

Libraries » jnicklas/capybara (master) » Index » File: README

Capybara

build error dependencies up-to-date code climate 3.1

Capybara helps you test web applications by simulating how a real user would interact with your app. It is agnostic about the driver running your tests and comes with Rack::Test and Selenium support built in. WebKit is supported through an external gem.

Need help? Ask on the mailing list (please do not open an issue on GitHub): <http://groups.google.com/group/ruby-capybara>

Key benefits

- **No setup** necessary for Rails and Rack application. Works out of the box.
- **Intuitive API** which mimics the language an actual user would use.
- **Switch the backend** your tests run against from fast headless mode to an actual browser with no changes to your tests.
- **Powerful synchronization** features mean you never have to manually wait for asynchronous processes to complete.

Setup

Capybara requires Ruby 1.9.3 or later. To install, add this line to your Gemfile and run `bundle install`:

```
gem 'capybara'
```

If the application that you are testing is a Rails app, add this line to your test helper file:

```
require 'capybara/rails'
```

Note: In rails 4.0/4.1 The default rails test environment (`config/environments/test.rb`) is not threadsafe – see <https://github.com/rails/rails/issues/15089> If you experience random errors about missing constants, adding `config.allow_concurrency = false` to `config/environments/test.rb` should solve the issue.

If the application that you are testing is a Rack app, but not Rails, set `Capybara.app` to your Rack app:

```
Capybara.app = MyRackApp
```

If you need to test JavaScript, or if your app interacts with (or is located at) a remote URL, you'll need to [use a](#)

Table of Contents

1. **Key benefits**
2. **Setup**
3. **Using Capybara with Cucumber**
4. **Using Capybara with RSpec**
5. **Using Capybara with Test::Unit**
6. **Using Capybara with MiniTest::Spec**
7. **Drivers**
 1. Selecting the Driver
 2. RackTest
 3. Selenium
 4. Capybara-webkit
 5. Poltergeist
8. **The DSL**
 1. Navigating
 2. Clicking links and buttons
 3. Interacting with forms
 4. Querying
 5. Finding
 6. Scoping
 7. Working with windows
 8. Scripting
 9. Modals
 10. Debugging
9. **Matching**
 1. Exactness
 2. Strategy
10. **Transactions and database setup**
11. **Asynchronous JavaScript (Ajax and friends)**
12. **Using the DSL elsewhere**
13. **Calling remote servers**
14. **Using the sessions manually**
15. **XPath, CSS and selectors**
16. **Beware the XPath // trap**
17. **Configuring and adding drivers**
18. **Gotchas:**
19. **Development**