

[Advanced Search](#)

Search...

- [Explore](#)
- [Gist](#)
- [Blog](#)
- [Help](#)

[pmq20](#)

- [Notifications](#)
- [Account Settings](#)
- [Log Out](#)

[cavalle](#) / [steak](#)

- [Watch](#) [Unwatch](#)
- [Fork](#)
- - [728](#)
 - [32](#)
- [Code](#)
- [Network](#)
- [Pull Requests 0](#)
- [Issues 5](#)
- [Wiki 2](#)
- [Stats & Graphs](#)

The delicious combination of RSpec and Capybara for Acceptance BDD — [Read more](#)

- [🍏 Clone in Mac](#)
- [ZIP](#)
- [HTTP](#)
- [Git Read-Only](#)

Read-Only access

- [Tags 25](#)
- [Downloads 0](#)
- [branch: master](#)
Switch Branches/Tags
Filter branches/tags



- [Branches](#)
- [Tags](#)

[gh-pages](#)

[master](#)

[steak-1](#)

[steak-2](#)

- [Files](#)
- [Commits](#)
- [Branches 4](#)

Latest commit to the **master** branch

[Merge pull request #48 from laserlemon/patch-1 ...](#)

Fix build status and add dependency status to README

[commit 284be93089](#)



[cavalle](#) authored January 09, 2012

[steak](#) /

	name	age	history message
	lib	June 14, 2011	Version 2.0.0 [cavalle]
	spec	June 14, 2011	Version 2.0.0 [cavalle]
	.gitignore	April 24, 2011	Make spec faster by using project fixtures (instead of generating new... [cavalle]
	.travis.yml	July 16, 2011	1.8.6 is not supported [cavalle]
	Gemfile	June 14, 2011	Version 2.0.0 [cavalle]
	Gemfile.lock	June 14, 2011	Version 2.0.0 [cavalle]
	LICENSE	March 12, 2011	Steak 2 rewrite driven by a new README [cavalle]
	README.md	January 09, 2012	Fix build status and add dependency status to README [laserlemon]
	Rakefile	April 24, 2011	Make spec faster by using project fixtures (instead of generating new... [cavalle]
	steak.gemspec	June 14, 2011	Version 2.0.0 [cavalle]
	README.md		

Steak build status passing dependency status out-of-date

The delicious combination of RSpec and Capybara for Acceptance BDD



What is Steak?

Steak is a minimal extension of RSpec-Rails that adds several conveniences to do acceptance testing of Rails applications using Capybara. It's an alternative to Cucumber in plain Ruby. This is how an acceptance spec looks like in Steak:

```
feature 'Main page' do

  background do
    create_user :login => 'jdoe'
  end

  scenario 'should show existing quotes' do
    create_quote :text => 'The language of friendship is not words, but meanings',
                  :author => 'Henry David Thoreau'

    login_as 'jdoe'
    visit '/'

    within('.quote') do
      page.should have_content('The language of friendship is not words, but meanings')
      page.should have_content('Henry David Thoreau')
    end
  end
end
```

No givens, whens or thens. No steps, no english, just Ruby. The delicious combination of RSpec and Capybara. That's Steak!

Why Steak?

If, after all, this is just RSpec + Capybara, why does Steak even exist? Do I really need it?

Basically Steak exists for three reasons:

1. **Making a point.** First of all, Steak proposes that using RSpec and Capybara for acceptance testing is a sensible alternative to Cucumber. It also sets a name to refer to that approach.
2. **Adding convenience.** As a gem, Steak aims to make the experience as convenient as possible. It provides Rails integration testing support and several generators and rake tasks so that setting up a new project or creating and running specs are quick and seamless tasks. A natural extension of RSpec-Rails.
3. **Building a community.** No development approach or ruby gem is really valuable without an active community behind it. The mailing list, the IRC channel, the wiki or the twitter account are useful tools to build a community of users that help each other by sharing knowledge, resources and best practices.

Getting Started

NOTE: The current version of Steak (2.0) assumes that you're testing a Rails 3 application, with RSpec 2 and Capybara. For Rails 2, RSpec 1 or Webrat you should use [Steak 1](#) (or consider upgrading to non-obsolete technologies ;P)

It's super-easy to get you started. Just add the gem to your Gemfile...

```
group :test, :development do
  gem 'steak'
  # ...
```

...and then install the gem and run the generator:

```
$ bundle
$ rails g steak:install
```

You're now set up! Note that Steak is the only dependency you really need, you can safely remove any reference to capybara, rspec-rails or rspec from your Gemfile, they will be included by Steak. Also note that, unless previously executed, Steak will run the RSpec generator so you don't need to invoke it.

Take a look now at the default directory structure you've got under spec/acceptance. It will help you organize the helpers and configurations for your acceptance specs.

Creating and running specs

You can create your specs by hand, however you may prefer to use the generator:

```
$ rails g steak:spec my_first_feature
```

To run your acceptance specs you just do like with any other spec...

```
$ bundle exec rspec spec/acceptance/my_first_feature_spec.rb
```

...or using rake:

```
$ rake spec:acceptance
```

Resources

- [Source code](#) – Lurking and forking
- [Issues](#) – Bugs and feature requests

- [Google Group](#) & IRC channel – Help and discussion
- [Twitter](#) – Announcements and mentions
- [Wiki](#) – Tutorials, docs and other resources

Credits

Steak was created by [Luismi Cavallé](#) and improved thanks to the love from:

- Álvaro Bautista
- Felipe Talavera
- Paco Guzmán
- Jeff Kreeftmeijer
- Jaime Iniesta
- Emili Parreño
- Andreas Wolff
- Wincent Colaiuta
- Falk Pausen
- Francesc Esplugas
- Raúl Murciano
- Enable Interactive
- Vojto Rinik
- Fred Wu
- Sergio Espeja
- Joao Carlos

Copyright © 2009 - 2011 Luismi Cavallé, released under the MIT license

GitHub Links

GitHub

- [About](#)
- [Blog](#)
- [Features](#)
- [Contact & Support](#)
- [Training](#)
- [GitHub Enterprise](#)
- [Site Status](#)

Tools

- [Gauges: Analyze web traffic](#)
- [Speaker Deck: Presentations](#)
- [Gist: Code snippets](#)
- [GitHub for Mac](#)
- [Issues for iPhone](#)
- [Job Board](#)

Extras

- [GitHub Shop](#)
- [The Octodex](#)

Documentation

- [GitHub Help](#)
- [Developer API](#)
- [GitHub Flavored Markdown](#)
- [GitHub Pages](#)
- [Terms of Service](#)
- [Privacy](#)
- [Security](#)

© 2012 GitHub Inc. All rights reserved.



Powered by the [Dedicated Servers](#) and [Cloud Computing](#) of Rackspace Hosting®

Markdown Cheat Sheet

Format Text

Headers

```
# This is an <h1> tag
## This is an <h2> tag
##### This is an <h6> tag
```

Text styles

```
*This text will be italic*
_This will also be italic_
**This text will be bold**
__This will also be bold__

*You **can** combine them*
```

Lists

Unordered

```
* Item 1
* Item 2
  * Item 2a
  * Item 2b
```

Ordered

```
1. Item 1
2. Item 2
3. Item 3
  * Item 3a
  * Item 3b
```

Miscellaneous

Images

```
![GitHub Logo](/images/logo.png)  
Format: ![Alt Text](url)
```

Links

```
http://github.com - automatic!  
[GitHub](http://github.com)
```

Blockquotes

As Kanye West said:

```
> We're living the future so  
> the present is our past.
```

Code Examples in Markdown

Syntax highlighting with [GFM](#)

```
```javascript  
function fancyAlert(arg) {
 if(arg) {
 $.facebox({div:'#foo'})
 }
}
```
```

Or, indent your code 4 spaces

Here is a Python code example
without syntax highlighting:

```
def foo:  
    if not bar:  
        return true
```

Inline code for comments

I think you should use an
`<addr>` element here instead.

Something went wrong with that request. Please try again. [Dismiss](#)