



RSpec, rspec-puppet, and beaker-rspec

David.Schmitt@puppet.com

[@dev_el_ops](#)



Contents

Basic Specifications

Quickly Testing Catalogs

Thoroughly Testing Systems

Basic Specifications

Basic Specifications

A bicycle
On a flat surface
When releasing it
Will topple over

A bicycle
On a flat surface
When pushing it
Will roll on

Encoding

```
describe 'a bicycle' do
  context 'on a flat surface' do
    context 'when releasing it' do
      it 'topples over'
    end
    context 'when pushing it' do
      it 'rolls on'
    end
  end
end
```

Output

```
$ rspec --format documentation --color spec/bicycle_spec.rb
```

```
a bicycle
```

```
  on a flat surface
```

```
    when releasing it
```

```
      topples over (PENDING: Not yet implemented)
```

```
    when pushing it
```

```
      rolls on (PENDING: Not yet implemented)
```

Adding Details - Data

```
describe 'a bicycle' do
  let(:owner) { 'Testuser' }
  subject { Bicycle.new(owner) }
```

Adding Details - Behaviour

```
context 'when pushing it' do
  before :each do
    subject.push
    subject.release
  end
end
```


Adding Details - Behaviour

```
context 'with expensive calculations' do
  before :all do
    @expensive = get_expensive_value()
  end
end
```

Adding Details - Matchers

```
class Bicycle
  attr_accessor :rolling?, :speed

  it { is_expected.to be_rolling? }

  it { is_expected.to have_attributes(speed: (a_value > 0)) }

  it { expect(subject.speed).to be > 0 }
```

Error output

1) a bicycle on a flat surface when pushing it should have attributes `{:speed => (a value > 0)}`

Failure/Error: it { is_expected.to have_attributes(speed: (a_value > 0)) }

expected #<Bicycle:0x0000000111d400 @owner="Testuser", @speed=-1, @held=false, @pushed=true, @rolling=true> to have attributes `{:speed => (a value > 0)}` but had attributes `{:speed => -1}`

Diff:

`@@ -1,2 +1,2 @@`

`-:speed => (a value > 0),`

`+:speed => -1,`

`# ./spec/encoding1.rb:45:in `block (4 levels) in <top (required)>'`

Complex Specifications

```
it 'fulfills a complex condition' do
  # more code here
  expect(some_value).to eq expected_value
end
```

Sharing Examples

```
RSpec.shared_examples 'a rolling bicycle' do
  it { is_expected.to be_rolling? }
  it { is_expected.to have_attributes(speed: (a_value > 0)) }
end
```

```
it_behaves_like 'a rolling bicycle'
```

Tagging and filtering

```
describe 'a bicycle' do
  context 'on a flat surface' do
    context 'when releasing it', :focus do
      it 'topples over'
    end
    context 'when pushing it' do
      it 'rolls on'
    end
  end
end
```

```
$ rspec --format documentation --color spec/bicycle_spec.rb --tag focus
```

Resources

<https://relishapp.com/rspec/>

<http://betterspecs.org/>

Quickly Testing Catalogs

The Easiest and most Important

```
# spec/classes/scanner_spec.rb

describe 'scanner' do
  it { is_expected.to compile.with_all_deps }
end
```

The Dirty Truth

```
describe 'scanner' do
  let(:facts) {{
    osfamily: 'Debian'
  }}
  let(:params) {{
    key: 'value'
  }}
  let(:pre_condition) {
    '# more puppet code'
  }
  it { is_expected.to compile.with_all_deps }
end
```

Expecting Resources

```
file { ['/var/spool/scanner': require => [Class['Scanner::Install'], #...

it {
  is_expected.to contain_file('/var/spool/scanner')
    .that_requires(['Package[scanner]', 'Package[scanner-daemon]'])
    .that_notifies('Service[scanner]')
    .with(
      'ensure' => 'directory',
      'group'  => 'scanner',
      'mode'   => '0750',
      'owner'  => 'scanner',
    )
}
```

Other Checks

- Defines: like classes, add: `let(:title) { 'foo' }`
- Functions: `is_expected.to run.with_params(...).and_return(...)`
- Types/Providers: **test the ruby code!**
- Exported Resources: `expect(exported_resources).to contain_file('foo')`

Resources

<https://github.com/rodjek/rspec-puppet/>

<http://betterspecs.org/>

<https://github.com/puppetlabs/puppetlabs-ntp>

https://github.com/puppetlabs/modulesync_configs

Thoroughly Testing Systems

Beaker and Friends

- Beaker: Manage VMs for testing
- Beaker DSL: talk to the VMs
- Beaker-RSpec: Hook up RSpec with VMs

The Simplest Nodeset

```
# spec/acceptance/nodesets/default.yml
HOSTS:
  testhost:
    roles:
      - default
    hypervisor: vagrant
    box: puppetlabs/debian-8.2-64-nocm
    platform: debian-8-amd64
CONFIG:
  log_level: debug
  type: git
```


The Simplest Setup

```
# spec/spec_helper_acceptance.rb

require 'beaker-rspec'

install_puppet_agent_on default

copy_module_to(default, source: '.', module_name: 'scanner')
on default, puppet('module', 'install', 'puppetlabs-stdlib')
```

The Simplest Test

```
require 'spec_helper_acceptance'

describe 'scanner' do
  before(:all) do
    @manifest = 'class { scanner: ...}'
    @result = apply_manifest_on default, @manifest,
      acceptable_exit_codes: (0...256), expect_changes: true
  end

  it 'runs with changes' do
    expect(@result.exit_code).to eq 2
  end
end
```

The Important Test

```
it 'runs idempotently' do
  result = apply_manifest_on default, @manifest,
    acceptable_exit_codes: (0...256), expect_changes: true
  expect(result.exit_code).to eq 0
end
```

serverspec

```
describe service('apache2'), :if => fact('osfamily') == 'debian' do
  it { should be_enabled }
  it { should be_running }
end
```

Where to Go Now?

How do you know that your service works?

Resources

<https://github.com/puppetlabs/beaker-rspec/>

<http://www.rubydoc.info/github/puppetlabs/beaker/Beaker/DSL>

<http://betterspecs.org/>

https://github.com/puppetlabs/modulesync_configs

<https://github.com/puppetlabs/beaker/blob/master/docs/Beaker-Libraries.md>

