



Sencha

Unit Testing Web and Mobile Applications

January 2013 Webinar

Arthur Kay

@arthurakay

Solutions Engineer, Sencha Inc.

Mats Bryntse

@bryntum

CEO, Bryntum

Agenda

- Unit Testing: Overview
- Challenges for Web Developers
- JavaScript Syntax
- Logical Tests (Unit)
- Integration Tests (UI)
- Resources
- Q&A



Unit Testing: Overview



What are Unit Tests?

- A way to gain confidence in your code
- Verification that bugs exist
- Passing tests \neq no bugs
- Passing tests \neq working code



Unit Testing: Challenges for the Web

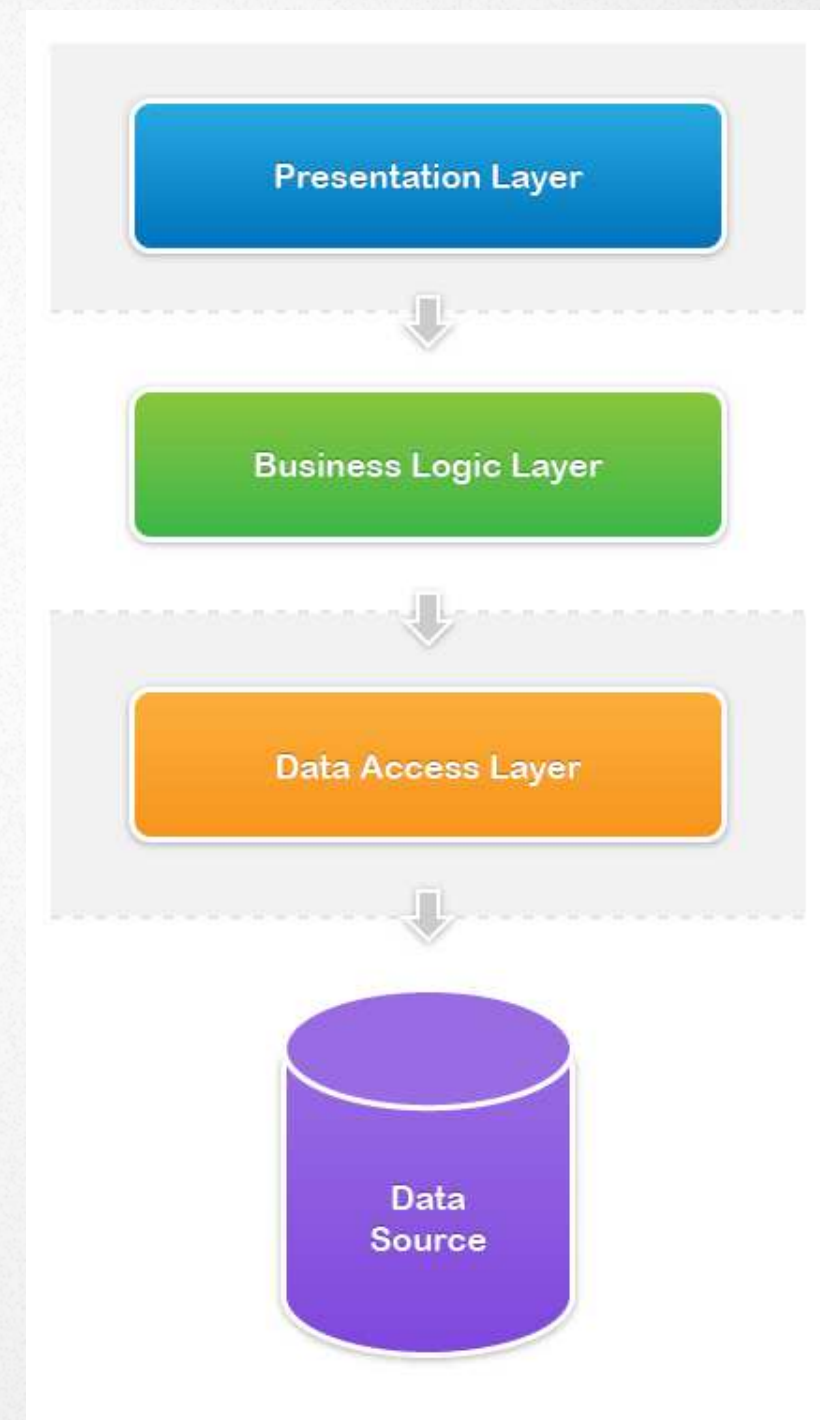


What Do I Test?



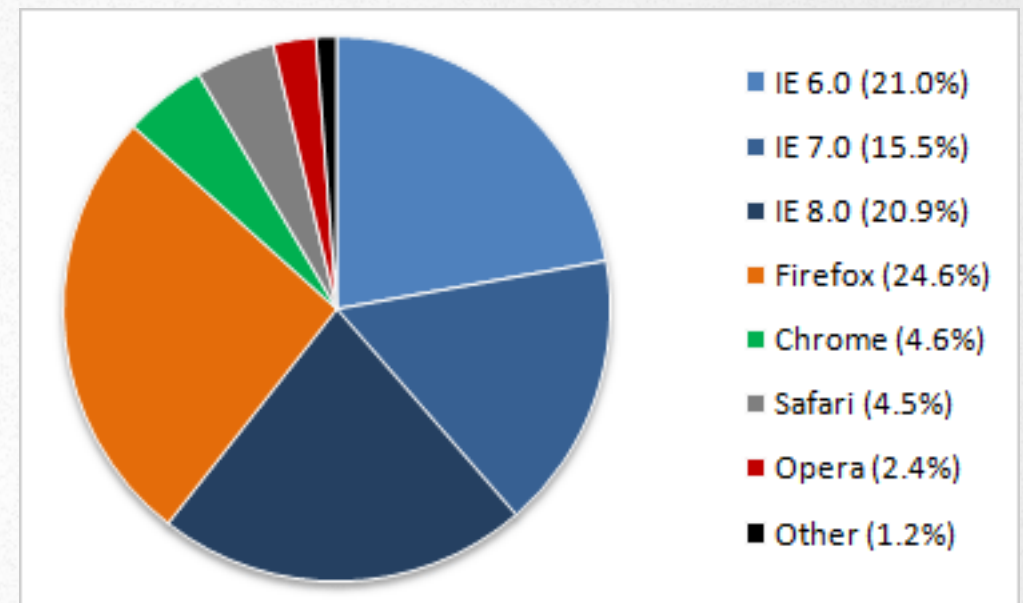
Typical Enterprise App

- . N-Tier apps
- . Strongly typed languages
- . Only one environment
- . QA Team for releases



Unit Tests on the Web

- . Spaghetti code
- . Poorly defined “units”
- . Test against many browsers and platforms

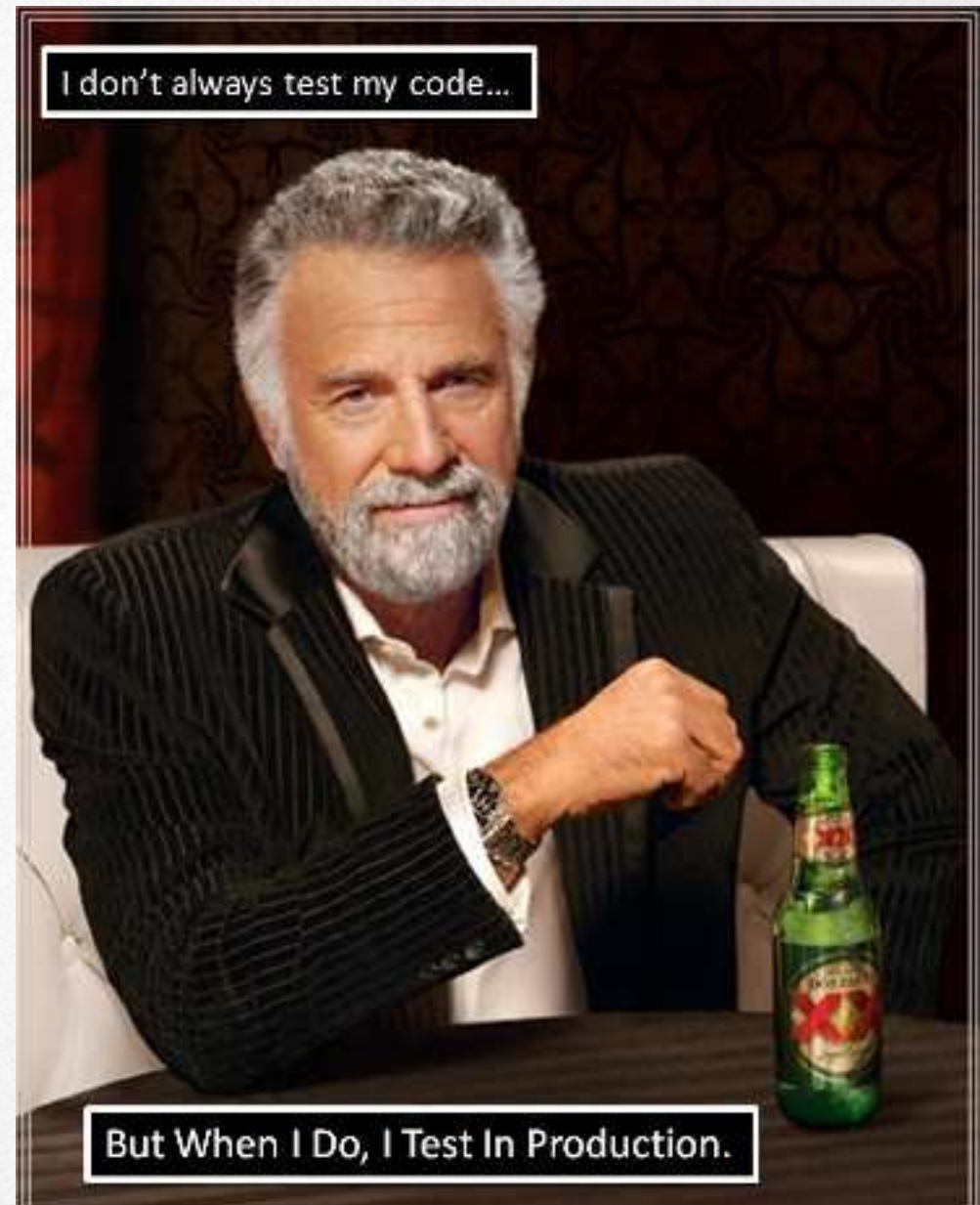


Automation



Automation or Bust

- . JavaScript is not compiled
- . Run tests regularly
 - . Continuous Integration
 - . Version control hooks
 - . Nightly builds, etc.



Tools



Choosing the Right Tools

- . Approved software/tools
- . No “one size fits all” tool
- . Compatibility with existing systems and tools



Sample App



Syntax



JavaScript Syntax

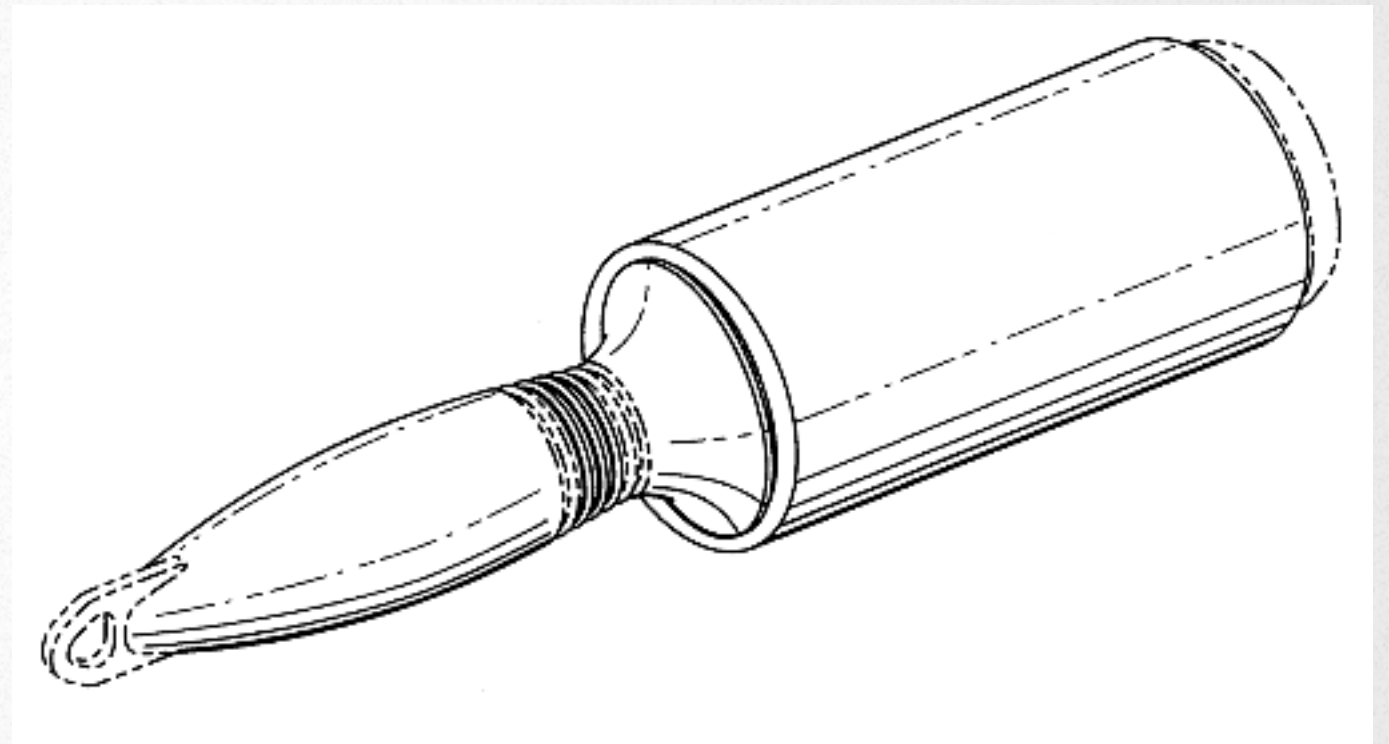
- . Prevent runtime bugs
- . Enforce best practices
- . JSLint, JSHint, etc.
- . PhantomLint...



Example: LintRoller

- . Runs on Node.js
- . Installed via NPM
- . Currently supports:
 - . JSLint
 - . JSHint
 - . Esprima

. <http://arthurakay.github.com/LintRoller/>



Challenges

- . Software approval
- . Version control features

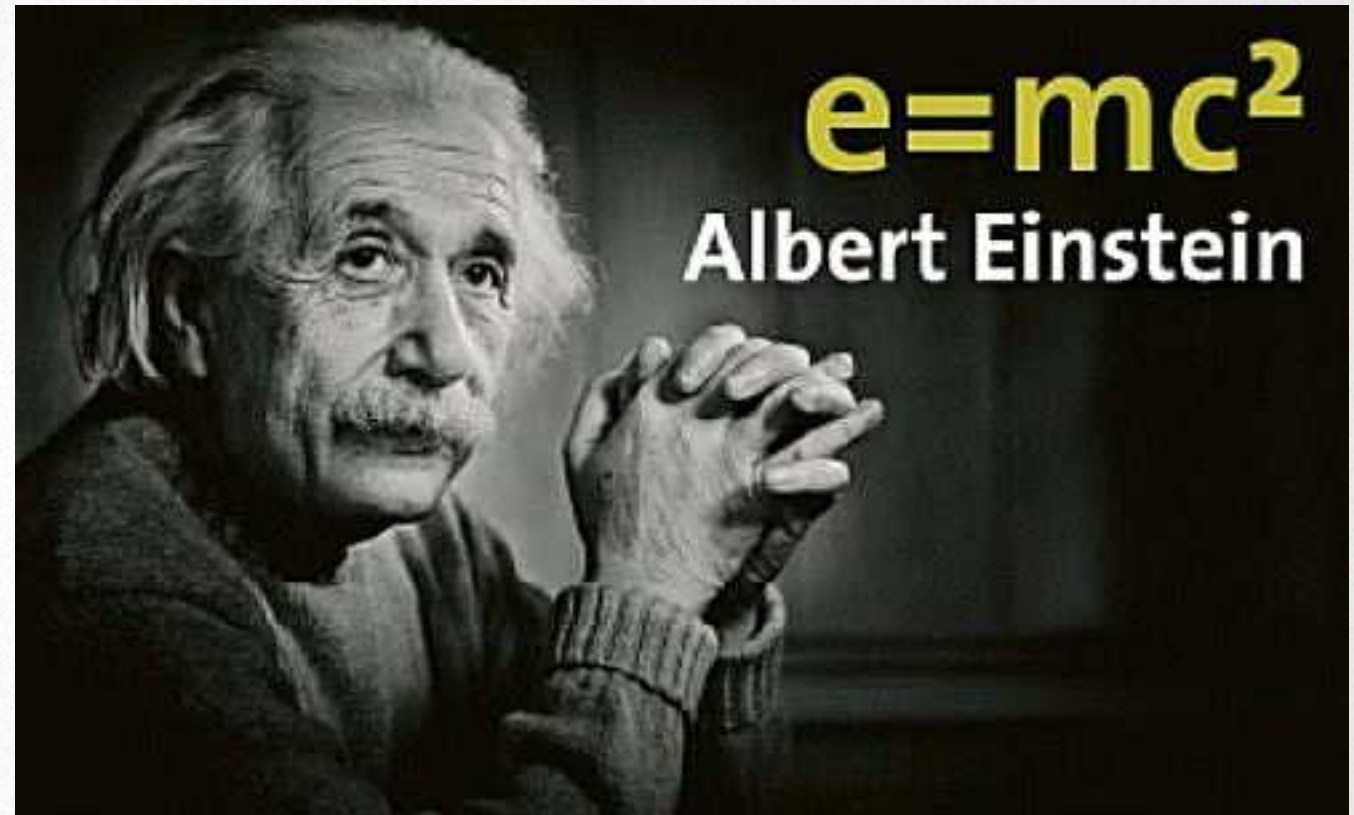


Logic Tests (Unit)



Logic Tests (Unit)

- . Assertions & Expectations
- . Objective in nature
- . Think of mathematical proofs
- . What do I test?



Example: Jasmine

- . Very readable API
- . Tests run in browser
- . Plugins available for IDEs
 - . JSTestDriver
- . PhantomJS



<http://pivotal.github.com/jasmine/>



Challenges

- . Deciding what to test
- . Having “testable” code
- . Mocking data / APIs
- . Avoiding bugs in the tests



Integration Tests (UI)



Integration Tests (UI)

- . Validate appearance or behavior
- . Subjective in nature
- . Simulate real-world use
- . Isolating widgets, QA



Example: Siesta

- . Run in browser or IDE
- . Automation tools
- . Easy integration with Sencha frameworks
- . click, type, drag, tap, etc



<http://www.bryntum.com/products/siesta/>



2 aspects of UI testing

- . Isolated UI Unit (preferred)
 - . + Unit test approach, test 1 Component API at a time
 - . + Safe to refactor the Component internals
 - . + Develop TDD style
- . Black box (“QA-testing”, Selenium)
 - . + Does my application really work?
 - . - More complex
 - . - Doesn't provide exact failure context



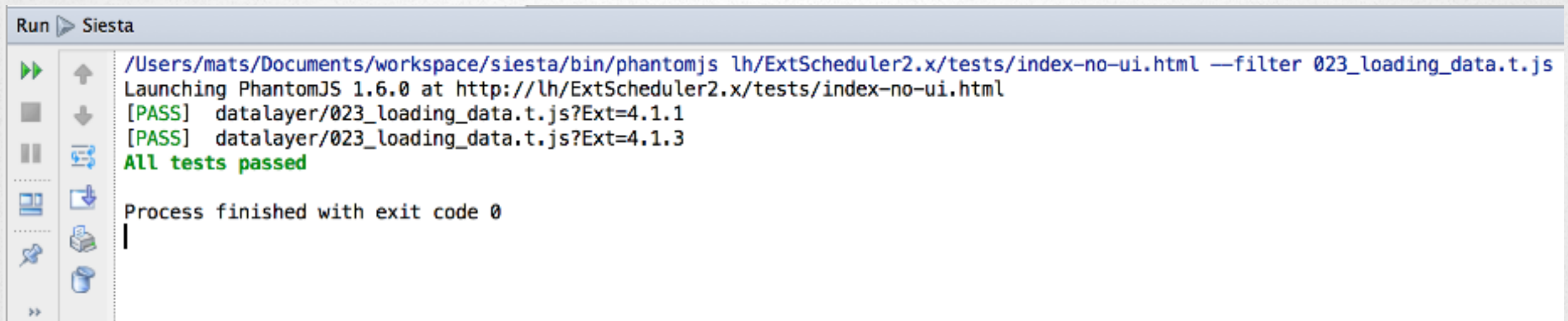
Siesta speaks 'Sencha'

- . Knows about Sencha components, stores, CQ etc.
- . waitForRowsVisible(myGrid)
- . waitForStoresToLoad(userStore, adminStore, callback)
- . click('button[cls=foo]')
- . type(textfield, 'Hello World')
- . willFireNTimes(myButton, 1, 'click')



Run tests inside your IDE

- . 1. Pick awesome keyboard shortcut
- . 2. Map to Phantom JS Siesta launcher
- . 3. Enjoy!



The screenshot shows a terminal window titled "Run ▶ Siesta". On the left is a vertical toolbar with icons for running, debugging, and other IDE functions. The terminal text displays the command to launch PhantomJS with specific filters, followed by two "[PASS]" messages for individual test cases, a green "All tests passed" message, and a final message stating "Process finished with exit code 0".

```
Run ▶ Siesta
/Users/mats/Documents/workspace/siesta/bin/phantomjs lh/ExtScheduler2.x/tests/index-no-ui.html --filter 023_loading_data.t.js
Launching PhantomJS 1.6.0 at http://lh/ExtScheduler2.x/tests/index-no-ui.html
[PASS] datalayer/023_loading_data.t.js?Ext=4.1.1
[PASS] datalayer/023_loading_data.t.js?Ext=4.1.3
All tests passed
Process finished with exit code 0
```









UI test considerations

- . UI tests are slower, more fragile
- . Tests should be run on many browsers
- . Automation is a must
- . Simulate wisely, consider other options?



Testing Element Pinch

Name									
	0:00	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00
 Employee #0 Web Developer									
 Employee #1 Web Developer									
 Employee #2 Web Developer			 Some Task						
 Employee #3 Web Developer									
 Employee #4 Web Developer									



Options

- . Figure out how to correctly simulate multi-finger pinch (hard)
- . or
- . Simply mock the events and call methods directly (easy)

```
1 Ext.define('Sch.feature.EventPinch', {  
2  
3     constructor: function () {  
4         this.scheduler.on({  
5             pinchstart : this.onPinchStart,  
6             pinch      : this.onPinch,  
7             pinchend   : this.onPinchEnd,  
8             scope      : this  
9         });  
10    },  
11  
12    onPinchStart: function (e) {  
13    },  
14  
15    onPinch: function (e) {  
16    },  
17  
18    onPinchEnd: function (e) {  
19    }  
20 });  
21
```


Example: Siesta



Resources



Resources

- . Sencha Blog: Automating Unit Tests (8/2/12)
 - . <http://www.sencha.com/blog/automating-unit-tests>
- . Sencha Blog: UI Testing a Sencha App (1/13)
 - . <http://www.sencha.com/blog/ui-testing-a-sencha-app>
- . Sample app:
 - . <https://github.com/SenchaProSvcs/UnitTestDemo>
- . Siesta
 - . <http://bryntum.com/products/siesta/>



Resources

- . *Test Driven JavaScript Development* - Christian Johansen
- . “Writing Testable JavaScript” - Rebecca Murphey
 - . <http://www.youtube.com/watch?v=OzjogCFO4Zo>
 - . <https://speakerdeck.com/rmurphey/writing-testable-javascript>
- . “Behavior Driven Testing with Jasmine” - Davis Frank
 - . <http://docs.sencha.com/ext-js/4-1/#!/video/18100173>
- . “JavaScript Code Analysis with Esprima” - Ariya Hidayat
 - . <http://www.youtube.com/watch?v=ACYZFkvq0Sk>
- . “Write Code that Works” - Reid Burke
 - . <http://www.yuiblog.com/blog/2013/01/10/yuiconf-2012-talk-reid-burke-on-write-code-that-works/>

Advice

Decide what tests should accomplish

Don't expect 100% coverage

Tests should compliment process



Q & A



