TestRunner Design.graffle

June 04, 2009

Wireframes

Browser Title

http://myapp.com   **Google**

Loading Targets...

⊙**TestRunner**   ☐ Continuous Integration   Run Tests

Empty

Page 3 of 16
Modified Thu Jun 04 2009 – Prepared by Charles Jolley

Browser Title

http://myapp.com

Google

▼ **APPS**

  apples

  bananas

▼ **FRAMEWORKS**

  ▼ core_bananas

      oranges

  pears

▼ **SPROUTCORE**

  tests

  docs

  runtime

  foundation

Select a Target

TestRunner

☐ Continuous Integration

Run Tests

Browser Title

http://myapp.com

Google

▼ **APPS**

　apples

　bananas

▼ **FRAMEWORKS**

　▼ core_bananas

　　oranges

　pears

▼ **SPROUTCORE**

　tests

　docs

　runtime

　foundation

Loading Tests...

TestRunner

☐ Continuous Integration

Run Tests

List Loading

http://myapp.com

Google

▼ APPS

    apples

    bananas

▼ FRAMEWORKS

  ▼ core_bananas

      oranges

    pears

▼ SPROUTCORE

  tests

  docs

  runtime

  foundation

▼ **core**
not yet run — Default state

▼ **controllers**
not yet run   Run — Show on hover

  ▼ **foo**
  subtest has warnings — Group with child status

    foo/actionName
    not yet run — When clicked

    foo/propertyName
    23 tests run, 230 assertions OK, 3 warnings — Warning result

▼ **models**
running... — When running

foo
23 tests run, 233 of 233 assertions OK — Success result

bar
23 tests run, 200 assertions OK, 10 failures, 5 errors, 3 warnings — Failure result

TestRunner

☐ Continuous Integration

Run Tests

List

http://myapp.com

Google

APPS

🔲 **apples**

🔲 bananas

FRAMEWORKS

🔲 core_bananas

🔲 oranges

🔲 pears

SPROUTCORE

🔲 tests

🔲 docs

🔲 runtime

🔲 foundation

All Tests

controllers/foo/actionName

🟢 **TestRunner**

☐ Continuous Integration

Architecture

Target: SC.Record

name: String
parent: Target
children: [Target]
tests: [Test]
isExpanded: Boolean
displayName: String
kind: String

Test: SC.Record

filename: String
url: String
state: STATE
report: String (HTML)
tests: Number
assertions: Number
ok: Number
failures: Number
errors: Number
warnings: Number

Test States
READY
WAITING
RUNNING
PASS
FAIL
WARN
CANCELLED

(see Test Statechart)

Model

[Target]

targetsController:
SC.ArrayController

content: [Targets]

targetRoot

B

sourceController:
SC.TreeController

content: TargetRoot

selection

didSelectTarget()

mainController

state: STATE

selectTarget()

selectTest()

back()

gotoRoute()

mainController States
EMPTY
LIST
DETAIL

(see Main Statechart)

testRunner States
READY
RUNNING

(see Test Runner Statechart)

testRunner

state: STATE

test: Test (curtest)

hasTests: Boolean

run(test)

cancel(test)

State Charts

selectTarget/**set(target)**
selectTest/**set(test)**

LIST

selectTarget [target != null]

didLoadTargets
[hasTargets=YES,
target != null,
test = null]

selectTarget [target=null]

selectTarget [target != null]

selectTest

START
enter/**loadTargets()**

didLoadTargets
[hasTargets=YES,
target = null]

READY
selectTarget [target=null]

didLoadTargets
[hasTargets=NO]

didLoadTargets
[hasTargets=YES,
target != null,
test != null]

NO_TARGETS

selectTarget [target != null]
selectTest [test=null]
back

selectTarget [target=null]

DETAIL

EVENTS
selectTarget
selectTest
back
didLoadTargets

selectTest [test != null]

LIST

IDLE

RUNNING

EVENTS
selectTarget
selectTest
back
didLoadTargets

READY
stop/**ignore()**
targetDidChange/**ignore()**
ok/**ignore()**
warn/**ignore()**
error/**ignore()**
failure/**ignore()**

run

RUNNING
enter/**set(curtest), set(testState=RUNNING), loadTest()**
exit/**set(testState), unloadTest()**

run [test = curtest]/**ignore()**
ok [test != curtest]/**ignore()**
error [test != curtest]/**ignore()**
warn [test != curtest]/**ignore()**
failure [test != curtest]/**ignore()**

run [test != curtest]/**addTest()**
stop [test != curtest]/**removeTest**()

SUCCESS
ok/**ignore()**

warn [test=curtest]

failure [test=curtest]
error [test=curtest]

WARN
warn/**ignore()**
ok/**ignore()**

failure [test=curtest]
error [test=curtest]

FAIL
warn/**ignore()**
ok/**ignore()**
failure/**ignore()**
error/**ignore()**

done [hasTests=NO]
stop [test=curtest, hasTests=NO]
targetDidChange/**removeAllTests()**

done [hasTests=YES]
stop [test = curtest, hasTests = YES]

EVENTS	run(test)	stop(test)	done	targetDidChange
ok	warn	failure	error	

READY

didSchedule

CANCELLED

didSchedule

WAITING

didBegin

didStop

RUNNING

didPass

didFail   didWarn

DONE

FAILED

PASSED

WARNED

didSchedule