

# SQDG

## AUTOMATION SHOWCASE

*“Besides black art, there is only automation  
and mechanization. Federico Garcia Lorca”*

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# THANKS TO OILDEX FOR HOSTING

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# ROB EDWARDS

# PROTRACTOR / PAGE OBJECTS

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara

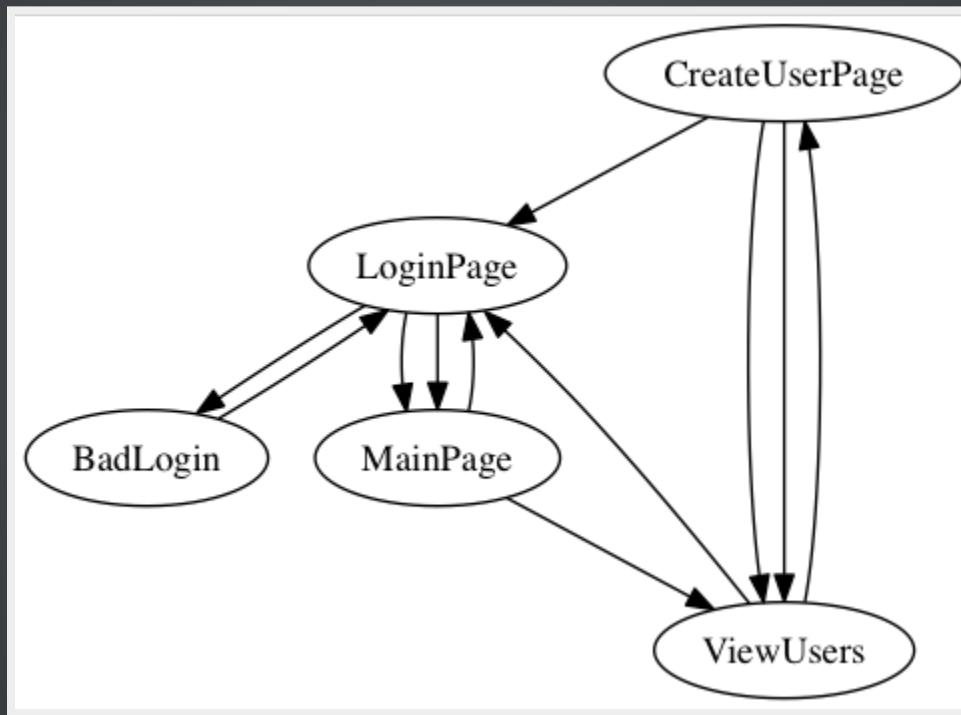
# **PAUL ROGERS**

## **MODEL BASED TESTING**

Github example

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

See - [How to Build Your Own Robot Army](#) for an intro into Model Based Testing



- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara

# go.rb

```
require 'model'
require 'fileutils'
require 'selenium-webdriver'

def browser
    $browser
end

$browser = Selenium::WebDriver.for(:firefox )
$new_users = []

require_relative "login_page_controls"
require_relative "create_users_page"
require_relative "bad_login"
require_relative "login_page"
require_relative "main_page"
require_relative "view_users"
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# bad\_login.rb

```
class BadLogin < Model::WebNode

  transition :goto_login , :goes_to => :LoginPage

  include LoginPageControls

  def goto_login
    browser.find_element(:id => "logout").click
  end

  def on_page?
    error_section.displayed? and error_section.text == "Username was incorrect"
  end
end
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# create\_users\_page.rb

```
class CreateUserPage < Model::WebNode

  transition :logout, :goes_to => :LoginPage
  transition :view_users, :goes_to => :ViewUsers
  transition :create_user, :goes_to => :ViewUsers , :verification_method

  def logout
    browser.find_element(:id => "logout").click
  end

  def view_users
    browser.find_element(:id => "users").click
  end

  def create_user( name = nil)
    if name.nil?
      name = "New_" + Time.now.to_i.to_s
    end
  end

```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# login\_page.rb

```
require_relative 'login_page_controls'
require_relative 'main_page'

class LoginPage < Model::WebNode

  transition :bad_login, :goes_to => :BadLogin
  transition :login, :goes_to => :MainPage
  transition :login_with_created_user, :goes_to => :MainPage, :if => :any_user?
  has_url "/"

  include LoginPageControls

  def bad_login
    login("bad_" + Time.now.to_i.to_s)
  end

  def any_users_created?
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# login\_page\_controls.rb

```
module LoginPageControls

  def username_text_field
    browser.find_element(:id => "username" )
  end

  def login_button
    browser.find_element(:id => "login" )
  end

  def title_section
    browser.find_element(:id => "title")
  end

  def error_section
    browser.find_element(:id => "page_body")
  end
end
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# main\_page.rb

```
require_relative "login_page"

class MainPage < Model::WebNode

  transition :logout , :goes_to => :LoginPage
  transition :view_users , :goes_to => :ViewUsers

  def logout
    browser.find_element(:id => "logout").click
  end

  def view_users
    browser.find_element(:id => "users").click
  end

  def on_page?
    message_span = browser.find_element(:id => "message")
    ( message_span.displayed? and message_span.text.match(/You are now s|
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# view\_users.rb

```
require_relative 'login_page'
require_relative 'create_users_page'

class ViewUsers < Model::WebNode

  transition :logout , :goes_to => :LoginPage
  transition :create_user , :goes_to => :CreateUserPage

  def logout
    browser.find_element(:id => "logout").click
  end

  def create_user
    browser.find_element(:id => "create_user").click
  end

  def list_users
    all = [
      
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

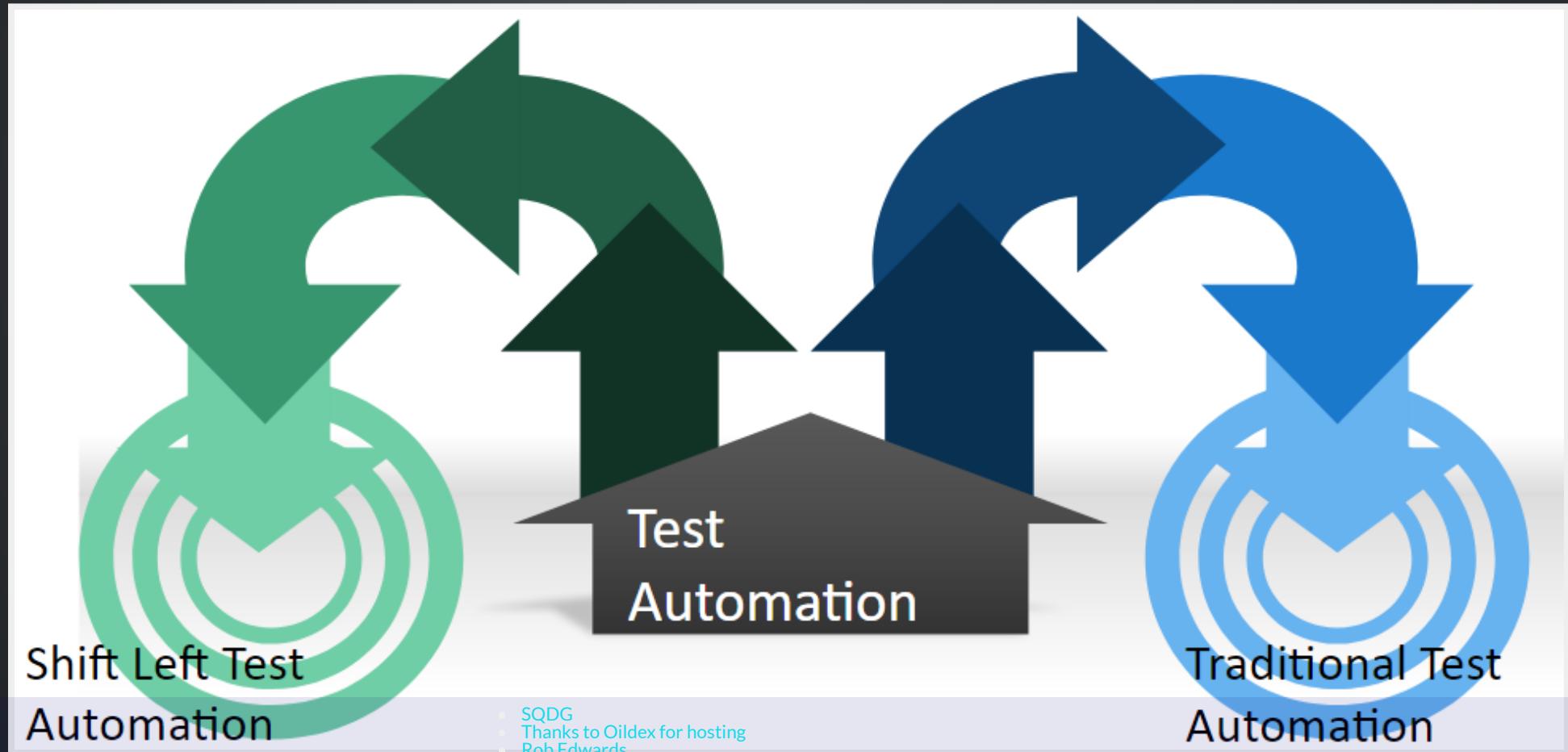
# PRANJAL DUTTA

## NEXT GENERATION FUNCTIONAL TEST AUTOMATION FRAMEWORK

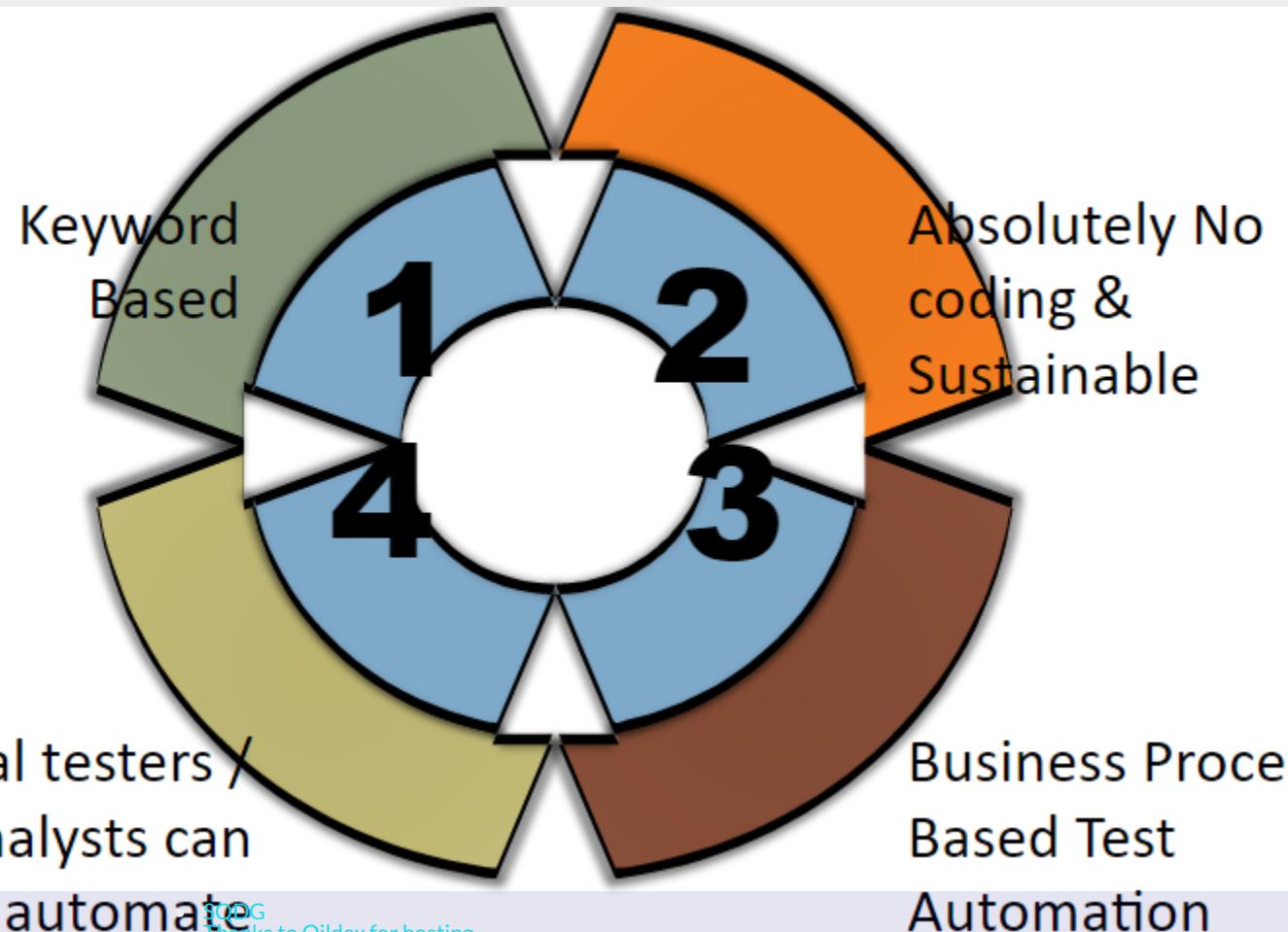
- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara

# CURRENT INDUSTRY TRENDS – DEVOPS

**AUTOMATE → AUTOMATE → AUTOMATE = BUT SUSTAINABLE**



# KEY FEATURES



# CAPYBARA

*Acceptance test framework for web  
applications*

---

Website <http://teamcapybara.github.io/capybara/>

---

Language ruby

---

Drivers [poltergeist/phantomJS](#), webkit

---

Frameworks selenium, RackTest

- [SQDG](#)
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- [Capybara](#)
- ...

# INSTALLATION

- GemFile and Bundler
- Drivers can be installed locally

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# CONFIG/FOLDER STRUCTURE

```
.  
| project  
|   cucumber.yml  
|   Gemfile  
|   Gemfile.lock  
|  
\---features  
    |   login.feature  
    |  
    +---step_definitions  
    |       login_steps.rb  
    |  
    \---support  
        env.rb
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# EXAMPLE

## env.rb

```
require 'capybara'
require 'capybara/cucumber'
require 'capybara/poltergeist'
require 'capybara/rspec'

Capybara.run_server          = false
Capybara.default_driver      = :poltergeist
Capybara.default_max_wait_time = 30
Capybara.app_host
= "https://automation-showcase-app.herokuapp.com"
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# login.feature

Feature: Verify that the correct users can login

In order to verify that the test app login works

As a test user

I need to sign in as each of the test users

Scenario Outline: Sign in, sign out

Given I can sign in as <user>

Then I should be signed in and see <message>

And I can sign out

Examples:

| user | message |

| alpha | You are now signed |

| test | You are now signed |

| error | Username was incorrect. |

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# login\_steps.rb

```
Given %r{^I can sign in as (.*)$} do |user|
  visit "https://automation-showcase-app.herokuapp.com"

  expect(page).to have_title 'Test App'

  user == 'test' ? user = ' test ' : user
  fill_in 'username', :with => user
  click_button 'Login'
end

Then %r{^I should be signed in and see (.*)$} do |message|
  expect(page).to have_content message
end

Then %r{^I can sign out$} do
  click_link 'logout'
end
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# SUMMARY

- Simple installation
- Simple setup
- Can get large and difficult to manage
- Can end up with lots of duplicated code
- Works well with JavaScript and waiting

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# CAPYBARA SITE\_PRISM

*Acceptance test framework for web  
applications*

---

Website <http://teamcapybara.github.io/capybara/>

---

Language ruby

---

Drivers poltergeist/phantomJS, webkit

---

Framework selenium, RackTest, Page-object

- [SQDG](#)
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# INSTALLATION

- GemFile and Bundler
- Drivers can be installed locally

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# CONFIG/FOLDER STRUCTURE

```
.  
project  
|   cucumber.yml  
|   Gemfile  
|   Gemfile.lock  
|   readme.md  
|  
\---features  
|   login.feature  
|  
+---pages  
|   error_page.rb  
|   home_page.rb  
|   index_page.rb  
|  
+---step_definitions  
|   login_steps.rb  
|
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# EXAMPLE

## env.rb

```
require 'capybara'
require 'capybara/cucumber'
require 'capybara/poltergeist'
require 'site_prism'

Capybara.run_server
  = false
Capybara.default_driver          = :poltergeist
Capybara.default_max_wait_time  = 30
Capybara.app_host
  = "https://automation-showcase-app.herokuapp.com"
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# login.feature

Feature: Verify that the correct users can login  
In order to verify that the test app login works  
As a test user  
I need to sign in as each of the test users

Scenario Outline: Sign in, sign out  
Given I can sign in as <user>  
Then I should be signed in and see <message>  
And I can sign out

Examples:

user	message
alpha	You are now signed!
test	You are now signed!
error	Username was incorrect.

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# login\_steps.rb

```
Given %r{^I can sign in as (.*)\$} do |user|
  user == 'test' ? user = ' test ' : user

  @index = IndexPage.new
  @index.load
  expect(@index.title).to eq 'Test App'

  @index.username.set(user)
  @index.login.click
end

Then %r{^I should be signed in and see (.*)\$} do |message|
  @logged_in = message =~ /You are now signed/ ? true : false
  @page      = @logged_in ? HomePage.new : ErrorPage.new
  expect(@page.message.text).to match(message)
end

Then %r{^I can sign out\$} do
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# index\_page.rb

```
class IndexPage < SitePrism::Page

  set_url "https://automation-showcase-app.herokuapp.com"

  element :login,      '#login'
  element :username,   '#username'

end
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

## error\_page.rb

```
class ErrorPage < SitePrism::Page

  set_url "https://automation-showcase-app.herokuapp.com/error"

  element :message, '#message'
  element :logout, '#logout'

end
```

- [SQDG](#)
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- [Capybara](#)

# home\_page.rb

```
class HomePage < SitePrism::Page

  set_url "https://automation-showcase-app.herokuapp.com/home"

  element :message, '#message'
  element :logout, '#logout'

end
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# SUMMARY

- Simple installation
- Simple setup
- Good for JS apps and ajax
- Page objects are easy to create
- Good default checker configuration

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# GHOSTINSPECTOR

*Automated, UI Testing, Browser Testing, End-to-end Testing, Visual Testing*

---

Website <https://ghostinspector.com/>

---

Language Web interface, selenium json import, chrome recorder

---

Drivers CasperJS?

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara
- ..

# INSTALLATION

- Install the chrome recorder if you want

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# SUMMARY

- Simple installation for Chrome
- Simple setup
- Can get large and difficult to manage
- Can end up with lots of duplicated code
- Nice reports
- Can be a bit slow as waiting isn't good

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# NIGHTWATCH.JS

*easy to use Node.js based E2E testing solution  
for browser based apps and websites*

---

Website <http://nightwatchjs.org/>

---

Language JavaScript (Node.js)

---

Drivers Selenium/WebDriver

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara
- ..

# INSTALLATION

- npm

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# CONFIG/FOLDER STRUCTURE

```
.  
project  
|   nightwatch.json  
|  
+---reports  
|  
\---tests  
    test.js
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# EXAMPLE

## nightwatch.json

```
{  
  "src_folders" : ["tests"],  
  "output_folder" : "reports",  
  "custom_commands_path" : "",  
  "custom_assertions_path" : "",  
  "page_objects_path" : "",  
  "globals_path" : "",  
  
  "selenium" : {  
    "start_process" : true,  
    "cli_args" : {  
      "webdriver.chrome.driver" : "./Users/clancey/dev/src/vindex/path/chromedriver"  
    }  
  },  
  
  "test_settings" : {  
    "default" : {  
      "selenium_port" : 4444,  
      "selenium_host" : "localhost",  
      "silent" : true  
    }  
  }  
}
```

• SQDG9515  
• Thanks to Oildex for hosting  
• Rob Edwards  
• Paul Rogers  
• Pranjal Dutta  
• Capybara ..

# test.js

```
module.exports = {
  'Verify title' : function (browser) {
    browser
      .url('https://automation-showcase-app.herokuapp.com')
      .waitForElementVisible('body', 1000)
      .assert.title('Test App');
  },
  'Valid user alpha can log in' : function (browser) {
    browser
      .clearValue('#username')
      .setValue('#username', 'alpha')
      .click('#login')
      .pause(1000)
      .assert.containsText('#message', 'You are now signed')
      .click('#logout');
  },
}
```

- [SQDG](#)
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- [Capybara](#)

# SUMMARY

- Took a bit to get it configured
- Seems a bit verbose

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# PROTRACTOR

*an e2e test framework for Angular*

---

Website	<a href="http://www.protractortest.org/#/">http://www.protractortest.org/#/</a>
Language	JavaScript (Node.js)
Drivers	Selenium/WebDriver, SauceLabs
Framework	Jasmine

---

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara
- ..

# INSTALLATION

- npm

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# CONFIG/FOLDER STRUCTURE

```
.  
project  
|   conf.js  
|   spec.js
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# EXAMPLE

## conf.js

```
exports.config = {
  framework: 'jasmine',
  seleniumAddress: 'http://localhost:4444/wd/hub',
  specs: ['spec.js'],
  getPageTimeout: 15000,
  jasmineNodeOpts: {defaultTimeoutInterval: 15000}
};
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# spec.js

```
describe('Automation Showcase App', function() {
  it('should have a title', function() {
    browser.waitForAngularEnabled(false);

    browser.get('https://automation-showcase-app.herokuapp.com/');

    expect(browser.getTitle()).toMatch('Test App');
  });
  it('should not allow a login for invalid users', function() {
    browser.waitForAngularEnabled(false);

    browser.get('https://automation-showcase-app.herokuapp.com/');

    element(by.id('username')).clear().then(function() {
      element(by.id('username')).sendKeys('error');
    });

    element(by.id('login')).click().then(function() {
      expect(element(by.id('error')).getText()).toEqual('error');
    });
  });
});
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# SUMMARY

- Took a bit to get it configured
- Seems a bit verbose
- The way that it handles promises is different from other automation libraries, it is more in line with Angular development.

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# ROBOT FRAMEWORK

*Generic test automation framework for  
acceptance testing and ATDD*

---

Website <http://robotframework.org/>

---

Language python

---

Drivers too many to list

- SQDG
- Thanks to Oiledx for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# INSTALLATION

## • PIP

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# CONFIG/FOLDER STRUCTURE

```
.  
project  
\---tests  
    invalid_login.robot  
    resource.robot  
    valid_login.robot
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# EXAMPLE

## resource.robot

```
*** Settings ***
Documentation      A resource file with reusable keywords and variables.
...
...
The system specific keywords created here form our own
...
domain specific language. They utilize keywords provided
...
by the imported Selenium2Library.
Library           Selenium2Library

*** Variables ***
${SERVER}          automation-showcase-app.herokuapp.com
${BROWSER}         Chrome
${DELAY}            0
${VALID USER}      alpha
${LOGIN URL}       http://${SERVER} /
${HOME URL}        https://${SERVER}/home
• Thanks to Oiledx for hosting
• Rob Edwards
• Paul Rogers
• Pranjal Dutta
• Capybara ..
```

# invalid\_login.robot

```
*** Settings ***
Documentation      A test suite with a single test for invalid login.
...
...
    This test has a workflow that is created using keywords in
...
    the imported resource file.
Resource          resource.robot

*** Test Cases ***
Valid Login
    Open Browser To Login Page
    Input Username      error
    Submit Credentials
    Error Page Should Be Open
    Verify Message      Username was incorrect.
Logout
[Tearardown]      Close Browser
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# 'valid\_login.robot'

```
*** Settings ***
Documentation      A test suite with a single test for valid login.
...
...
    This test has a workflow that is created using keywords in
...
    the imported resource file.
Resource          resource.robot

*** Test Cases ***
Valid Login
    Open Browser To Login Page
    Input Username    alpha
    Submit Credentials
    Welcome Page Should Be Open
    Verify Message    You are now signed
    Logout
    Login Page Should Be Open
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# SUMMARY

- Simple to configure and setup
- Python format is quick and easy to learn
- Seems like a lot of *magic* going on (I kinda like it!)

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# WATIR

*An open source Ruby library for automating tests*

---

Website <http://watir.github.io/>

---

Language ruby

---

Drivers Selenium/WebDriver, SauceLabs, PhantomJS

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara
- ..

# INSTALLATION

\*

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# CONFIG/FOLDER STRUCTURE

```
.  
project  
|   cucumber.yml  
|   Gemfile  
|   Gemfile.lock  
|  
\---features  
|   login.feature  
|  
+---step_definitions  
|       login_steps.rb  
|  
\---support  
  
env.rb  
  
hooks.rb
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# EXAMPLE

## env.rb

```
require 'watir'

$browser = Watir::Browser.new :chrome
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# hooks.rb

```
Before do
  @browser = $browser
end

at_exit do
  $browser.close
end
```

- [SQDG](#)
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- [Capybara](#) ..

# login.feature

Feature: Verify that the correct users can login

In order to verify that the test app login works

As a test user

I need to sign in as each of the test users

Scenario Outline: Sign in, sign out

Given I can sign in as <user>

Then I should be signed in and see <message>

And I can sign out

Examples:

| user | message |

| alpha | You are now signed |

| test | You are now signed |

| error | Username was incorrect. |

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# login\_steps.rb

```
Given %r{^I can sign in as (.*)$} do |user|
  @browser.goto('https://automation-showcase-app.herokuapp.com')

  expect(@browser.title).to eq('Test App')

  user == 'test' ? user = ' test ' : user
  @browser.text_field(id: 'username').set(user)
  @browser.button(id: 'login').click
end
```

```
Then %r{^I should be signed in and see (.*)$} do |message|
  page_message = @browser.span(id: 'message').text
  expect(page_message).to match(message)
end
```

```
Then %r{^I can sign out$} do
  logout = @browser.link(id: 'logout')
  expect(logout.present?).to be true
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# SUMMARY

- Simple installation and setup
- Can get large and difficult to manage
- Can end up with lots of duplicated code
- Works well with javascript and waiting

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# WATIR WITH PAGE-OBJECT

*An open source Ruby library for automating tests*

---

Website <http://watir.github.io/>

---

Language ruby

---

Drivers Selenium/WebDriver, SauceLabs, PhantomJS

---

Framework Page-object

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara
- ...

# INSTALLATION

- bundler

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# CONFIG/FOLDER STRUCTURE

```
.  
project  
|   cucumber.yml  
|   debug.log  
|   Gemfile  
|   Gemfile.lock  
|  
\---features  
|   login.feature  
|  
+---pages  
|   error_page.rb  
|   home_page.rb  
|   index_page.rb  
|  
+---step_definitions  
|   login_steps.rb  
|
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# EXAMPLE

## env.rb

```
require 'page-object'
require 'page-object/page_factory'
require 'watir'

World(PageObject::PageFactory)

$browser = Watir::Browser.new :chrome
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# hooks.rb

```
Before do
  @browser = $browser
end

at_exit do
  $browser.close
end
```

- [SQDG](#)
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- [Capybara](#) ..

# login.feature

Feature: Verify that the correct users can login

In order to verify that the test app login works

As a test user

I need to sign in as each of the test users

Scenario Outline: Sign in, sign out

Given I can sign in as <user>

Then I should be signed in and see <message>

And I can sign out

Examples:

| user | message |

| alpha | You are now signed |

| test | You are now signed |

| error | Username was incorrect. |

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# login\_steps.rb

```
Given %r{^I can sign in as (.*)$} do |user|
  user == 'test' ? user = ' test ' : user

  visit(IndexPage) do |page|
    page.has_expected_title?
    page.username = user
    page.login
  end
end

Then %r{^I should be signed in and see (.*)$} do |message|
  @logged_in = message =~ /You are now signed/ ? true : false
  page_message = @logged_in ? on(HomePage).message : on(ErrorPage).message
  expect(page_message).to match(message)
end

Then %r{^I can sign out$} do
  @logged_in ? on(HomePage).logout : on(ErrorPage).logout
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# index\_page.rb

```
class IndexPage

  include PageObject

  page_url "https://automation-showcase-app.herokuapp.com"
  expected_title = "Test App"

  button :login, id: 'login'
  text_field :username, id: 'username'

end
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# error\_page.rb

```
class ErrorPage

  include PageObject

  page_url "https://automation-showcase-app.herokuapp.com/error"

  span :message, id: 'message'
  link :logout, id: 'logout'

end
```

- [SQDG](#)
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- [Capybara](#)

# home\_page.rb

```
class HomePage

  include PageObject

  page_url "https://automation-showcase-app.herokuapp.com/home"

  span :message, id: 'message'
  link :logout, id: 'logout'

end
```

- [SQDG](#)
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- [Capybara](#)

# SUMMARY

- Simple installation and setup
- Can take some time to setup pages
- It can enable TDD
- Works well with javascript and waiting

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# WEBDRIVER.IO

*WebDriver bindings for Node.js*

---

Website <http://webdriver.io>

---

Language JavaScript (Node.js)

---

Drivers webdriver

---

Framework Jasmine, Mocha, Cucumber

- [SQDG](#)
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- [Capybara](#)
- [Selenium](#)
- [PhantomJS](#)
- [Geckodriver](#)
- [Chromedriver](#)
- [IEDriverServer](#)
- [SafariDriver](#)
- [AndroidDriver](#)
- [iOSDriver](#)
- [Appium](#)
- [Karma](#)
- [Mocha](#)
- [Jasmine](#)
- [Cucumber](#)
- [Protractor](#)
- [Nightwatch](#)
- [Selenium Grid](#)
- [Selenium RC](#)
- [Selenium IDE](#)
- [Selenium WebDriver](#)
- [Selenium Grid](#)
- [Selenium RC](#)
- [Selenium IDE](#)
- [Selenium WebDriver](#)

# INSTALLATION

- npm

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# CONFIG/FOLDER STRUCTURE

```
.  
project  
|   readme.md  
|   wdio.conf.js  
|  
+---errorShots  
|  
+---node_modules  
|  
\---test  
    \---specs  
  
test.js
```

*Note* The node\_modules folder contains all of the installation files, not listed here.

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# EXAMPLE

## wdio.conf.js

```
exports.config = {  
  specs: [  
    './test/specs/**/*.js'  
,  
  exclude: [  
],  
  maxInstances: 10,  
  capabilities: [{  
    maxInstances: 5,  
    browserName: 'chrome'  
}],  
  sync: true,  
  logLevel: 'silent',  
  coloredLogs: true,  
  bail: 0,  
  screenshotPath: './errorShots/',  
  baseUrl: 'https://automation-showcase-app.herokuapp.com',  
  waitforTimeout: 10000  
};
```

- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara

# testc.js

```
var assert = require('assert');
describe('Automation Showcase App', function() {
  it('should have a title', function () {
    browser.url('https://automation-showcase-app.herokuapp.com');
    var title = browser.getTitle();
    assert.equal(title, 'Test App');
  });
  it('should not allow a login for invalid users', function() {
    browser.url('https://automation-showcase-app.herokuapp.com/');
    browser.clearElement('#username');
    browser.setValue('#username', 'error');
    browser.click('#login');
    var message = browser.getText('#message');
    assert.equal(message, 'Username was incorrect.');
  });
  it('should allow a login for valid user test', function() {
    browser.url('https://automation-showcase-app.herokuapp.com/');
    browser.clearElement('#username');
```

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# SUMMARY

- Simple installation and setup
- Can get large and difficult to manage (needs page objects)
- Can be a bit verbose

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..

# THANKS!

- SQDG
- Thanks to Oildex for hosting
- Rob Edwards
- Paul Rogers
- Pranjal Dutta
- Capybara ..