

# Video Outline:

## 1. introduction

- a. Caleb Terry, CSCE 431 Fall 2022 student
- b. making this because my group had a lot of trouble figuring out how to test certain aspects of the application
- c. I will be covering many aspects of testing in Ruby applications, specifically what are Integration, Request and Unit tests and what are each of these tests used for.

## 2. environment setup

- a. using default rails app that students are using to start Book assignment and project from
- b. update local docker image:
  - i. **docker pull dmartinez05/ruby\_rails\_postgresql:latest**
- c. mount and run the docker image:
  - i. NOTE: you must be in the current path to use this command - C:\RailsTest431
  - ii. **docker run --rm -it --volume "\${PWD}:/RailsTest431" -e DATABASE\_USER=test\_app -e DATABASE\_PASSWORD=test\_password -p 3000:3000 dmartinez05/ruby\_rails\_postgresql:latest**
  - iii. if on a Mac, replace {PWD} with (pwd)
- d. move to shared folder - "RailsTest431"
  - i. **cd RailsTest431**
- e. create a Rails test app
  - i. **rails new test\_app -d=postgresql**
- f. move into test\_app folder for app
  - i. **cd test\_app**
- g. update the config file for the postgresql DB
  - i. **vim config/database.yml**
  - ii. [vim Cheatsheet](#)
  - iii. add the following 3 lines:

host: localhost

username: <%= ENV['DATABASE\_USER'] %>

password: <%= ENV['DATABASE\_PASSWORD'] %>

```

default: &default
  adapter: postgresql
  encoding: unicode
  # For details on connection pooling, see Rails configuration guide
  # https://guides.rubyonrails.org/configuring.html#database-pooling
  pool: <%= ENV.fetch("RAILS_MAX_THREADS") { 5 } %>
  host: localhost
  username: <%= ENV['DATABASE_USER'] %>
  password: <%= ENV['DATABASE_PASSWORD'] %>

development:

```

- h. instantiate the DB
  - i. **rails db:create**
  - ii. **rails db:migrate**
- i. to run the app locally (<http://127.0.0.1:3000>)
  - i. **rails server --binding=0.0.0.0**

### 3. Create a Book entity in the DB

- a. create the Book entity with a title and author attribute
  - i. **rails g scaffold Book title:string author\_id:integer**
- b. migrate the DB to add pending changes ^
  - i. **rails db:migrate**
- c. create an Author entity to store information about authors
  - i. **rails g scaffold Author name:string**
- d. add a belongs\_to relation to the book model
  - i. **belongs\_to :author**
- e. the author also has many books, so add a **has\_many :book** to the author's model

### 4. Update Gemfile for testing with RSpec

- a. add **gem 'rspec'** to the list of gems

```

21 # gem 'redis', '~> 4.0'
22 # Use Active Model has_secure_password
23 # gem 'bcrypt', '~> 3.1.7'
24
25 # Use Active Storage variant
26 # gem 'image_processing', '~> 1.2'
27
28 # Reduces boot times through caching; required in config/boot.rb
29 gem 'bootsnap', '>= 1.4.4', require: false
30 | gem 'rexml'

```

- b. add `gem 'rspec-rails'` under the `group :development, :test do` tab

```

32 ✓ group :development, :test do
33   # Call 'byebug' anywhere in the code to stop execution and get a debugger console
34   gem 'byebug', platforms: [:mri, :mingw, :x64_mingw]
35   | gem 'rspec-rails'
36 end

```

## 5. Write unit tests

- a. <https://www.geeksforgeeks.org/unit-testing-software-testing/>
- b. a book should have an author and title before being created
  - i. add `validates :title, :author_id` to book model
- c. an author should have a name before being created
  - i. `validates :name`

## 6. Integration Testing

- a. <https://www.geeksforgeeks.org/software-engineering-integration-testing/?ref=lbp>

## 7. View Testing

- a. <https://guides.rubyonrails.org/testing.html#testing-views>

## 8. Controller Testing

- a. <https://guides.rubyonrails.org/testing.html#functional-tests-for-your-controllers>