Behaviour-Driven Development with Hera

Koen De Hondt

koen@all-objects-all-the-time.st

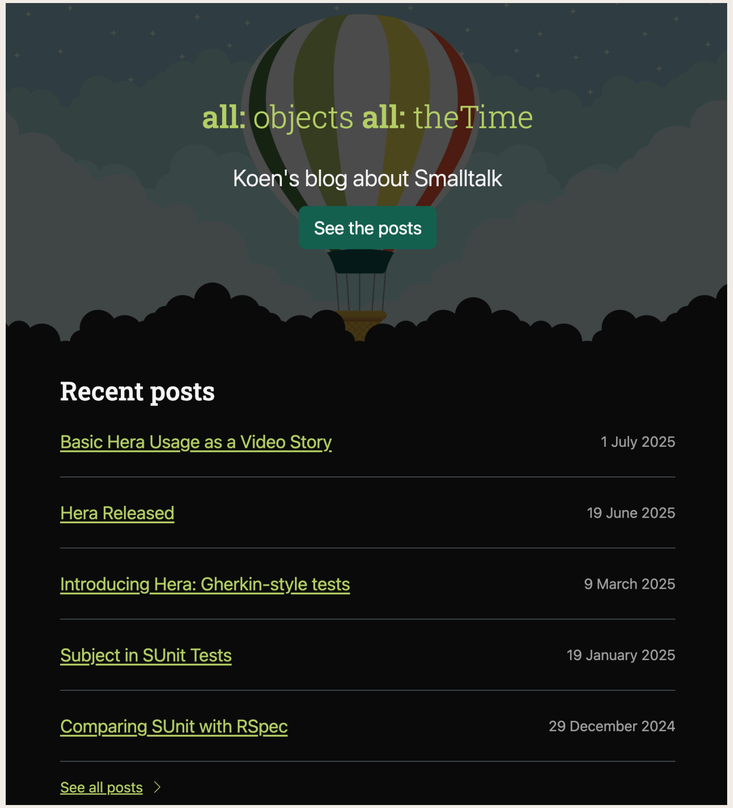
ESUG’25 — Gdansk, Poland

About me

Long-time Smalltalker

Pharo contributor

Passionate tool builder



https://all-objects-all-the-time.st

Behaviour-driven Development

BDD is a way for software teams to work that **closes the gap between business people and technical people** by:

- Encouraging collaboration across roles to build shared understanding of the problem to be solved
- Working in rapid, small iterations to increase feedback and the flow of value
- **Producing system documentation that is automatically checked against the system's behaviour**

We do this by focusing collaborative work around concrete, real-world examples that illustrate how we want the system to behave. We use those examples to guide us from concept through to implementation, in a process of continuous collaboration.

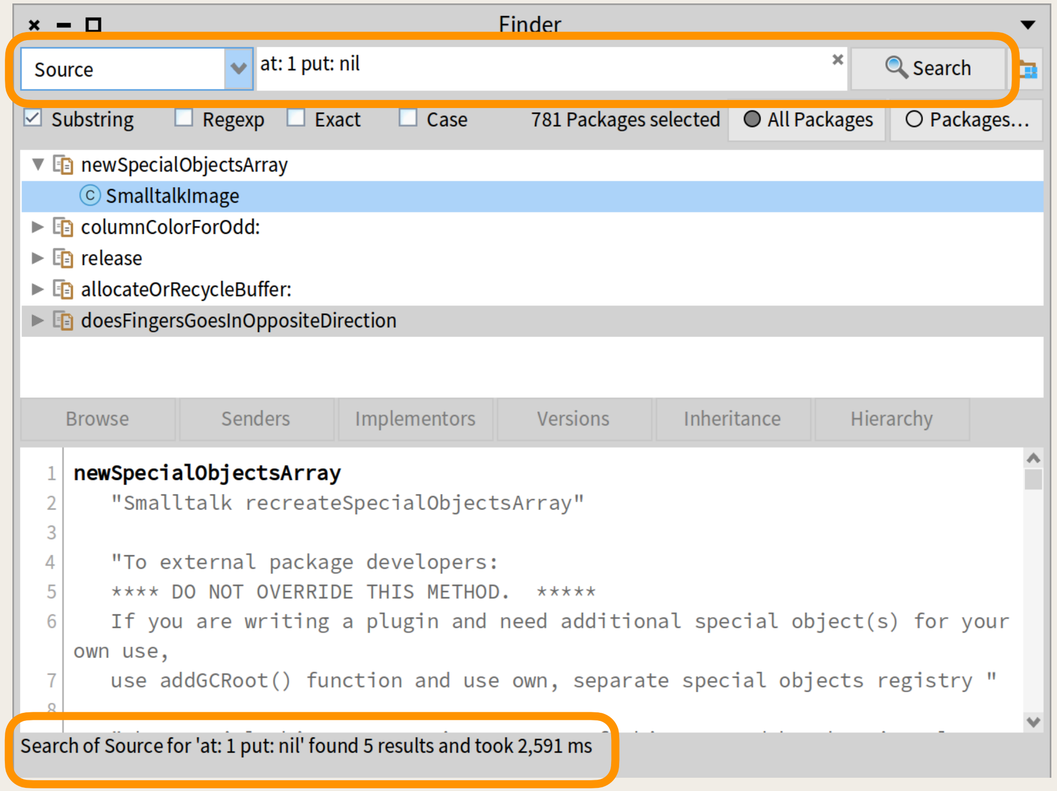
Source: https://cucumber.io/docs/bdd

What’s out There?

Gherkin uses a set of special keywords to give structure and meaning to executable specifications

See https://cucumber.io/docs/gherkin

Describing the Finder (1)



Feature: Finding

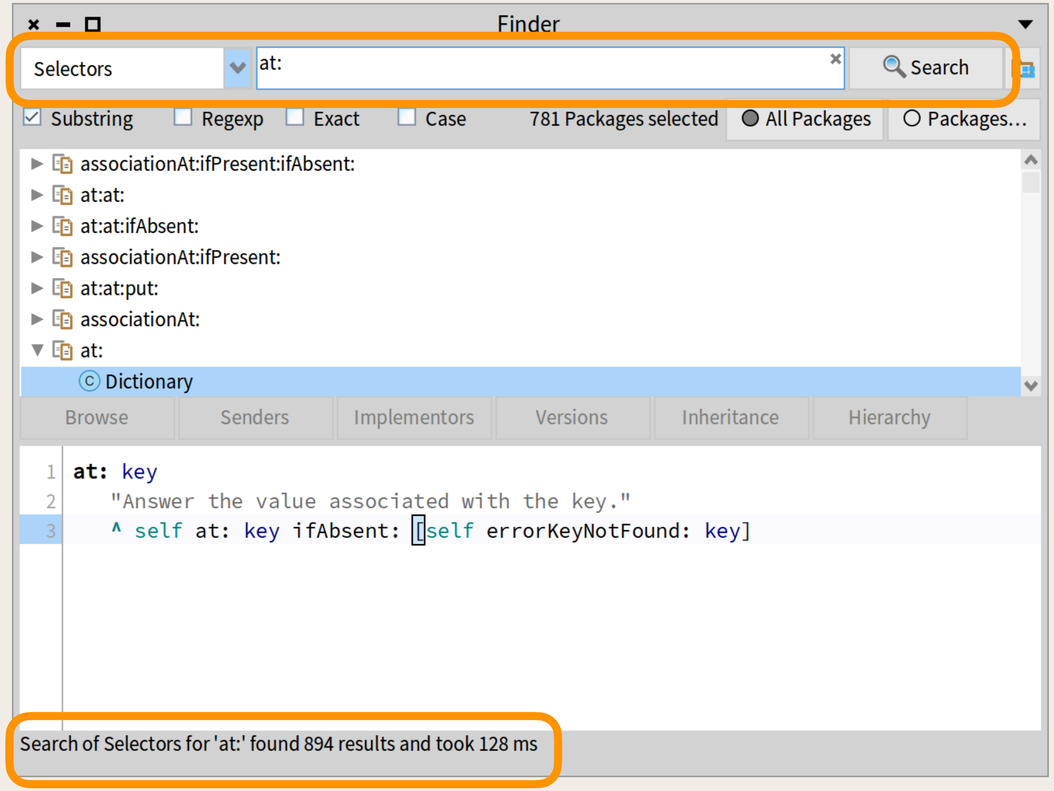
Scenario: Find in source

Given the Finder is open

When I search in Source for "at: 1 put: nil"

Then I see 5 matching methods

Describing the Finder (2)



Feature: Finding

Scenario: Find in selectors

Given the Finder is open

When I search in Selectors for "at:"

Then I see 894 matching methods

Given–When–Then Scenarios

Given
a precondition, context, initial state

When
an action or event

Then
an observable output

Steps and Step Definitions

Feature: Finding

Scenario: Find in source

Given the Finder is open

When I search in Source for "at: 1 put: nil"

Then I see 5 matching methods

Feature: Finding

Scenario: Find in selectors

Given the Finder is open

When I search in Selectors for "at:"

Then I see 894 matching methods

```
step 'the Finder is open' do
  ...
end

step 'I search in Source for "at: 1 put: nil"' do
  ...
end

step 'I see 5 matching methods' do
  ...
end

step 'I search in Selectors for "at:"' do
  ...
end

step 'I see 894 matching methods' do
  ...
end
```

Exact Match

Feature: Finding

Scenario: Find in source

Given the Finder is open

When I search in Source for "at: 1 put: nil"

Then I see 5 matching methods

```
step 'I search in Source for "at: 1 put: nil"' do
  ...
end

step 'I see 5 matching methods' do
  ...
end
```

Feature: Finding

Scenario: Find in selectors

Given the Finder is open

When I search in Selectors for "at:"

Then I see 894 matching methods

```
step 'I search in Selectors for "at:"' do
  ...
end

step 'I see 894 matching methods' do
  ...
end
```

Cucumber Expression Match

Feature: Finding

Scenario: Find in source

Given the Finder is open

When I search in Source for "at: 1 put: nil"

Then I see 5 matching methods

Feature: Finding

Scenario: Find in selectors

Given the Finder is open

When I search in Selectors for "at:"

Then I see 894 matching methods

```
step 'I search in {word} for {string}' do | scope searchString |
  ...
end

step 'I see {int} matching methods' do | quantity |
  ...
end
```

{string}, {word}, {int}, {float}, {}

Example with Data Table and Doc String

```
Feature: Browsing

  Scenario: Navigating to a method to see the source

    When I select:

      | Package  | Collections-Unordered  |

      | Tag      | Dictionaries           |

      | Class    | Dictionary             |

      | Protocol | accessing              |

      | Method   | at:                    |

    Then I see the source:

      """

      at: key

        "Answer the value associated with the key."

        ^ self at: key ifAbsent: [self errorKeyNotFound: key]

      """
```

```
step 'I select' do | dataTable |
  ...
end

step 'I see the source' do | docString |
  ...
end
```

What’s out There?

Cucumber runs Gherkin scripts by matching steps with step definitions in your favourite programming language

See https://cucumber.io

Cucumber-JS	JavaScript	OFFICIAL
Cucumber-JVM	Java	OFFICIAL
Cucumber-Ruby	Ruby	OFFICIAL
Android™	Java	OFFICIAL
Cucumber-JVM	Kotlin	OFFICIAL
Cucumber-Lua	Lua	OFFICIAL
Cucumber-Scala	Scala	OFFICIAL
Cucumber.cpp	C++	OFFICIAL
Cucumber.ml	OCaml	OFFICIAL
Godog	Go	OFFICIAL
Behat	PHP	SEMI-OFFICIAL
Behave	Python	SEMI-OFFICIAL
Cucumberish	iOS (Swift/ObjC)	SEMI-OFFICIAL
Pytest-BDD	Python	SEMI-OFFICIAL
Reqnroll	.NET (C#, F#, VB)	SEMI-OFFICIAL
Test::BDD-Cucumber	Perl	SEMI-OFFICIAL
Xunit.Gherkin.Quick	.NET (C#, F#, VB)	SEMI-OFFICIAL
gocuke	Go	SEMI-OFFICIAL
Cucumber-Rust	Rust	UNOFFICIAL
GoBDD	Go	UNOFFICIAL
Unencumbered	D	UNOFFICIAL
cwt-cucumber	C++	UNOFFICIAL
Cucumber-Clojure	Clojure	UNMAINTAINED
Cucumber-Gosu	Gosu	UNMAINTAINED
Cucumber-Groovy	Groovy	UNMAINTAINED
Cucumber-JRuby	JRuby	UNMAINTAINED
Cucumber-Jython	Jython	UNMAINTAINED
Cucumber-Tcl	Tcl	UNMAINTAINED

Source: https://cucumber.io/docs/installation/

Hera

Set of tools to write and run Gherkin scripts

Feature scripts are not stored as text, but as objects answered by methods

Feature scripts are grouped in classes, called acceptance test classes

Step definitions are methods

When running scenarios, Hera matches scenario steps with step definition methods

Hera Feature Browser

FinderAcceptanceTest
Finding in selectors

```
Feature: Finding in selectors

   Background:
      Given the Finder is open

   Scenario: Substring search
      When I search for "indexOf:startingAt:ifAbsent
      Then I see the following matching methods:
         | indexOf:startingAt:ifAbsent:       |
```

When | I check "Case" | =

```
stepICheckCase
```

Hera Step Browser

All | Given | When | Then | Duplicates | Empty | Unused

Step
I check "Case"
I uncheck "Substring"
I check "Exact"
I search for {string}

When | I check "Exact" | =

```
stepICheckExact

   self check: { #finder . #searchOptions .
#exactCheckBox }
```

Defined in FinderAcceptanceTest

Hera Runner

3 | 3 | 0 | 0 | Less than a second | (2025-06-25 22:00)

```
         | lastIndexOf:startingAt:ifAbsent:    |
         | indexOf:startingAt:ifAbsent:using: |

   Scenario: Exact search
      Given the Finder is open [Background]
      When I uncheck "Substring"
      And I check "Exact"
      And I search for "indexOf:startingAt:ifAbsent:"
      Then I see the following matching methods:
         | indexOf:startingAt:ifAbsent: |

   Scenario: Case-sensitive search
      Given the Finder is open [Background]
      When I uncheck "Substring"
      And I check "Case"
      And I search for "indexOf:startingAt:ifAbsent:"
      Then I see the following matching methods:
         | indexOf:startingAt:ifAbsent:       |
         | indexOf:startingAt:ifAbsent:using: |

Total: 3. Passed: 3. Failed: 0. Error: 0.
```

Hera Feature Method

Feature: Finding

Scenario: Find in selectors

Given the Finder is open

When I search in Selectors for "at:"

Then I see 894 matching methods

```
featureFinding

  <heraFeature: 'Finding'>

  ^ (self feature: 'Finding')
        scenarios: {
              (self scenario: 'Find in selectors')
                    given: 'the Finder is open';
                    when: 'I search in Selectors for "at:"';
                    then: 'I see 894 matching methods' }
```

Hera Step Definition Method

Feature: Finding

Scenario: Find in selectors

Given the Finder is open

When I search in Selectors for "at:"

Then I see 894 matching methods

```
stepTheFinderIsOpen

  <heraStepDefinition: #(Given 'the Finder is open')>

  self openPresenterAs: #finder with: [ StFinderPresenter open ]
```

```
stepISearchIn: scope for: searchString

  <heraStepDefinition: #(When match 'I search in {word} for {string}')>

  | searchBar index |
  searchBar := (self presenterAt: #finder) searchBar.
  index := self indexOf: scope in: searchBar searchModeDropList.
  searchBar searchModeDropList selectIndex: index.
  searchBar searchInput text: searchString.
  searchBar searchButton click
```

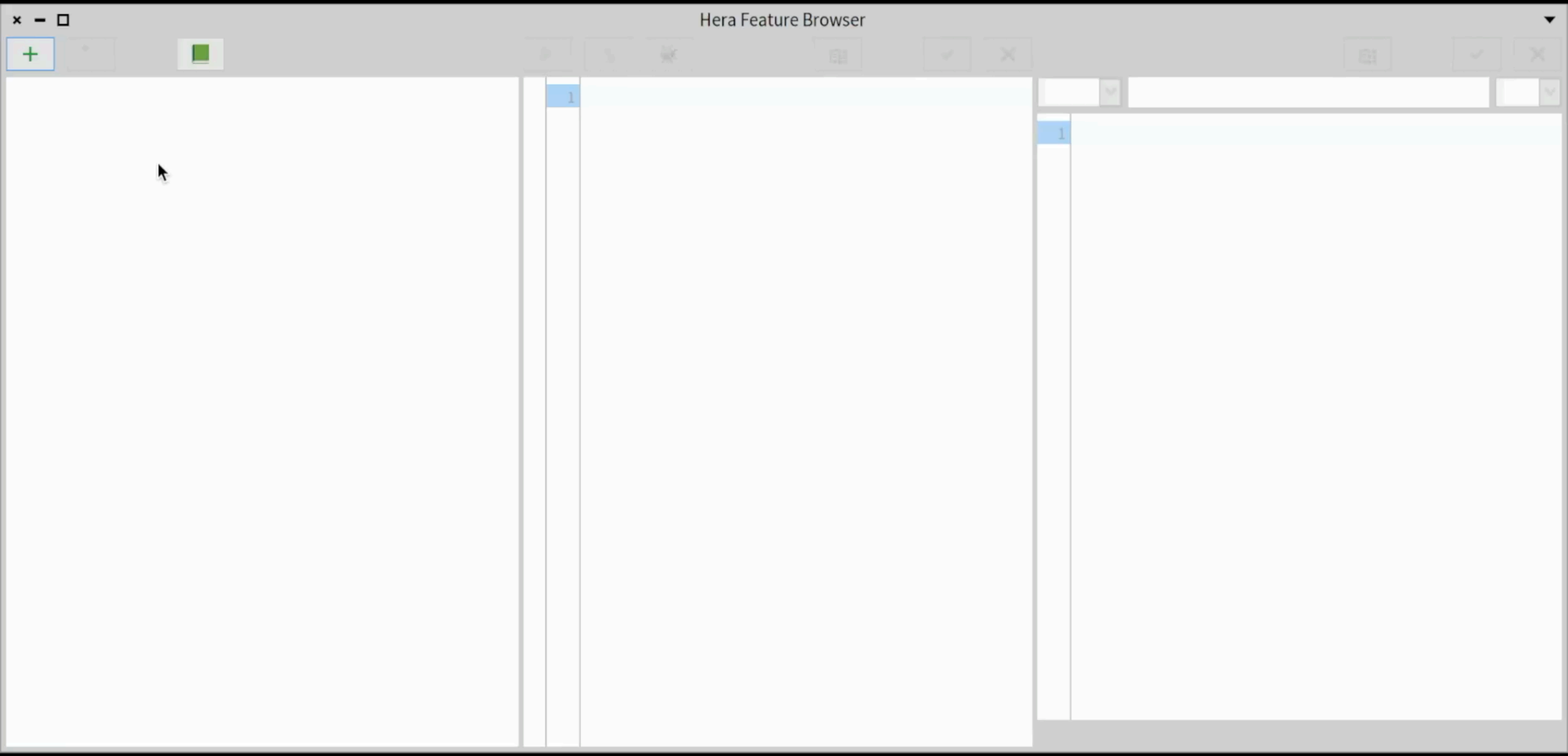
```
stepISeeMatchingMethods: quantity

  <heraStepDefinition: #(Then match 'I see {int} matching methods')>

  | resultTree |
  resultTree := (self presenterAt: #finder) resultTree.
  self assert: resultTree roots size equals: quantity
```

The videos in the following 13 slides
and the accompanying explanations are available at

https://all-objects-all-the-time.st/#/blog/posts/15



Video 1/13

Hera Feature Browser

- FinderAcceptanceTest
  - Finding in selectors

```
1 Feature: Finding in selectors
```

```
1
```

Hera Feature Browser

- FinderAcceptanceTest
  - Finding in selectors

```
Feature: Finding in selectors

   Scenario: Substring search
      Given the Finder is open
```

Given | the Finder is open | =

```
stepTheFinderIsOpen

   self openPresenterAs: #finder with: [ StFinderPresenter open ]
```

Defined in FinderAcceptanceTest

Hera Feature Browser

- FinderAcceptanceTest
  - Finding in selectors

```
Feature: Finding in selectors

  Scenario: Substring search
    Given the Finder is open
```

Hera Feature Browser

Hera Runner

1 1 0 0 Less than a second (2025-06-25 21:11)

```
Feature: Finding in selectors

   Scenario: Substring search
      Given the Finder is open

Total: 1. Passed: 1. Failed: 0. Error: 0.
```

Given the Finder is open =

```
stepTheFinderIsOpen

   self openPresenterAs: #finder with: [ StFinderPresenter open ]
```

Defined in FinderAcceptanceTest

Hera Feature Browser

Hera Runner

1 1 0 0 7 seconds (2025-06-25 21:13)

```
Feature: Finding in selectors

   Scenario: Substring search
      Given the Finder is open
      When I pause

Total: 1. Passed: 1. Failed: 0. Error: 0.
```

Given the Finder is open =

```
stepTheFinderIsOpen

   self openPresenterAs: #finder with: [ StFinderPresenter open ]
```

Defined in FinderAcceptanceTest

Hera Feature Browser

- FinderAcceptanceTest
  - Finding in selectors

```
Feature: Finding in selectors

   Scenario: Substring search
      Given the Finder is open
      When I search for
"indexOf:startingAt:ifAbsent:"
      When I pause
```

When | I search for "indexOf:startingAt:ifAbsent:" | =

```
stepISearchForindexOfstartingAtifAbsent

   self
      presenterAtPath: { #finder . #searchBar }
      do: [ :searchBar |
         searchBar searchInput text: 'indexOf:startingAt:ifAbsent:'.
         searchBar searchButton click ]
```

Defined in FinderAcceptanceTest

Hera Feature Browser

- FinderAcceptanceTest
  - Finding in selectors

```
Feature: Finding in selectors

   Scenario: Substring search
      Given the Finder is open
      When I search for "indexOf:startingAt:ifAbsent:"
      When I pause
```

When | I search for "indexOf:startingAt:ifAbsent:" | =

```
stepISearchForindexOfstartingAtifAbsent

   self
      presenterAtPath: { #finder . #searchBar }
      do: [ :searchBar |
         searchBar searchInput text: 'indexOf:startingAt:ifAbsent:'.
         searchBar searchButton click ]
```

Defined in FinderAcceptanceTest

Hera Feature Browser

- FinderAcceptanceTest
  - Finding in selectors

```
Feature: Finding in selectors

   Scenario: Substring search
      Given the Finder is open
      When I search for "indexOf:startingAt:ifAbsent:"
      Then I see the following matching methods:
         | indexOf:startingAt:ifAbsent:       |
         | lastIndexOf:startingAt:ifAbsent:   |
         | indexOf:startingAt:ifAbsent:using: |
      When I pause
```

Then | I see the following matching methods: | =

```
stepISeeTheFollowingMatchingMethods: aDataTable

   | expectedSelectors resultTree roots |
   expectedSelectors := aDataTable asArray.
   resultTree := self presenterAtPath: { #finder . #resultTree }.
   roots := resultTree roots.
   self assert: roots size equals: expectedSelectors size.
   expectedSelectors withIndexDo: [ :each :index |
      self assert: (roots at: index) displayString equals: each ]
```

Defined in FinderAcceptanceTest

Hera Feature Browser

FinderAcceptanceTest
Finding in selectors

```
Feature: Finding in selectors

   Scenario: Substring search
      Given the Finder is open
      When I search for "indexOf:startingAt:ifAbsent:"
      Then I see the following matching methods:
         | indexOf:startingAt:ifAbsent:       |
         | lastIndexOf:startingAt:ifAbsent:   |
         | indexOf:startingAt:ifAbsent:using: |
```

When | I search for {string} | {}

```
stepISearchFor: aSearchString

   self
      presenterAtPath: { #finder . #searchBar }
      do: [ :searchBar |
         searchBar searchInput text: aSearchString.
         searchBar searchButton click ]
```

Defined in FinderAcceptanceTest

Hera Feature Browser

Hera Runner

2 1 0 1 Less than a second (2025-06-25 21:45)

```
Feature: Finding in selectors

   Scenario: Substring search
      Given the Finder is open
      When I search for "indexOf:startingAt:ifAbsent:"
      Then I see the following matching methods:
         | indexOf:startingAt:ifAbsent:       |
         | lastIndexOf:startingAt:ifAbsent:   |
         | indexOf:startingAt:ifAbsent:using: |

   Scenario: Exact search
      When I uncheck "Substring"
         ERROR Presenter with name "finder" not found.

Total: 2. Passed: 1. Failed: 0. Error: 1.
```

Given I check "Exact" =

```
stepICheckExact

   self check: { #finder . #searchOptions . #exactCheckBox }
```

Defined in FinderAcceptanceTest

Hera Feature Browser

- FinderAcceptanceTest
  - Finding in selectors

```
Feature: Finding in selectors

   Background:
      Given the Finder is open

   Scenario: Substring search
      When I search for "indexOf:startingAt:ifAbsent:"
      Then I see the following matching methods:
         | indexOf:startingAt:ifAbsent:       |
         | lastIndexOf:startingAt:ifAbsent:   |
         | indexOf:startingAt:ifAbsent:using: |

   Scenario: Exact search
      When I uncheck "Substring"
      And I check "Exact"
      And I search for "indexOf:startingAt:ifAbsent:"
      Then I see the following matching methods:
         | indexOf:startingAt:ifAbsent: |
```

Given | I check "Exact" | =

```
stepICheckExact

   self check: { #finder . #searchOptions . #exactCheckBox }
```

Defined in FinderAcceptanceTest

Hera Feature Browser

- FinderAcceptanceTest
  - Finding in selectors

```
Feature: Finding in selectors

   Background:
      Given the Finder is open

   Scenario: Substring search
      When I search for "indexOf:startingAt:ifAbsent:"
      Then I see the following matching methods:
         | indexOf:startingAt:ifAbsent:       |
         | lastIndexOf:startingAt:ifAbsent:   |
         | indexOf:startingAt:ifAbsent:using: |

   Scenario: Exact search
      When I uncheck "Substring"
      And I check "Exact"
      And I search for "indexOf:startingAt:ifAbsent:"
      Then I see the following matching methods:
         | indexOf:startingAt:ifAbsent: |

   Scenario: Case-sensitive search
      When I uncheck "Substring"
      And I check "Case"
      And I search for "indexOf:startingAt:ifAbsent:"
      Then I see the following matching methods:
         | indexOf:startingAt:ifAbsent:       |
         | indexOf:startingAt:ifAbsent:using: |
```

Given | I check "Case" | =

```
stepICheckCase

   self check: { #finder . #searchOptions . #caseCheckBox }
```

Defined in FinderAcceptanceTest

Video 13/13

Gherkin Compatibility

Supported

Feature, Rule, Background, Scenario, Given, When, Then, And, But, *

Data table, doc string, tag

Free-form text descriptions in feature, rule, and scenario

{string}, {word}, {int}, {float}, {}

Not supported yet

Scenario Outline

Free-form descriptions in background

Not supported

Comments with #

Technical Limitations

Describing modal dialogs

Next Steps

Gather feedback from the community and improve Hera

Better integration with the Pharo tools

Refactoring support

CI support

Create a DSL for describing the behaviour of Spec applications

Use Hera on a wide scale

Use Hera to describe the behaviour of Atlas

Resources

Introduction: https://all-objects-all-the-time.st/#/blog/posts/13

Release of Hera 1.0: https://all-objects-all-the-time.st/#/blog/posts/14

Documentation: https://all-objects-all-the-time.st/#/projects/hera

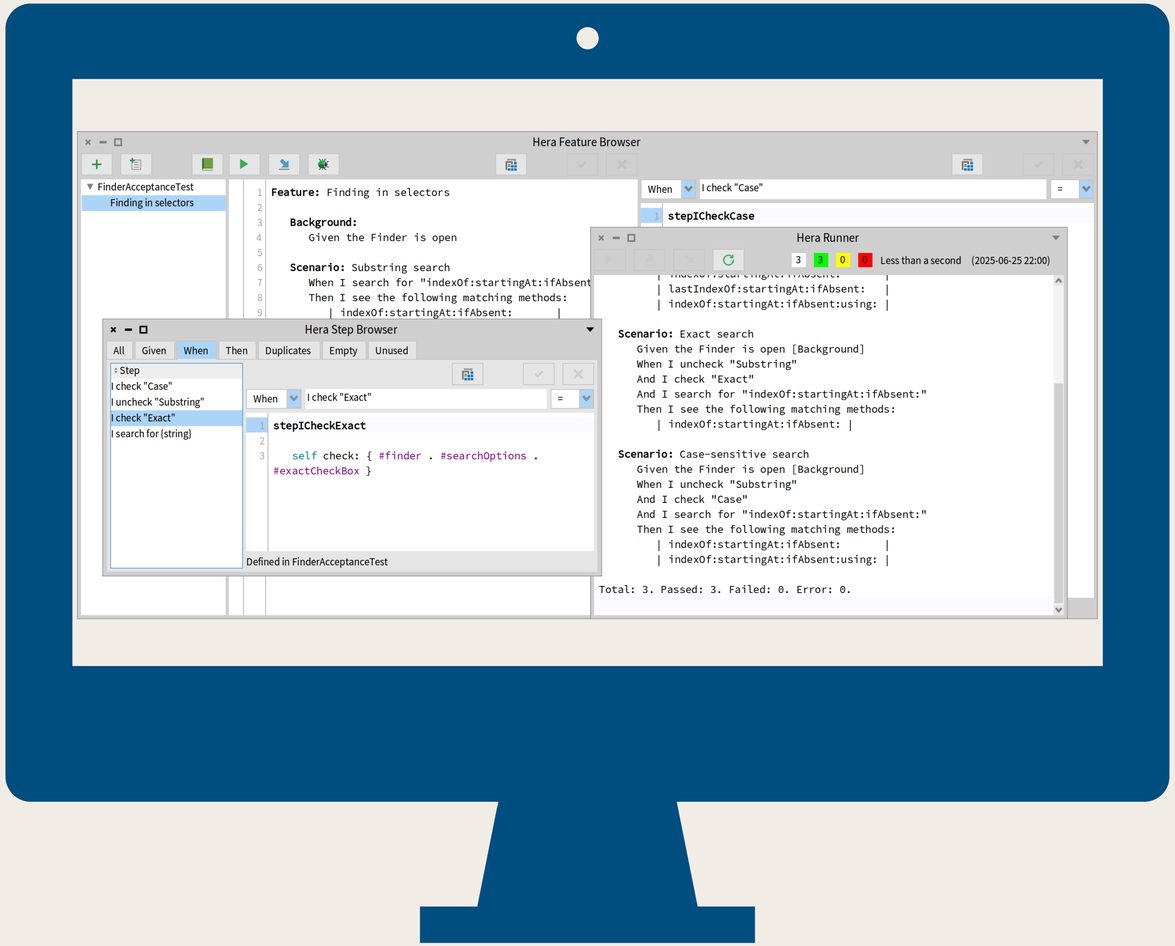
Repo: https://github.com/koendehondt/hera-for-pharo

Issues and feature requests: https://github.com/koendehondt/hera-for-pharo/issues

Innovation Technology Awards

Come and see Hera in action

Demo at 17:30 today



Hera Feature Browser

HeraWebAcceptanceTest
HeraWebAcceptanceTestExample
Describing web page behavior

```
Feature: Describing web page behavior

  Scenario: Search Pharo on Wikipedia

    When I open https://en.wikipedia.org/
    And I enter "Pharo" in the "Search Wikipedia" field
    Then the current URL is https://en.wikipedia.org/wiki/Pharo
    And I see a page with main header "Pharo"
    When I pause
```

When | I open {} | {}

```
stepIOpenURL: url

  driver get: url
```

Defined in HeraWebAcceptanceTest

Pharo - Wikipedia

en.wikipedia.org/wiki/Pharo

Chrome is being controlled by automated test software.

WIKIPEDIA The Free Encyclopedia | Search Wikipedia | Search | Donate | Create account | Log in

Pharo

9 languages

Article | Talk | Read | Edit | View history | Tools

From Wikipedia, the free encyclopedia

For other uses, see Pharo (disambiguation).
Not to be confused with the similar term Pharaoh (disambiguation).

Pharo is a cross-platform implementation of the classic Smalltalk-80 programming language and runtime system.[3] It is based on the OpenSmalltalk virtual machine (VM) named Cog,[4][5][6][7]:16 which evaluates a dynamic, reflective, and object-oriented programming language with a syntax closely resembling Smalltalk-80. It is free and open-source software, released under a mix of MIT, and Apache 2 licenses.

Pharo is shipped with source code compiled into a *system image* that contains all software needed to run Pharo.[7]:16 Like the original Smalltalk-80, Pharo provides several live programming features such as immediate object manipulation, live updates, and just-in-time compilation (JIT). The system image includes an integrated development environment (IDE) to modify its components.

Pharo was forked from Squeak v3.9 in March 2008.[8][3][7][7]:10[9]

Pharo	
Pharo logo with lighthouse	
Paradigm	object-oriented
Family	Smalltalk: Squeak
Developer	Pharo community

Contents hide

(Top)
Overview
Key features
Virtual machine
Built-in software
Language features
Relation to Smalltalk
Language syntax
History
Use of Pharo
Companies and consultants
Performance and virtual machine (VM)
See also
References
External links

Behaviour-Driven Development with Hera

Thank you!