Overview - RSpec for Rails

Overview

In the last submodule we covered how to write RSpec tests for Ruby classes and methods. In this submodule we are going learn how to write RSpect for Ruby on Rails.

MiniTest vs RSpec

Rails comes prepackaged with a test framework called MiniTest (previously called Test::Unit). When you created a new rails app in the past, you may have noticed that a *"/test"* directory is generated for you. You may also have noticed that when you generate a model, rails adds a *"<mymodelname>_test.rb"* to that *"/test"* directory. That *"<mymodelname>_test.rb"* file is MiniTest.

RSpec is a bit more popular in industry so we will be teaching you how to write RSpec tests rather than MiniTest tests. Both are very similar though, so once you have learned the concepts of one, you should have no problem picking up the other.

What we will be testing in RSpec for Rails

The Rails functionality than you can test with RSpec is quite expansive. If you look at the rspec-rails documentation https://github.com/rspec/rspec-rails you will see that you can write spec tests for models, controllers, requests, mailers, jobs, views, routes, and helpers. This is practically your entire rails application!

Of all the spec tests that we could teach you how to write, we are going to focus on just teaching you how to write Model specs and Feature specs. Model specs are the most important because in real world applications the code base of the Model tends to be the largest (Fat Model, Skinny Controller). This means that there are the most spec tests to write for Models, and also that the functionality being tested by Model specs tends to be the most complicated and the most important. We teach you feature specs because it gives us the opportunity to introduce you to Capybara, a very popular gems for extending RSpec functionality.

Don't worry about not learning how to write spec tests for the other Rails functionality. Writing those spec tests is very similar, and after learning how to write spec tests for Models and Features, you will be in good shape to quickly and easily pick up how to write those other spec tests on your own.