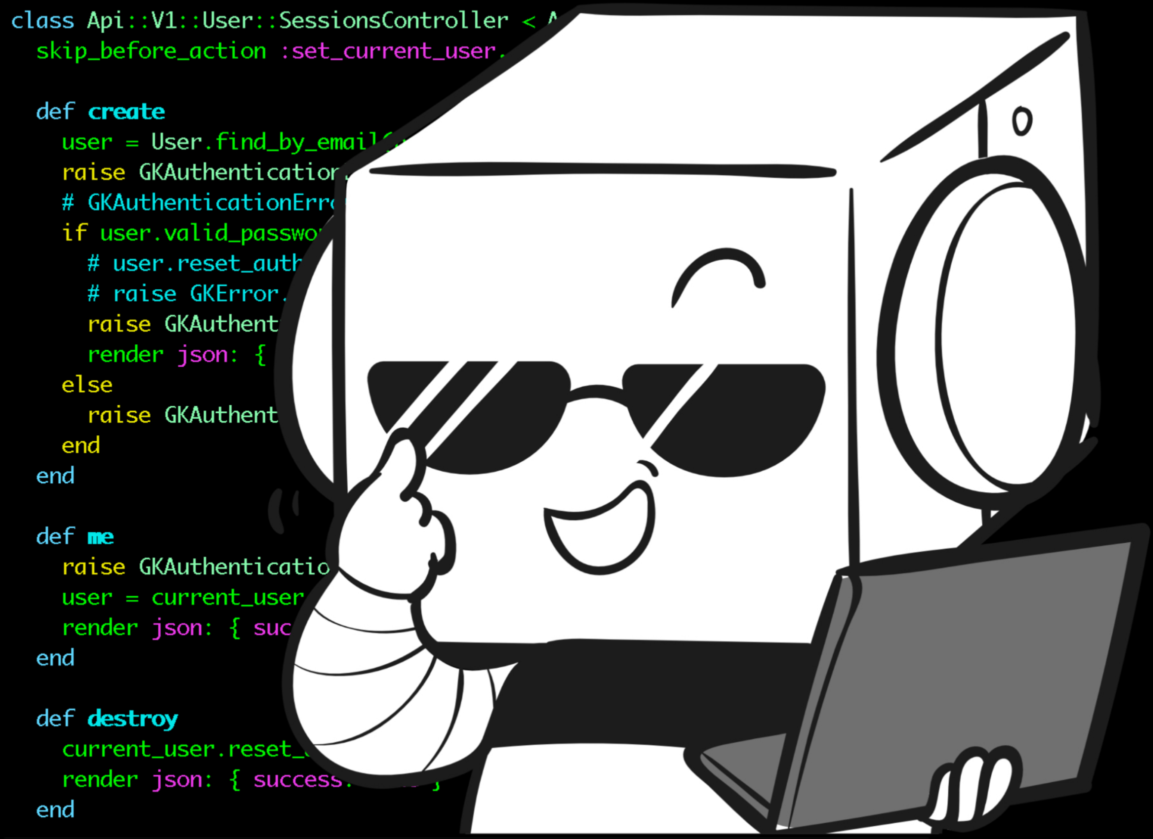
Soon to write API Using Ruby on RAILS

It's so easy, what are you waiting for?



```
class Api::V1::User::SessionsController < A
  skip_before_action :set_current_user,

  def create
    user = User.find_by_email(
    raise GKAuthenticatior
    # GKAuthenticationErr
    if user.valid_passwo
      # user.reset_auth
      # raise GKError.
      raise GKAuthent
      render json: {
    else
      raise GKAuthent
    end
  end

  def me
    raise GKAuthenticatio
    user = current_user
    render json: { suc
  end

  def destroy
    current_user.reset_
    render json: { success:
  end
```

Written by Tanasak Tantitarntong

Soon to be Ruby on Rails –API programmer

Building API could be fund with Ruby on Rails. Let Rails help you with easy and fast coding while you only need to worry about requirements and features.

SaKKo Sama

This book is for sale at http://leanpub.com/soon-to-be-ror-programmer

This version was published on 2021-04-08

Tweet This Book!

Please help SaKKo Sama by spreading the word about this book on Twitter!

The suggested hashtag for this book is #soontoberorprogrammer.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

#soontoberorprogrammer

Contents

1. Introduction

Hi, my name is Tanasak. I started Ruby on Rails over 9 years ago and is addicted to it. I'm not a native English speaker, so please bare with me in grammars and spellings. I will try my best so that it won't hurt the reading experience.

This book is for

- Anyone who's interested in Ruby on Rails
- Anyone who just want to build pure API as backend. All API will be consumed by other frontend such as, Nuxtjs, Nextjs, and Mobile Applications.

All APIs will be tested via Hoppscotch (aka Postwoman), Postman, or curl.

1.1 A little intro about Ruby on Rails

Rails is a server-side web application framework written in Ruby. It follows Model View Controller design pattern. It provides lots of production ready libraries and is supported by a large community.

I personally love rails because of these reasons

1. It is optimized for programmer's happiness. Everything is based on convention over configuration.
2. Ruby is very easy to read and write. I rather spend time writing my application than spending time learning a programming language.
3. The standards that Rails uses is production ready. Studying Rails and following the convention will provide standard security for the application.

There are lots of softwares using Ruby on Rails such as, GitHub, dev.to, Basecamp, HEY, Shopify, Airbnb, Twitch, SoundCloud, Hulu, Zendesk, Square, Cookpad

And there are lots of startup that's using Ruby on Rails as well.

The ugly part that people like to bring up about Ruby on Rails is its speed. I would say that it is good enough for most applications. Unless you application is expected to have very heavy load and concurrency, then you should avoid Ruby on Rails.

However, studying a framework (especially Ruby on Rails) will provide you with a lot of good conventions where you can probably apply them to other frameworks you are using.

This book requires you to have atleast this much tools

Mac (Apple Computer with Intel CPU) highly recommended

Ubuntu Linux Computer (a little more advance) but if you are using then you probably already know what you are doing.

Windows 10 with Window Subsystem for Linux (Ubuntu) installed

I find it difficult to develop Ruby on Rails application Windows, so please get a Mac or get a used/refurblished ones if you can afford it.

I'm writing this book based on my personal experience. I'm quite confidence about many things I write and used. However, if any part is wrong, please do not hesitate to contact me via saklism@gmail.com.

2. Setup your development environment

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

2.1 RVM

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

2.2 Ruby

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

2.3 NVM (optional)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

2.4 NodeJS (optional)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

2.5 Ruby on Rails

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

2.6 Text Editor

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

Atom

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

Sublime Text 3

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

VSCode

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

2.7 Postman / Hoppscotch

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

3. Creating Ruby on Rails Project

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

4. Folder structure

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

4.1 .ruby-version

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

4.2 .ruby-gemset (new)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

4.3 Gemfile & Gemfile.lock

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

4.4 config folder

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

4.5 config/database.yml

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

4.6 Environments files `config/environments/*.rb`

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

4.7 app folder

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

4.8 app/controllers/application_controller.rb

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

4.9 app/models/application_record.rb

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

4.10 Other folders and files

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

5. Ruby

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

5.1 Numbers

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

5.2 BigDecimal (Rails only)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

5.3 Strings

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

5.4 Comment

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

5.5 Print

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

5.6 Object Methods

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

5.7 Array

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

5.8 Hash

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

5.9 Blocks

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

5.10 Modify Built-in Class

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

6. Controller for Beginner

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

6.1 HTTP Request / Actions

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

6.2 Static GET API

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

Cross origin request (rack-cors)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

6.3 POST PUT PATCH DELETE for Static

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

7. Models and Migration

Workin on backend usually mean interacting with database. Whether it is about

1. User Authentication
2. Storing customer data
3. Storing application data

Ruby on Rails provide very easy ways to interact with database. Let's learn the basic in this chapter.

Models

Rails is a Model View Controller framework. The only reasonable part to interact with database is Model. All the codes are located at `app/models/*.rb`

ApplicationRecord

`app/models/application_record.rb` should be super class of all your models.

ApplicationRecord is a subclass of ActiveRecord. ApplicationRecord was recently used in only Rails 5 onwards. So, if you are looking for help, searching ActiveRecord might give you more relevant answer in some cases.

ActiveRecord is an ORM which is provided by Rails. You can define multiple things in its subclass. Mainly,

1. data / columns of the Model
2. Models' relationship
3. Model/Class inheritance
4. Data validation before writing to database
5. Writing data into database object oriented style

Naming Model is as important as Naming anything in Rails. There are conventions to follow.

Naming Model

All model should be TitleCase such as, User, UserRole, DeliveryLocation. And they should be singularized.

File name of all model should be snake_case and singularized. such as, user.rb, user_role.rb, delivery_location.rb

The table that store model's data should be snake_case and pluralized. It should always match the model's name. (if not, you can still override it later) such as, users, user_roles, delivery_locations. The reason is because a table can store multiple records so it should be pluralized.

To make it clear, here are the example

Filename	Class name	Table name
user.rb	User	users
user_role.rb	UserRole	user_roles
delivery_location.rb	DeliveryLocation	delivery_locations
mouse.rb	Mouse	mice
person.rb	Person	people

Naming in Rails is very very important. If you don't follow the conventions, you will run into a lot of weird problems. But if every one follow these conventions, reading other people code will become much easier.

Rails usually require these field/columns

These fields are default requirment for ActiveRecord.

Columns	Type
id	bigint(primary_key)
created_at	datetime
updated_at	datetime

If you named them correctly, created_at and updated_at should be automatically updated for you.

ℹ Rails support code first database natively. So don't worry about creating database just yet. They will all be handled automatically if you use rails generator.

Columns in Database

They should be in snake_case as a default convention.

But if you don't like it, you can use any other cases.

I'm going to use snake_case since it's widely used in Rails.

7.1 Creating Model

We will be using `~/ror/project1` from the previous chapter.

Creating Model is simple. We will be using the generator provided by rails.

> I always use the generator and migration in rails. I never directly migrate the database in the past 9 years. So, basically almost all cases in database migration can be done in code.

When you use code first database, switching database should be as easy as adding gems and changing `config/database.yml`. There might be some different in database but it shouldn't be difficult to do so.

Let's start from creating Model called User

Generate User Model

```
1 rails generate model User first_name:text last_name:text
2 # Running via Spring preloader in process 95681
3 #       invoke  active_record
4 #       create    db/migrate/20201119074454_create_users.rb
5 #       create    app/models/user.rb
6 #       invoke    test_unit
7 #       create      test/models/user_test.rb
8 #       create      test/fixtures/users.yml
```

`1 # rails generate model User first_name:text last_name:text`

`rails generate model` is a command to generate model

`User` is the model name and it should be TitleCase + singularized.

`first_name:text last_name:text` are attributes/columns in database with types.

5 # create app/models/user.rb

This model file is used to communicate directly with the database.

4 # create db/migrate/20201119074454_create_users.rb

Database migration file. This is created according to Model name and attributes you input.

Migration file is versioned by date/time of the generated date.

Let's take a look at migration file.

'db/migrate/20201119074454_create_users.rb'

```ruby
 1 class CreateUsers < ActiveRecord::Migration[6.0]
 2   def change
 3     create_table :users do |t|
 4       t.text :first_name
 5       t.text :last_name
 6 
 7       t.timestamps
 8     end
 9   end
10 end
```

The migration file is written in Ruby. The command may look weird in the beginning but you will get used to it later.

3 # create_table

Create a table called `users`. With this command, you will always get `id` as primary_key

t.timestamps

This command is inside create_table block which means it will be creating `create_at` and `updated_at` for `users` table. Its type would be DateTime.

t.text

This command is inside create_table block which means it will be creating `first_name` and `last_name` as text.

In Ruby there are "String" and :symbol which could be confusing in the beginning.

- :first_name is symbol type. It could be converted to string.
- "first_name" is string and it could be converted to symbol as well.
- Symbol is immutable. If you created :first_name and you want to change it to :first_name1, it's not doable.
- Symbol use less resource than string and it has better performance than string.

7.2 Migrating Database

Migrating database is also done via `rails` command.

```
rails db:migrate
# == 20201119074454 CreateUsers: migrating =====================
# -- create_table(:users)
#    -> 0.0021s
# == 20201119074454 CreateUsers: migrated (0.0021s) ============
```

`rails db:migrate`

will read files from `db/migrate/DATETIME_**.rb`

will check whether each file has been migrated or not.

will only migrate the file which has not been migrated.

The logs after migration

You should be seeing DATETIME of the migration file.

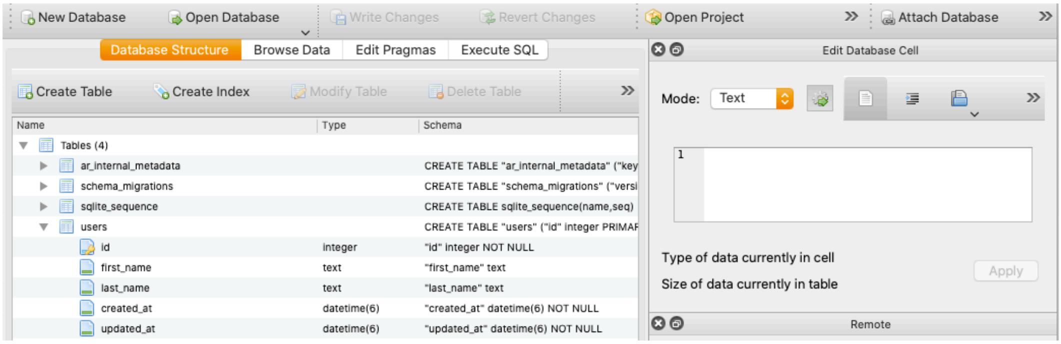
Time it takes and the operations it did.

If the migration failed, only the last migration will be cancelled and reverted. So if you have 10 files and #5 failed then #1-4 will not be reverted. You need to fix #5 and run `rails db:migrate` again. If #5 success then the migration will continue to the next file.

If you try to run `rails db:migrate` again and again, nothing will happen since migration file will only run once and will not run again.

After db migration, since we are using sqlite3, you might want to try browsing the db by using this application. https://sqlitebrowser.org/dl/

The database file should be located here `db/development.sqlite3`



Name	Type	Schema
Tables (4)		
ar_internal_metadata		CREATE TABLE "ar_internal_metadata" ("key
schema_migrations		CREATE TABLE "schema_migrations" ("versi
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
users		CREATE TABLE "users" ("id" integer PRIMAF
id	integer	"id" integer NOT NULL
first_name	text	"first_name" text
last_name	text	"last_name" text
created_at	datetime(6)	"created_at" datetime(6) NOT NULL
updated_at	datetime(6)	"updated_at" datetime(6) NOT NULL

Mode: Text

1

Type of data currently in cell

Size of data currently in table

You should be seeing `users` table and the columns we specified in our migration file.

If you want to add columns to this Model such as `title`, it's very easy. Just run,

'Add columns to User'

```
rails generate migration AddTitleToUser
# Running via Spring preloader in process 14084
#       invoke  active_record
#       create    db/migrate/20201119083726_add_title_to_user.rb
```

This is how you create pure migration file without Model. With this file, you need to add some code.

‘db/migrate/20201119083726_add_title_to_user.rb’

```ruby
class AddTitleToUser < ActiveRecord::Migration[6.0]
  def change
    add_column :users, :title, :text
  end
end
```

Now let’s run migrate again.

```
rails db:migrate
# == 20201119083726 AddTitleToUser: migrating ==================
# -- add_column(:users, :title, :text)
#    -> 0.0017s
# == 20201119083726 AddTitleToUser: migrated (0.0017s) =========
```

Try to open the database with sqlite3 viewer again, you should now be seeing `title` column.

If you did something wrong in your migration and you already migrated your database, do not change the migration file.

- You can either create a new migration file and fix those mistakes or
- Run `rails db:rollback` which will rollback migration one by one. Until you reach the migration you want to change. Only after you rollback, you can change the migration file and run `rails db:migrate` again.

Important
Try to keep all your migration file as correct as possible. So that if you want to start your database from clean state, you can just drop the database and start again by simply run `rails db:migrate`.

8. Rails Console

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

8.1 Ruby

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

8.2 The power of Rails console

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

8.3 Rails powered by Ruby is fun

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

9. CRUD API

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

9.1 Review

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

9.2 RESTful APIs

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

9.3 API GET users

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

9.4 API GET users/:id

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

9.5 API POST /users create

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

9.6 API PATCH /users/:id update

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

9.7 API DELETE /users/:id destroy

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

9.8 Resourceful routes

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

9.9 Final words on CRUD API

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

10. The Power of Scaffolding

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

10.1 rails g scaffold

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

10.2 Controller from scaffold

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

11. Model Relationship

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

11.1 ER Diagram

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

11.2 Starting small project

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

11.3 Scaffold APIs

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

11.4 The Missing APIs

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

11.5 Model's Relationship

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

11.6 Testing Model's Relationship

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

11.7 Add missing API

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

11.8 Extra

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

12. Sending Emails

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

12.1 Action Mailer Basic

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

12.2 HTML Email Template Layouts

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.

12.3 Test sending email

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/soon-to-be-ror-programmer.