

VCR

WAR STORIES

by Jan Zdráhal

WHAT'S VCR?



WHAT'S VCR?

- records your test suite's HTTP interactions and replays them during future test runs for fast, deterministic, accurate tests
- <https://github.com/vcr/vcr>

EXAMPLE

```
require 'vcr'

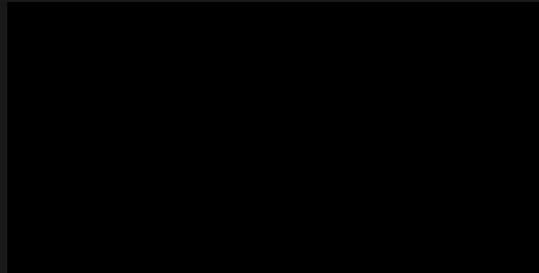
VCR.configure do |c|
  c.cassette_library_dir = 'cassettes'
  c.hook_into :webmock
  c.configure_rspec_metadata!
  c.default_cassette_options = { :record => :new_episodes }
end

describe "VCR", :vcr do
  it 'records an http request' do
    5.times do
      puts Net::HTTP.get_response('localhost', '/', 4567).body
    end
  end
end
```

EXAMPLE SERVER

```
require 'sinatra'  
  
get '/' do  
  sleep 1  
  'Hello Ruby Stories!'  
end
```

DEMO



HOW IT WORKS?

- checks HTTP method, URI, host, path, body and headers
- matchers for nondeterministic URIs

```
random_id_matcher = lambda do |actual, recorded|
  random_id_regex = %r{/uploads/.+}
  actual.parsed_uri.path.match random_id_regex
end
```

WHY WE USED IT?

45 MINUTE CI BUILD

✓ # 4049.4	 staging1 - sdk integration tests - base	🕒 46 min 29 sec	🔒
✓ # 4049.5	 staging1 - sdk integration tests - project	🕒 19 min 12 sec	🔒
✓ # 4049.6	 staging2 - lcm integration tests	🕒 19 min 39 sec	🔒

- caused by a lot of integration tests
- ran on every PR
- 45 minutes to find out your code is broken
- every glitch of the staging server fails the build
- high load on the staging server
- hitting request limits

SOLUTION ATTEMPT #1

- running unit tests with every PR and integration tests only once a day
- too many bugs getting to the develop branch
- we would learn about our bugs the next day (after switching context)

THE SOLUTION

With VCR, the same test suite finishes in under 5 minutes. That's 88% faster!

Build ID	Job Name	Duration
✓ # 4063.1	pronto code review	54 sec
✓ # 4063.2	unit tests 2.3	1 min 25 sec
✓ # 4063.3	unit tests 2.4	1 min 15 sec
✓ # 4063.4	unit tests 2.5	1 min 42 sec
✓ # 4063.5	unit tests 2.6	1 min 27 sec
✓ # 4063.6	unit tests jruby-9.1.14	3 min 36 sec
✓ # 4063.7	sdk integration (vcr) tests - base	4 min 38 sec
✓ # 4063.8	sdk integration (vcr) tests - project	2 min 38 sec
✓ # 4063.9	lcm integration (vcr) tests - e2e	1 min 46 sec

- live tests running every day

PROBLEMS

*Making your tests run under VCR is not
a walk in the park.*

INITIAL INVESTMENT

- VCRizing the tests took two weeks (the dev was new to the project and language though)
- all blocks not recorded by default
- parallelism

TIME FOR A RIDDLE

```
describe 'all block riddle' do
  before(:all) do
    user_id = server.create_user
  end

  it 'does something' do
    server.do_something_with_user user_id
  end

  it 'does something else' do
    server.do_something_else_with_user user_id
  end
end
```

DEVELOPMENT OVERHEAD

- re-recording cassettes with every change
- recording cassettes could be automated
- writing matchers cannot be automated
- problems with recording from the middle of the test suite (e.g. cached resources)
- where to store 390M of cassettes?

WHEN TO USE SOMETHING ELSE

- emulators (e.g. Google Datastore emulator)
- mock libraries (e.g. redis-mock)
- manual mocks

UNIT TESTS?

- it's tempting not to write unit tests
- unit tests are needed for covering edge cases
- if it's hard to write tests, the code is probably too complex
- fortunately test coverage can be checked automatically (e.g. Coveralls)

SUMMARY