Testing Seaside Components

C. David Shaffer
Department of Mathematics and Computer Science
Westminster College

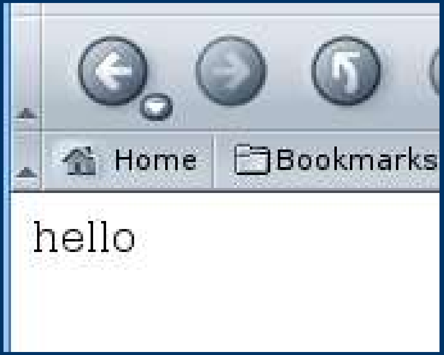
Features

- Tests run on server = access to component being tested
- Uses Smalltalk debugger
- Web test runner
- Available for Squeak and VisualWorks (thanks to Michel Bany for VW port!)

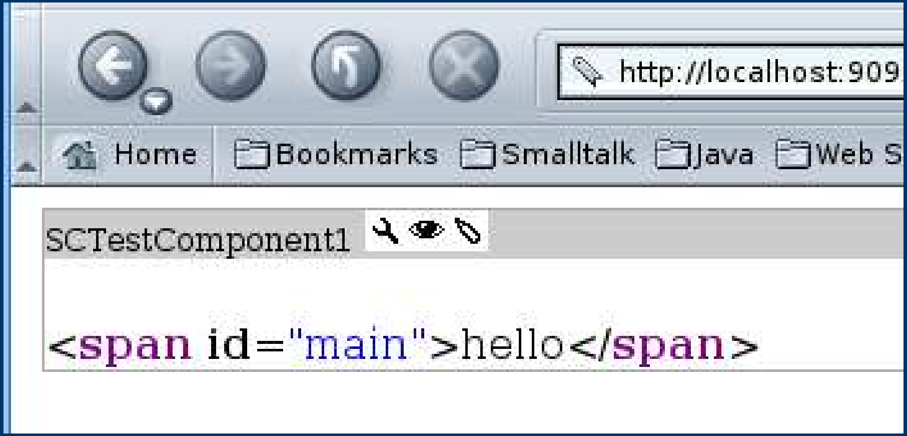
First example

Class SCTestComponent1

```
renderContentOn: html
  html cssId: 'main'.
  html span: 'hello'
```



hello



SCTestComponent1

```
<span id="main">hello</span>
```

First example

```
SCComponentTestCase subclass: #SCSampleComponentTest
   instanceVariableNames: ''
   classVariableNames: ''
   poolDictionaries: ''
   category: 'SeasideTesting-Examples'
```

First example

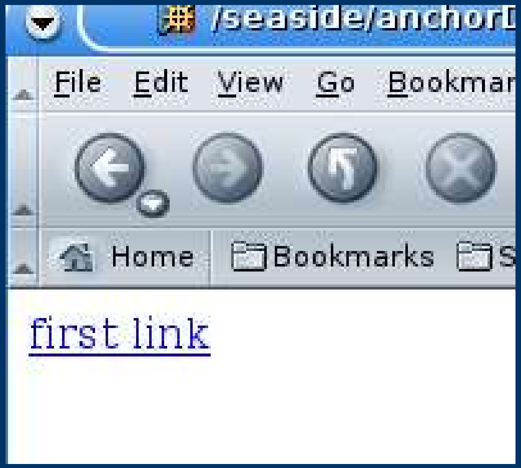
```
testComponent1
   self newApplicationWithRootClass: SCTestComponent1.
   self establishSession.
   self assert: (self lastResponse
      stringWithId: 'main') = 'hello'
```

Following anchors

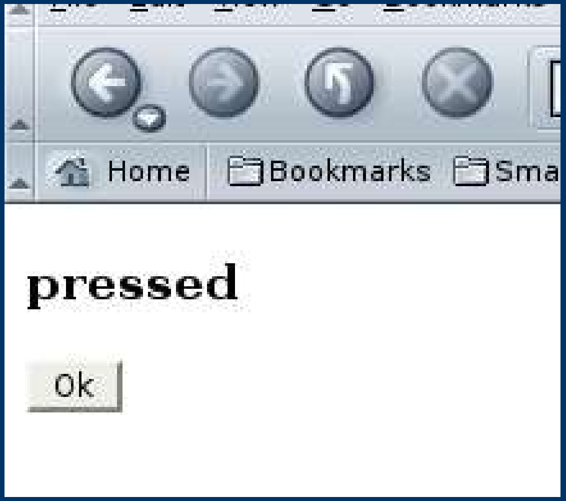
Class SCAnchorDemo

```
renderContentOn: html
  html cssId: 'first'.
  html
    anchorWithAction: [self firstPressed]
    text: 'first link'.



firstPressed
  self inform: 'pressed'
```

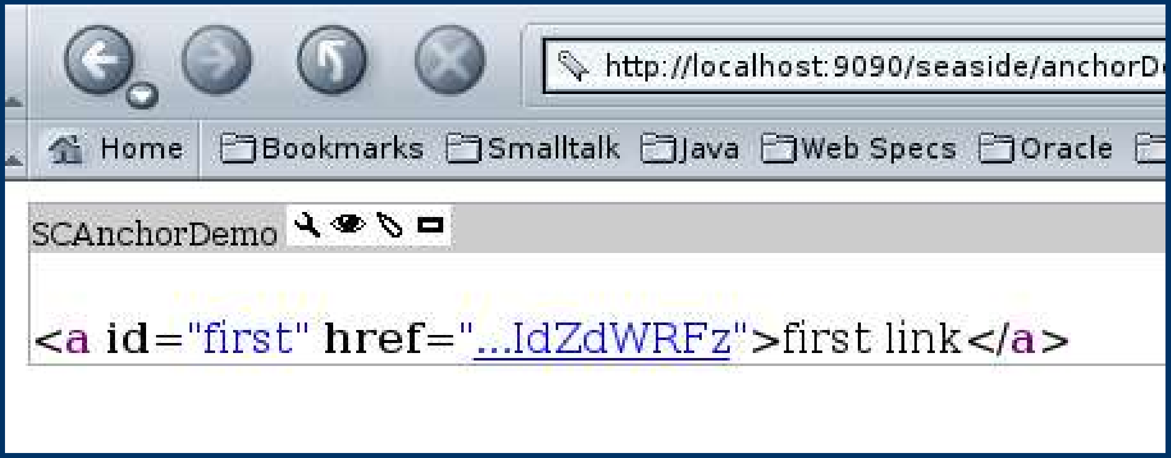
SCAnchorDemo



first link



pressed


http://localhost:9090/seaside/anchorD
Home Bookmarks Smalltalk Java Web Specs Oracle
SCAnchorDemo

```
<a id="first" href="...IdZdWRFz">first link</a>
```

Following anchors

```
testAnchor
   self newApplicationWithRootClass: SCAnchorDemo.
   self establishSession.
   self followAnchor: (self lastResponse
      anchorWithId: 'first').

   self assert: (self lastResponse
      containsString: 'pressed')

   “alternatively”
   self assert: (self lastResponse
      elementsNamed: 'h3') first
         contentString = 'pressed'
```

lastResponse

Response parsed --> XML DOM (XMLElement)

Wrapped in SCSeasideResponse:

- Conveience methods for searching for XML elements by id, class or name (tag)
- Method for wrapping parts in subclasses of SCXMLElementWrapper. For example:
  - SCSubmitButtonHtmlInput
  - SCTextAreaHtmlInput
  - SCSeasideForm
  - SCSeasideAnchor

Finding anchors

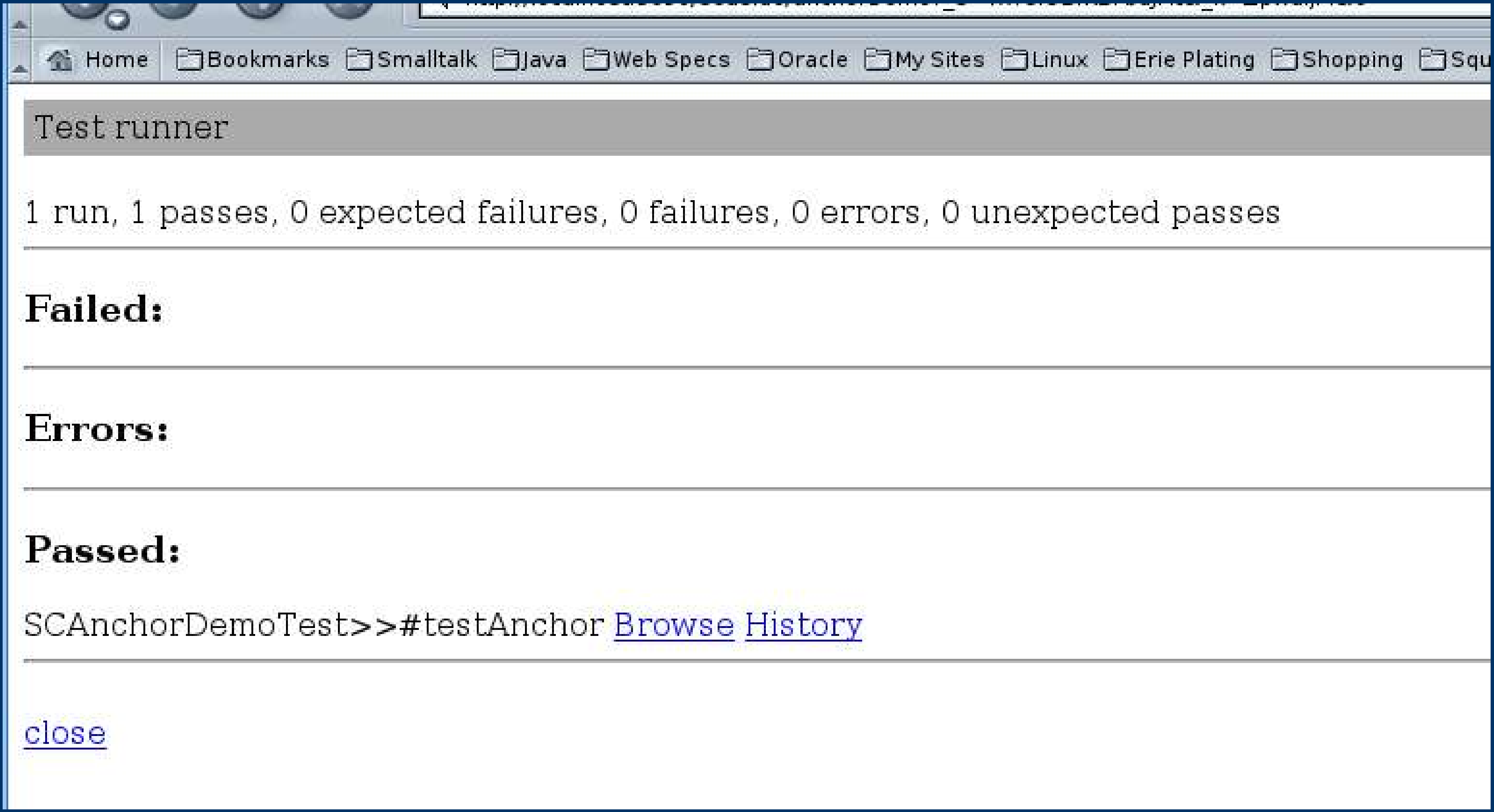
Selected methods of SCSeasideResponse which return SCSeasideAnchor(s)

anchorWithId: -- uses CSS id

anchorWithLabel: -- text inside A tag

anchors – collection of anchors in order of occurrence

Web TestRunner



Test runner

1 run, 1 passes, 0 expected failures, 0 failures, 0 errors, 0 unexpected passes

Failed:

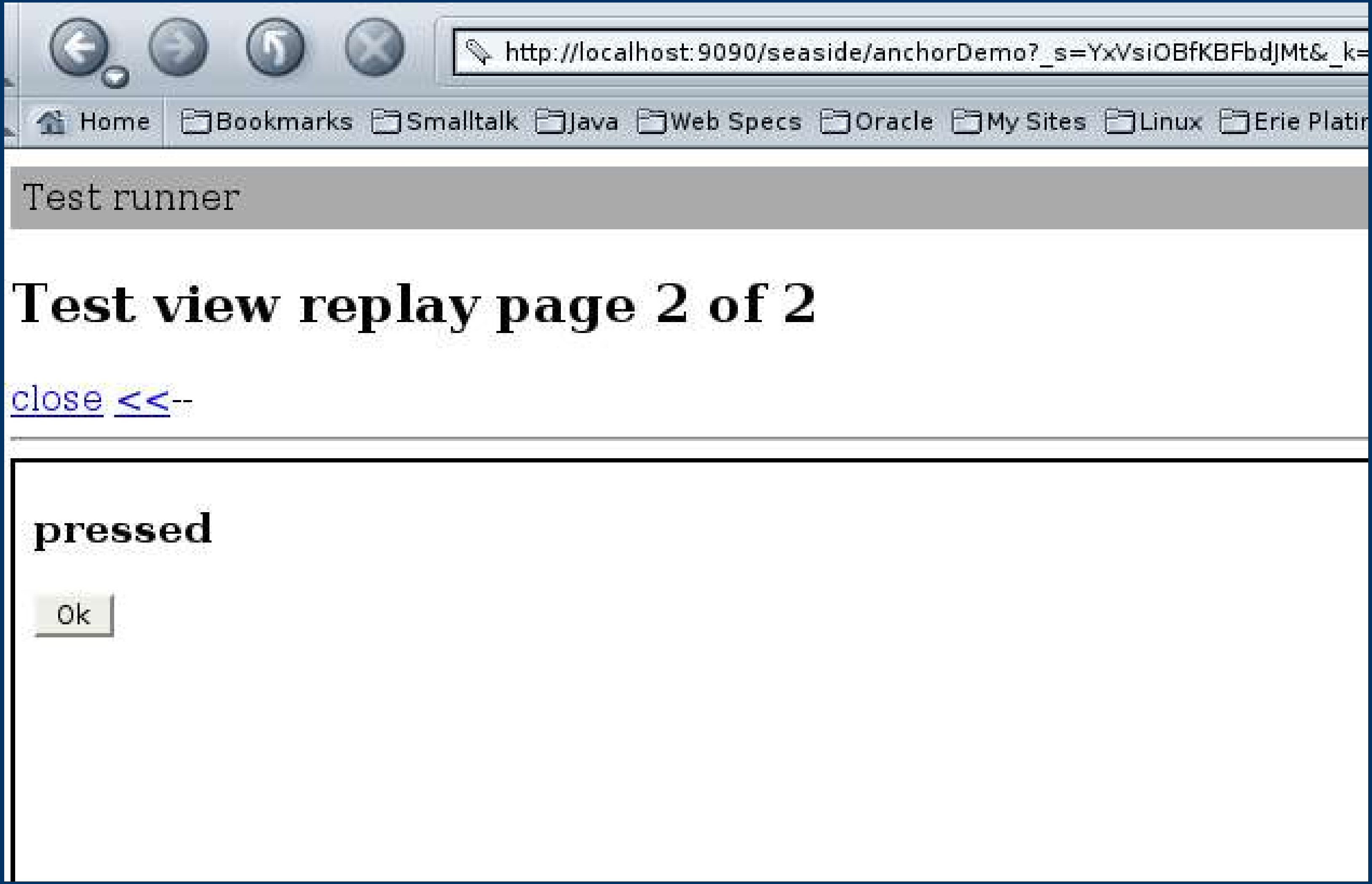
Errors:

Passed:

SCAnchorDemoTest>>#testAnchor Browse History

close

Web TestRunner



Test runner

Test view replay page 2 of 2

close <<--

pressed

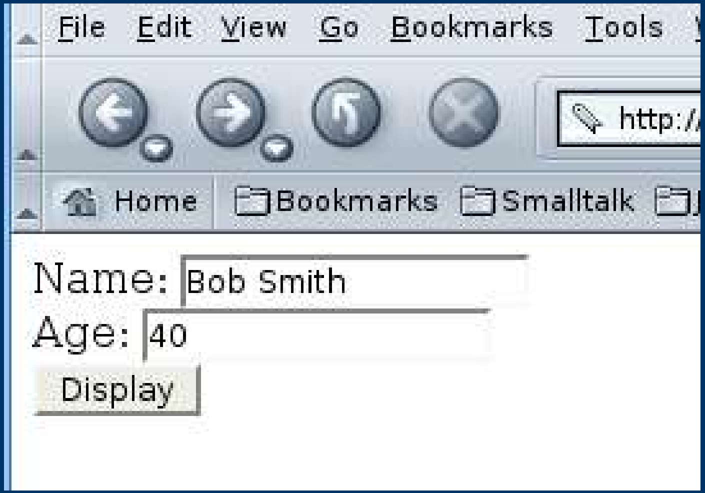
Ok

Forms

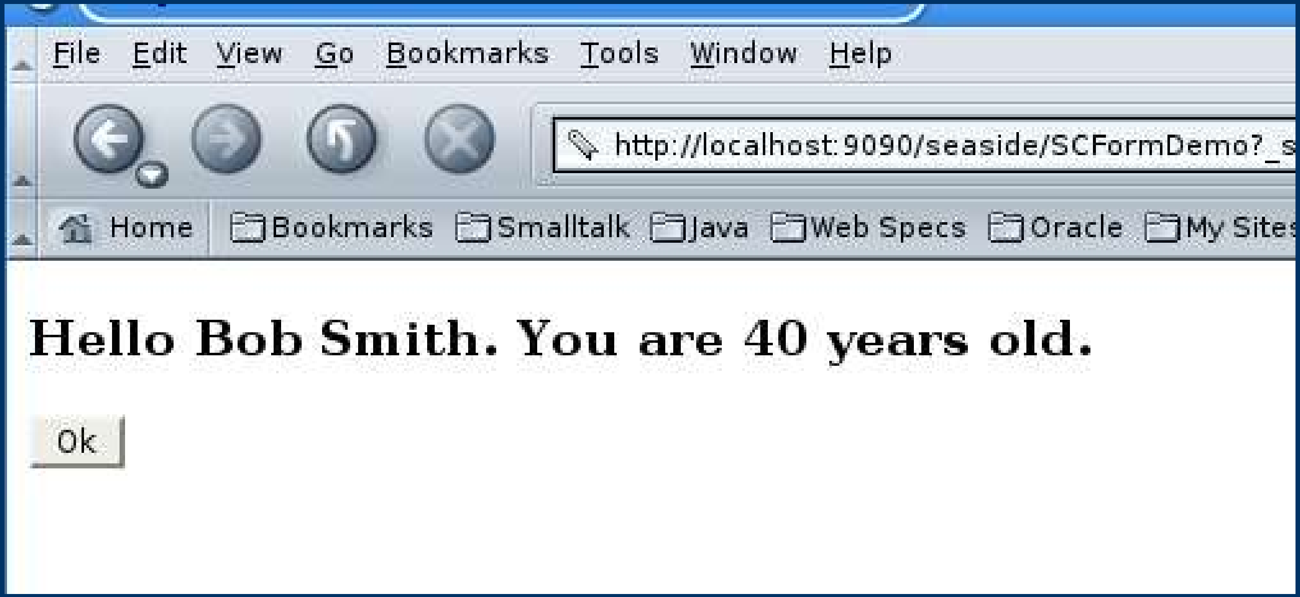
Class SCFormDemo

```
renderContentOn: html
  html form: [
    html text: 'Name: ';
      cssId: 'name';
      textInputOn: #name of: self; br;
      text: 'Age: ';
      cssId: 'age';
      textInputOn: #age of: self; br;
      submitButtonWithAction:
        [self displayInfo]text:'Display']
```

Class SCFormDemo



Name: Bob Smith
Age: 40
Display


File Edit View Go Bookmarks Tools Window Help
Home Bookmarks Smalltalk Java Web Specs Oracle My Sites

http://localhost:9090/seaside/SCFormDemo?_s

Hello Bob Smith. You are 40 years old.

Ok

Forms

```
testDisplay
   | form |
   self newApplicationWithRootClass: SCFormDemo.
   self establishSession.
   form := self lastResponse forms first.
   form textInputWithId: 'name' value: 'Bob Smith'.
   form textInputWithId: 'age' value: '40'.
   self
      submitForm: form
      pressingButton: form buttons first.

   self assert: (self lastResponse
         elementsNamed: 'h3') first
            contentString =
            'Hello Bob Smith.  You are 40 years old.'
```

What do we test?

Seaside component more than just visual display

- State: often tests of state are less brittle that tests of displayed content
- “answer”: components that provide a Seaside answer
- Callbacks: components that provide hooks

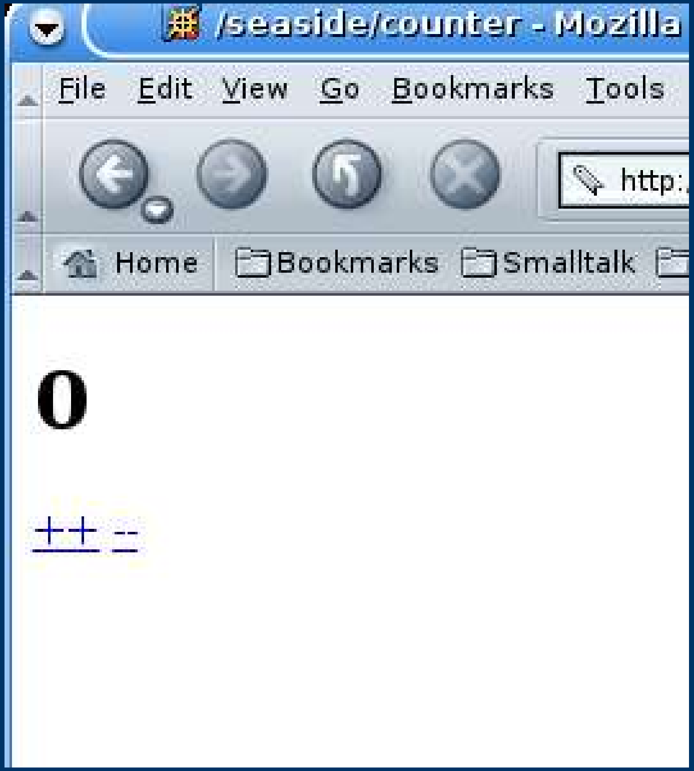
Need access to Seaside component

Testing the Counter (WACounter)

#component answers the instance of the component used to satisfy the **last** request

Seaside may be keeping track of several “versions” of that component registered with #registerObjectForBacktracking:

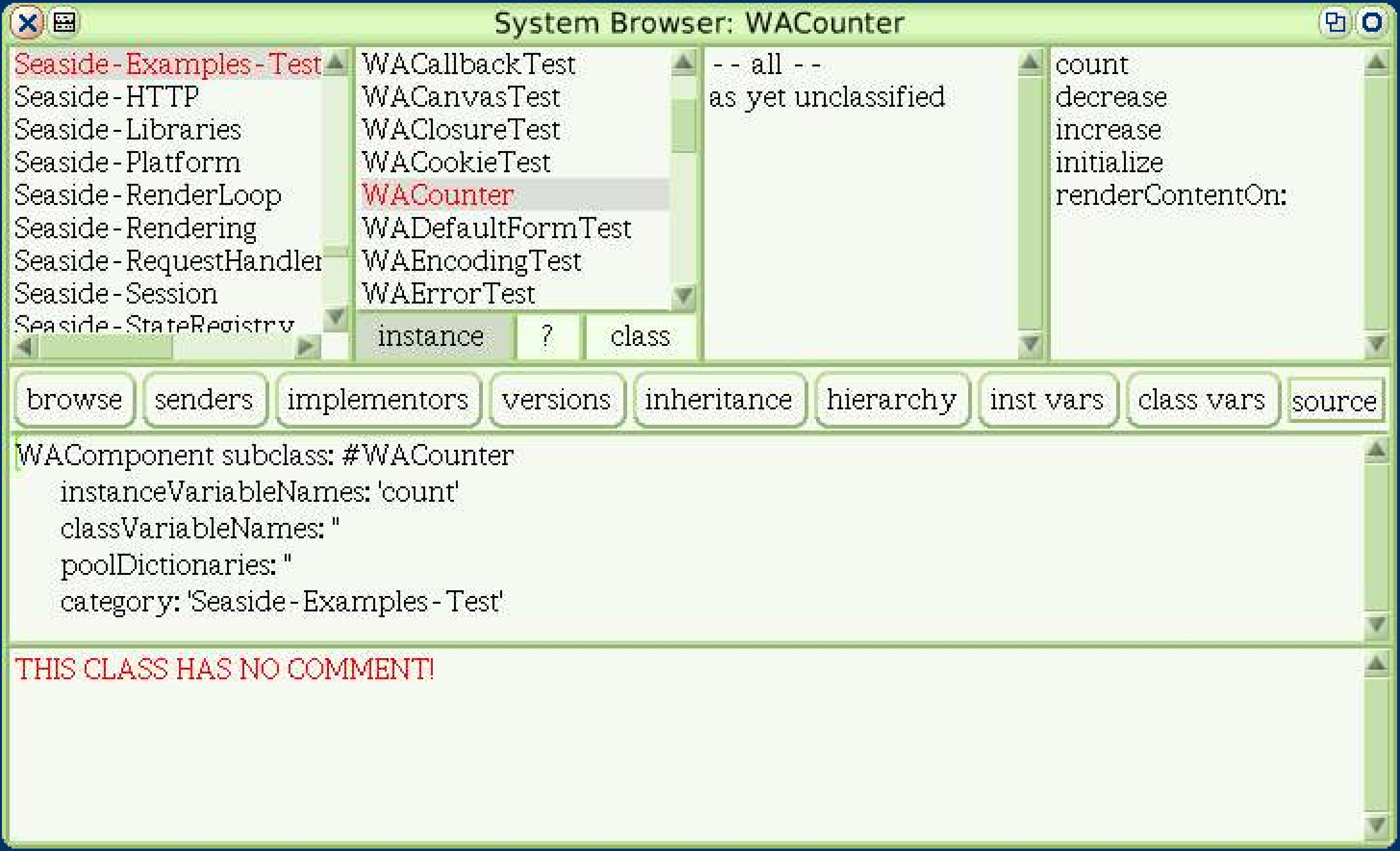
Testing the Counter (WACounter)



0

++ --

Testing the Counter (WACounter)



```
WAComponent subclass: #WACounter
    instanceVariableNames: 'count'
    classVariableNames: ''
    poolDictionaries: ''
    category: 'Seaside-Examples-Test'
```

THIS CLASS HAS NO COMMENT!

Testing the Counter (WACounter)

```
testBack
  self newApplicationWithRootClass: WACounter.
  self establishSession.
  self followAnchor: (self lastResponse
      anchorWithLabel: '++').
  self followAnchor: (self lastResponse
      anchorWithLabel: '++').
```

Testing the Counter (WACounter)

```
testBack
  self newApplicationWithRootClass: WACounter.
  self establishSession.
  self followAnchor: (self lastResponse
      anchorWithLabel: '++').
  self followAnchor: (self lastResponse
      anchorWithLabel: '++').
  self assert: self component count = 2.
```

Testing the Counter (WACounter)

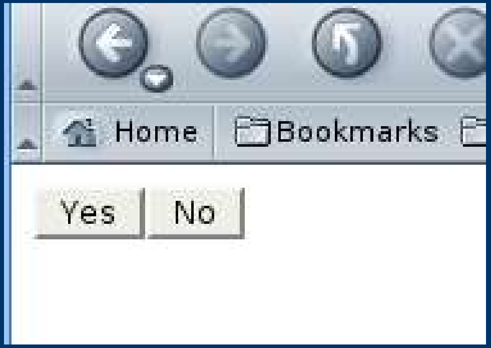
```
testBack
  self newApplicationWithRootClass: WACounter.
  self establishSession.
  self followAnchor: (self lastResponse
      anchorWithLabel: '++').
  self followAnchor: (self lastResponse
      anchorWithLabel: '++').
  self assert: self component count = 2.
  self back.
```

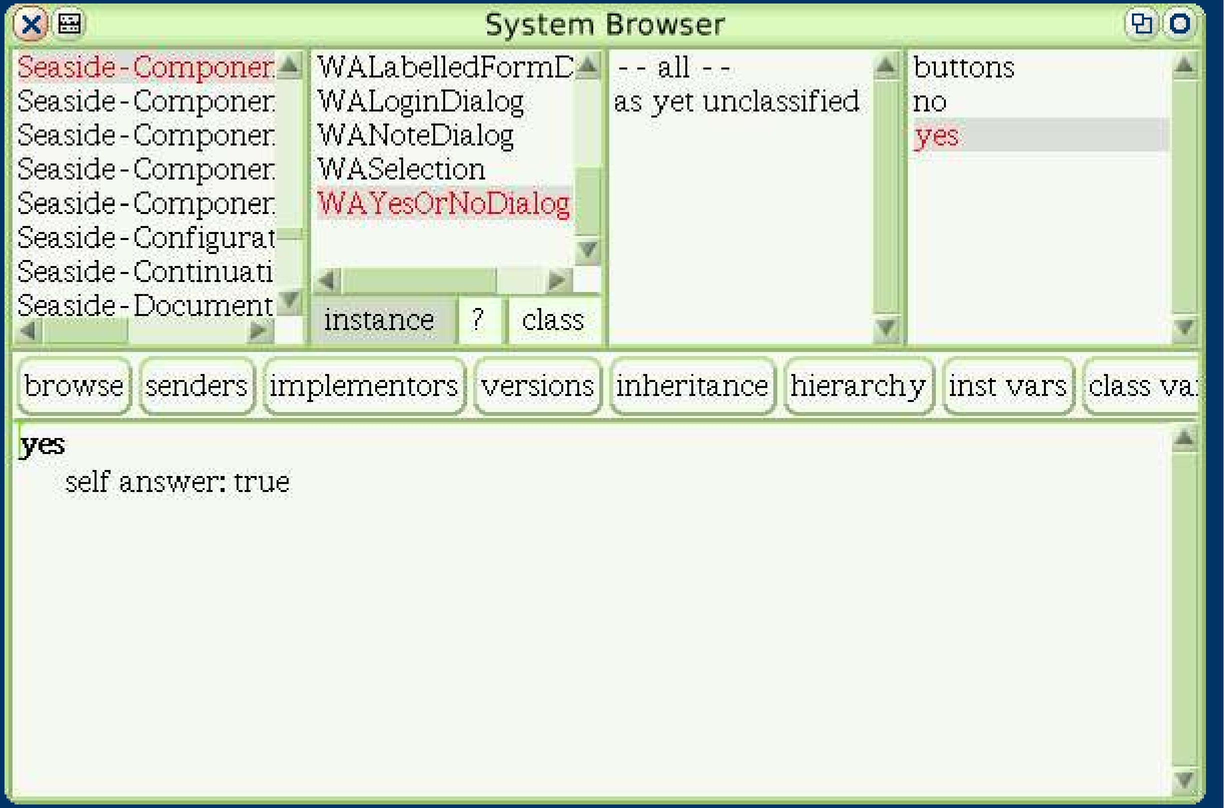
Testing the Counter (WACounter)

```
testBack
  self newApplicationWithRootClass: WACounter.
  self establishSession.
  self followAnchor: (self lastResponse
      anchorWithLabel: '++').
  self followAnchor: (self lastResponse
      anchorWithLabel: '++').
  self assert: self component count = 2.
  self back.
  self
    followAnchor: (self lastResponse
      anchorWithLabel: '++').
  self assert: self component count = 2
```

Detecting answers

Detecting answers





```
yes
	self answer: true
```

Detecting answers

```
testYes
    | form |
    self newApplicationWithRootClass: WAYesOrNoDialog.
    form := self establishSession forms first.
    self
        submitForm: form
        pressingButton: (form buttonWithValue: 'Yes').
    self assert: self answer
```

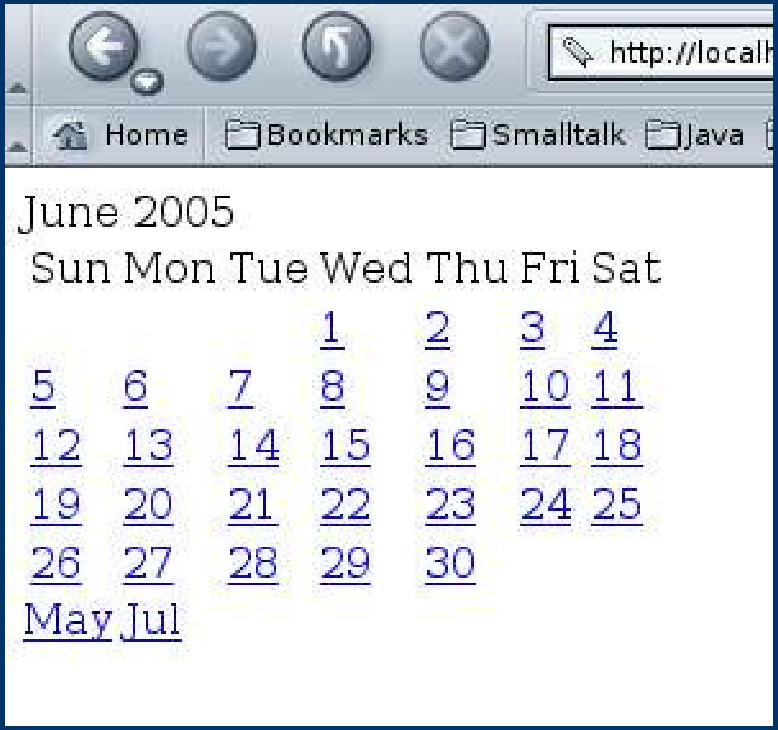
Detecting answers

Related methods:

- answer – components last answer (error if none)
- componentAnswered – boolean, has the component answered?
- componentAnswered: value – did the component answer the specified value?

Testing callbacks

WAMiniCalendar



June 2005

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

May Jul

Must supply canSelectBlock to instance – server creates instance!

Optional selectBlock callback

WAMiniCalendar

```
testSelectedDate
   | selected anchors |
   self
      newApplicationWithRootClass: WAMiniCalendar
      initializeWith: [:cal |
         cal canSelectBlock: [:date | true].
         cal selectBlock: [:date | selected := date]].
```

WAMiniCalendar

```
testSelectedDate
  | selected anchors |
  self
    newApplicationWithRootClass: WAMiniCalendar
    initializeWith: [:cal |
      cal canSelectBlock: [:date | true].
      cal selectBlock: [:date | selected := date]].
  self establishSession.
  self assert: selected isNil.
  anchors := self lastResponse
    anchorsWithLabel:
      (Date today dayOfMonth printString).
  self followAnchor: anchors last.
  self assert: selected = Date today.
```

Other topics...

- Session also available
- Hook for configuring application
- History in Web TestRunner is “live”
- Marking interactions for visual inspection
- http://www.cdshaffer.com/david/Seaside

Issues

- Visual appearance not tested
  - Support for storing snapshots of pages for human testers to view
- Client scripting (Javascript/DHTML) not tested – Squelenium Demo?

Other free frameworks

- SmallHttpUnit
  - VW, runs “outside” server
  - Excellent API for accessing page elements
- HttpUnit
  - Java, runs “outside” server
- Cactus
  - Java, designed to run in-container like SeasideTesting
- StrutsTestCase – like Cactus+HttpUnit

Conclusions

- Test components in isolation or in larger application
- Can interact directly with component/session to test state
- Can test back button behavior