To-do List Application using Ruby on Rails

Piyush Jain Goda (godap@sunypoly.edu)

Abstract

To-do list is the simple web application that allows the user to add the task which are to be completed. Since there is always a requirement for the accomplishment of small task, consequently there is always possibility of missing out some of them. The aim is to understand the Ruby language with the help of this simple web application.

Ruby

Introduction

Ruby is a dynamic, interpreted, reflective, object-oriented, and general-purpose programming language. It was designed and developed in the mid-1990s by Yukihiro "Martz" Matsumoto in Japan.[1] Rudy draws inspiration from Lisp, Smalltalk, and Perl. Ruby is a pure object-oriented language, but it is also suitable for procedural and functional programming styles. Ruby is considered to follow the principle of POLA (principle of least astonishment). It means that the language behaves in such a way to minimize the confusion for experienced users. The top website that uses Ruby are Airbnb, Github, Hulu, Bloomberg, etc., this shows the popularity of the language.

Features

- Open source and is freely available on the web.
- General-purpose, interpreted programming language.
- A fully object-oriented programming language.
- Flexibility
- Mixins
- Dynamic typing and Duck typing
- Server-side scripting language similar to Python and Lisp. [2]

Ruby Installation

Ruby has differently installed on different operating systems.

- On Linux/UNIX, you can use the package management system of your distribution or third-party tools (rbenv and RVM).
- On macOS machines, you can use third-party tools (rbenv and RVM).
- On Windows machines, you can use RubyInstaller. [3]

As we installed Ruby in the Windows OS. The following steps to install Ruby in Windows.

Download RubyInstaller. The latest version is Ruby 2.6.1-1[4]

RubyInstaller is the easiest and most used Ruby environment on Windows-based installer that includes Ruby language, an execution environment, documentation and more.

- Download a zipped file having the latest version of Ruby.
- After having downloaded the Ruby archive, unpack it and change into the newly created directory.
- Double-click the Ruby2.6.1-1.exe file.
- The Ruby installation wizard starts and click Next to move to the Important Information page of the wizard and keep moving till Ruby installer completes installing Ruby.

There are many online editor and compiler. For example, Repl.it (https://repl.it/languages/ruby), Rextexter (https://rextester.com/l/ruby_online_compiler), Jdoodle (https://www.jdoodle.com/execute-ruby-online) and more.

Interactive Ruby (IRb)

Interactive Ruby (IRb) provides a shell for experimentation. Within the Irb shell, you can immediately view expression results, line by line. This tool comes along with Ruby installation. Just type 'irb' at your command prompt and an Interactive Ruby Session will start.

For example:
C:\User\irb
irb(main):001:0> "hello world"
=>hello world

All the ruby files will have extension .rb.

Implementation

The Ruby interpreter from ruby-lang.org is the reference implementation that defines the language. It is sometimes known as MRI, or "Matz's Ruby Implementation."

There are a number of alternative implementations of Ruby, including JRuby, IronRuby, Rubinius, Cardinal, and mruby.

Packages

Ruby's package management system is known as RubyGems, and packages or modules distributed using RubyGems are called "gems." RubyGems makes it easy to install Ruby software and can automatically manage complex dependencies between packages.

Ruby on Rails

Introduction

Ruby on Rails is an open source framework for the programing language Ruby. Rails is a web framework that helps make web applications by providing classes for saving to the database, handling URLs and displaying HTML. It provides a large, flexible library of functions and features that have already been developed, which can easily and quickly be implemented into almost any application. Rails is a gem or a Ruby library. [6]

A framework is a collection of code, tools & utilities that give you a specific structure to work with. This structure makes your code more organized. It saves you a lot of work when you learn to use it correctly!

Advantages of Ruby on Rails

You can move extremely fast with Rails. You can get from idea to implementation in a matter of days, or weeks. The other benefits of rails are:

Tooling: Rails provides fantastic tooling that helps you to deliver more features in less time. It provides a standard structure for web apps, where all the common patterns are taken care of for you.

Libraries: There's a gem (3rd party module) for just about anything you can think of.

Code Quality: The quality of the Ruby code to be significantly higher than their PHP or NodeJS equivalents.

Test Automation: The Ruby community is big in to testing and test automation. The Ruby libraries are so great is the reason that helping to provide good quality software.

Large Community: Ruby community that runs regular meetups. It's one of the most popular languages on social coding site Github.

Productivity: Ruby is an eloquent and succinct language, which when combined with the plethora of 3rd party libraries, enables you to development features incredibly fast.

Disadvantages

Runtime Speed: When compared to the runtime speed of NodeJS, Rails is slow.

Boot Speed: Depending on the number of gem dependencies and files, it can take a significant amount of time to start, which can hinder developer performance. This reduces the boot speed of the Rails framework.

Multithreading: Rails supports multithreading, though some of the IO libraries do not, as they keep hold of the GIL (Global Interpreter Lock).

ActiveRecord: AR is used heavily within the Ruby on Rails world and is a hard dependency for many of the RubyGems. Although we think it's a great design pattern, the biggest drawback we see is that your domain becomes tightly coupled to your persistence mechanism.

Rails Philosophy

The Rails philosophy includes 2 guiding principles[8]:

- **Don't Repeat Yourself (DRY):** DRY is a principle of software development which states that "Every piece of knowledge must have an authoritative, unambiguous, single representation within a system. If same piece of code will not repeat again and again, code will be more maintainable, extensible and less buggy.
- **Convention Over Configuration (CoC):** It provides different opinions for the best way to do many things in a web application.

Ruby on Rails Installation

There are three methods to install Ruby:

- Using rbenv
- Using rvm
- From source

Ruby on Rails can be used with either a simple text editor or with an IDE. Some of the Rails IDE are Eclipse, IntellijIDEA, NetBeans, E, RubyMine, Heroku, etc.

Rails Script

Rails provides us some excellent tools that are used to develop Rails application. These tools are packaged as scripts from command line.Following are the most useful Rails scripts used in Rails application:

Rail Console:

The Rails console is a command line utility which runs Rails application from the command line. The Rails console is an extension of Ruby irb. It provides all the features of irb along with the ability to auto-load Rails application environment, including all its classes and components. It helps you to walk through your application step-by-step.

WEBrick Web server:

Rails is configured to automatically use WEBrick server. This server is written in pure Ruby and supports almost all platforms like Windows, Mac or Unix.

Generators:

The Rails include code generation scripts, which are used to automatically generate model and controller classes for an application. By running generator command, skeleton files for all your model and controller classes will be generated. It also generates database migration files for each model it generates.

Migrations:

Migrations bring Rails DRY feature to life. It is a pure Ruby code that define the structure of a database. You don't have to use SQL to write your code while using migration.

Ruby on Rails MVC

Rails is also based on MVC pattern. It basically works as following[8]:

Requests first come to the controller, controller finds an appropriate view and interacts with model which in turn interacts with database and send response to controller. The controller gives the output to the view based on the response.

Model:

The models are classes in Rails. They interact with database, store data, handles validation, transaction, etc. This subsystem is implemented in **ActiveRecord** library. This library provides an interface between database tables and Ruby program code that manipulates database records. Ruby method names are automatically generated from database tables field names.

View:

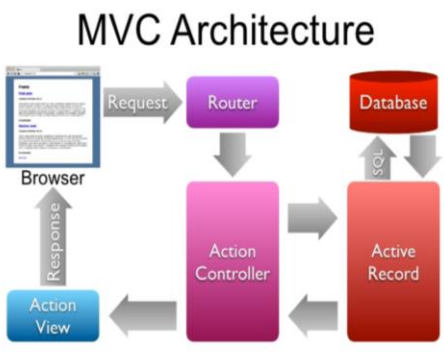
View represent data in a particular format in an application for the users. It handles HTML, CSS, JavaScript, and XML in an application. They do what controller tells them. This subsystem is implemented in **ActionView** library. This library is an Embedded Ruby (Erb) based system which define presentation templates for data presentation.

Controller:

Controller directs traffic to views and models. It query models for data from the database and display the desired result with the help of view in an application. This subsystem is implemented in **ActionController** library. This library is a data broker sitting between ActiveRecord and ActionView.

Routers:

Routers map HTTP requests from the browser to a Controller via the HTTP verb (GET, POST, PUT, DELETE) and the URI of the HTTP request.RVM stands for Ruby Version Manager. It is a command line tool which allows you to easily install, manage and work with different Ruby environments. With RVM, you can easily install different versions of Ruby and easily switch between them.



Ruby on Rails RVM

RVM stands for Ruby Version Manager. It is a command line tool which allows you to easily install, manage and work with different Ruby environments. With RVM, you can easily install different versions of Ruby and easily switch between them.

RVM is maintained by the GitHub community through pull requests sent to the project repository.

Ruby on Rails Bundler

In Rails, bundler provides a constant environment for Ruby projects by tracking and installing suitable gems that are needed. It manages an application's dependencies through its entire life, across many machines, systematically and repeatably.

To use bundler, we have install it by using command

"gem install bundler."

Ruby on Rails Migrations

Migrations are a way to alter database schema over time in a consistent and organized manner. They use a Ruby DSL through which there is no need to write SQL by hand.

Each migration is a new version of the database. Each migration modifies database by adding or removing tables, `columns or entries. Active record will update your db/schema.rb file to match up-to-date structure of your database.

Migration file contains a specific set of instructions for how a database should be created. When this file is run, Rails will make changes in the database automatically. Gradually, the migration file will act as a versioned history of how database has changed. It implies that you will be able to recreate the database from the set of instructions file.

Syntax to create a migration file:

"rails generate migration table_name"

After creating all the required migration files you need to execute them. To execute migration file against database, run the following code:

"rake db:migrate"

Ruby on Rails Scaffolding

Scaffolding is a quick way to produce some major pieces of an application. For auto-generating, a set of models, views, and controllers for a new resource in a single operation, scaffolding is used. Scaffolding is a technique supported by MVC frameworks in which programmers can specify how application database may be used.

To generate a scaffold for the post resource, enter the following command:

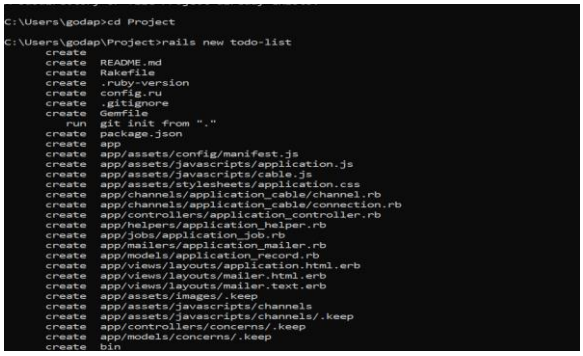
"rails generate scaffold Post name:string title:string content: text"

The scaffold generator will build several files in your application with some folders.

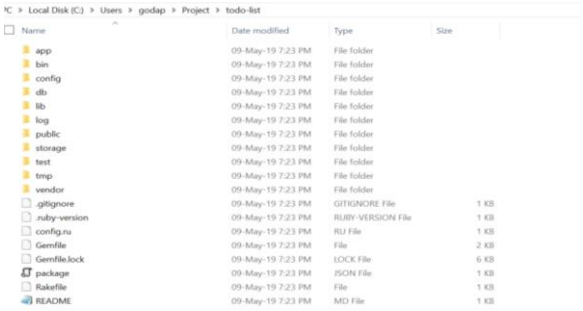
Implementation

As we started learning Ruby and Ruby on Rails, we built few applications using Rails such as Hello World, Welcome, Sign Up page and To-do List. As we installed all the requirement such as ruby and ruby on rails. Now, we provide the steps towards how we have developed this project.

Step 1: Open up command prompt, enter following code:
"rails new todo-list"
This will create rails application named todo-list.

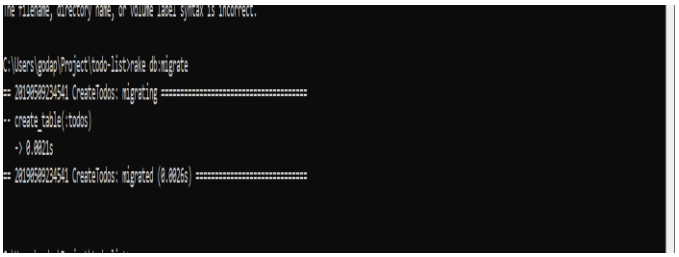


The following is the files that are build by creating rails by above code.

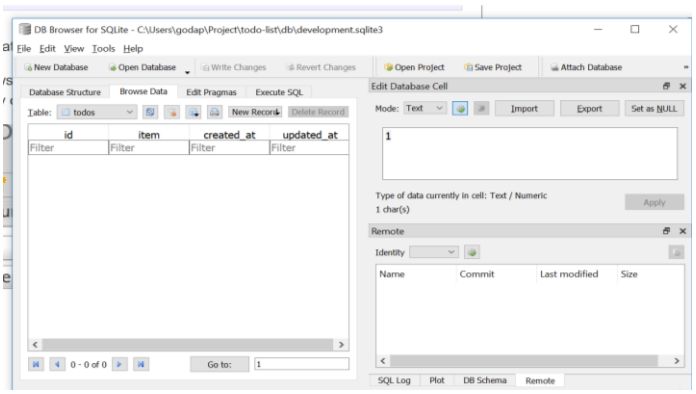


Step 2: Models are special ruby classes that talks to database, store and validates data.
"rails generate model Todo item:string"
The above command will create migration create_todos.

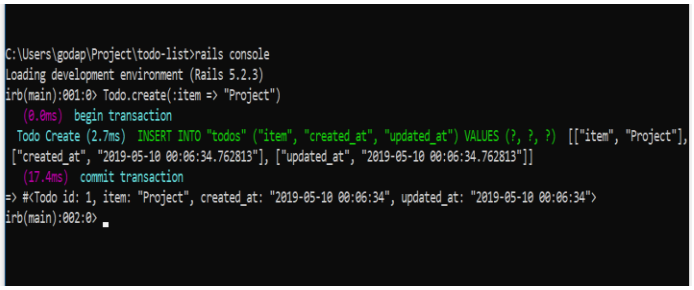
Step 3: Now, we migrate our table to database using migrate.
"rake db:migrate"
This will create new database with table todos and column item.



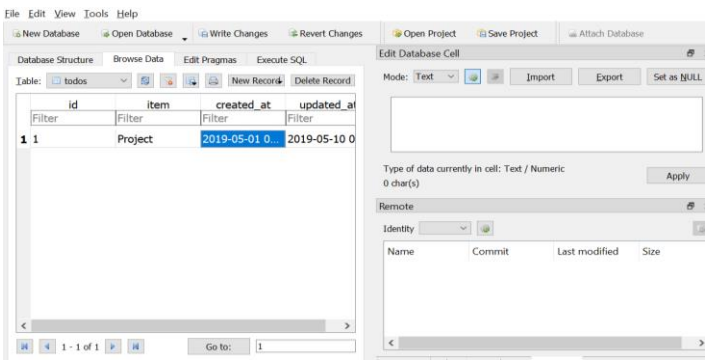
The below is the table that we created in our database. You should see empty database with table todos and columns.



Step 4: Now, we insert records in the database.
Open rails console using "rails console".
Then we add the some data to the table in the database. The following is the image of inserting the data into the database.



The below is the image that shows the data that is produced from the above code.



Step 5: We have our records in the database. Now, we have to show this data in the browser. To show the data in the browser we need to create to parts controllers and views which are present at the location C:Project/todo-list/app.

Controllers are user request to database. It can be GET request and POST request that is stored in a database and the browser.

Views are HTML files that user see through browser. We'll use slim for that. To use rails effectively we need to use ruby code in HTML templates. Default, it is ERB.

The below is the code to create the controller named todos.

"rails generate controller todos"

```
irb(main):002:0> exit

C:\Users\godap\Project\todo-list>rails generate controller todos
      create  app/controllers/todos_controller.rb
      invoke  erb
      create    app/views/todos
      invoke  test_unit
      create    test/controllers/todos_controller_test.rb
      invoke  helper
      create    app/helpers/todos_helper.rb
      invoke    test_unit
      invoke  assets
      invoke    coffee
      create      app/assets/javascripts/todos.coffee
      invoke    scss
      create      app/assets/stylesheets/todos.scss

C:\Users\godap\Project\todo-list>
```

Step 6: Now, we start creating actions that will show the records which we created to browser.

We add the command in the todo controller ac"app/controllers/todos_controller.rb" that will checkout database for all todos and saves into instance variable todos.

```
def index
    @todos = Todo.all
End
```

Now, we need to add route and we do by following code:

```
-   @todos.each do |todo|
ul
    li = todo.item
```

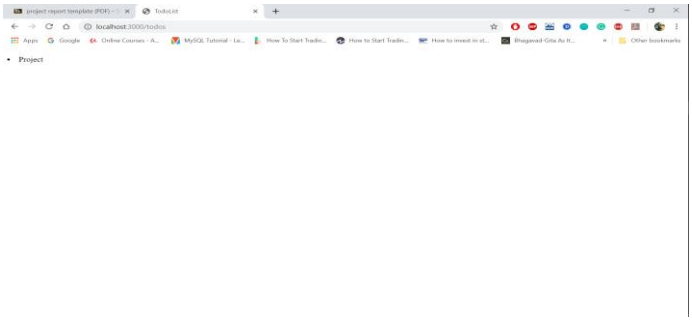
Step 7: Now, we run the server by following command:

"rails server"

```
C:\Users\godap\Project\todo-list>rails server
=> Booting Puma
=> Rails 5.2.3 application starting in development
=> Run `rails server -h` for more startup options
*** SIGUSR2 not implemented, signal based restart unavailable!
*** SIGUSR1 not implemented, signal based restart unavailable!
*** SIGHUP not implemented, signal based logs reopening unavailable!
Puma starting in single mode...
* Version 3.12.1 (ruby 2.5.3-p105), codename: Llamas in Pajamas
* Min threads: 5, max threads: 5
* Environment: development
* Listening on tcp://localhost:3000
Use Ctrl-C to stop
Started GET "/" for ::1 at 2019-05-09 20:59:49 -0400
  (0.4ms)  SELECT "schema_migrations"."version" FROM "schema_migrations" ORDER BY "schema_migrations"."version" ASC
  ↳ C:/Ruby25-x64/lib/ruby/gems/2.5.0/gems/activerecord-5.2.3/lib/active_record/log_subscriber.rb:98
Processing by Rails::WelcomeController#index as HTML
  Rendering C:/Ruby25-x64/lib/ruby/gems/2.5.0/gems/railties-5.2.3/lib/rails/templates/rails/welcome/index.html.erb
  Rendered C:/Ruby25-x64/lib/ruby/gems/2.5.0/gems/railties-5.2.3/lib/rails/templates/rails/welcome/index.html.erb (10.6
ms)
Completed 200 OK in 173ms (Views: 83.6ms | ActiveRecord: 0.0ms)


Started GET "/todos" for ::1 at 2019-05-09 20:59:57 -0400
Processing by TodosController#index as HTML
  Rendering todos/index.html.erb within layouts/application
  Rendered todos/index.html.erb within layouts/application (0.4ms)
```

Then open the browser with the following link "http://localhost:3000/todos"



The above is the required task that we have to do.

Issues encountered

While implementing this project we faced a few issues, of which the predominant ones are stated below:

The below is the error that is produced due to syntax error in the index.

The below is the error due to syntax error during the migrating.

We had occurred various challenges in the other web application other than this.

Conclusion

Ruby on Rails is a web application development framework It is designed to make programming web applications easier by making assumptions about what every developer needs to get started. It allows you to write less code while accomplishing more than many other languages and frameworks. Using Ruby on Rails, we can easily build a simple web application comparing to other languages. We had a fun during implementing the examples in rails. We have gained knowledge how to build a Rails application with the kind of functionality which is actually required to it. We not only learnt how to build the application but also the background process of the implantation. For example, while creating the new application using rails command, in the background it creates the files such as controller, routes, Erb files, etc., that are require to process further.

References

[1]https://en.wikipedia.org/wiki/Ruby_(programming_language)
[2] https://www.javatpoint.com/ruby-tutorial
[3] https://www.ruby-lang.org/en/downloads/
[4] https://rubyinstaller.org/
[5]Book "The Ruby Programming Language by David Flanagan, Yukihiro Matsumoto."
[6] https://guides.rubyonrails.org/getting_started.html
[7] https://www.javatpoint.com/ruby-on-rails-tutorial
[8]https://medium.com/@1sherlynn/ruby-on-rails-beginner-adventures-simple-blog-application-6dffd6dcb11a
[8]http://masteruby.github.io/weekly-rails/2014/03/22/how-to-create-todo-list-app-static-pages.html#.XNTcaI5Kjb0