Ruby on Rails

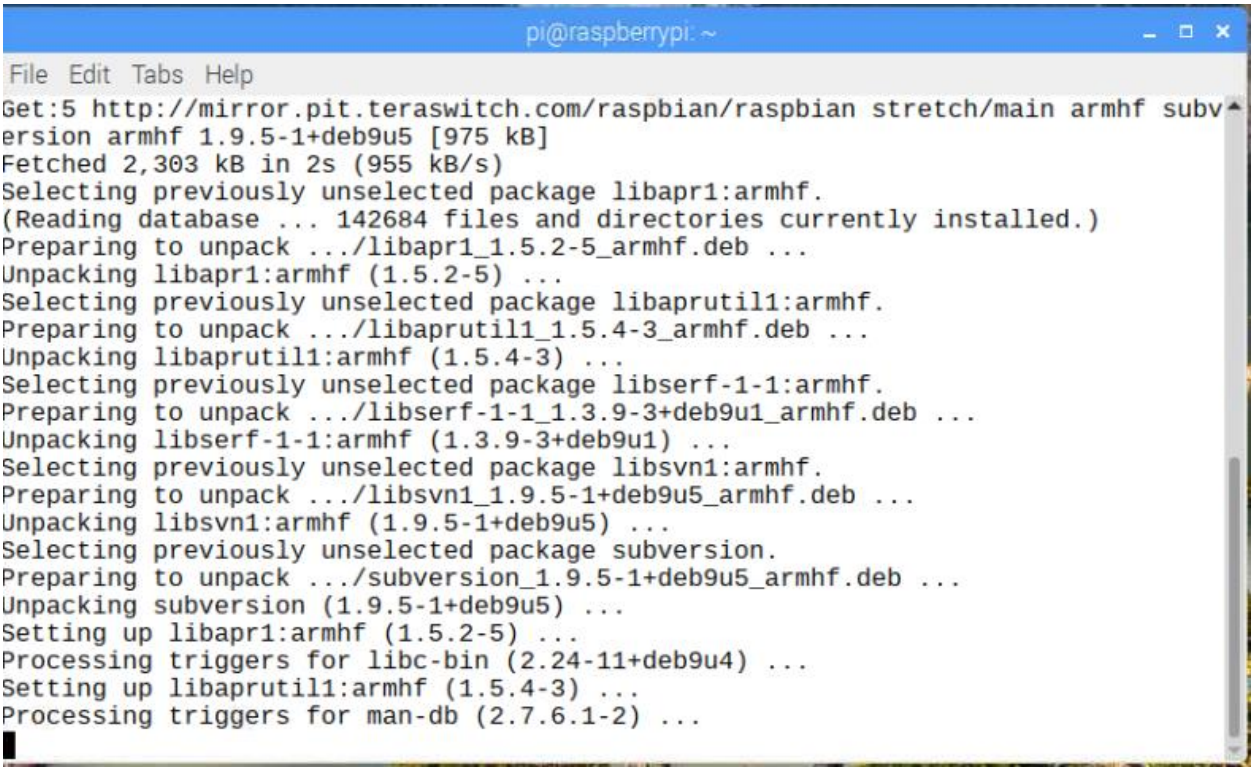
System Environment:

- Raspberry Pi 4
- Raspbian OS
- Ruby Installed

Installation

1. Open the command terminal and run the following command to install Ruby on Rails:

sudo apt-get install -y git curl zlib1g-dev subversion



If an error occurs, try updating your RPi with the following command:

sudo apt-get update

Then try step 1 again.

2. Install additional dependencies:

sudo apt-get install -y openssl libreadline6-dev git-core zlib1g libssl-dev

sudo apt-get install -y libyaml-dev libsqlite3-dev sqlite3

sudo apt-get install -y libxml2-dev libxslt-dev

sudo apt-get install -y autoconf automake libtool bison

3. Now we will install RVM (Ruby Version Manager) Run the following command:

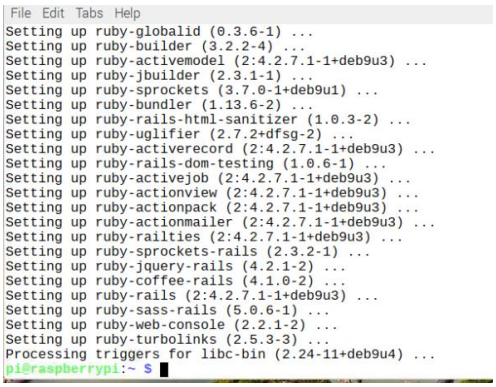
curl -L get.rvm.io | bash -s stable --rails

4. Run this script to enable Ruby:

source ~/.rvm/scripts/rvm

5. Some Raspberry Pis might run into an issue when using the method above. If such an error occurs, use the following command to install rails instead:

Sudo apt-get install ruby-rails



6. Ensure you have Ruby and Rails by running the following commands:

ruby- v

rails -v

```
pi@raspberrypi:~ $ ruby -v
ruby 2.3.3p222 (2016-11-21) [arm-linux-gnueabihf]
pi@raspberrypi:~ $ rails -v
Rails 4.2.7.1
pi@raspberrypi:~ $
```

7. We will also need nodeJS:

sudo apt-get install nodejs

8. Let's test rails by making an example server:

rails new myExample

```
Jsing rails-dom-testing 1.0.6
Jsing activejob 4.2.7.1
Jsing activerecord 4.2.7.1
Jsing actionview 4.2.7.1
Jsing actionpack 4.2.7.1
Jsing actionmailer 4.2.7.1
Jsing railties 4.2.7.1
Jsing sprockets-rails 2.3.2
Jsing coffee-rails 4.1.0
Jsing jquery-rails 4.2.1
Jsing rails 4.2.7.1
Jsing sass-rails 5.0.6
Jsing web-console 2.2.1
Jsing turbolinks 2.5.3
Bundle complete! 12 Gemfile dependencies, 54 gems now installed.
Jse `bundle show [gemname]` to see where a bundled gem is installed.
         run  bundle exec spring binstub --all
* bin/rake: spring inserted
* bin/rails: spring inserted
pi@raspberrypi:~ $
```

9. Go into your example directory:

Cd myExample

10. Let's add some data:

rails g scaffold user name age:integer

```
pi@raspberrypi:~/myExample $ rails g scaffold user name age:integer
Warning: Running `gem pristine --all` to regenerate your installed gemspecs (and deleting t
ng.
               active_record
      create     db/migrate/20200809223248_create_users.rb
      create     app/models/user.rb
                 test_unit
      create       test/models/user_test.rb
      create       test/fixtures/users.yml
               resource_route
       route     resources :users
               scaffold_controller
      create     app/controllers/users_controller.rb
                 erb
      create       app/views/users
      create       app/views/users/index.html.erb
      create       app/views/users/edit.html.erb
      create       app/views/users/show.html.erb
      create       app/views/users/new.html.erb
      create       app/views/users/_form.html.erb
                 test_unit
      create       test/controllers/users_controller_test.rb
                 helper
      create       app/helpers/users_helper.rb
                   test_unit
                 jbuilder
      create       app/views/users/index.json.jbuilder
      create       app/views/users/show.json.jbuilder
               assets
                 coffee
      create       app/assets/javascripts/users.coffee
                 scss
      create       app/assets/stylesheets/users.scss
               scss
      create     app/assets/stylesheets/scaffolds.scss
pi@raspberrypi:~/myExample $
```

11. Now let's migrate the database in order to initialize our DB:

rake db:migrate

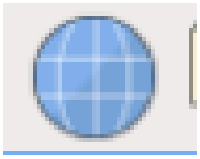
```
pi@raspberrypi:~/myExample $ rake db:migrate
== 20200809223248 CreateUsers: migrating ======================================
-- create_table(:users)
   -> 0.0055s
== 20200809223248 CreateUsers: migrated (0.0059s) =============================

pi@raspberrypi:~/myExample $
```

12. Let's also initialize our server:

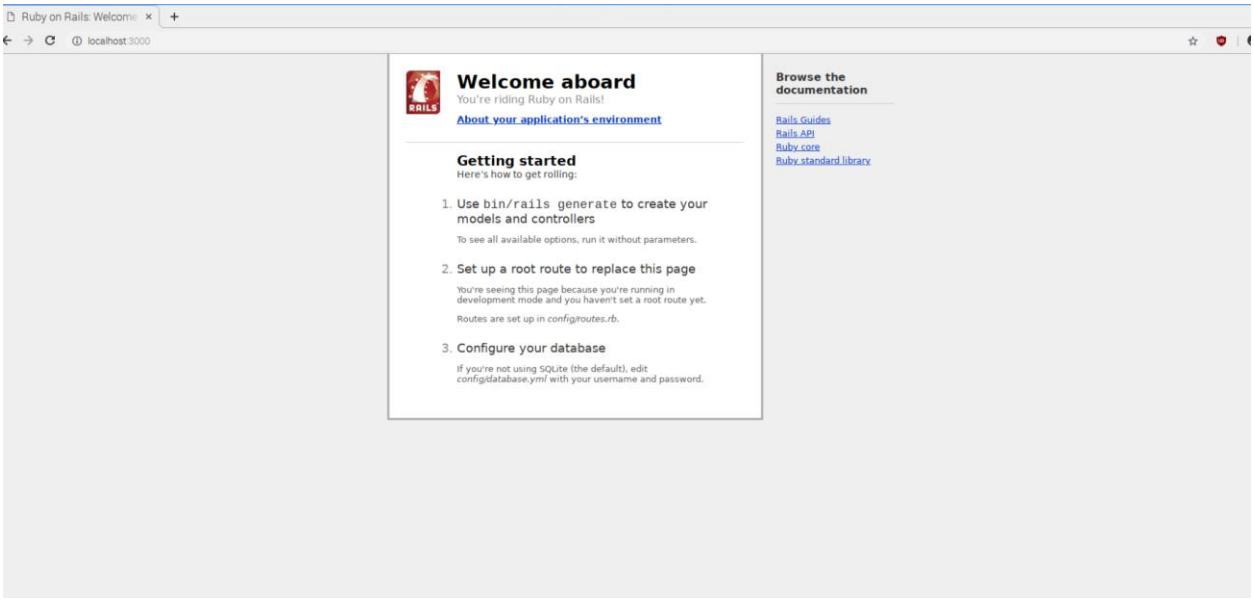
rails server

Our server is now up and running. Make sure to leave the command prompt open.

13. At the top right of the Raspberry Pi navbar there is an internet browser tab. Open the browser.



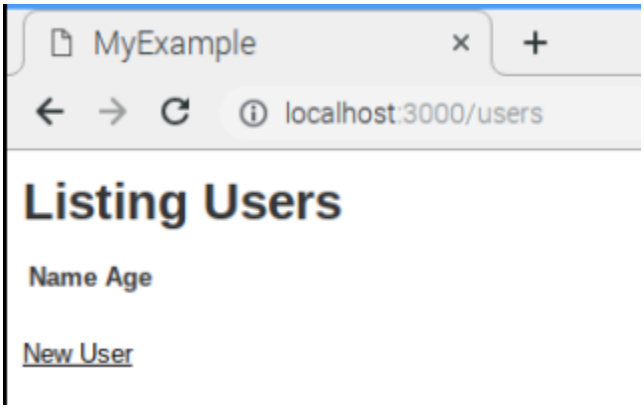
14. On the URL type in localhost:3000 to access our server. You should see a Rails icon followed by "Welcome abord." If this is the case, then we have successfully installed Rails and initialized a server.



15. Each time a user navigates to the URL, a prompt should appear on the console where we ran our server:

```
  ActiveRecord::SchemaMigration Load (0.8ms)  SELECT "schema_migrations".* FROM "schema_migrations"
Processing by Rails::WelcomeController#index as HTML
  Rendered /usr/lib/ruby/vendor_ruby/rails/templates/rails/welcome/index.html.erb (9.4ms)
Completed 200 OK in 125ms (Views: 66.7ms | ActiveRecord: 0.0ms)
```

16. Go back to the Internet browser and navigate to localhost:3000/users The page should reload and you should see the database we created.



17. We can click on new user to enter a new record into the database:

User was successfully created.

Name: John

Age: 21

Edit | Back

18. The new record should appear when we go back to our users directory: