

Manually testing Associations

in the console

```
LibrarySystem.connection  
ls = LibrarySystem.create name: "King County"  
LibrarySystem.first  
  
Branch.connection  
Branch.create library_system: LibrarySystem.first, name: "Wedgwood"  
Branch.first.library_system
```

Automate It

rspec-rails

TDD/BDD library that makes rspec functionality available in rails.

following screenshots from:
<https://github.com/rspec/rspec-rails>

Add it to the Gemfile

```
group :development, :test do  
  gem 'rspec-rails', '~> 3.0'  
end
```

initialize

Download and install by running:

```
bundle install
```

Initialize the `spec/` directory (where specs will reside) with:

```
rails generate rspec:install
```

run the tests

Use the `rspec` command to run your specs:

```
bundle exec rspec
```

test structure

describe

test block

it

individual test

for a model

```
require "rails_helper"

RSpec.describe User, :type => :model do
  it "orders by last name" do
    lindeman = User.create!(first_name: "Andy", last_name: "Lindeman")
    chelimsky = User.create!(first_name: "David", last_name: "Chelimsky")

    expect(User.ordered_by_last_name).to eq([chelimsky, lindeman])
  end
end
```

for a controller

```
RSpec.describe TeamsController do
  describe "GET index" do
    it "assigns @teams" do
      team = Team.create
      get :index
      expect(assigns(:teams)).to eq([team])
    end

    it "renders the index template" do
      get :index
      expect(response).to render_template("index")
    end
  end
end
```

we'll do this next time

pending tests

```
describe "an example" do
  it "is a pending example"
end
```

```
describe "an example" do
  it "is implemented but waiting" do
    pending("something else getting finished")
    this_should_not_get_executed
  end
end
```

<https://relishapp.com/rspec/rspec-core/v/2-0/docs/pending/pending-examples>

tagging

```
describe "group with tagged specs" do
  it "example I'm working now", :focus => true do; end
  it "special example", :type => 'special' do; end
  it "slow example", :skip => true do; end
  it "ordinary example", :speed => 'slow' do; end
  it "untagged example" do; end
end
```

o filter examples with a simple tag

When I run "rspec . --tag focus"

Then the output should contain "Run filtered using {:focus=>true}"

And the output should contain "1 example, 0 failures"

o filter examples with a name:value tag

When I run "rspec . --tag type:special"

Then the output should contain:

```
Run filtered using {:type=>"special"}
```

And the output should contain "1 example, 0 failures"

<https://relishapp.com/rspec/rspec-core/v/2-3/docs/command-line/tag-option>

assertion library

provides methods for comparing
expected values to results

<http://www.rubydoc.info/gems/rspec-expectations>

install (in rails)

comes with rspec-rails!

why?

calls to `assert_equal`

```
def assert_equal x, y  
  fail "values not equal" unless x == y  
end
```

become

```
expect(x).to eq(y)
```

easier to write

this

```
it "" do
  assert_less_than y(2), 12.5664
  assert_greater_than y(2), 12.56
end
```

becomes

```
it "" do
  expect(y(2)).to be_within(.001).of(12.566)
end
```

easier to read errors

```
expected: #<Money @value=5.55 @currency=:USD>  
got: #<Money @value=1.11 @currency=:USD>
```

and you don't have to provide an error message yourself

consistency

across applications

watch task

automatically run rake tasks when one of a set of specified files changes.

(setting one up is part of the homework)

tutorial time

[http://tutorials.jumpstartlab.com/topics/capybara/
capybara_practice.html](http://tutorials.jumpstartlab.com/topics/capybara/capybara_practice.html)

Readings

[https://github.com/rspec/rspec-rails/blob/master/
README.md](https://github.com/rspec/rspec-rails/blob/master/README.md)

<http://robots.thoughtbot.com/how-we-test-rails-applications>

Watch this Video

https://www.youtube.com/watch?v=BG_DDUD4M9E

Homework

- Frontend (UI) for your apps.
- Add model tests. Use spec expectations.
- Add integration tests. Use capybara.
- Add a watch task that runs your specs automagically when you update your rails app. Use <https://github.com/guard/guard-rspec>