



*Cappuccino*

+

**testing**

#CappCon2016



# Alexandre Wilhelm

Engineering Degree in  
Embedded Software and Mobile  
Computing (UTBM - 2012)

Software Engineer (Mountain  
View - CA)

Member of the Cappuccino Core  
Team

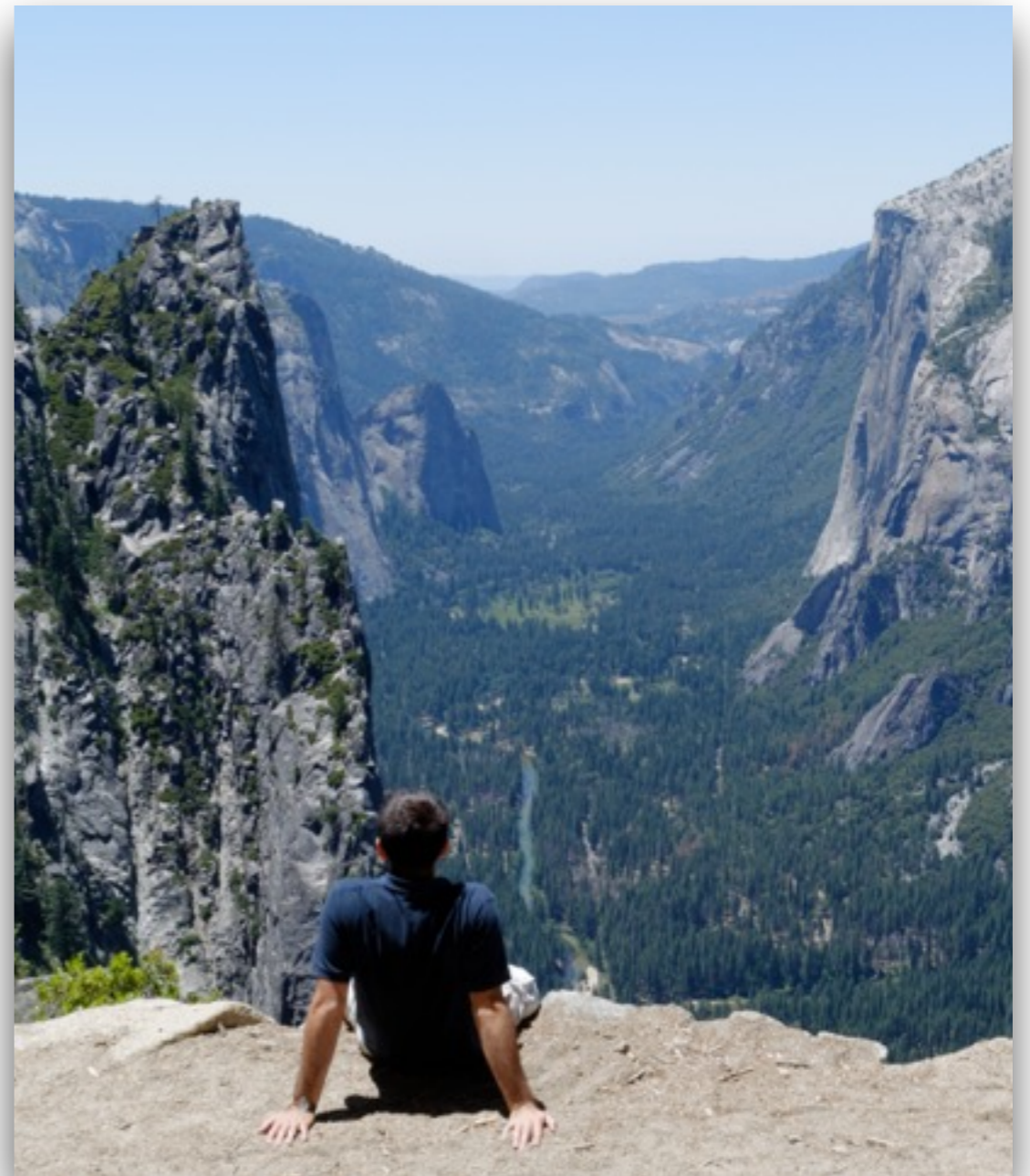
Backpacking during my free  
time



@WilhelmAlex



@Dogild





# what do you need to test a Cappuccino Application?

An unit testing framework

An integration/functional testing framework

Possibility to test the DOM in your tests

Possibility to use these frameworks in a CI (Jenkins, TravisCI, CircleCI...)



# summary

OJTest, a collection of unit testing Frameworks

Cucapp, an interface between Cucumber and Cappuccino

Demo (if it works)



# OJTest

Collection of Frameworks for testing a Cappuccino Application

Perfect for unit testing

Tool written in Objective-J

Does not use a web browser, run in a shell with Rhino or Narwhal

Does not work/test DOMElements

Provides the command ojtest

<https://github.com/cappuccino/OJTest>



# installation

OJTest is installed with  
Cappuccino

Installed version is  
linked to the last  
release of Cappuccino

Install OJTest manually  
if you want the latest  
modifications

```
Terminal — -bash — 60x23
[Dogild:OJTest (master)]$ jake install
(in /Users/Dogild/Project/OJTest)
OJTest installation done! Have fun in testing now :)
Build took 390 milliseconds
[Dogild:OJTest (master)]$ ojtest-h
-bash: ojtest-h: command not found
[Dogild:OJTest (master)]$ ojtest -h
Usage: ojtest [OPTIONS] INPUT_FILE ...
Runs the given suite of tests.
Usage: ojtest [OPTIONS] INPUT_FILE:NAME_OF_THE_TEST
Runs only the given test of the given file
  -x: Stop running tests after the first error or failure
  -c: Runs test coverage along with your OJUnit tests.
  -t THRESHOLD: Set the threshold for test coverage to the sp
ecified percentage. Implies -c. Default: 0.8
  -l LEVEL: Set the minimum log level. (fatal, error, warn, i
nfo, debug, trace)
  -s: Run with OJSpec enabled (experimental). When this is st
able, OJSpec will be enabled by default.
  -p: Run with Rhino-enabled parallel test running on 10 thre
ads.
  -h --help: displays usage information (final option)
[Dogild:OJTest (master)]$
```



# OJUnit

A xUnit framework

Perfect tool to do assertions

Can be used to test public or private methods

Static/instance setUp and tearDown

Can be used to test uncompiled code

Lot of examples in Cappuccino itself



# OJMoq

A mock framework

Perfect tool to mock an external resource, for instance a ReST server

Perfect tool to test the behavior of a library using datasource or delegate





“Demo”

–Hope it’s going to work...



# cucapp

Interface between Cucumber and Cappuccino (third-party library)

Perfect tool to create functional tests

Tool written in Ruby

Works with Chrome/Firefox/Opera/PhantomJS

Possibility to do black-box testing or white-box testing

Possibility to record a set of events and play them again and again

CLI in the javascript console for debugging

<https://github.com/cappuccino/cucapp>



# cucumber

Cucumber is for Behavior-Driven Development

Language Gherkin is used to define test cases

Provides a huge amount of features/plugins

Easy to read and use

Large community

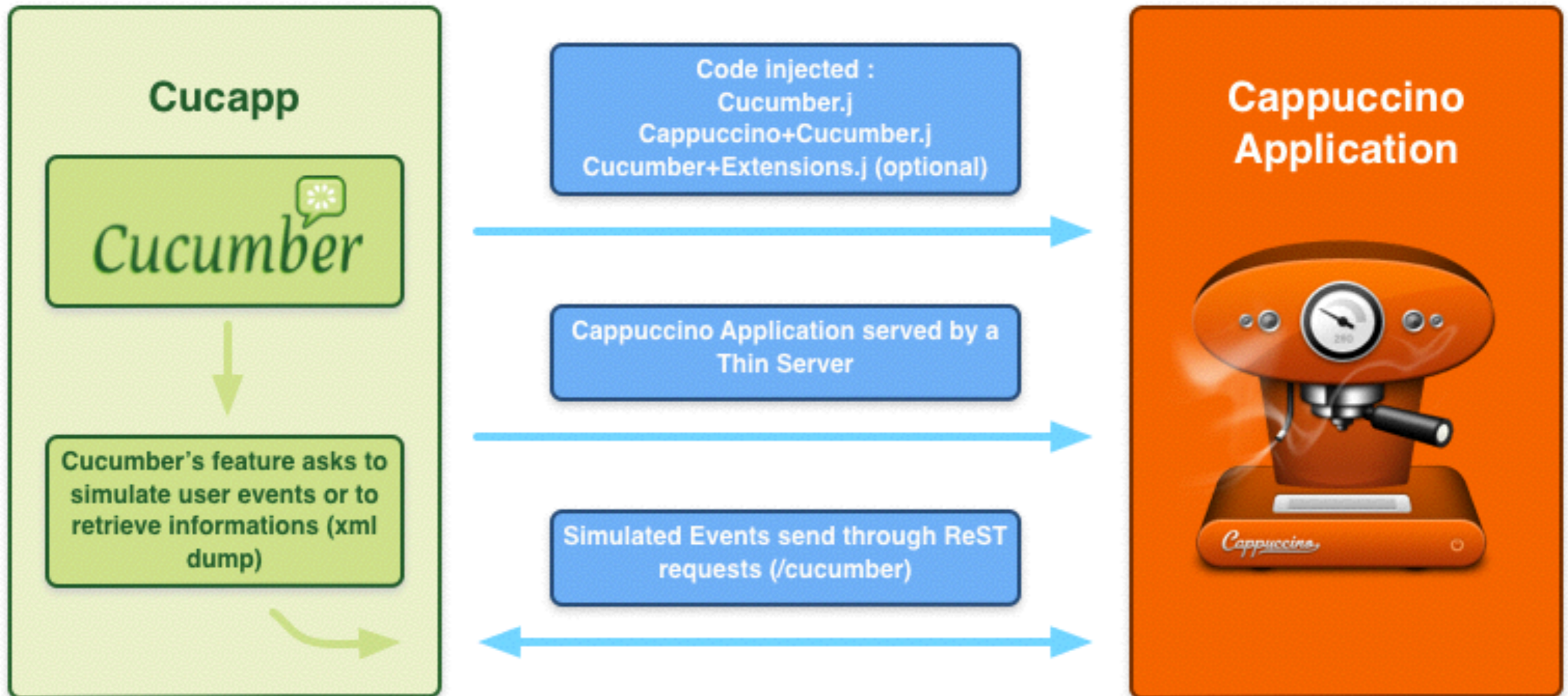
```
Scenario: eat 5 out of 12
  Given there are 12 cucumbers
  When I eat 5 cucumbers
  Then I should have 7 cucumbers

Scenario: eat 5 out of 20
  Given there are 20 cucumbers
  When I eat 5 cucumbers
  Then I should have 15 cucumbers
```

cucumber 



# at a glance





# installation

Cucapp does not come with Cappuccino

Ruby needs to be installed

Clone the Cucapp repository and do `jake install`

Install the gems Cucumber - Nokogiri - JSON - Thin -  
Watir-Webdriver

Install the command `cucapp` on your system



# cucapp philosophy

Cucapp provides a set of abstract steps to simulate any **behavior** an user could do :

- Click on the mouse
- Hit keys on the keyboard
- Drag and drop
- Scrolling

Cucapp creates new DOMEvents and dispatches them to the main document of the Cappuccino Application. The chain events of Cappuccino is preserved and Cucapp covers the exact same amount of code as a real DOMEvent.

Cucapp provides a mechanism to **retrieve** informations from the Cappuccino Application. For instance value of a CPTextField, the date of a CPDatePicker etc...

Cucapp provides an easy mechanism to add new **behavior** or new methods to **retrieve** information

Add the ivar cucappIdentifier to the class CPResponder (unique identifier for the test context)

Water WebDriver is used to control the Browser

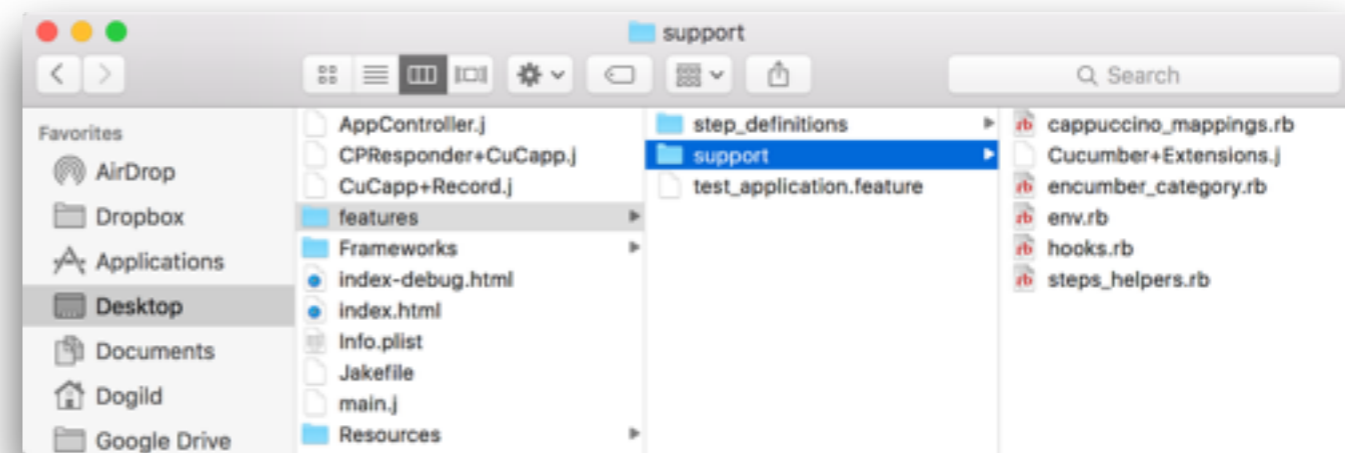


# cucapp Command

Initialize a cappuccino application with cucapp

Add a command `jake cucumber-test` to your Jakefile

Add the needed files to create new tests



```
186 task ("cucumber-test", function()
187 {
188     var SYSTEM = require("system");
189
190     OS.system("ln -s " + SYSTEM.prefix + "/packages/cucapp/Cucapp Cucapp")
191     var code = OS.system("cucumber");
192     OS.system("rm -f Cucapp; rm -f cucumber.html")
193     OS.exit(code);
194 });
```



# cucapp CLI

Cucapp CLI in the javascript console

Possibility to search CProspender's through the javascript console

Useful for the QA team for getting information about CProspender data

Useful for the development team for debugging





“Demo”

–Hope it’s going to work again...



# jakefile enhancements

```
72 task("test", ["test-only"]);
73
74 task("test-only", function()
75 {
76     ENV["OBJJ_INCLUDE_PATHS"] = "Frameworks";
77
78     var tests = new FileList('Test/*Test.j'),
79         cmd = ["ojtest"].concat(tests.items()),
80         cmdString = cmd.map(OS.enquote).join(" "),
81         code = OS.system(cmdString),
82         manualTests = FILE.list('Test/Manual');
83
84     if (code !== 0)
85         OS.exit(code);
86
87     manualTests.forEach(function(manualTest)
88     {
89         if (manualTest.indexOf("Test") !== -1)
90         {
91             code = OS.system("cd Test/Manual/" + manualTest + "; capp gen -fl -F GraphCappuccino . --force; jake cucumber-test")
92
93             if (code !== 0)
94                 OS.exit(code);
95         }
96     });
97 });
```



# the future of Cucapp

Create better test results with screen shots or videos

Create an automated generator of documentation with the positive test cases (videos recording)

Integrate Cucapp in the Cappuccino project (currently working on that)



# Questions ?

(And sorry for my poor english...)

–Alexandre Wilhelm



@WilhelmAlex



@Dogild