

Rspec Rails Cheat Sheet

```
$ script/generate rspec
  exists lib/tasks
  create lib/tasks/rspec.rake
  create script/autospec
  create script/spec
  create spec
  create spec/rcov.opts
  create spec/spec.opts
  create spec/spec_helper.rb
```

```
$ script/generate # display docs and list of generators
```

```
$ script/generate rspec_model Person
```

```
$ script/generate rspec_scaffold thing name:string description:text
```

```
rake -T spec # lists all rspec rake tasks
```

```
rake spec # run all specs
```

```
rake spec SPEC=spec/models/mymodel_spec.rb SPEC_OPTS="-e 'should do something'" #run single spec
```

```
require 'spec_helper'

describe Person do
  before do
    @valid_attributes = {
      :first => "value for first",
      :last => "value for last",
      :present => false
    }
  end

  it "should create a new instance given valid attributes" do
    Person.create!(@valid_attributes)
  end
end
```

```
describe ThingsController do

  def mock_thing(stubs={})
    @mock_thing ||= mock_model(Thing, stubs)
  end

  describe "GET index" do
    it "assigns all things as @things" do
      Thing.stub(:find).with(:all).and_return([mock_thing])
      get :index
      assigns[:things].should == [mock_thing]
    end
  end
end
```

Verifying changes to a model

```
proc { thing.approve! }.should change(thing, :status).  
  from(Status::AWAITING_APPROVAL).  
  to(Status::APPROVED)
```

```
proc { thing.destroy }.should change(Thing, :count).by(-1)
```

Mocks and Stubs

```
user_mock = mock "User"  
user_mock.should_receive(:authenticate).with("password").and_return(true)  
user_mock.should_receive(:coffee).exactly(3).times.and_return(:americano)  
user_mock.should_receive(:coffee).exactly(5).times.and_raise(NotEnoughCoffeeException)
```

```
people_stub = mock "people"  
people_stub.stub!(:each).and_yield(mock_user)  
people_stub.stub!(:bad_method).and_raise(RuntimeError)
```

```
user_stub = mock_model("User", :id => 23, :username => "pat", :email => "pat@example.com")
```

```
require 'spec_helper'  
  
describe "/things/index.html.erb" do  
  include ThingsHelper  
  
  before do  
    assigns[:things] = [  
      stub_model(Thing,  
        :name => "value for name",  
        :description => "value for description"  
      ),  
      stub_model(Thing,  
        :name => "value for name",  
        :description => "value for description"  
      )  
    ]  
  end  
  
  it "renders a list of things" do  
    render  
    response.should have_tag("p", "value for name".to_s, 2)  
    response.should have_tag("p", "value for description".to_s, 2)  
  end  
end
```

Note: TextMate wants an absolute path:

```
require File.expand_path(File.dirname(__FILE__) + '/../spec_helper')
```