Current Version
Link:https://docs.google.com/document/d/1pHRImSS6Z26shCdsn5NRig4NQUWmoFZbDiC2qfmlxzk/edit?tab=t.0#heading=h.5cc5muqvwan5

How to generate custom databases and using fixtures to test your model

Small notes:
Connecting back to docker container VS Code Studios:

- **Run container in docker desktop**
- **View -> Command Palate -> "Docker" -> Docker Attach -> Container**

Before beginning these are the requirements:

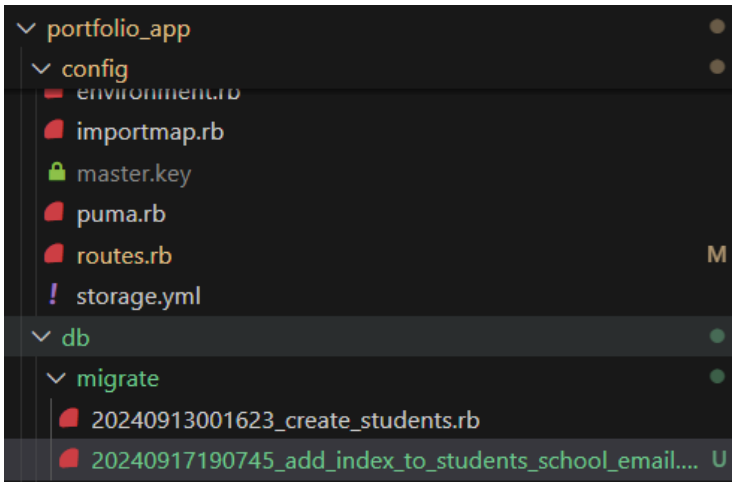
- IDE environment and Docker setup
- Run image and creation of container
- Install of Porfoilo_app
- Portfoilo_app must have scaffolding of student

Step 1: Custom Database Migration for Student_Emails being unique

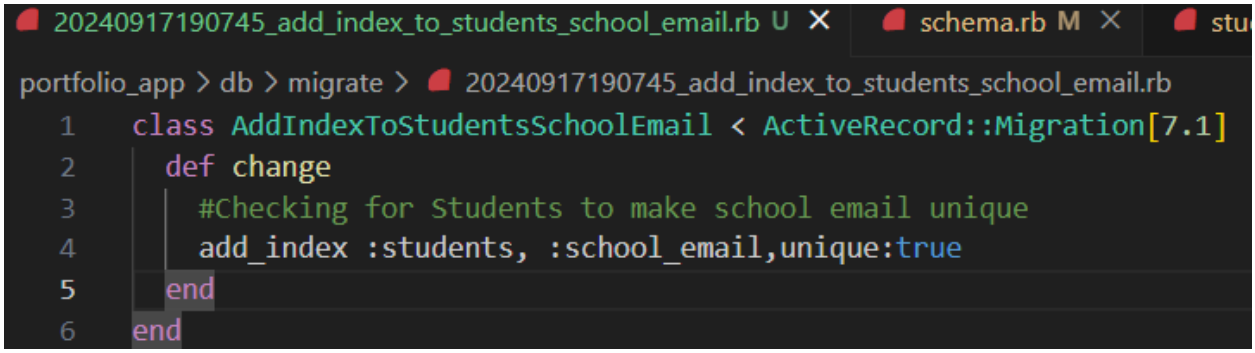
1. In Ruby, run this command in terminal**: " rails generate migration AddIndexToStudentsEmail"**
    a. The file generated will be used as a Database

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

# rails generate migration AddIndexToStudentsEmail
```

2. This command to create a file name **"####..._add_index_to_students_email**
    a. The file will be within "**db"** under "**migrate"**



3. Next with **"####..._add_index_to_students_email"** file, enter this: **add_index :students, :school_email, unique:true**
    a. This command is used to set the rules for school_email to be unique



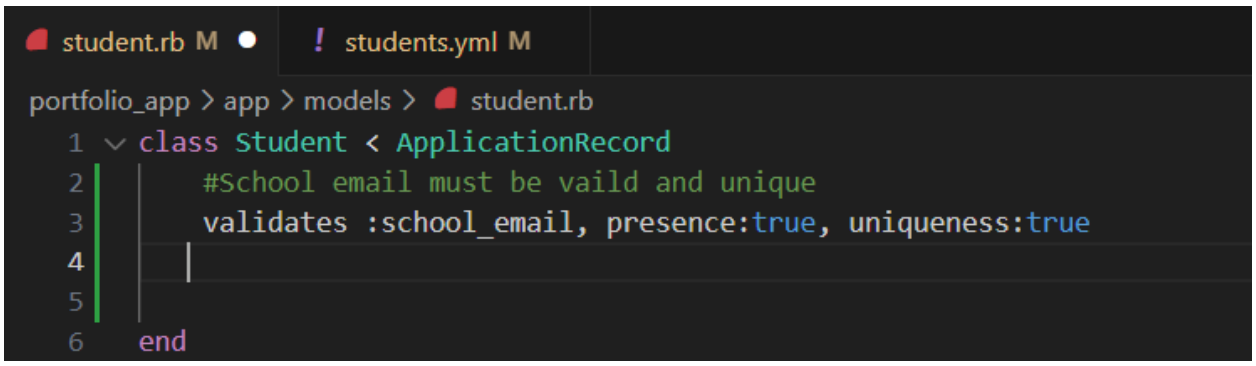
4. Following entering that command into the file, run "**rails db:migrate"** in order to update the **schema file:**
(Before: Schema File command and db:migrate)

```ruby
ActiveRecord::Schema[7.1].define(version: 2024_09_20_003926) do
  create_table "students", force: :cascade do |t|
    t.string "name"
    t.string "school_email"
    t.string "major"
    t.string "minor"
    t.date "graduation_date"
    t.datetime "created_at", null: false
    t.datetime "updated_at", null: false
  end
```

(After: Schema File command and db:migrate)

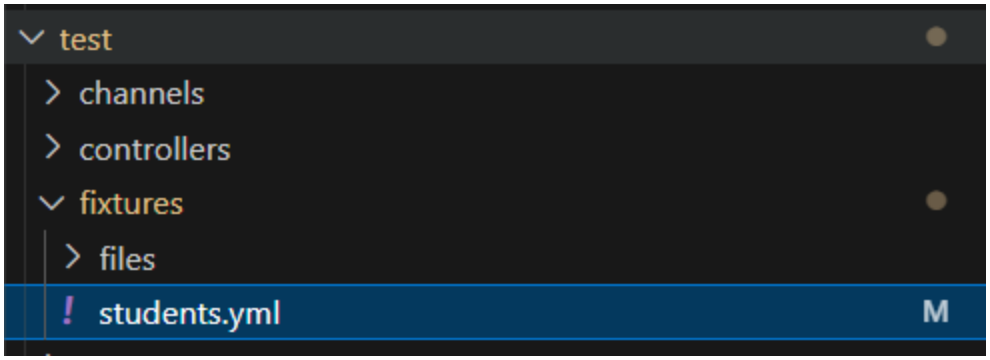
```ruby
ActiveRecord::Schema[7.1].define(version: 2024_09_20_012705) do
  create_table "students", force: :cascade do |t|
    t.string "name"
    t.string "school_email"
    t.string "major"
    t.string "minor"
    t.date "graduation_date"
    t.datetime "created_at", null: false
    t.datetime "updated_at", null: false
    t.index ["school_email"], name: "index_students_on_school_email", unique: true
  end

end
```

5. Next in your Portfoilo_app, head over to “app/models/student.rb” and put the following command: “**validates :school_email, presence:true, uniqueness:true”**
    a. Validates = Checks for category “school_email”
    b. Presence = Ensuring that the school email tested must exist otherwise error
    c. Uniqueness = Must be only email name of it’s kind, otherwise error

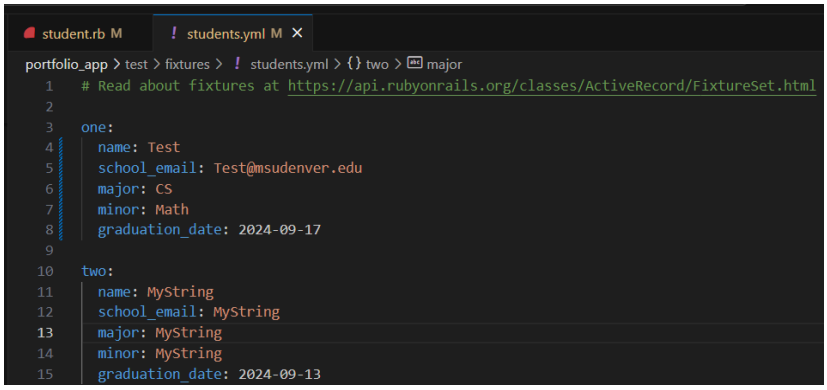


Step 2: Fixtures: Testing our code with dummy test code

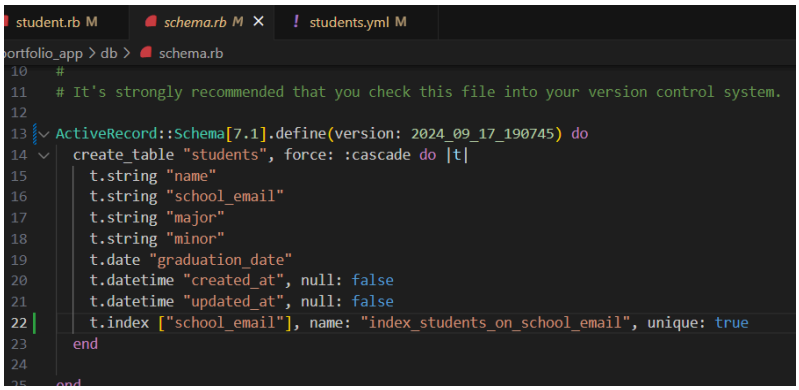
1. Fixtures are YAML Files located in test/fixtures and we will be editing students for this case



2. Within students.yml, exists a template for our Student record. This is where we can make dummy test code.
    a. The dummy code is based off our "schema.rb" file in "app/db"
    b. Ensure you change the default code in student.yml to fit your code for testing purposes



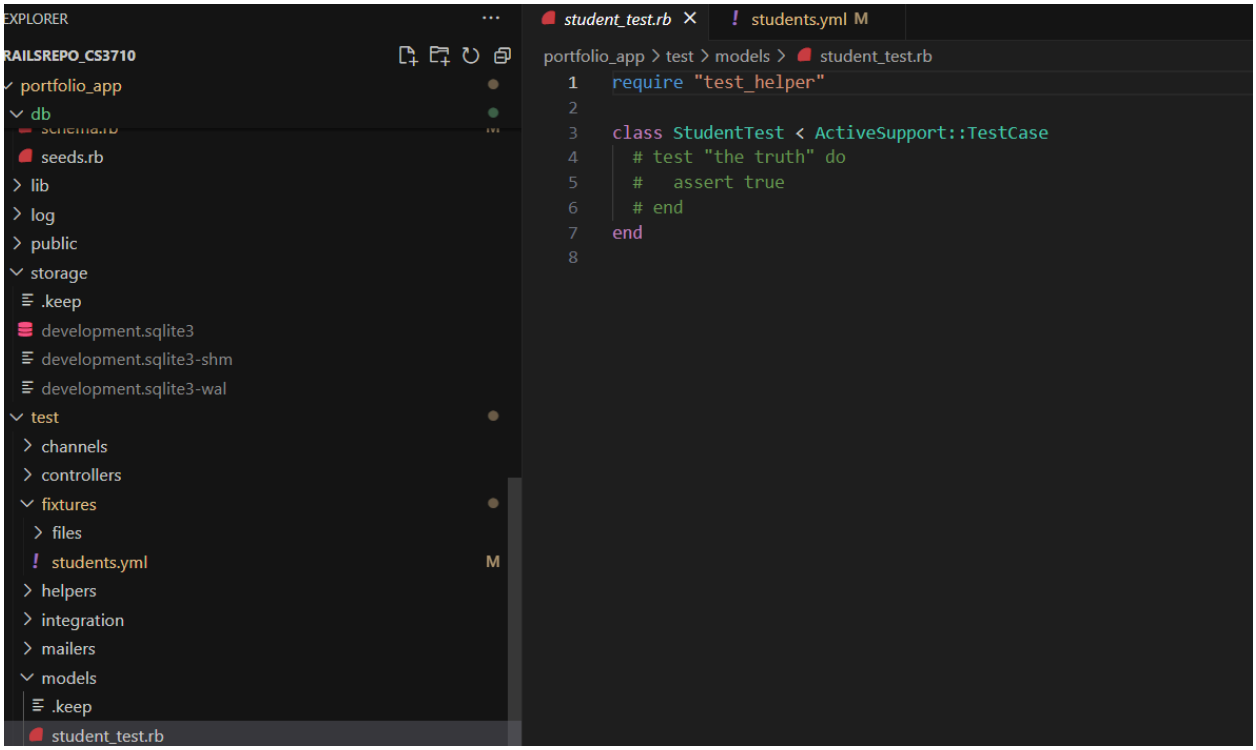
(Student.yml above, schema.rb below)



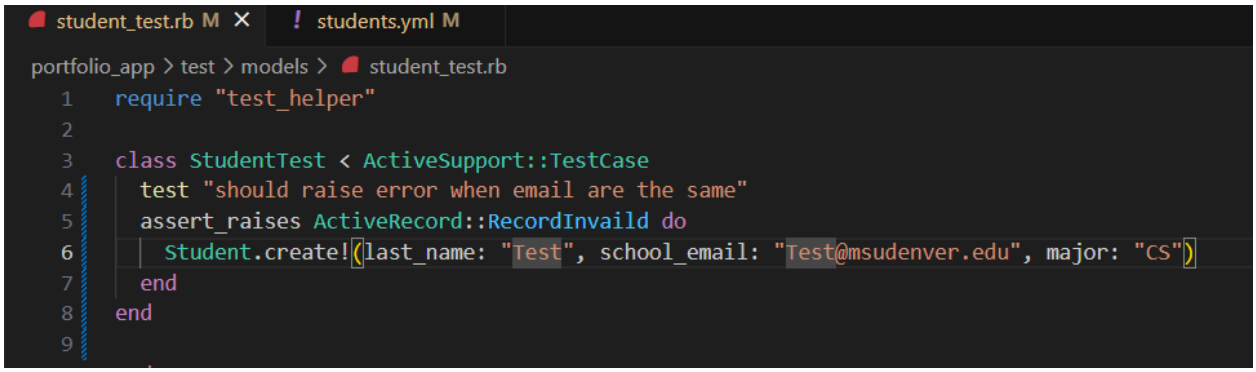
Step 3: Making the Student Model Test

Time to make a model test to verify what you are looking for. In this case, making sure "student_email" is unique

1. Start by looking in **"test/models/student_test.rb".** This file is designed to test your environment safety isolated



2. Within “student_test.rb” input the code that would be used for testing
    a. In this case, testing for unique emails would mean my test will try to create another user with the same email



Step 4: Running Tests in the model:

1. Once the test code is put in, testing the fixture is next. This command how to initialize the test: **rails test test/models/student_test.rb -v**
    a. Notice that for testing the command is broken down as **“rails test [file path to tested file]”**

```
root@3f875cd868f7:/workspace/portfolio_app# rails test test/models/student_test.rb -v
Running 1 tests in a single process (parallelization threshold is 50)
Run options: -v --seed 1744

# Running:

StudentTest#test_should_raise_error_when_saving_student_without_first_name = 2.34 s = F


Failure:
StudentTest#test_should_raise_error_when_saving_student_without_first_name [test/models/student_test.rb:5]:
[ActiveRecord::RecordInvalid] exception expected, not
Class: <ActiveModel::UnknownAttributeError>
Message: <"unknown attribute 'last_name' for Student.">
---Backtrace---
test/models/student_test.rb:6:in `block (2 levels) in <class:StudentTest>'
test/models/student_test.rb:5:in `block in <class:StudentTest>'
--------------
```

2. **An important note is to remember WHAT YOU ARE TESTING FOR.**
    a. Please take a look at your acceptance criteria for what needs to be tested for acceptable and bad data

Made a mistake or not looking correct?

Here are some troubleshooting methods your code doesn't look correct during the documentation:

1. To restart from the start of this document, do the command:

    rails destroy scaffold Students

    - If you do this command, make sure to remake the scaffold with:
        - **rails generate scaffold Student name:string school_email:string major:string minor:string graduation_date:date**
        - Return to the top of document

2. Messed up a "db:migrate"? You can rollback with this command:

    rails db:rollback

    - This will rollback the most recent "db:migrate"
    - Remember that you rollback further back adding "STEP=[Number]"
        - Ex: **rails db:rollback STEPS=3**
    - **Remember to delete the file of the generated DB file in "db/migrate" before creating a new one to avoid duplicates or diluting the folder with files**

END!