Testing JavaScript

- Jasmine
- Karma

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

JavaScript Testing Issues

- Like programs in any language, JavaScript programs should have **unit tests** that test the basic function of functions and classes
- The notion of unit testing began with the Java **jUnit** framework, but has been adopted by other languages, including JavaScript

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

JavaScript Testing Tools

- There are several popular JavaScript tools to facilitate unit testing:
    - **Jasmine** – Unit testing framework
    - **QUnit** – Unit testing framework
    - **Mocha** – Unit testing framework
    - **Karma** – A test runner for many environments, including browsers and Node.js. It can integrate with other frameworks

---

In this chapter, we will cover Jasmine and Karma.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Introduction to Jasmine

- **Jasmine** is a unit testing framework formerly known as jsUnit
- The home page for Java is at jasmine.github.io



AngularJS uses Jasmine to test its own code.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Installing Jasmine

- Jasmine provides several installation options, including:
    - Standalone
    - Node.js
    - Ruby
    - Java (Maven)
- You can download the standalone installation of Jasmine from https://github.com/jasmine/jasmine/releases

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Jasmine Fundamentals

- Jasmine provides:
    - **describe()** – a function you can use to group related tests, i.e. a **test suite**
    - **it()** – a function that contains one or more tests, i.e. a **spec**
    - **expect()** – a function that execute an assertion
    - **matchers** – functions that act as assertions
- Jasmine provides several matchers and you can write your own

---

Jasmine has several other features, including setup and teardown support to build test fixtures, "spies" that let you mock a function for testing and so forth.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

A Simple Test

- Here's a simple test that verifies array sorting

```
1    // sortUsers() is our function we want to test
2    describe('sorting the list of users', function()
3    {
4      it('sorts in descending order by default', function()
5      {
6        var users = ['jack', 'igor', 'jeff'];
7        var sorted = sortUsers(users);
8        expect(sorted).toEqual(['jeff', 'jack', 'igor']);
9      });
10   });
```

The strings passed as first parameters to describe() and it() document the test.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Built-in Matchers

Matcher	Description
toBe()	Compares using ===
toEqual()	Compares using ==
toMatch()	Matches a regex
toBeUndefined()	Compares against JavaScript 'undefined' value
toBeNull()	Compares against JavaScript null
toBeTruthy()	Checks for truthiness
toBeFalsy()	Checks for falsyness
toContain()	Determines if an item is in an array
toBeLessThan()	Numeric comparison
toBeGreaterThan()	Numeric comparison
toBeCloseTo()	Numeric comparison with a fudge factor
toThrow()	Nxpect an exception

The expect() function also provides a "not" property that you can use to negate a matcher, for example: expect(....).not.toThrow(...).

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

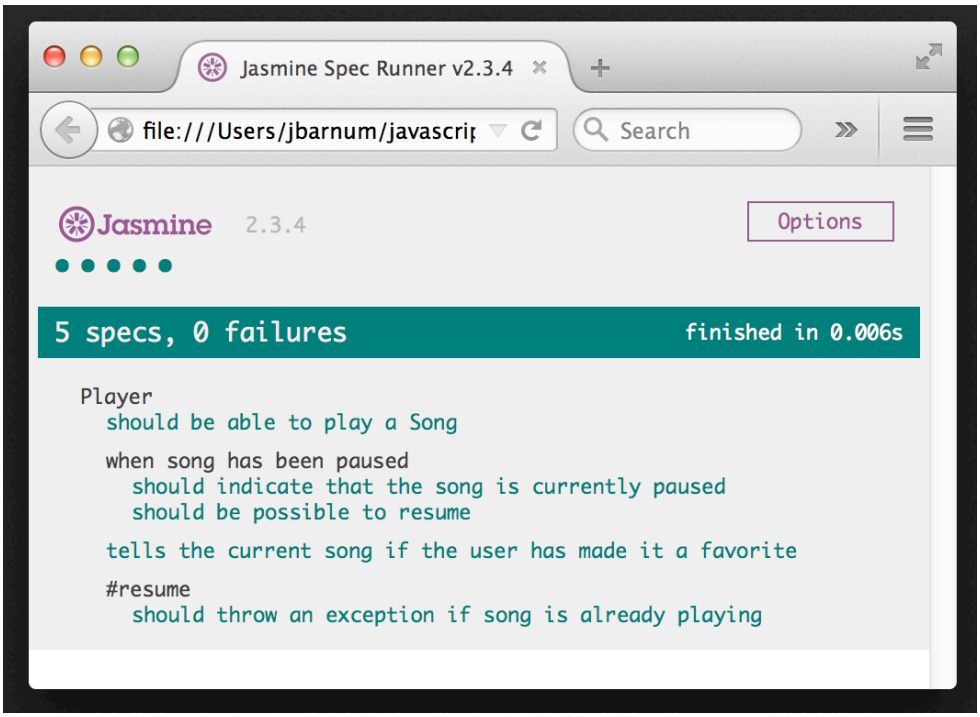
Writing a Test Runner HTML Page

- To run tests, you can write a testing Web page that includes your JavaScript code, your tests (specs) and the Jasmine framework
- Jasmine provides CSS to style test results for easy reading

```
1    <link rel="shortcut icon" type="image/png"
2       href="lib/jasmine-2.3.4/jasmine_favicon.png">
3    <link rel="stylesheet" href="lib/jasmine-2.3.4/jasmine.css">
4
5    <script src="lib/jasmine-2.3.4/jasmine.js"></script>
6    <script src="lib/jasmine-2.3.4/jasmine-html.js"></script>
7    <script src="lib/jasmine-2.3.4/boot.js"></script>
8
9    <!-- code to test -->
10   <script src="src/Player.js"></script>
11   <script src="src/Song.js"></script>
12
13   <!-- Jasmine test scripts (specs) -->
14   <script src="spec/SpecHelper.js"></script>
15   <script src="spec/PlayerSpec.js"></script>
```

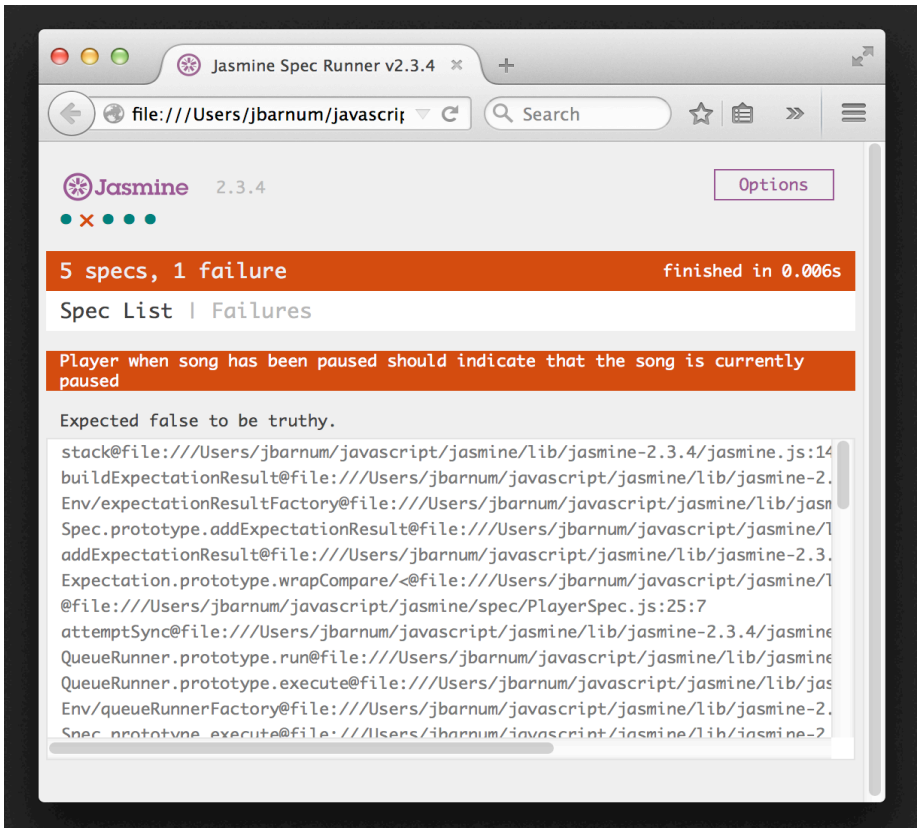

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Writing a Test Runner HTML Page, cont'd



Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Writing a Test Runner HTML Page, cont'd



Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Introduction to Karma

- Karma is a **test runner** that makes it easy to run tests written in Jasmine or other unit-testing framework
- Karma runs your tests in various browsers or Node.js and reports pass/fail to the command line
- The home page for Karma is at: karma-runner.github.io



Karma was formerly known as Testacular.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Installing Karma

- Karma runs as a command-line tool using the **Node Package Manager** (npm)
- After installing Node.js, you can install Karma:

```
npm install karma --save-dev

sudo npm install -g karma-cli
```

You can dowload Node.js itself from:

`https://nodejs.org/download`

The Node.js install will also install npm and optionally update your computer's PATH to reference it.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

The Karma Configuration File

- Before running Karma, you must create a configuration file that specifies where your tests are and which browsers you want to test under
- Karma provides an **init** script you can run to create the configuration file

```
karma init my.conf.js
```

my.conf.js

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

```
1    // Karma configuration
2    // Generated on Wed Jun 17 2015 12:08:02 GMT-0700 (PDT)
3
4    module.exports = function(config) {
5      config.set({
6
7        // base path that will be used to resolve all patterns (eg. files, exclude)
8        basePath: '',
9
10
11       // frameworks to use
12       // available frameworks: https://npmjs.org/browse/keyword/karma-adapter
13       frameworks: ['jasmine'],
14
15
16       // list of files / patterns to load in the browser
17       files: [
18         '/Users/jbarnum/javascript/jasmine/**/*.js'
19       ],
20
21
22       // list of files to exclude
23       exclude: [
24       ],
25
26
27       // preprocess matching files before serving them to the browser
28       // available preprocessors: https://npmjs.org/browse/keyword/karma-preprocesso
29       preprocessors: {
30       },
31
32
33       // test results reporter to use
34       // possible values: 'dots', 'progress'
35       // available reporters: https://npmjs.org/browse/keyword/karma-reporter
36       reporters: ['progress'],
37
38
39       // web server port
40       port: 9876,
41
42
43       // enable / disable colors in the output (reporters and logs)
44       colors: true,
45
```

```
46
47      // level of logging
48      // possible values: config.LOG_DISABLE || config.LOG_ERROR || config.LOG_WARN
49      logLevel: config.LOG_INFO,
50
51
52      // enable / disable watching file and executing tests whenever any file change
53      autoWatch: false,
54
55
56      // start these browsers
57      // available browser launchers: https://npmjs.org/browse/keyword/karma-launche
58      browsers: ['Chrome'],
59
60
61      // Continuous Integration mode
62      // if true, Karma captures browsers, runs the tests and exits
63      singleRun: false
64    });
65  };
```

Karma and Browsers

- Karma supports many browsers for testing (referred to as **capturing** a browser)
- Karma will start the brower(s), run the tests, kill the browser and then report the pass/fail to the command line
- You can also start a browser manually for testing, even a browser running on a mobile device

---

To use a device's browser, on the device, you browse to the URL where the Karma server is running. For details, see: http://karma-runner.github.io/0.12/config/browsers.html

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Running Karma Tests

- To run tests, from the command-line:

```
karma start my.conf.js
```


Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Using Karma and Jasmine With AngularJS

- It's common to use Karma and Jasmine to test AngularJS artifacts such as **controllers** and **directives**

There is a simple, but instructive tutorial at:

`https://blog.logentries.com/2015/01/unit-testing-with-karma-and-jasmine-for-angularjs`

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Chapter Summary

In this chapter, you learned:

- About popular JavaScript testing frameworks
- How to get started with Jasmine and Karma

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum