Assignment-Ruby-on-Rails-Book-Collection 2

Objective

The primary objective of this assignment is to help you get familiar with TDD testing using the simple CRUD Application you built previously.

Task description

Taking the initial Book Collection Application you developed, implement Test-driven development (TDD) using RSPEC by writing unit and integration tests, after which you will resolve bugs.

Submission

1. Deadline: See Canvas
2. You will use your own Github for development, however you will push your source code to the Github Classroom repository (link to be provided)
3. Submit the report (described below) to Canvas.

Jaka Bernat

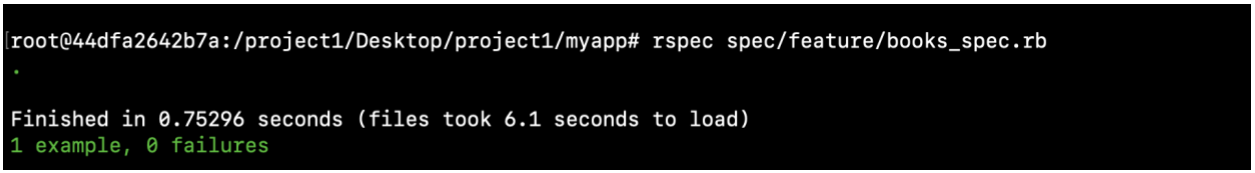
730009309

Link to GitHub repo:
https://github.com/FA23-CSCE431-SoftEng/assignment-1-jakabernat.git

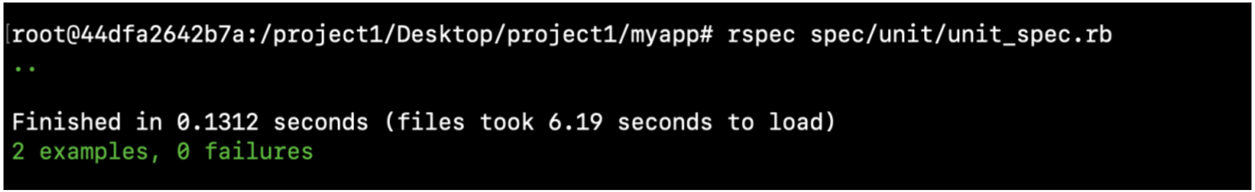
Assignment Instructions

1. Using the code you wrote for "Lab-BookCollection 1," write unit/integration tests

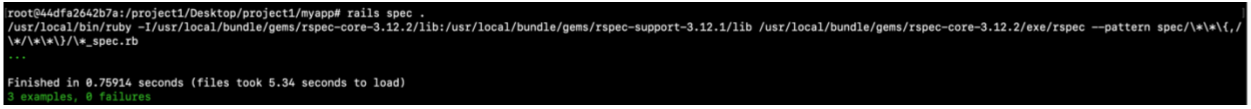
   ○ Write a unit using RSPEC to test the model for 'book' and an integration test to make sure that the book is added successfully to the database as indicated by the correct 'flash notice.' Likewise, test a 'rainy-day' scenario by trying to save a book with 'title' blank, where you look for the corresponding 'flash notice.' The tests should pass. **Take a screenshot (1) of the results. You'll need to put the screenshot into your report later.**



```
root@44dfa2642b7a:/project1/Desktop/project1/myapp# rspec spec/feature/books_spec.rb
.

Finished in 0.75296 seconds (files took 6.1 seconds to load)
1 example, 0 failures
```

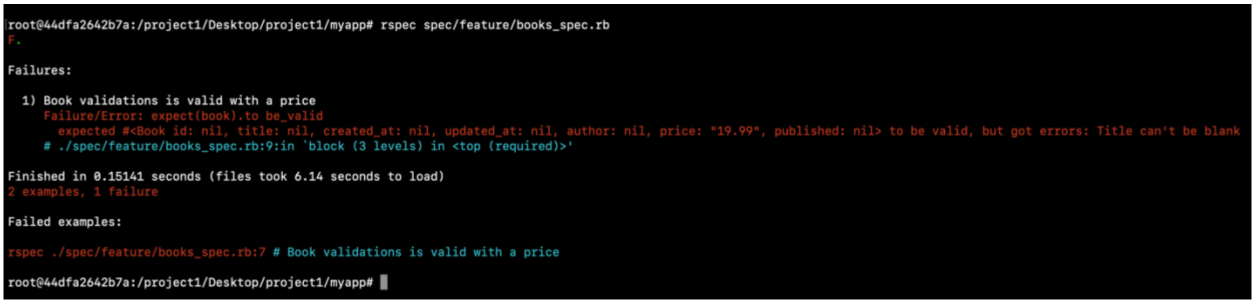


```
root@44dfa2642b7a:/project1/Desktop/project1/myapp# rspec spec/unit/unit_spec.rb
..

Finished in 0.1312 seconds (files took 6.19 seconds to load)
2 examples, 0 failures
```

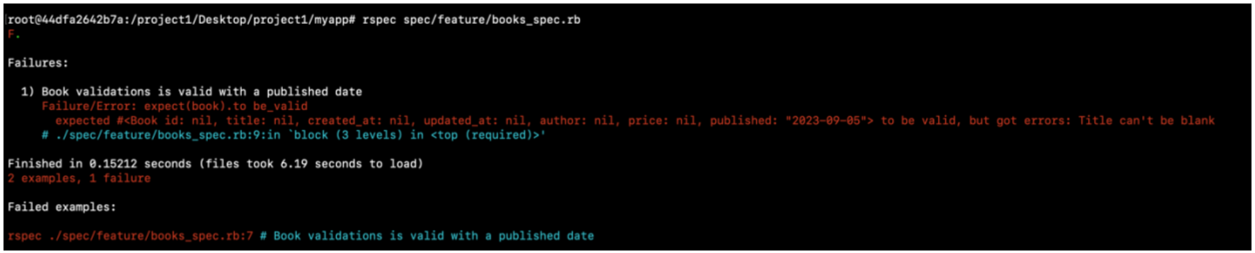


```
root@44dfa2642b7a:/project1/Desktop/project1/myapp# rails spec .
/usr/local/bin/ruby -I/usr/local/bundle/gems/rspec-core-3.12.2/lib:/usr/local/bundle/gems/rspec-support-3.12.1/lib /usr/local/bundle/gems/rspec-core-3.12.2/exe/rspec --pattern spec/\*\*\{,/
\*/\*\*\}/\*_spec.rb
...

Finished in 0.75914 seconds (files took 5.34 seconds to load)
3 examples, 0 failures
```

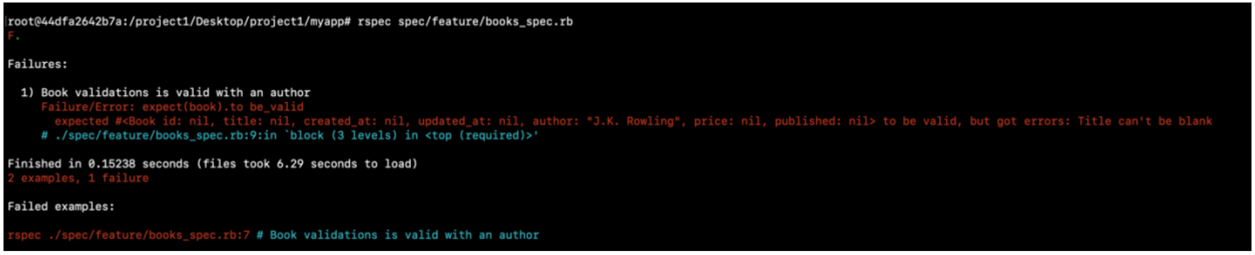
   ○ Next, you need to write more tests on your own for new attributes – author (String), price (Number), and published date (Date). For each new attribute, you need to write a unit test. After you write all the tests, run these tests . Your tests should **not** pass at this moment because you haven't started coding for these other attributes yet. **Take a screenshot (2) of the results. You'll need to put the screenshot into your report later.**



```
root@44dfa2642b7a:/project1/Desktop/project1/myapp# rspec spec/feature/books_spec.rb
F.

Failures:

  1) Book validations is valid with a price
     Failure/Error: expect(book).to be_valid
       expected #<Book id: nil, title: nil, created_at: nil, updated_at: nil, author: nil, price: "19.99", published: nil> to be valid, but got errors: Title can't be blank
     # ./spec/feature/books_spec.rb:9:in `block (3 levels) in <top (required)>'

Finished in 0.15141 seconds (files took 6.14 seconds to load)
2 examples, 1 failure

Failed examples:

rspec ./spec/feature/books_spec.rb:7 # Book validations is valid with a price

root@44dfa2642b7a:/project1/Desktop/project1/myapp# 
```

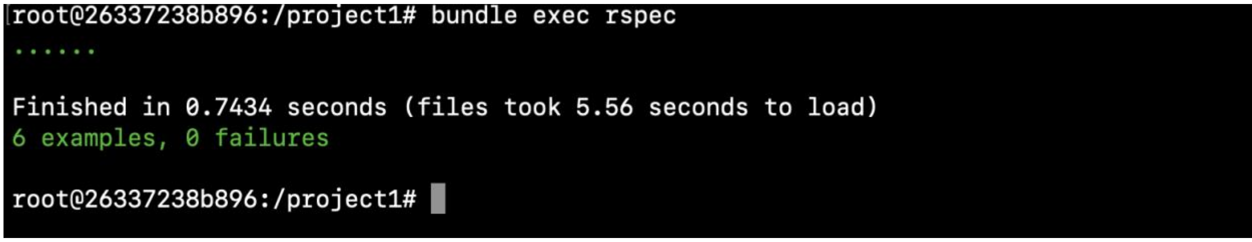


```
root@44dfa2642b7a:/project1/Desktop/project1/myapp# rspec spec/feature/books_spec.rb
F.

Failures:

  1) Book validations is valid with a published date
     Failure/Error: expect(book).to be_valid
       expected #<Book id: nil, title: nil, created_at: nil, updated_at: nil, author: nil, price: nil, published: "2023-09-05"> to be valid, but got errors: Title can't be blank
     # ./spec/feature/books_spec.rb:9:in `block (3 levels) in <top (required)>'

Finished in 0.15212 seconds (files took 6.19 seconds to load)
2 examples, 1 failure

Failed examples:

rspec ./spec/feature/books_spec.rb:7 # Book validations is valid with a published date
```



```
root@44dfa2642b7a:/project1/Desktop/project1/myapp# rspec spec/feature/books_spec.rb
F.

Failures:

  1) Book validations is valid with an author
     Failure/Error: expect(book).to be_valid
       expected #<Book id: nil, title: nil, created_at: nil, updated_at: nil, author: "J.K. Rowling", price: nil, published: nil> to be valid, but got errors: Title can't be blank
     # ./spec/feature/books_spec.rb:9:in `block (3 levels) in <top (required)>'

Finished in 0.15238 seconds (files took 6.29 seconds to load)
2 examples, 1 failure

Failed examples:

rspec ./spec/feature/books_spec.rb:7 # Book validations is valid with an author
```

- Insert code for the additional attributes author (String), price (Numerical), and published-date (Date/Datetime). **Take a screenshot (3) of the results of the tests passing after the successful inclusion of the attributes. You'll need to put the screenshot into your report later.**



```
root@26337238b896:/project1# bundle exec rspec
......

Finished in 0.7434 seconds (files took 5.56 seconds to load)
6 examples, 0 failures

root@26337238b896:/project1# 
```

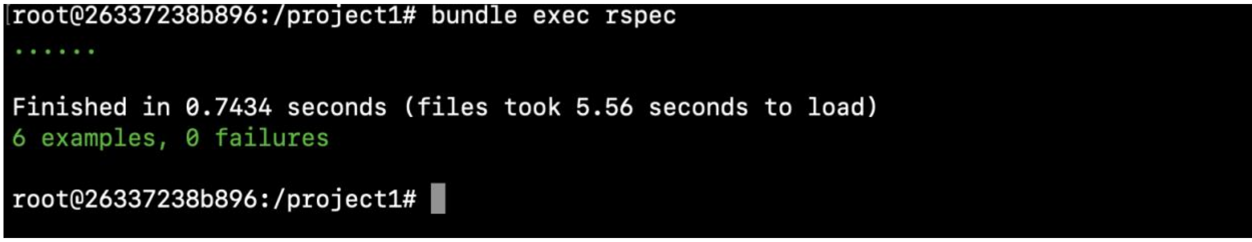
Note: published-date should be a drop-down menu.

- Remember to always make the migrations to create a database table

- Make sure you follow Rails convention and architecture (MVC architectural design pattern).

- After completing all of the requirements, run all the tests again, they should all pass. If not, refactor your code until they all pass. When all of them pass, **take a screenshot (4)**

of the results. You’ll need to put the screenshot into your report later.



```
root@26337238b896:/project1# bundle exec rspec
......

Finished in 0.7434 seconds (files took 5.56 seconds to load)
6 examples, 0 failures

root@26337238b896:/project1# 
```

2. Create a new branch test

   $ git checkout -b test

3. Save all code

   $ git add .

   $ git rm node_modules (if node_modules is in your .gitignore, this step is not necessary)

   $ git commit -m "your commit message"

4. Push the changes to your Github repository. To check the remote, use $ git status.

   $ git push origin test

Report

Your report should include the following content:

1. Link to your Github repository where you committed the working code.

2. Screenshots. Please add them sequentially from (1) to (n). For each screenshot, explain what it is about.

Save your report as FirstName_LastName_Lab-BookCollection 2-TestDrivenDev(TDD).pdf and submit it to Canvas.

References: Linkedin Trainings free of charge with your TAMU credentials

[Ruby on Rails 7 Essential Training]()

Article:

[Ruby On Rails - HTTP, MVC and Routes]()