -moz-inline-box -moz-inline-box" Line no: "510"
14=> ".grid_action_clp_asc: Value Error : -moz-inline-box is not a display value : -moz-inline-box -moz-inline-box" Line no: "518"
15=> ".grid_action_clp_dasc: Value Error : -moz-inline-box is not a display value : -moz-inline-box -moz-inline-box" Line no: "526"
16=> ".icon_status_active, .icon_status_inactive: Value Error : -moz-inline-box is not a display value : -moz-inline-box -moz-inline-box" Line no: "606"
No of errors: 16
Validation report generated: cssValidateReportPath.json

Done, without errors.
```

2.5 YSLOW REPORT

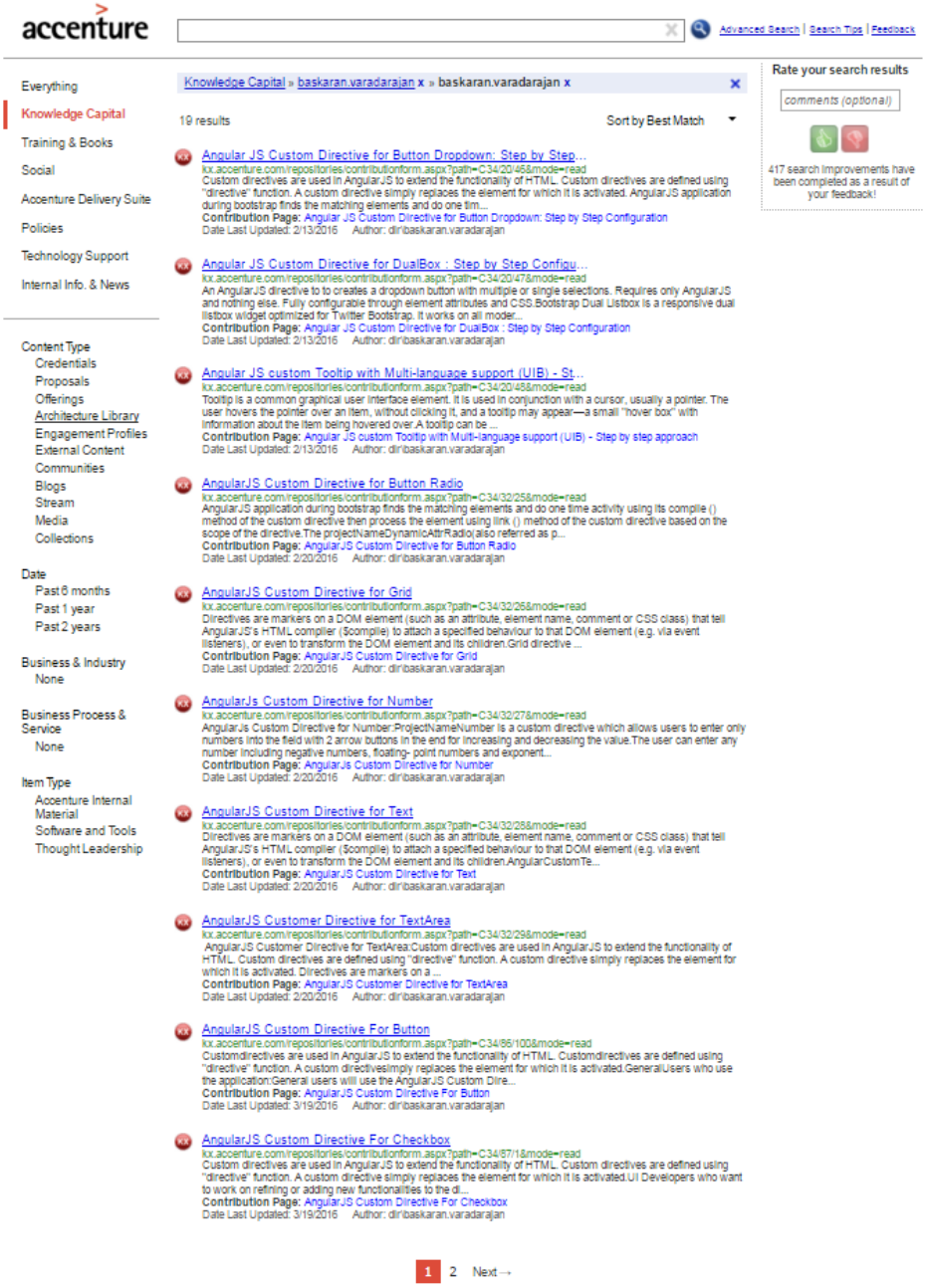
2.5.1 Sample performance report of YSlow application



3. REFERENCES LINK IN KX SITE

3.1 KX SITE LINKS:

https://search.accenture.com/search.aspx?aid=IKN&k=%20&s=KX_Contributions&a=(CredentialChampion:baskaran.varadarajan)OR(Contacts:baskaran.varadarajan)&v=date&origin=kxHome&suborigin=Content



https://search.accenture.com/search.aspx?aid=IKN&k=%20&s=KX_Contributions&a=(CredentialChampion:baskaran.varadarajan)OR(Contacts:baskaran.varadarajan)&p=2&origin=searchwithin&suborigin=resultsikn