

Minitest

@slashmili

Bit background

- System & network administrator
- Python
- Perl
- PHP



Let's learn Ruby



Enjoyed it a lot



You don't learn it until build it

We love test

Setup testing framework



Categories by group [\(Show by name\)](#)

Active Record Plugins

Active Record DB Adapters [activerecord-jdbc-adapter](#), [activerecord-sqlserver-adapter](#), [activerecord-jdbcmysql-adapter](#), [activerecord-jdbcpostgresql-adapter](#), [activerecord-jdbcderby-adapter](#), and 3 more

Active Record Default Values [Default value for](#) and [Active record defaults](#)

Active Record Enumerations [enumerize](#), [Enumerated attribute](#), [Simple enum](#), [Active Enum](#), [Classy Enum](#), and 8 more

Active Record Index Assistants [foreigner](#), [Rails indexes](#), [schema_plus](#), [lol_dba](#), [Immigrant](#), and 3 more

Active Record Named Scopes [ransack](#), [Searchlogic](#), [Pacecar](#), [randumb](#), [Can search](#), and 7 more

Active Record Nesting [Awesome nested set](#), [Ancestry](#), [Acts as tree](#), [Closure Tree](#), [Better nested set](#), and 2 more

Active Record Sharding [Octopus](#), [Db-charmer](#), [masochism](#), [multi_db](#), [Data fabric](#), and 2 more

Active Record Soft Delete [rails3_acts_as_paranoid](#), [paranoia](#), [Acts as paranoid](#), [Acts as archive](#), [Permanent records](#), and 7 more

Active Record Sortables [Acts as list](#), [ranked-model](#), [Sortable](#), [Resort](#), and [acts_as_restful_list](#)

Active Record User Stamping [Userstamp](#), [User stamp](#), [Acts as audited](#), [Record activities](#), [Acts as loggable](#), and 2 more

Active Record Value Cleanup [Attribute normalizer](#), [Strip attributes](#), [Nilify blanks](#), [auto_strip_attributes](#), [Scrubby](#), and 2 more

Active Record Versioning [PaperTrail](#), [audited](#), [Vestal Versions](#), [Acts as versioned](#), [Acts as revisable](#), and 11 more

Pagination [will_paginate](#), [Kaminari](#), [sorted](#), [ajax_pagination](#), [Paged scopes](#), and 3 more

Rails Comments [juvia](#), [Acts as commentable with threading](#), [Acts as commentable](#), [Disqus](#), [opinio](#), and 3 more

Rails DB Bootstrapping [Seed fu](#), [Regular seedbank](#), [Bootstrapper](#), [Db seedlets](#), and 3 more

SPONSORED BY YOU!

Running the Ruby Toolbox takes a lot of work and dedication from **my** spare time. Please consider a donation!

♥ [DONATE](#)

Announcements

The Rails Girls Summer of Code needs your support! 4 Jun 2013

See you at RailsConf 2013! 29 Apr 2013

Fix your Rails 2.3/3.0 and Devise apps! 28 Jan 2013

Signin bug smashing 26 Jan 2013

Stop what you're doing and fix all your Rails apps right now! 10 Jan 2013

Recent Comments

BornALegend commented on **forem**
a day ago

BornALegend commented on **socialization**
5 days ago

rcp140875 commented on **palava_machine**
6 days ago

 **Bryan Hanks, PMP** commented on **Fix your Rails 2.3/3.0 and Devise apps!**
18 days ago

RSpec is cool

Why MiniTest

Seattle.rb / minitest

branch: master minitest / lib / minitest / +

+ Thanks to minitest-gcstats got a 5-10% speedup via reduced GC! ...

zenspider authored on Sep 3 latest commit 2c6e0fd7ab

..

assertions.rb	+ Thanks to minitest-gcstats got a 5-10% speedup via reduced GC!	a month ago
autorun.rb	Oh god... here we go.	2 years ago
benchmark.rb	- Fixed up benchmark spec doco and added a test to demonstrate. (bhen...	9 months ago
expectations.rb	+ Added clarification about the use of thread local variables in expe...	a year ago
hell.rb	Fixed require in minitest/hell. (locks)	10 months ago
mock.rb	+ Mocked methods can take a block to verify state. Seattle.rb 12 bday...	7 months ago
parallel.rb	- Fixed 2 cases still using MiniTest. (mikesea)	7 months ago
pride.rb	+ Minitest.init_plugins passes down options.	a year ago
pride_plugin.rb	killed dead code (teoljungberg)	9 months ago
spec.rb	restyled / renamed vars in last patch	a month ago
test.rb	+ Extract teardown method list.	a month ago
utils.rb	+ Improved warning message about minitest/autorun teardown address. (bhen...	a month ago

Only Handful of files

```
1 require 'minitest/autorun'
2
3 class Foo
4   def add(x,y)
5     x + y
6   end
7 end
8
9 describe Foo do
10   describe '#add' do
11     it 'adds two numbers' do
12       Foo.new.add(4, 2).must_equal 6
13     end
14   end
15 end
```

```
[ 01:43:25 ] > ruby /tmp/t1.rb
Run options: --seed 41329
```

```
# Running tests:
```

```
.
```

```
Finished tests in 0.001788s, 559.2841 tests/s, 559.2841 assertions/s.
```

```
1 tests, 1 assertions, 0 failures, 0 errors, 0 skips
```

Ruby >= 1.9

MiniTest::Spec

```
describe User do
  it "must set default params during creation" do
    params = {name: "Sam Rast", email: "sam-rast@gmail.com", password: "SECUREPASS"}
    user = User.create params
    user.username.must_equal "sam-rast"
    user.initials.must_equal "SR"
  end
end
```

```
describe Api::V1::CommentsController do
  before do
    sign_in user
  end

  after do
    kill user
  end

  it 'shows comment' do
    comment = create(:comment, card_id: card.id, user_id: user.id)
    get :show, id: comment
    assert_response :success
  end
end
```



```
describe Api::V1::CommentsController do
  describe 'sing in user' do
    before do
      sign_in user
    end
    it 'shows comment' do
      comment = create(:comment, card_id: card.id, user_id: user.id)
      get :show, id: comment
      assert_response :success
    end
  end
end
```

```
describe Api::V1::CommentsController do
  let(:user) { create(:user) }
  let(:board) { create(:board, user: user) }
  let(:list_done) { board.lists.first }
  let(:card) { create(:card, list_id: list_done.id)}

  before do
    sign_in user
  end

  it "creates new" do
    post :create, comment: { text: 'This is my text', card_id: card.id}
    assigns(:comment).text.must_equal 'This is my text'
    assigns(:comment).user.must_equal user
    assigns(:comment).card.must_equal card
  end
end
```

```
describe Api::V1::CommentsController do
  subject { create(:user) }

  before do
    sign_in subject
  end

  it "creates new" do
    post :create, comment: { text: 'This is my text', card_id: card.id }
    assigns(:comment).text.must_equal 'This is my text'
    assigns(:comment).user.must_equal subject
  end
end
```

```
describe Card do
  let(:user) { create(:user) }
  let(:board) { create(:board, user_id: user.id) }
  let(:card) { create(:card, list_id: board.lists.first.id) }

  it "must be able to save members ad member_ids" do
    card.members = [10]
    card.member_ids.must_equal [10]
    skip 'for some reason the below test doesnt pass'

    card.member_ids = [20]
    card.members.must_equal [20]
  end
end
```

Expectations


```
describe Membership do
  let(:board_access) { Membership.new }

  it "must be valid" do
    board_access.must_be :valid?
  end

  it "must have three roles" do
    Membership::ROLES.must_equal %w[owner admin normal]
  end

  it "must save a role" do
    user = create(:user)
    board = create(:board, user_id: user.id)
    board_access.roles= %w[normal]
    board_access.roles.wont_include %w[guest]
  end

  it "wont allow to have the double membership" do
    user = create(:user)
    new_user = create(:user)
    board = create(:board, user_id: user.id, public: true)

    proc {
      mem = Membership.create(board: board, user: user, roles: %w[normal])
    }.must_raise ActiveRecord::RecordNotUnique
  end
end
```

Minitest::Unit

Mocking/Stubbing

```
describe Something do
  let(:something) { MinitTest::Mock.new }

  it "does something" do
    something.expect(:name, 'Widget')
    something.name.must_equal 'Widget'
    something.verify
  end
end
```

```
describe Something do
  let(:something) { Something.new(name: 'Widget') }

  it "does something" do
    something.stub(:name, 'Foo') do
      something.name.must_equal 'Foo'
    end
  end
end
```

Struct, OpenStruct

Or use Mocha, RR, FlexMock

Rails - MiniTest

**Should I convert to
MiniTest ?**

It's light

Familiar syntax

9X% of Rspec

No dependency

```
rspec (2.14.1)
  rspec-core (~> 2.14.0)
  rspec-expectations (~> 2.14.0)
  rspec-mocks (~> 2.14.0)
rspec-core (2.14.7)
rspec-expectations (2.14.4)
  diff-lcs (>= 1.1.3, < 2.0)
rspec-mocks (2.14.4)
rspec-rails (2.14.0)
  actionpack (>= 3.0)
  activesupport (>= 3.0)
  railties (>= 3.0)
  rspec-core (~> 2.14.0)
  rspec-expectations (~> 2.14.0)
  rspec-mocks (~> 2.14.0)
```

Ships with Ruby

Thanks

@slashmili