CS1632, LECTURE 14: SYSTEMS TESTING THE WEB WITH SELENIUM

Bill Laboon

Background

- So far, all of our testing has been with specific, often text-based, input and output
- Turns out not everybody uses a text-based interface
- GUIs, web pages, mobile applications, etc.

Testing Techniques are Similar

We can actually use many of the same tools and techniques to test more complicated interfaces. Now that you understand the basics of automated testing, it's possible to take what you've learned and apply it to a more complex interface.

But this remains:

EXPECTED BEHAVIOR vs OBSERVED BEHAVIOR

Testing the Web

- Just an example – similar ideas for testing other graphical or non-textual / mathematical interfaces
- Keep in mind that we are going to *expect* certain things to occur or be seen, and then *observe* whether or not they occur or are seen.

Web = text

- Specially formatted and displayed text, but text!
- If your computer can process it, it’s just 1s and 0s, which can be represented as text

Theoretically, we could test web pages like so…

```
// Any downsides to this?

@Test
public void testWeb() {
   String expectedHtml = “<head></head><body><strong>Hello, world!
</strong></body>";
   String pageText = getPage("http://example.com");
   assertEquals(expectedHtml, pageText);
}
```

Downsides

1. Change the page, change the entire test
   Fragile tests!
2. What about JavaScript?
   Also check that the right JS is on page?
3. Unreadable
4. Simplistic and low-level
   Kind of like programming in assembly
5. No semantic understanding (e.g. of links, textboxes)

Web Testing Frameworks

Think of these as a higher-lever level programming language for testing web pages.

Sure, you could program everything in assembly, but this is rarely ideal.

Another way to think of it – just extra libraries so you don’t have to program everything yourself.

What is Selenium?

- An open-source web testing framework.
- Battle-hardened.
- Works with Windows, OS X, Linux, other OSes.
- Works with Java, Ruby, Python, other languages.
- Works with most modern web browsers.
- Has its own IDE.
- Can also be used for quick scripting.

Why the name Selenium?

One of their competitors was "Mercury.“
Mercury poisoning is cured by Selenium pills.

Selenium is a whole ecosystem

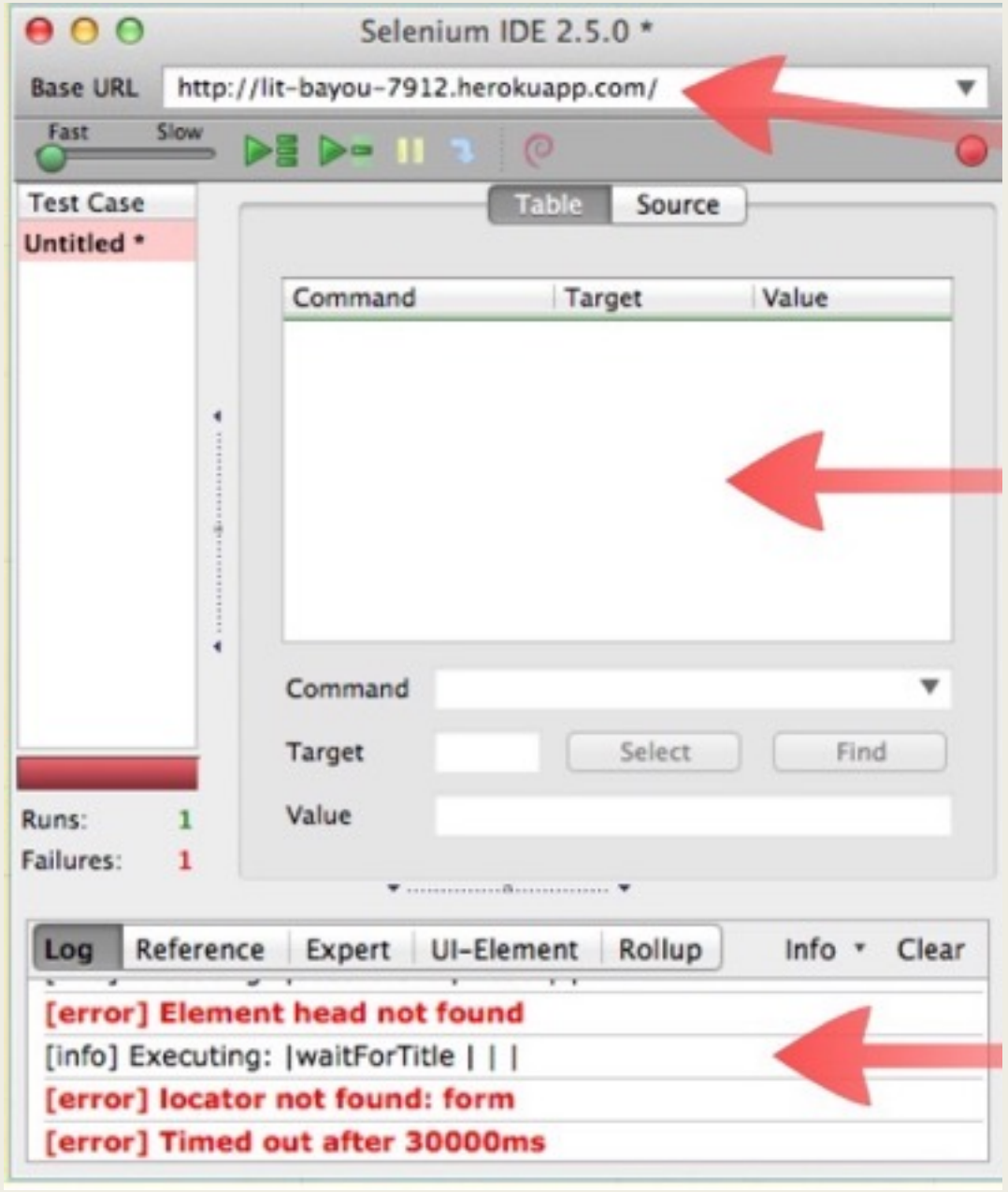
Selenium is a very complex, complete framework. We're really just going to be seeing the tip of the iceberg in this class.

It has other uses aside from testing, as well:

1. Web macros
2. Scraping
3. Load testing or parallel testing with Selenium Grid

Getting Started with Selenium

1. Download Firefox (getting Selenium IDE working in Chrome is possible, but more prone to bugs)
2. Go to http://docs.seleniumhq.org/
3. Download and install Selenium4.
4. Click on the "Se" icon in the upper right-hand corner, or go to Tools -> Selenium IDE



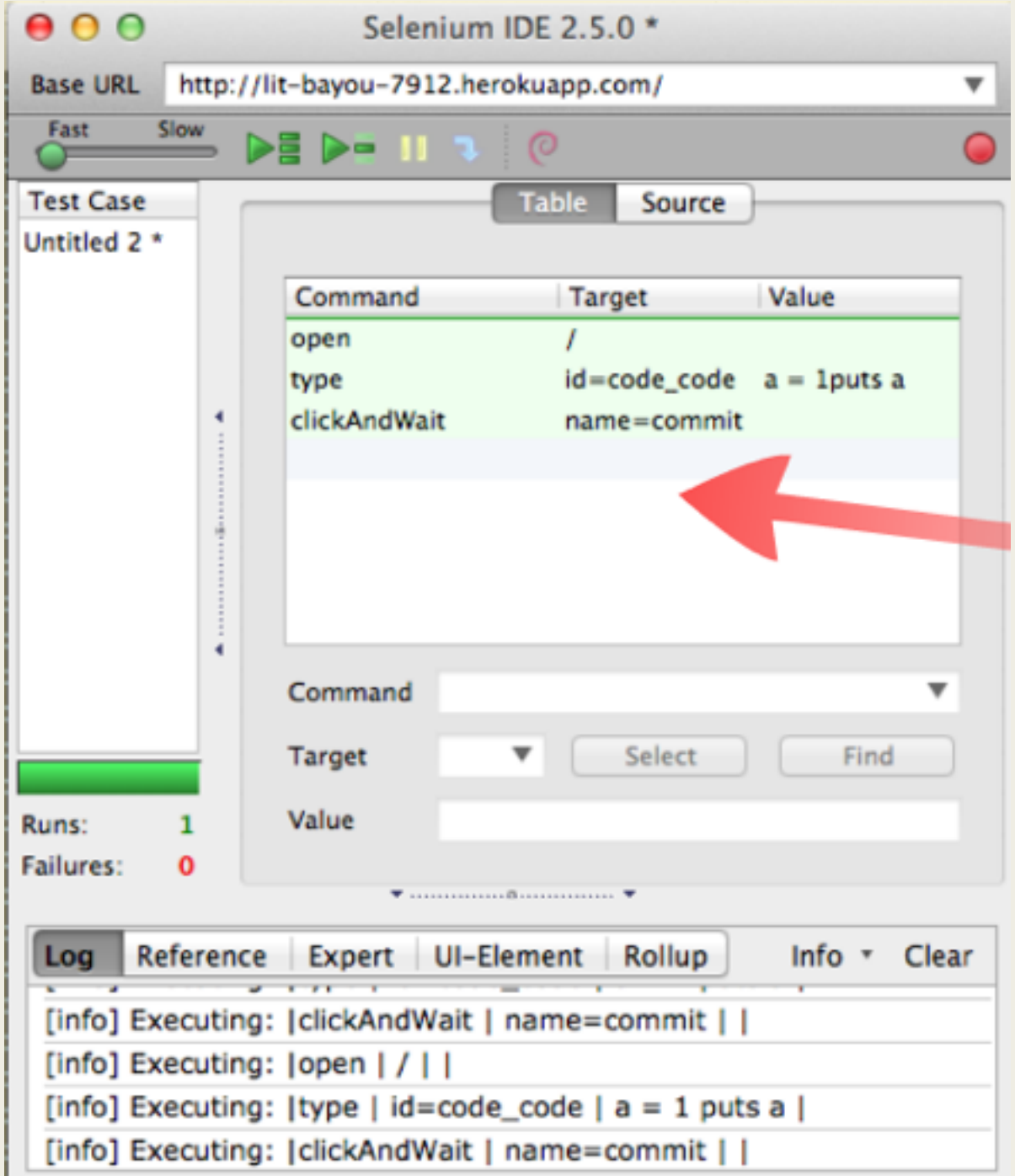
Starting URL

Commands

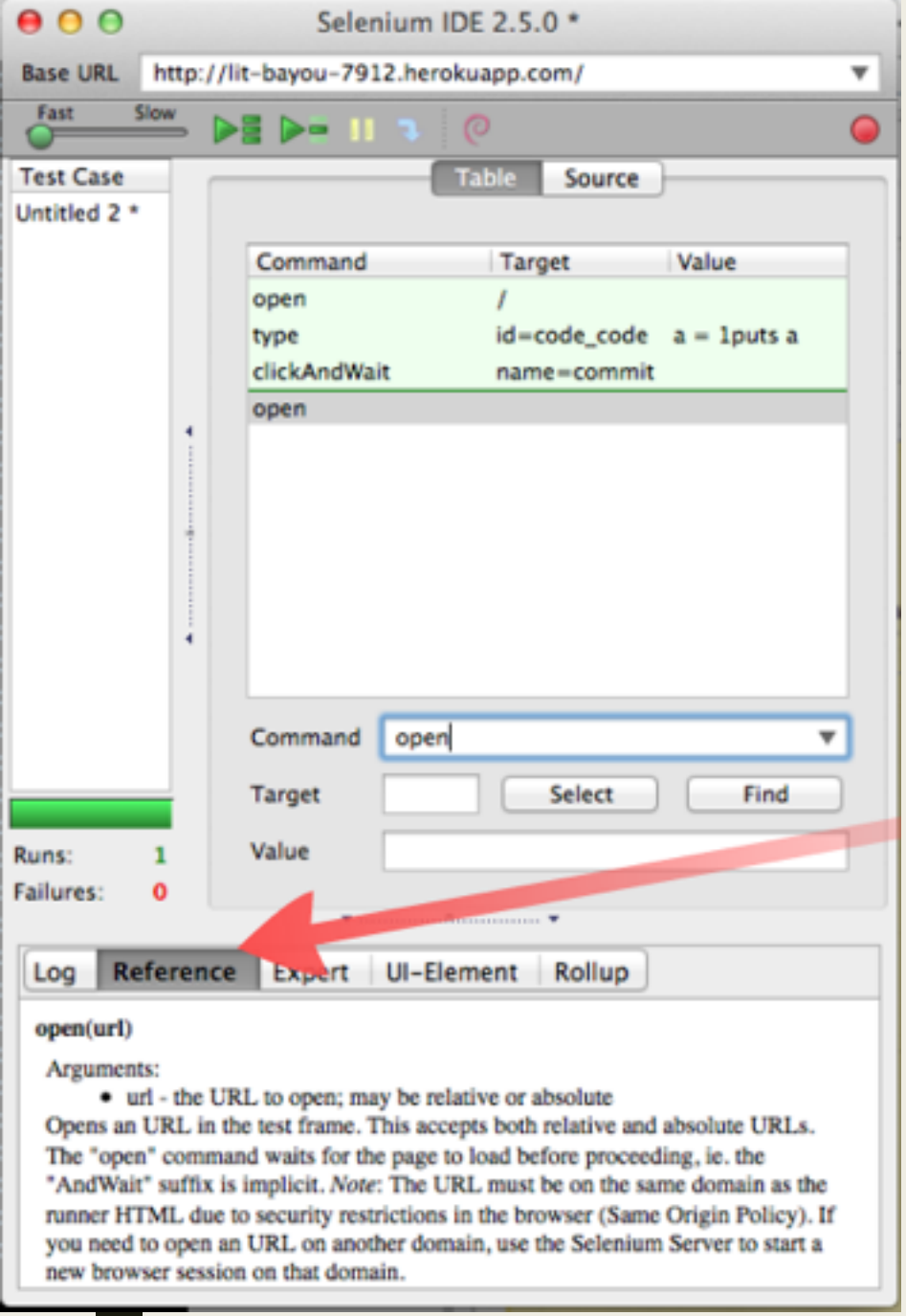
Logging output

Simple Scripting

1. File... New Test Case
2. Record an operation (red circle)
3. Stop recording
4. Run test suite (green triangle)


Selenium IDE 2.5.0 *
Base URL
http://lit-bayou-7912.herokuapp.com/
Fast
Slow
Test Case
Untitled 2 *
Table
Source
Command
Target
Value
open
/
type
id=code_code
a = 1puts a
clickAndWait
name=commit
Command
Target
Select
Find
Value
Runs:
1
Failures:
0
Log
Reference
Expert
UI-Element
Rollup
Info
Clear
[info] Executing: |clickAndWait | name=commit | |
[info] Executing: |open | / | |
[info] Executing: |type | id=code_code | a = 1 puts a |
[info] Executing: |clickAndWait | name=commit | |

Our code

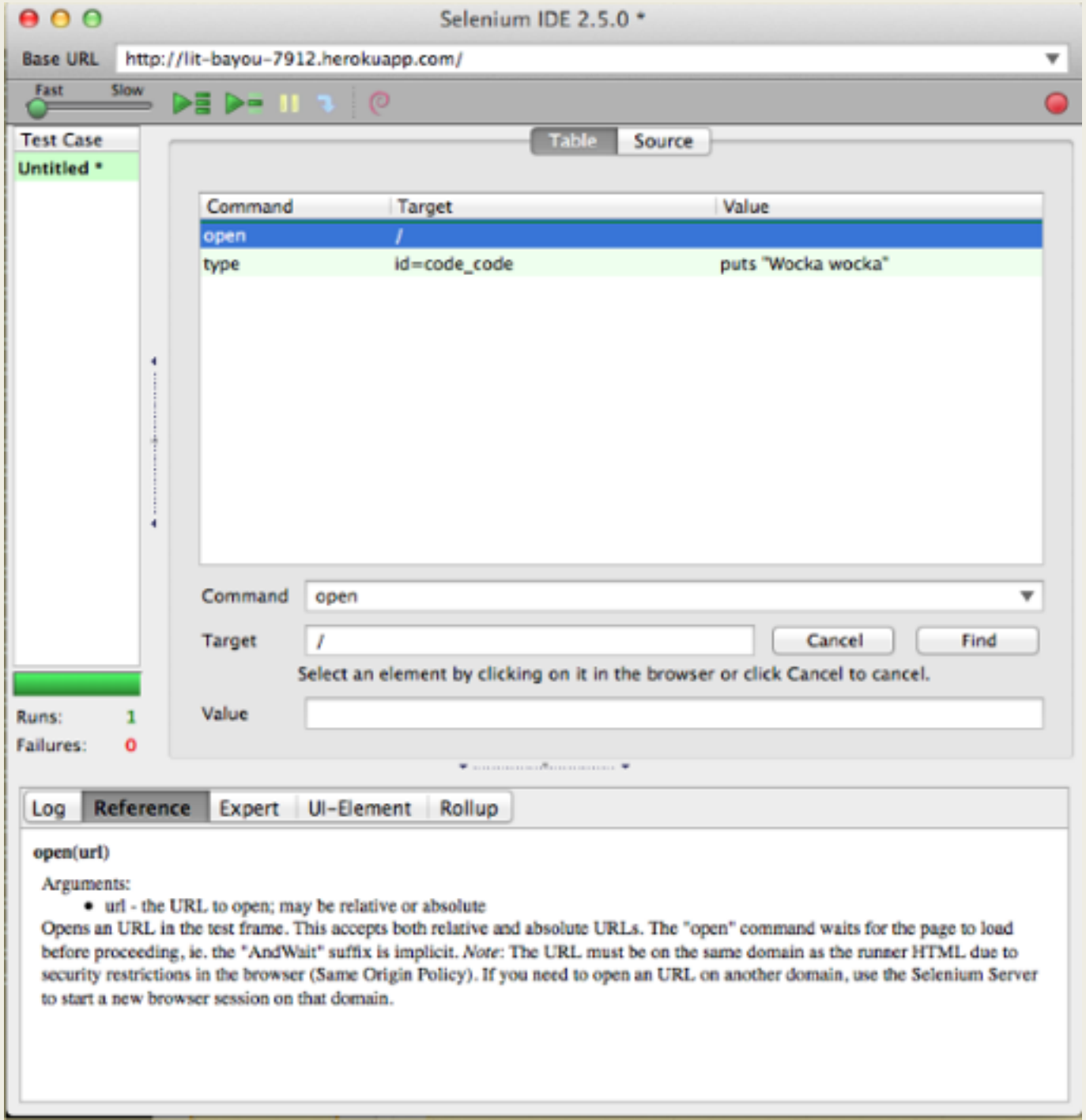


open <URL> - Opens either an absolute or relative URL

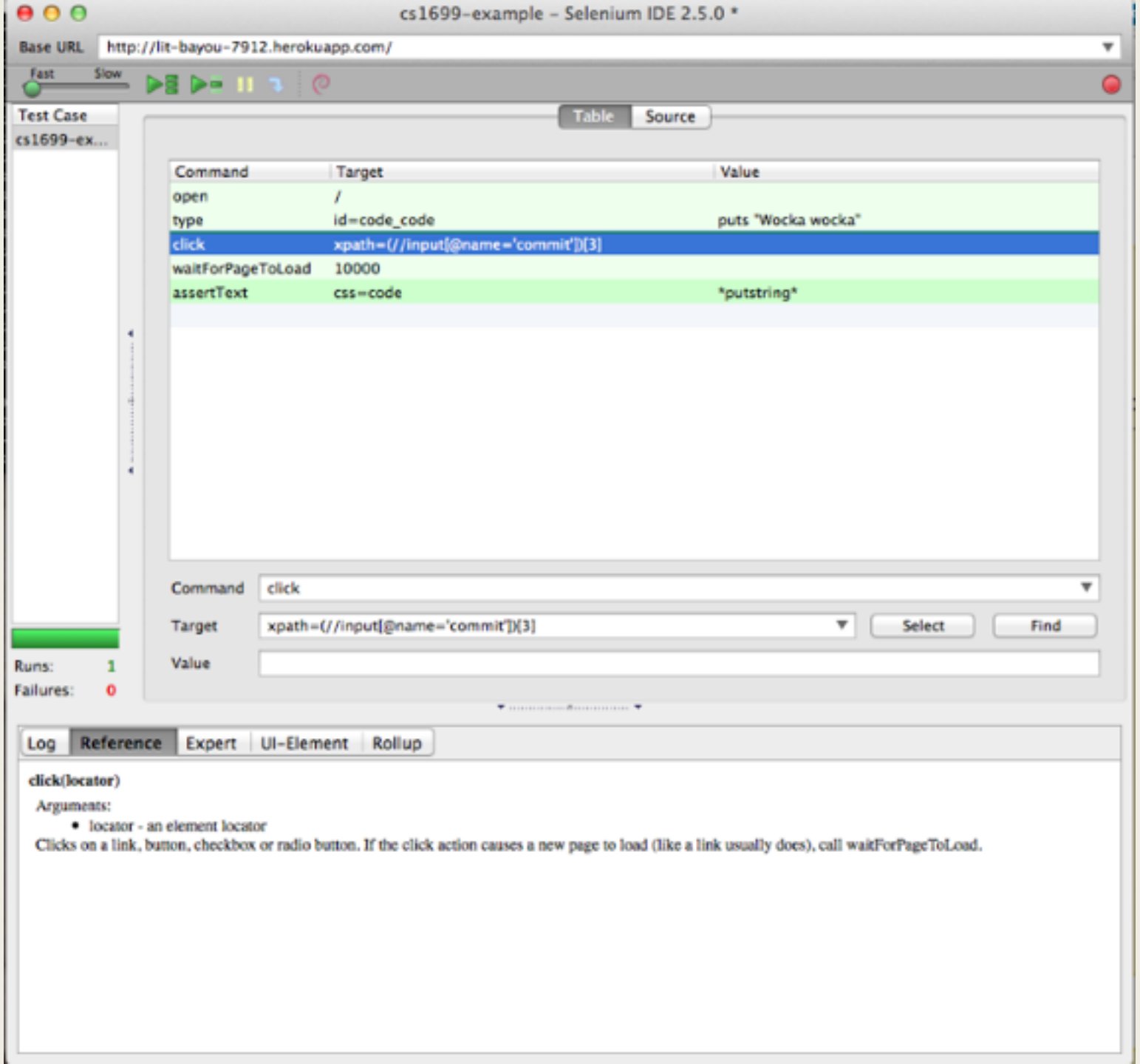
Note that you can check what an operation does by clicking on the "Reference" tab at the bottom

Also note…

- …you can move steps around by just clicking and dragging
- … you can add comments by selecting “Add new comment”
- … the textbox is not a traditional textbox
- … commands may or may not have arguments



The type command will type in a given target (e.g. textbox)



Click <target> will click on an element on the web page

Avoiding Intermittent Failures

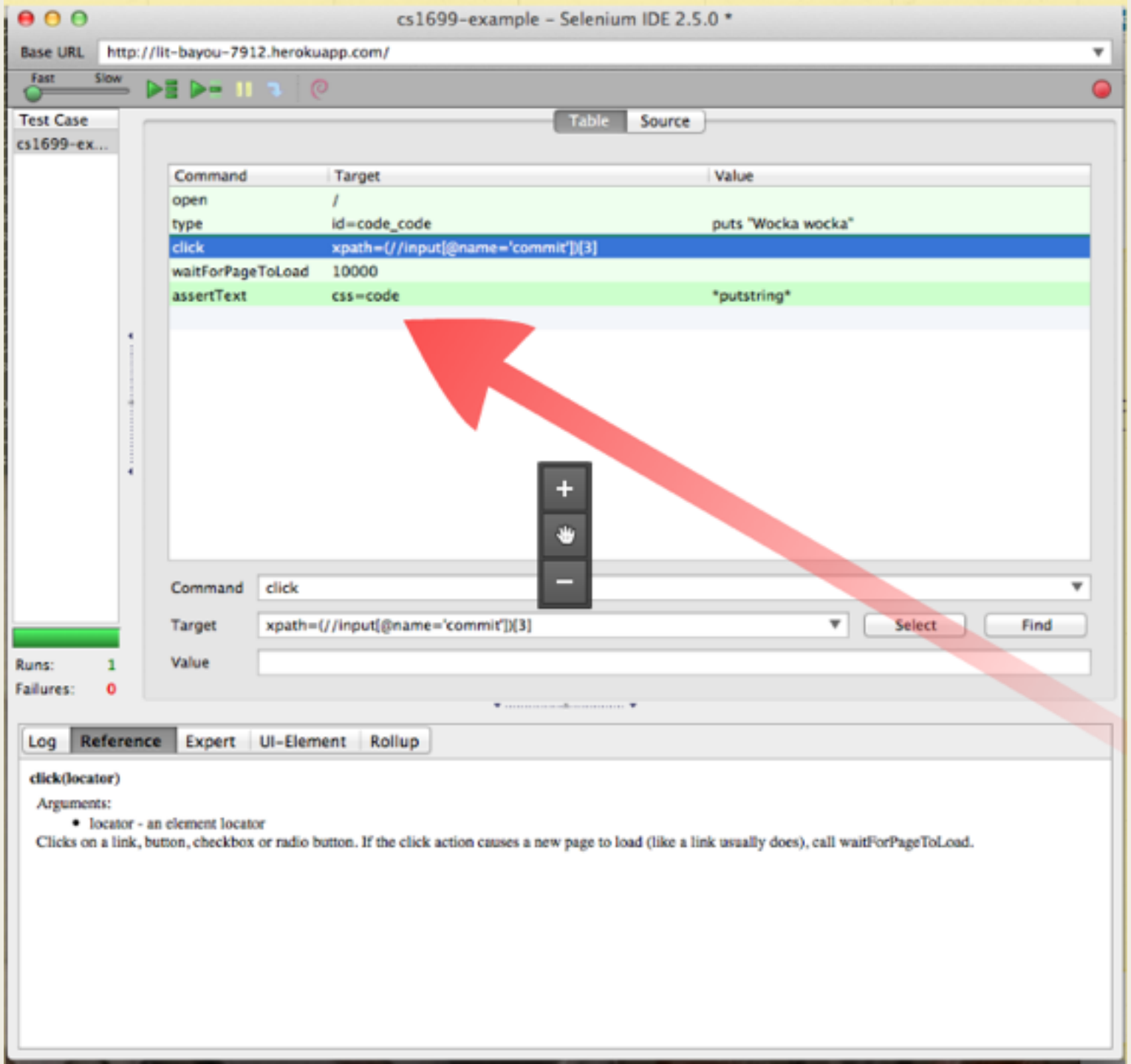
- waitForPageToLoad - Waits for another page to load, with an optional timeout.
- If you don't wait, Selenium goes as fast as it can, and may do a check before the page is ready.
- This means that your assertions may fail simply because the page hasn’t finished loading!
- Intermittent (a/k/a non-deterministic or ND failures) are a common symptom of poorly written web tests.

…andWait

- clickAndWait -> a combination of "click" and "waitForPageToLoad“
- There are other "...andWait" variants since waiting for a page to load is very common.

Hello, assertions, my old friend… I've come to assert you again..

- We are going to use assertions to specify expected behavior
- Same concept as traditional Junit assertions, just at a different level of abstraction


cs1699-example - Selenium IDE 2.5.0 *
Base URL
http://lit-bayou-7912.herokuapp.com/
Fast
Slow
Test Case
cs1699-ex...
Table
Source
Command
Target
Value
open
/
type
id=code_code
puts "Wocka wocka"
click
xpath=(//input[@name='commit'])[3]
waitForPageToLoad
10000
assertText
css=code
putstring
Command
click
Target
xpath=(//input[@name='commit'])[3]
Select
Find
Value
Runs:
1
Failures:
0
Log
Reference
Expert
UI-Element
Rollup
click(locator)
Arguments:
locator - an element locator
Clicks on a link, button, checkbox or radio button. If the click action causes a new page to load (like a link usually does), call waitForPageToLoad.

assertion

Select can be helpful to find a way to specify a target

- Lots of ways to specify an element on a webpage
- Select can help you find one that works

Lots of fun assertions…

- assertText / assertTextPresent - Assert that text exists (on an element (former) or entire page (latter)). Note that this is a regex!
- assertCookie - Assert that a cookie exists.
- assertElementPresent - Assert that an element exists somewhere on the page.
- assertAlert - Assert that an alert took place.
- assertEditable - Assert that an element is editable.
- assertEval - Evaluate some JavaScript and assert the result.