



Test Automation

Tony Godwin
(Day 2 of 4)



Test Automation Class Overview

- Less than twelve hours to learn “Test Automation”
- Automation of Browser Client
 - Record and Playback (Day 1)
 - Record and Playback (Day 2) <= YOU ARE HERE
 - Write code (Day 3)
 - Write code (Day 4)



Remember: Define a Test Strategy

A great way to increase your chances of automated web testing success is to focus your efforts by mapping out a testing strategy.

The best way to do that is to answer four questions:

1. How does your business make money (or generate value for the end-user)?
2. How do your users use your application?
3. What browsers are your users using?
4. What things have broken in the application before?"

-- "*Selenium Bootcamp*" by Dave Haeffner – Sauce Labs



Record and Playback

- Did you read the step-by-step?
 - <https://medium.com/katalon-studio/a-sample-web-automation-test-project-9c532237c2bd>
 - What did it say in “Step 4”?
- On your own
 - What Application?
 - <http://demoaut.katalon.com>
 - <http://store.demoqa.com/>
 - What is your “use case”
 - Add it as a record and playback



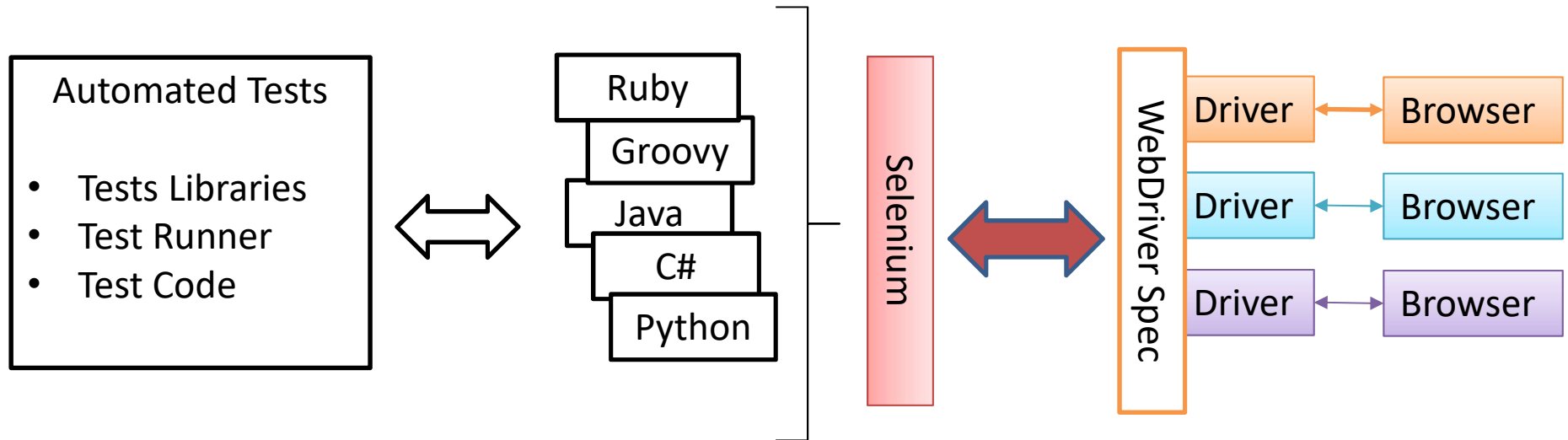
QUIZ

What questions help you create a test strategy to improve your test automation success?

- A. ____ What does my boss need fixed?
- B. ____ How does your business make money (or generate value for the end-user)?
- C. ____ Do I have enough time to write code for automation?
- D. ____ How do your users use your application?
- E. ____ What things have broken in the application before?
- F. ____ Can I get a developer to write a Unit Test?



Browser Tool Stack



- Browser Vendors create automation drivers which conform to the W3C WebDriver Specification
- Selenium has bindings for lots of programming languages and talks to the browser specific driver
- Automated Tests are “test libraries,” a “test runner,” and “test code.”



Elements of the tool stack are:

- Client (application in a browser)
 - Product under test
 - Browser and version
- Interact with the Browser
 - WebDriver / Selenium
 - “Web Driver Wrapper” (WATIR / Capybara)
- Test Code
 - How does a test look like a use case?
 - How do you separate what changes from what doesn’t change?
 - What do you have for asserts?
- Test Harness / Test Runner
 - How to select tests to run
 - How to verify/assert success or failure
 - Test execution log
 - Jenkins Job
- Test Runs
 - Test result reporting
 - Test suite execution log
- Test Repository
 - Test Rail



Where is the tool stack?

Item	Object	Input
→ 1 - Open Browser		""
→ 2 - Navigate To Url		"https://katalon-demo-cura.herokuapp.com/"
→ 3 - Click	a_Make Appointment	
→ 4 - Set Text	input_Username_username	"John Doe"
→ 5 - Set Encrypted Text	input_Password_password	"mnkJqkPPKg1friGnJQNHA=="
→ 6 - Click	button_Login	
→ 7 - Verify Element Text	p_Login failed Please ensure the username	"Login failed! Please ensure the username and password are valid."
→ 8 - Click	i_CURA Healthcare_fa fa-bars	
→ 9 - Click	a_Home	
→ 10 - Close Browser		

```
⊕ import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint
⊖ WebUI.openBrowser('')
WebUI.navigateToUrl('https://katalon-demo-cura.herokuapp.com/')
WebUI.click(findTestObject('Object Repository/Page_CURA Healthcare Service/a_Make Appointment'))
WebUI.setText(findTestObject('Object Repository/Page_CURA Healthcare Service/input_Username_username'), 'John Doe')
WebUI.setEncryptedText(findTestObject('Object Repository/Page_CURA Healthcare Service/input_Password_password'), 'mnkJqkPPKg1friGnJQNHA==')
WebUI.click(findTestObject('Object Repository/Page_CURA Healthcare Service/button_Login'))
WebUI.verifyElementText(findTestObject('Object Repository/Page_CURA Healthcare Service/p_Login failed Please ensure the u
    'Login failed! Please ensure the username and password are valid.'))
WebUI.click(findTestObject('Object Repository/Page_CURA Healthcare Service/i_CURA Healthcare_fa fa-bars'))
WebUI.click(findTestObject('Object Repository/Page_CURA Healthcare Service/a_Home'))
WebUI.closeBrowser()
```




Become a developer...

- Fork a GitHub repository
 - Login to your GitHub Account
 - Find my repo: <https://github.com/agodwin/v4q.git>
 - Click the “Fork” Button
 - A copy of my repo in your project
 - Clone your GitHub repo
 - A local repository you can edit and “push” to GitHub
- What did you just get?
 - Look at files with GUI
 - Open terminal and look at file in CLI
 - Look at files with Text Editor



Text Editor Check / Git Check

- Open the v4q/hello.txt file
- Add some text
- Save the file
- Commit the file (local repo)
 - git status
 - git add
 - git commit -m “Look ma, I can code”
- Push the changes (remote repo)
 - git status
 - git push
- Check GitHub for the changes



Ruby and Watir

- Check Ruby Version
 - `ruby -version`
- Check Chrome Version
- Check 'chromedriver'
 - Is 'chromedriver' in the PATH for CLI?
 - `chromedriver --version`
 - Is 'chromedriver' compatible with Chrome
 - Update to new 'chromedriver'



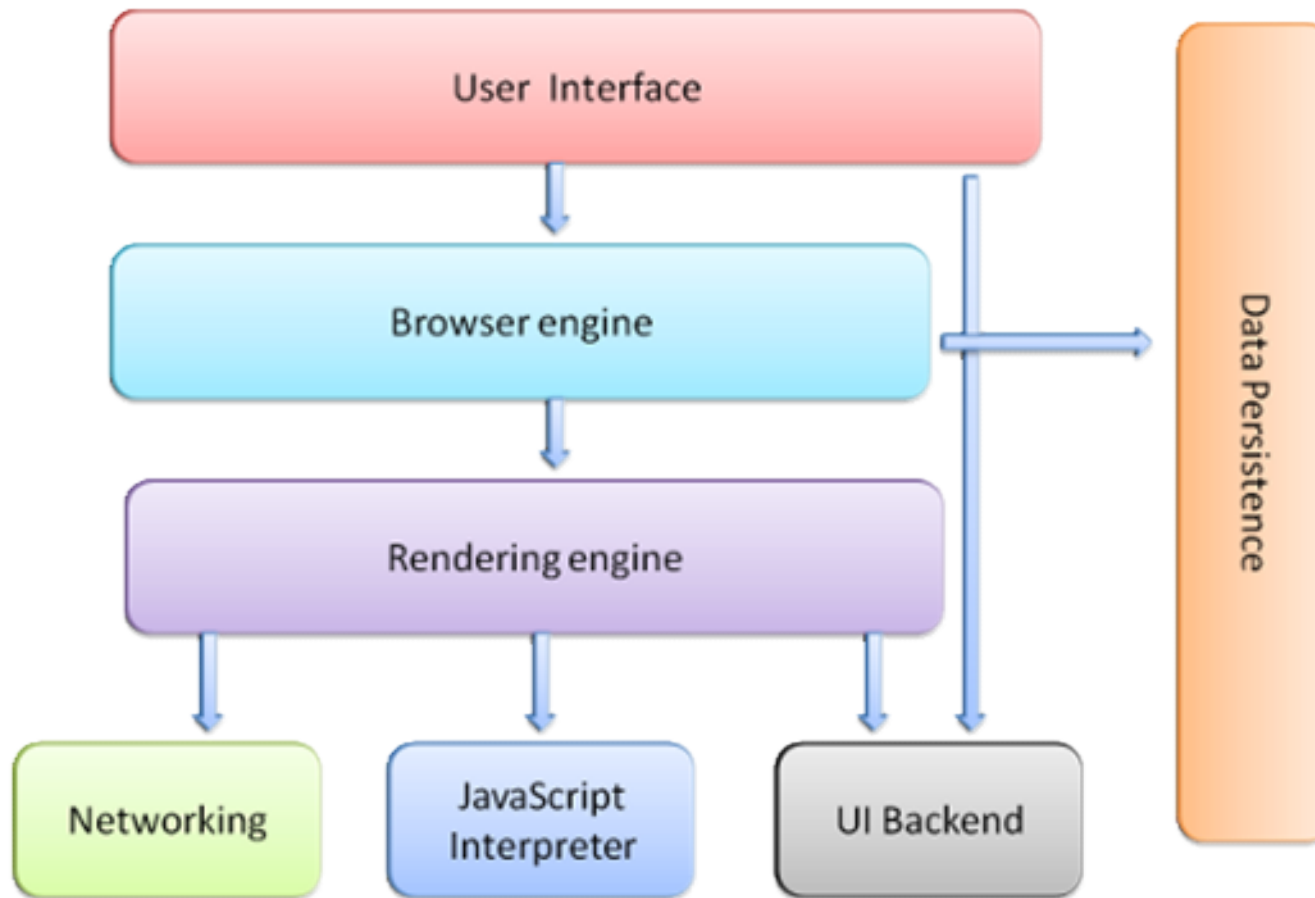
Ruby and Watir

- Add new libraries (gems in Ruby)
 - `gem -version`
 - `gem list`
 - `gem install watir`
 - `gem install rspec`
 - `gem install faker`
 - `gem install ffi`
 - `gem list`
- Open 'watir-examples.rb' in text editor
- Run the watir samples
 - `cd <local-repo-location>/v4q`
 - `ruby watir-examples.rb`



Browser Components

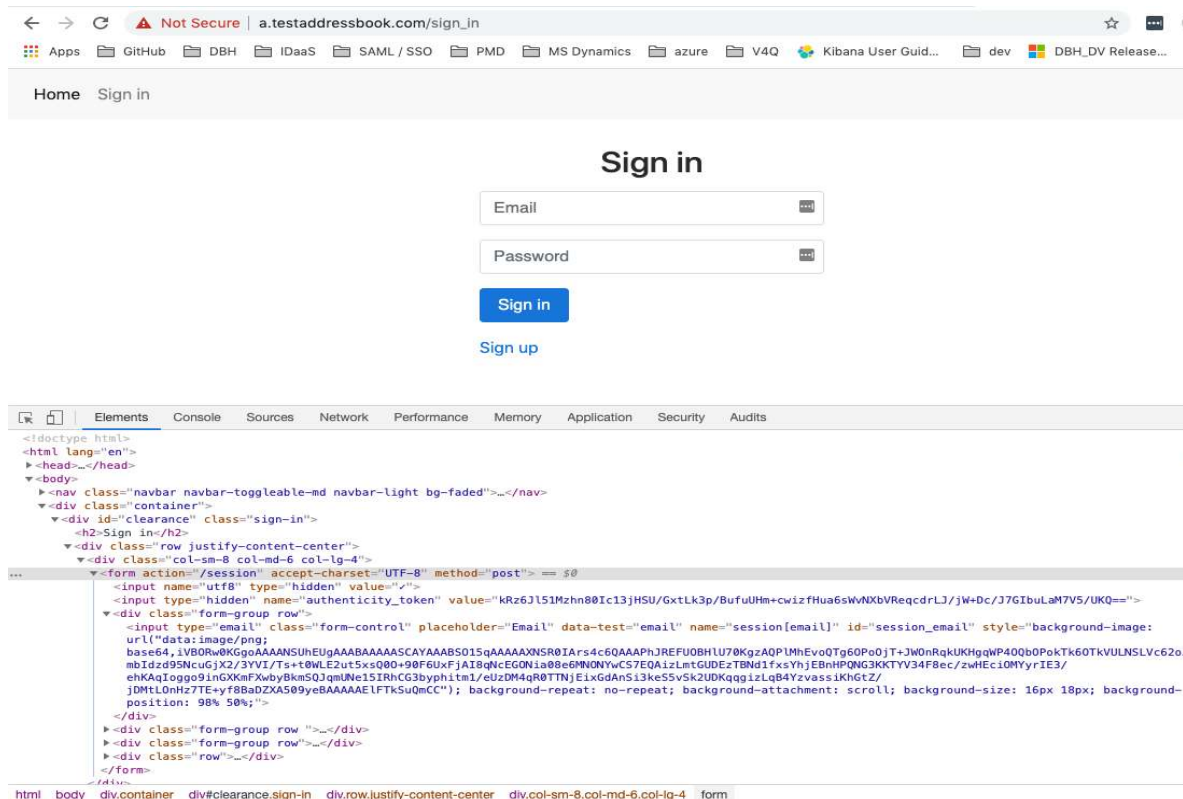
A picture is worth 1000 words





Demo Browser Developer Tools (again)

- Menu > More Tools > Developer Tools
- Dock at the Bottom





Ruby, Watir, and IRB

- IRB – Interactive Ruby
- Learn Watir and explore HTML/DOM
- Run the watir samples (V2)
 - `cd <local-repo-location>/v4q`
 - Open 'watir-examples.rb' in text editor
- Launch IRB and Load Watir “gem”
 - `irb`
 - `require 'watir'`

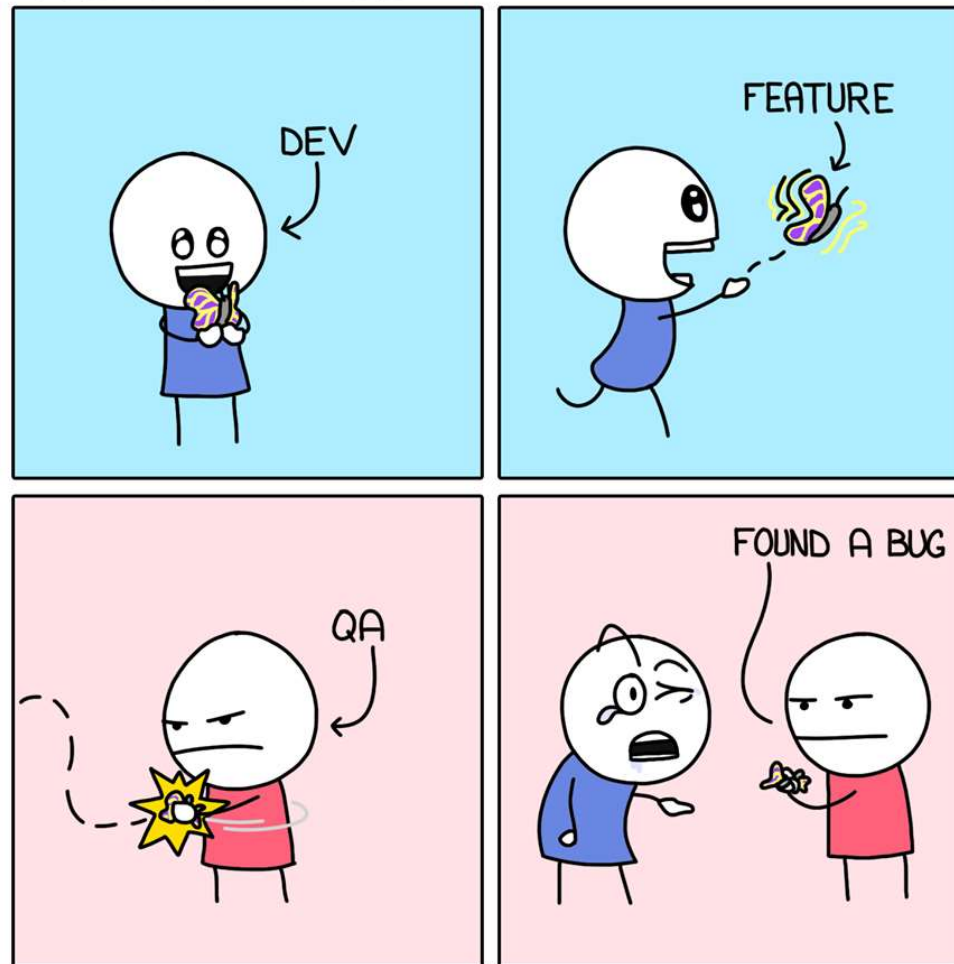


Question?

- Continuous to Learning:
 - Repeat the IRB exercise with:
 - <http://demoaut.katalon.com>
 - <http://store.demoqa.com/>
 - Try and watch Lynda / Linked-in Learning
 - <https://www.linkedin.com/learning/>
 - “Ruby Testing with RSPEC”
 - Introduction and sections 1, 2, 3, 4, and 5
 - Ruby Programming Language
 - <http://ruby-doc.com/docs/ProgrammingRuby/>



THE STRUGGLE



MONKEYUSER.COM