

Лекция 10. Тестирование веб-приложений

- Тесты серверного кода
 - Unit, RSpec,...
- Тесты на стороне браузера
 - curl, Selenium IDE
 - Ruby + Selenium Webdriver



Тестирование веб-приложений



09.11.21

- *Тестирование программ — проверка их функционирования в соответствии с заданными требованиями*
- *Тестирование функциональное*
 - *Модульное, интеграционное,...*
- *Тестирование не функциональное*
 - *Производительность, масштабируемость, предельные объемы,...*
- *Тестирование со стороны сервера*
- *Тестирование со стороны браузера*
- *...*

Тестирование Rails-приложения



09.11.21

- Серверное тестирование:
 - моделей
 - контроллеров и представлений с помощью функциональных тестов
 - правил маршрутизации
 - сопряжений
- <http://guides.rubyonrails.org/testing.html>

Тестирование моделей



09.11.21

- Шаблон test/models/user_test.rb

```
require 'test_helper'
```

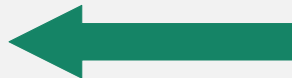
```
class UserTest < ActiveSupport::TestCase
```

```
  # test "the truth" do
```

```
    # assert true
```

```
  # end
```

```
end
```



Автоматически
созданная
заглушка

Тестирование моделей



09.11.21

```
require 'test_helper'
```

```
class UserTest < ActiveSupport::TestCase
  test 'should not save new empty User' do
    user = User.new
    assert !user.save
  end
```



Критерий
тестирования

```
  test 'should save not empty User' do
    user = User.new
    user.name = 'ivanov'
    user.email = 'ivanov@ivanov.com'
    assert user.save
  end
end
```



Критерий
тестирования

Добавление ограничений в модели (логическая целостность)



09.11.21

```
class User < ActiveRecord::Base  
  attr_accessible :email, :name
```

```
  validates :name, :presence => true
```



Проверить, что
атрибут **:name**
не пустой

```
end
```

Fixtures



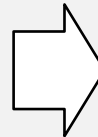
09.11.21

- “Арматура” данных

```
# fixtures/users.yml
```

```
david:
```

```
  name: David Heinemeier Hansson
  birthday: 1979-10-15
  profession: Systems development
```



```
steve:
```

```
  name: Steve Ross Kellock
  birthday: 1974-09-27
  profession: guy with keyboard
```

```
# Tests:
```

```
# this will return the User object
# for the fixture named david
```

```
users(:david)
```

```
# this will return the property
# for david called id
```

```
users(:david).id
```

```
# one can also access methods
# available on the User class
```

```
david = users(:david)
```

```
david.call(david.partner)
```

- <http://guides.rubyonrails.org/testing.html#the-low-down-on-fixtures>
- <http://api.rubyonrails.org/v5.1.4/classes/ActiveRecord/FixtureSet.html>

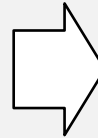
Factory Bot Rails



09.11.21

- Генераторы данных на Ruby для Ruby

```
# factories/users.rb
factory :user,
  aliases: [:author, :commenter] do
    first_name      "John"
    last_name       "Doe"
    date_of_birth { 18.years.ago }
  end
```



```
factory :post do
  author
  # instead of
  # association :author, factory: :user
  title "How to read a book effectively"
  body  "There are five steps involved."
end
```

```
# Tests:

# Returns a User instance that's not saved
user = build(:user)

# Returns a saved User instance
user = create(:user)

# Returns a hash of attributes that
# can be used to build a User instance
attrs = attributes_for(:user)

# Passing a block to any of the methods
# above will yield the return object
create(:user) do |user|
  user.posts.create(attributes_for(:post))
end
```

- https://github.com/thoughtbot/factory_bot_rails
- https://github.com/thoughtbot/factory_bot/blob/master/GETTING_STARTED.md#configure-your-test-suite

Тестирование контроллеров

Сгенерированный шаблон Rails < 5.1



09.11.21

- test/functional/calc_controller_test.rb

```
require 'test_helper'
```

```
class CalcControllerTest < ActionController::TestCase
```

```
  test "should get input" do
```

```
    get :input
```

```
    assert_response :success # Проверяет только факт выполнения
  end
```

```
  test "should get view" do
```

```
    get :view
```

```
    assert_response :success # Проверяет только факт выполнения
  end
```

```
end
```

Тестирование контроллеров

Сгенерированный шаблон Rails > 5.2



09.11.21

- test/functional/calc_controller_test.rb

```
require 'test_helper'
```

```
class CalcControllerTest < ActionDispatch::IntegrationTest
```

```
  test "should get input" do
```

```
    get calc_new_url
```

```
    assert_response :success # Проверяет только факт выполнения
  end
```

```
  test "should get edit" do
```

```
    get calc_view_url
```

```
    assert_response :success # Проверяет только факт выполнения
  end
```

```
end
```

Тестирование контроллеров

Функциональные тесты



09.11.21

```
require 'test_helper'

class UsersControllerTest <
  ActionDispatch::IntegrationTest
  setup do
    @user = users(:one)
  end

  test "should get index" do
    get users_url
    assert_response :success
  end

  test "should get new" do
    get new_user_url
    assert_response :success
  end

  test "should create user" do
    assert_difference('User.count') do
      post users_url, params: { user: { name:
        @user.name } }
    end

    assert_redirected_to user_url(User.last)
  end
```

```
  test "should show user" do
    get user_url(@user)
    assert_response :success
  end
```

```
  test "should get edit" do
    get edit_user_url(@user)
    assert_response :success
  end
```

```
  test "should update user" do
    patch user_url(@user),
      params: { user: { name: @user.name } }
    assert_redirected_to user_url(@user)
  end
```

```
  test "should destroy user" do
    assert_difference('User.count', -1) do
      delete user_url(@user)
    end

    assert_redirected_to users_url
  end
```

Тестирование контроллеров

Доступ к запросу и отклику



09.11.21

- **@controller** – контроллер, который обработал запрос
- **@request** – объект запроса
- **@response** – объект ответа

```
class ArticlesControllerTest < ActionDispatch::IntegrationTest
  test "should get index" do
    get articles_url

    assert_equal "index", @controller.action_name
    assert_equal "application/x-www-form-urlencoded", @request.media_type
    assert_match "Articles", @response.body
  end
end
```

- <http://guides.rubyonrails.org/testing.html#instance-variables-available>

Тестирование контроллеров

Доступ к переменным



09.11.21

```
class PostsController < ActionController::Base
  def index
    @posts = Post.all
  end
end
```

```
class PostControllerTest < ActionDispatch::IntegrationTest
  def test_index
    get post_index_url
    assert_equal Post.all, assigns(:posts)
  end
end
```

- В Rails 5 удалено из ядра. Необходимо добавить в Gemfile:
gem 'rails-controller-testing'
- <https://github.com/rails/rails-controller-testing>

Тестирование представлений

Функциональные тесты



09.11.21

- В составе теста контроллера

```
require 'test_helper'
```

```
class CalcControllerTest < ActionDispatch::IntegrationTest
  ...
end
```

```
  test "should get Unknown result" do
    get calc_view_url
    assert_select 'p#result', 'Unknown!'
  end
end
```

Тестирование представлений

`assert_select`



09.11.21

- `assert_select(selector, [equality], [message])`
- `assert_select(element, selector, [equality], [message])`

```
assert_select 'title', "Welcome to Rails Testing Guide"
```

```
assert_select 'ul.navigation' do  
  assert_select 'li.menu_item'  
end
```

```
assert_select "ol" do |elements|  
  elements.each do |element|  
    assert_select element, "li", 4  
  end  
end
```

```
assert_select "ol" do  
  assert_select "li", 8  
end
```

Тестирование правил маршрутизации



09.11.21

```
test "should route to post" do
  assert_routing '/posts/1',
    { :controller => "posts", :action => "show", :id => "1" }
end
```

- `assert_generates`
- `assert_recognizes`
- `assert_routing`
- <http://api.rubyonrails.org/classes/ActionDispatch/Assertions/RoutingAssertions.html>

Тестирование правил маршрутизации



09.11.21

```
class RoutingAccountTest < ActionController::IntegrationTest
  def test_account
    ["get", "post"].each do |method|
      assert_routing(
        { :method => method, :path => "/login" },
        { :controller => 'account', :action => 'login' }
      )
    end
    assert_routing(
      { :method => 'get', :path => "/logout" },
      { :controller => 'account', :action => 'logout' }
    )
    ...
  end
end
```

- <http://www.redmine.org/> -
http://www.redmine.org/projects/redmine/repository/changes/trunk/test/integration/routing/account_test.rb

Интеграционный тест

Связывание нескольких контроллеров



09.11.21

```
require 'test_helper'
class UserFlowsTest < ActionDispatch::IntegrationTest
  fixtures :users

  test "login and browse site" do
    https!           # login via https
    get "/login"
    assert_response :success

    post_via_redirect "/login", :username => users(:avs).username,
                                :password => users(:avs).password
    assert_equal '/welcome', path
    assert_equal 'Welcome avs!', flash[:notice]

    https!(false)
    get "/posts/all"
    assert_response :success
    assert assigns(:products)
  end
end
```

- <http://guides.rubyonrails.org/testing.html#integration-testing>

Функциональный тест для ActionMailer



09.11.21

```
require 'test_helper'

class UserControllerTest < ActionMailer::TestCase
  test "invite friend" do
    assert_difference 'ActionMailer::Base.deliveries.size', +1 do
      post :invite_friend, :email => 'friend@example.com'
    end
    invite_email = ActionMailer::Base.deliveries.last

    assert_equal "You have been invited by me@example.com",
      invite_email.subject
    assert_equal 'friend@example.com', invite_email.to[0]
    assert_match(/Hi friend@example.com/, invite_email.body)
  end
end
```

- <https://guides.rubyonrails.org/testing.html#testing-your-mailers>

Эмуляция внешнего сервера

Mock Objects



09.11.21

```
def expect_an_invalid_stop
  server = mock()
  server.expects(:CheckPlaceName).returns(0)
  MilesDavis.expects(:server).returns(server)
end
```

- <https://github.com/rspec/rspec-mocks>
- <https://github.com/freerange/mocha>
- <https://github.com/bblimke/webmock>
- <https://semaphoreci.com/community/tutorials/mocking-in-ruby-with-minitest>
- <https://mojomlingo.com/blog/2013/tests-mocks-rspec/>

Запуск тестов Rails



09.11.21

- rails test (ранее rake test)
- rails test:system
- rails db:test:prepare
- rake test:benchmark
- rake test:profile
- rails test:integration
- rails test:plugins
- rails test:units
-
- <https://guides.rubyonrails.org/testing.html>



- behavior driven development (BDD) framework
- domain-specific embedded language (DSEL)
- <http://rspec.info/>
- <http://rubydoc.info/gems/rspec-core>
- `gem install rspec`

Пример RSpec



09.11.21

```
# game_spec.rb
require 'bowling'
```

```
RSpec.describe Game do
  describe '#score' do
    it 'returns 0 for an all gutter game' do
      game = Game.new
      20.times { game.roll(0) }
      expect(game.score).to eq(0)
    end
  end
end
```

```
# game.rb

class Game
  def roll(pins)
  end

  def score
    0
  end
end
```

RSpec

Метод describe



09.11.21

- Группировка соответствующих по поведению ТЕСТОВ

```
describe BlogPost do
  it "does something" do
    blog_post = BlogPost.new :title => 'Hello'
    expect(blog_post).to ...
  end
  it "does something else" do
    blog_post = BlogPost.new :title => 'Hello'
    expect(blog_post).to ...
  end
end
```


RSpec

Метод let



09.11.21

```
describe BlogPost do
  let(:blog_post) { BlogPost.new :title => 'Hello' }
  it "does something" do
    expect(blog_post).to ...
  end
  it "does something else" do
    expect(blog_post).to ...
  end
end
```



```
$count = 0
```

```
describe "let" do
```

```
  let(:count) { print "'let 1'"; $count += 1 }
```

```
  let!(:count2) { print "'let 2!'; 1 }
```

```
  it "memoizes the value" do
```

```
    expect(count).to eq(1) # let вычисляет значение перед первым вызовом
```

```
    expect(count).to eq(1) # здесь то же значение count
```

```
  end
```

```
  it "is not cached across examples" do
```

```
    expect(count).to eq(2) # let вызывается второй раз для второго теста
```

```
    expect(count2).to eq(1) # let! уже вычислил значение ещё до обращения
```

```
  end
```

```
  it "is same as previous" do
```

```
    expect(count2).to eq(1)
```

```
  end
```

```
end
```

```
'let 2!'let 1!'let 2!'let 1!'let 2!'
```

RSpec

Метод context



09.11.21

- Эквивалент **describe**. Логически завершает описание

```
context "when there are no search results" do
  before do
    email_search_for(user, '123')
  end
  it "shows the search form" do
    expect(current_url).to == colleagues_url
  end
  it "renders an error message" do
    expect(response).to have_tag('.error',
      'No matching email addresses found.')
  end
end
```

RSpec

Метод specify



09.11.21

```
describe BlogPost do
  before { @blog_post = BlogPost.new :title => 'foo' }
  it "should not be published" do
    expect(@blog_post).not_to be_published
  end
end
```

- Замена before → let, it → specify

```
describe BlogPost do
  let(:blog_post) { BlogPost.new :title => 'foo' }
  specify { expect(blog_post).not_to be_published }
end
```

RSpec Expectations



09.11.21

- RSpec до версии 2.x включительно

`receiver.should(matcher)`

`receiver.should == expected`

`receiver.should =~ regexp`

- RSpec 3

`expect(actual).to ...`

`expect(actual).to_not ...`

RSpec

Примеры декларации ожидаемого значения



09.11.21

```
expect(actual).to eq(expected) # passes if actual == expected
expect(actual).to eql(expected) # passes if actual.eql?(expected)
expect(actual).not_to eql(not_expected) # passes if not(actual.eql?(expected))
expect(actual).to be(expected) # passes if actual.equal?(expected)
expect(actual).to equal(expected) # passes if actual.equal?(expected)
expect(actual).to be <= expected
expect(actual).to be < expected
expect(actual).to be_within(delta).of(expected)
expect(actual).to match(/expression/)
expect(actual).to be_xxx # passes if actual.xxx?
expect(actual).to have_xxx(:arg) # passes if actual.has_xxx?(:arg)
expect(actual).to include(expected)
expect(actual).to start_with(expected)
expect([1, 2, 3]).to contain_exactly(2, 3, 1)
expect([1, 2, 3]).to match_array([3, 2, 1])
...
```

- <http://www.rubydoc.info/gems/rspec-expectations/frames>

Подключение RSpec в Rails



09.11.21

- Gemfile должен содержать `gem 'rspec-rails'`
- `rails generate rspec:install`
- Установить переменную среды
`ENV["RAILS_ENV"] ||= 'test'`
- Вывод в формате **html**
`rspec <path_to_spec_file> --format html`

RSpec

Пример вывода в HTML



09.11.21

RSpec Code Examples

☒ Passed ☒ Failed ☒ Pending

PagesController

5 examples, 0 failures, 3 pending

Finished in **0.50000 seconds**

GET 'home'

returns http success

0.45312s

GET 'contact'

returns http success

0.01562s

pages/contact.html.erb

add some examples to (or delete) C:/Projects/scaffold_test1/spec/views/pages/contact.html.erb_spec.rb (PENDING: No reason given)

pages/home.html.erb

add some examples to (or delete) C:/Projects/scaffold_test1/spec/views/pages/home.html.erb_spec.rb (PENDING: No reason given)

PagesHelper

add some examples to (or delete) C:/Projects/scaffold_test1/spec/helpers/pages_helper_spec.rb (PENDING: No reason given)

Randomized with seed 63809

Cucumber



09.11.21

- behavior driven development (BDD) framework
- domain-specific language (DSL) Gherkin
- <http://cukes.info/>
- <https://github.com/cucumber/cucumber>
- `gem install cucumber`



Основные принципы Cucumber



09.11.21

- Сценарий теста на языке Gherkin
- Текст похож на естественный язык
- Поддерживается интернационализация
- Разбор действий, описанных на Gherkin, производится в Ruby-коде

Пример сценария



09.11.21

Feature: Some terse yet descriptive text of what is desired
Textual description of the business value of this feature
Business rules that govern the scope of the feature
Any additional information that will make the feature easier to understand

Scenario: Some determinable business situation
Given some precondition
And some other precondition
When some action by the actor
And some other action
And yet another action
Then some testable outcome is achieved
And something else we can check happens too

Scenario: A different situation

...

- <https://github.com/cucumber/cucumber/wiki/Gherkin>

Служебные слова Gherkin



09.11.21

- Feature – описание назначения теста
- Scenario – описание конкретного теста
 - Given - начальное состояние
 - When - действие пользователя
 - Then - наблюдаемый результат
- Для всех служебных слов есть переводы:
<https://github.com/cucumber/gherkin/blob/master/lib/gherkin/i18n.json>
 - "feature": "Функция|Функционал|Свойство",
 - "scenario": "Сценарий",
 - "given": "*|Допустим|Дано|Пусть",
 - "when": "*|Если|Когда",
 - "then": "*|То|Тогда",
 - "and": "*|И|К тому же|Также",
 - "but": "*|Но|А"
 - ...

Пример теста на русском языке



09.11.21

language: ru

Функция: 1.4 Открытие существующего проекта

У пользователя должна быть возможность открывать существующие проекты

@start

Сценарий: Вызвать окно открытия проекта через меню

Если я выбираю в выпадающем меню без скролла "1/1.png" пункт "1/2.png"

То я должен видеть "1/3.png"

И закрыть окно кликнув на "1/4.png"

Сценарий: Вызвать окно открытия проекта с панели инструментов

Если я кликаю на "1/5.png"

То я должен видеть "1/6.png"

Описание шагов на Ruby

features/step_definitions/*_steps.rb



09.11.21

- Регистрация обработчиков регулярных выражений + распознавание параметров

```
Given /^I have (\d+) cucumbers in my belly$/ do |cukes|  
  # Some Ruby code here  
end
```

Если (/выб[а-я]+ в выпадающем меню без скролла
(#{CAPTURE_PATTERN}) пункт (#{CAPTURE_PATTERN})/)

do |area, element|

```
  hover($DEFAULT_LOC)  
  sleep(SHORT_WAITING_TIME)  
  click(area)  
  sleep(SHORT_WAITING_TIME)  
  click(element)  
end
```

Пример отчёта



09.11.21

Cucumber Features

3 scenarios (3 failed)
19 steps (3 failed, 7 skipped, 9 passed)
Finished in 0m39.060s seconds
[Collapse All](#) [Expand All](#)

Feature: Change text properties

As a user
I want to change properties of selected text
In order to make my text less or more readable

Background

Given text fragment selected	features/step_definitions/cucumber_steps.rb:45
And menu opened by click on "1400188760887.png"	features/step_definitions/cucumber_steps.rb:12

Scenario: user changes the case

C:\Programming\Projects\cucumbers\features\wrong.feature:10

Given submenu opened by hover "1400350284909.png"	features/step_definitions/cucumber_steps.rb:17
When I click on "1400350576310.png"	features/step_definitions/cucumber_steps.rb:12
Then I should see "1401099724885.png"	features/step_definitions/cucumber_steps.rb:21

```
org.sikuli.script.FindFailed: can not find 1401099724885.png in S(0)[0,0 1920x1080] (StandardError)

./features/step_definitions/cucumber_steps.rb:22:in `/'^I should see ((?:"[0-9]+\.[png]"|Pattern.*))$/'
C:\Programming\Projects\cucumbers\features\wrong.feature:13:in `Then I should see "1401099724885.png"'

20
21 Given /^I should see ({CAPTURE_PATTERN})$/ do |img|
22   expect(wait(img,4)).to_not raise_exception
23 end
```

Scenario Outline: user chooses garniture and size by filling

C:\Programming\Projects\cucumbers\features\wrong.feature:15

Given window opened by click on "1401098332746.png"	features/step_definitions/cucumber_steps.rb:12
And tab opened by click on "1401096173233.png"	features/step_definitions/cucumber_steps.rb:12
When I fill in <field> with <string>	features/step_definitions/cucumber_steps.rb:29

Тестирование со стороны браузера



09.11.21

- Эмуляция запросов браузера
- Выполнение тестов при помощи браузера
- Скрипт на любом языке

Основные задачи тестирования браузером



09.11.21

- Проверка ссылок `<a>`, `<link>`, ``
- Проверка отображения элементов и соответствия пропорций
- Проверка работоспособности Javascript-кода

Основные средства клиентского тестирования



09.11.21

- cURL
 - Ruby-скрипты
 - Selenium
 - SikuliX
-
- AutomatedQA/SmartBear Software TestComplete
 - IBM Rational Robot
 - Microsoft Web Application Stress Tool



- Программа для получения файлов и эмуляции запросов
 - Авторизация
 - Произвольные запросы
 - Произвольные заголовки
 - Ответ в виде потока данных для дальнейшей обработки
- Libcurl
- <http://curl.haxx.se/>

cURL

Некоторые опции



09.11.21

Usage: curl [options...] <url>

Options: (H) means HTTP/HTTPS only, (F) means FTP only

- a/--append Append to target file when uploading (F)
- A/--user-agent <string> User-Agent to send to server (H)
 - anyauth Pick "any" authentication method (H)
- b/--cookie <name=string/file> Cookie string or file to read cookies from (H)
 - basic Use HTTP Basic Authentication (H)
- B/--use-ascii Use ASCII/text transfer
- c/--cookie-jar <file> Write cookies to this file after operation (H)
- C/--continue-at <offset> Resumed transfer offset
- d/--data <data> HTTP POST data (H)
 - data-ascii <data> HTTP POST ASCII data (H)
 - data-binary <data> HTTP POST binary data (H)
 - negotiate Use HTTP Negotiate Authentication (H)
 - digest Use HTTP Digest Authentication (H)
 - disable-eprt Inhibit using EPRT or LPRT (F)
 - disable-epsv Inhibit using EPSV (F)
- D/--dump-header <file> Write the headers to this file
 - egd-file <file> EGD socket path for random data (SSL)
 - tcp-nodelay Use the TCP_NODELAY option
- e/--referer Referer URL (H)
- E/--cert <cert[:passwd]> Client certificate file and password (SSL)

...

Скрипт на Ruby OpenURI



09.11.21

```
require 'open-uri'
```

```
open("http://www.ruby-lang.org/") { |f|  
  f.each_line { |line| p line }  
}
```

```
open("http://www.ruby-lang.org/en") { |f|  
  f.each_line { |line| p line }  
  p f.base_uri          # <URI::HTTP:0x40e6ef2 URL:http://www.ruby...  
  p f.content_type      # "text/html"  
  p f.charset           # "iso-8859-1"  
  p f.content_encoding  # []  
  p f.last_modified     # Thu Dec 05 02:45:02 UTC 2002  
}
```

- <http://www.ruby-doc.org/stdlib-2.4.1/libdoc/open-uri/rdoc/OpenURI.html>

Скрипт на Ruby

Net::HTTP



09.11.21

```
require 'net/http'
uri = URI('http://example.com/index.html')
params = { :limit => 10, :page => 3 }
uri.query = URI.encode_www_form(params)

res = Net::HTTP.get_response(uri)
puts res.body if res.is_a?(Net::HTTPSuccess)
```

- <https://ruby-doc.org/stdlib-3.0.2/libdoc/net/http/rdoc/Net/HTTP.html>



- Тестирование по изображениям
- Создана в MIT User Interface Design Group
<http://www.sikuli.org/>
- Поддержка и развитие сейчас: Raimund Hocke
<http://sikulix.com>
<https://github.com/RaiMan/SikuliX-2014>
- В 2014 в SikuliX IDE добавлен Ruby
- Работе из консоли с помощью gem sikulix
- Шаблоны для RSpec и Cucumber:
<https://github.com/rssdev10/sikulix-ide-templates>



Основные действия SikuliX



09.11.21

- Поиск изображений
find(img), findAll(img), wait(img), exists(img)
- Действия над изображениями
click(img), hover(img), dragDrop(img),...
- Действия с клавиатурой
type(img, text), paste(img, text)
- События
onAppear(img), onVanish(img), onChange(img)
- <http://sikulix-2014.readthedocs.org/>

Примеры специфических конструкций Ruby



09.11.21

- Поиск и обработка найденных объектов-изображений

```
findAll("123.png").each do |obj|  
  puts obj.getTarget.toString()  
end
```
- Ожидание исчезновения изображений

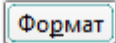
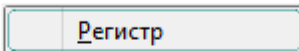
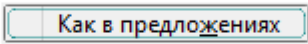


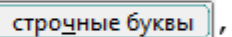
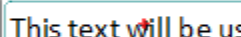
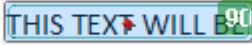
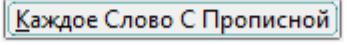
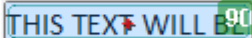
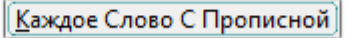
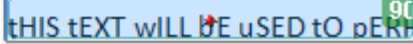
```
onVanish("1392466160677.png") do |e|  
  puts e.inspect # печатаем объект-событие  
end  
observe(10) # ожидаем события не более 10 секунд
```
- Регистрация обработчика «горячих клавиш»

```
addHotkey Key::F1, KeyModifier::ALT + KeyModifier::CTRL do  
  popup 'Hallo Welt!', 'Titel'  
  removeHotkey Key::F1, KeyModifier::ALT + KeyModifier::CTRL  
end
```

• Пример использования SikuliX + RSpec



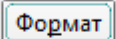
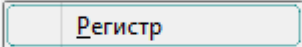
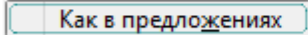
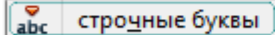
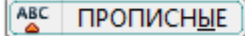

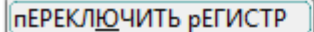
09.11.21

```
119 describe "Format menu", :open => :yes, :text_select => :yes do
120   include_context "open document"
121   include_context "text selection"
122   include_context "clickable container", 
123
124   describe "case selector submenu" do
125     include_context "self clean"
126     include_context "hoverable menu", 
127     describe "sentence case" do
128       it_behaves_like "clickable", , 
129     end
130     describe "lower case" do
131       it_behaves_like "clickable and hides something", , 
132     end
133     describe "upper case" do
134       it_behaves_like "clickable", , 
135     end
136     describe "camel case" do
137       it_behaves_like "clickable", , 
138     end
139     describe "switch case" do
140       it_behaves_like "clickable", , 
141     end
142   end
end
```

Пример использования SikuliX + Cucumber



09.11.21

```
66 Feature: Change text properties
67   As a user
68     I want to change properties of selected text
69     In order to make my text less or more readable
70
71   Background:
72     Given text fragment selected
73     And menu opened by click on 
74
75   Scenario Outline: user changes the case
76     Given submenu opened by hover 
77     When I click on <click img>
78     Then I should see <effect img>
79   Examples:
80     | click img | effect img |
81     |  | This text is apper case by default.  
This text is lowercase by-default.  
This should be Inverted. |
82     |  | apper case by |
83     |  | LOWERCASE BY |
84     |  | This Text Is Apper Case By Default.  
This Text Is Lowercase By-Default.  
This Should Be Inverted. |
85     |  | This Should Be iNVertED. |
```



- Selenium-IDE – расширение для Firefox
- Katalon Recorder – расширение для распространенных браузеров
- Selenium 1 (Selenium RC) – сервер для создания тестов
- Selenium WebDriver – библиотека для Ruby, Java, Javascript, Perl, Python,...
- <http://seleniumhq.org/>

Katalon Recorder, Selenium IDE



09.11.21

- Расширение для Firefox, записывающее последовательность действий пользователя

The screenshot displays the Katalon Automation Recorder interface. On the left, a web browser window titled "Calc" is open at the URL "localhost:3000/calc/input". The page contains two input fields, "Value 1" (containing "10") and "Value 2" (containing "15"), and a "Calc result" button. On the right, the Katalon Recorder extension window is visible. It shows a test suite named "Calc test" with three steps: "open" (http://localhost:3000/calc/input), "click" (id=v1), and "type" (id=v1, value=10). The bottom panel shows a log of the execution.

Test Suites	Command	Target	Value
Untitled Test Suite*	open	http://localhost:3000/calc/input	
Calc test *	click	id=v1	
Untitled Test Case	type	id=v1	10
Untitled Test Case	type	id=v2	15

Passed: 1 Failed: 0

Log Screenshots Variables Test Data Reference

```
[info] Playing test case Untitled Test Suite / Calc test
[info] Time: Wed Mar 14 2018 13:17:50 GMT+0300 (MSK) Timestamp: 1521822678976
[info] OS: macOS Version: 18.13
[info] Browser: Firefox Version: 58.0
[info] Executing: | open | http://localhost:3000/calc/input |
```

Selenium Webdriver



09.11.21

- Драйверы браузеров
 - HtmlUnitDriver — универсальный драйвер, не требующий браузера
 - FirefoxDriver
 - InternetExplorerDriver
 - ChromeDriver
 - OperaDriver
 - SafariDriver
 - AndriodDriver
 - IphoneDriver
- http://seleniumhq.org/docs/03_webdriver.html

Selenium + Ruby



09.11.21

- `gem install selenium-webdriver`
- Пример подключения к сайту `http://google.com`
require `'selenium-webdriver'`

```
driver = Selenium::WebDriver.for :firefox  
driver.get "http://google.com"
```

```
element = driver.find_element :name => "q"  
element.send_keys "Cheese!"  
element.submit  
puts "Page title is #{driver.title}"  
wait = Selenium::WebDriver::Wait.new(:timeout => 10)  
wait.until { driver.title.downcase.start_with? "cheese!" }  
puts "Page title is #{driver.title}"  
driver.quit
```

Selenium Webdriver

Некоторые методы



09.11.21

- Отправка запроса
 - `driver.get "http://www.google.com"`
- Поиск элемента
 - `element = driver.find_element(:id, "coolestWidgetEvah")`
 - `cheeses = driver.find_elements(:class, "cheese")`
 - `frame = driver.find_element(:tag_name, "iframe")`
 - `cheese = driver.find_element(:name, "cheese")`
 - `driver.find_element(:link, "cheese")`
 - `inputs = driver.find_elements(:xpath, "//input")`
- http://seleniumhq.org/docs/03_webdriver.html

Selenium Webdriver

Заполнение формы



09.11.21

- Традиционный способ

```
select = driver.find_element(:tag_name, "select")
all_options = select.find_elements(:tag_name, "option")
all_options.each do |option|
  puts "Value is: " + option.attribute("value")
  option.click
end
```

- Начиная с версии 2.14

```
select = Selenium::WebDriver::Support::Select.new
  ( driver.find_element (:tag_name, "select"))
select.deselect_all()
select.select_by(:text, "Edam")

driver.find_element(:id, "submit").click
```

Selenium Webdriver

Ожидание результата



09.11.21

```
require 'rubygems'
require 'selenium-webdriver'

driver = Selenium::WebDriver.for :firefox
driver.get "http://google.com"

element = driver.find_element :name => "q"
element.send_keys "Cheese!"
element.submit

puts "Page title is #{driver.title}"

wait = Selenium::WebDriver::Wait.new(:timeout => 10)
wait.until { driver.title.downcase.start_with? "cheese!" }

puts "Page title is #{driver.title}"
driver.quit
```

RSpec + Webdriver (U3 Selenium IDE)



09.11.21

```
require "selenium-webdriver"
require "rspec"
include RSpec::Expectations
describe "Calc" do
  before(:each) do
    @driver = Selenium::WebDriver.for :firefox
    @base_url = "http://localhost:3000"
    @driver.manage.timeouts.implicit_wait = 30
    @verification_errors = []
  end

  after(:each) do
    @driver.quit
    @verification_errors.should == []
  end

  it "test_calc" do
    @driver.get(@base_url + "/calc/input")
    @driver.find_element(:id, "v1").clear
    @driver.find_element(:id, "v1").send_keys "2"
    @driver.find_element(:id, "v2").clear
    @driver.find_element(:id, "v2").send_keys "10"
    @driver.find_element(:xpath, "(//input[@id='op__'])[2]").click
    @driver.find_element(:name, "commit").click
    verify { (@driver.find_element(:xpath, "//*[@id='result']").text).should == "20" }
  end

  def element_present?(how, what)
    @driver.find_element(how, what)
    true
  rescue
    Selenium::WebDriver::Error::NoSuchElementException
    false
  end

  def verify(&blk)
    yield
    rescue ExpectationNotMetError => ex
      @verification_errors << ex
    end
  end
end
```



- Средство для тестирования веб-приложений
- Встроенная поддержка Rack::Test and Selenium
- Возможность использования совместно с Unit::Test, RSpec и встроенного DSL
- `gem install capybara`
- <https://github.com/jnicklas/capybara>

Capbara with RSpec



09.11.21

```
describe "the signup process", :type => :request do
  before :each do
    User.make(:email => 'user@example.com', :password => 'caplin')
  end

  it "signs me in" do
    within("#session") do
      fill_in 'Login', :with => 'user@example.com'
      fill_in 'Password', :with => 'password'
    end
    click_link 'Sign in'
  end
end
```

Capbara DSL



09.11.21

```
feature "Signing up" do
  background do
    User.make(:email => 'user@example.com', :password => 'caplin')
  end

  scenario "Signing in with correct credentials" do
    within("#session") do
      fill_in 'Login', :with => 'user@example.com'
      fill_in 'Password', :with => 'caplin'
    end
    click_link 'Sign in'
  end
  given(:other_user) { User.make(:email => 'other@example.com', :password => 'rous') }

  scenario "Signing in as another user" do
    within("#session") do
      fill_in 'Login', :with => other_user.email
      fill_in 'Password', :with => other_user.password
    end
    click_link 'Sign in'
  end
end
```

Rails 6

System test



09.11.21

- Конфигурация selenium-драйвера
test/system/application_system_test_case.rb

```
require "test_helper"
```

```
class ApplicationSystemTestCase < ActionDispatch::SystemTestCase
  driven_by :selenium, using: :firefox, screen_size: [400, 400]
end
```

- Генерация каркаса новых тестов

```
rails generate system_test calcs
```

- Запуск системных тестов

```
rails test:system
```

- <https://guides.rubyonrails.org/testing.html#system-testing>

System test

Capbara style



09.11.21

```
require "application_system_test_case"
```

```
class CalcsTest < ApplicationSystemTestCase
```

```
  test "visiting the index" do
```

```
    visit calc_input_url
```

```
    fill_in "v1", with: "1"
```

```
    fill_in "v2", with: "5"
```

```
    choose option:"+"
```

```
    click_on "Calc result"
```

```
    assert_selector "#result", text: "6" # проверяем текст на текущей странице
```

```
  end
```

```
end
```


System test

Selenium driver style



09.11.21

```
require "application_system_test_case"
```

```
class CalcsTest < ApplicationSystemTestCase
```

```
  # получить объект драйвера Selenium из Capybara
```

```
  setup do
```

```
    @driver = Capybara.current_session.driver.browser
```

```
  end
```

```
  # выполнить экспортированный из Selenium IDE код с исправлением URL
```

```
  test "checking calc results with selenium" do
```

```
    @driver.get(calc_input_url) # установить адрес с помощью сгенерированного метода calc_input_url
```

```
    @driver.find_element(:id, 'v1').send_keys('10')
```

```
    @driver.find_element(:id, 'v2').send_keys('5')
```

```
    @driver.find_element(:id, 'op_-').click
```

```
    @driver.find_element(:name, 'commit').click
```

```
    assert_selector "#result", text: "5" # проверяем текст на текущей странице
```

```
  end
```

```
end
```



09.11.21

The
Pragmatic
Programmers

Scripted GUI Testing with Ruby



The Facets

The
Pragmatic
Programmers

The RSpec Book

Behaviour-Driven Development
with RSpec, Cucumber,
and Friends

David Chelmsky

with Dave Astels,
Zach Dennis,
Aslak Hellesøy,
Bryan Helmkamp,
and Dan North

Foreword by Robert C. Martin
(Uncle Bob)

Edited by Jacquelyn Carter

The Facets of Ruby Series

The
Pragmatic
Programmers

Cucumber Recipes

Automate Anything with
BDD Tools and Techniques



Ian Dees,
Matt Wynne,
and Aslak Hellesøy

Edited by Jacquelyn Carter

• Литература



09.11.21

-
- <http://guides.rubyonrails.org/testing.html>
 - <http://rspec.info/>
 - <http://cukes.info/>
 - <http://rubydoc.info/gems/rspec-core>
 - <http://seleniumhq.org/>
 - <https://github.com/jnicklas/capybara>
 - <http://sikulix.com>
-
- Obie Fernandez. The rails 5 way. 4-th edition. Addison-Wesley. 2017
 - Ian Dees. Scripted GUI Testing with Ruby. The Pragmatic Bookshelf. 2008
 - David Chelimsky, Dave Astels, Bryan Helmkamp, Dan North, Zach Dennis, Aslak Hellesoy. The RSpec Book: Behaviour Driven Development with Rspec, Cucumber, and Friends. Pragmatic Bookshelf, 2010
 - Ian Dees, Matt Wynne, Aslak Hellesoy. Cucumber Recipes: Automate Anything with BDD Tools and Techniques. Pragmatic Bookshelf, 2013