

Ruby on Rails

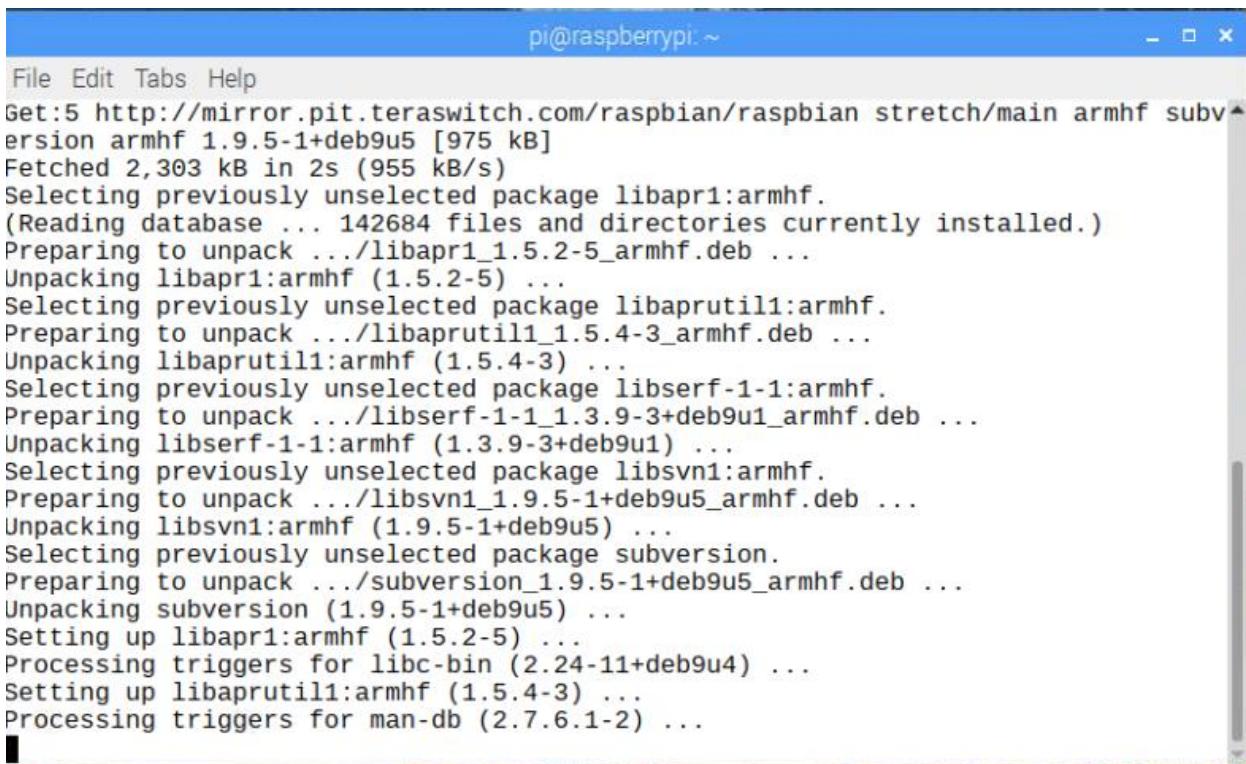
System Environment:

- 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
```



The screenshot shows a terminal window titled "pi@raspberrypi: ~". The window contains the output of the command "sudo apt-get install -y git curl zlib1g-dev subversion". The output shows the progress of the package download and installation, including file counts, sizes, and dependency resolution.

```
pi@raspberrypi: ~
File Edit Tabs Help
Get:5 http://mirror.pit.terawatch.com/raspbian/raspbian stretch/main armhf subversion armhf 1.9.5-1+deb9u5 [975 kB]
Fetched 2,303 kB in 2s (955 kB/s)
Selecting previously unselected package libapr1:armhf.
(Reading database ... 142684 files and directories currently installed.)
Preparing to unpack .../libapr1_1.5.2-5_armhf.deb ...
Unpacking libapr1:armhf (1.5.2-5) ...
Selecting previously unselected package libaprutil1:armhf.
Preparing to unpack .../libaprutil1_1.5.4-3_armhf.deb ...
Unpacking libaprutil1:armhf (1.5.4-3) ...
Selecting previously unselected package libserf-1-1:armhf.
Preparing to unpack .../libserf-1-1_1.3.9-3+deb9u1_armhf.deb ...
Unpacking libserf-1-1:armhf (1.3.9-3+deb9u1) ...
Selecting previously unselected package libsvn1:armhf.
Preparing to unpack .../libsvn1_1.9.5-1+deb9u5_armhf.deb ...
Unpacking libsvn1:armhf (1.9.5-1+deb9u5) ...
Selecting previously unselected package subversion.
Preparing to unpack .../subversion_1.9.5-1+deb9u5_armhf.deb ...
Unpacking subversion (1.9.5-1+deb9u5) ...
Setting up libapr1:armhf (1.5.2-5) ...
Processing triggers for libc-bin (2.24-11+deb9u4) ...
Setting up libaprutil1:armhf (1.5.4-3) ...
Processing triggers for man-db (2.7.6.1-2) ...
```

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:

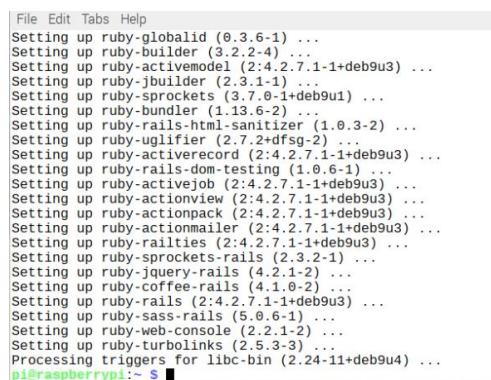
```
curl -L get.rvm.io | bash -s stable --rails
```

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



```
File Edit Tabs Help
Setting up ruby-globalid (0.3.6-1) ...
Setting up ruby-builder (3.2.2-4) ...
Setting up ruby-activemodel (2:4.2.7.1-1+deb9u3) ...
Setting up ruby-jbuilder (2.3.1-1) ...
Setting up ruby-sprockets (3.7.0-1+deb9u1) ...
Setting up ruby-bundler (1.13.6-2) ...
Setting up ruby-rails-html-sanitizer (1.0.3-2) ...
Setting up ruby-uglifier (2.7.2+dfsg-2) ...
Setting up ruby-active-record (2:4.2.7.1-1+deb9u3) ...
Setting up ruby-rails-dom-testing (1.0.6-1) ...
Setting up ruby-active-job (2:4.2.7.1-1+deb9u3) ...
Setting up ruby-action-view (2:4.2.7.1-1+deb9u3) ...
Setting up ruby-action-pack (2:4.2.7.1-1+deb9u3) ...
Setting up ruby-action-mailer (2:4.2.7.1-1+deb9u3) ...
Setting up ruby-railties (2:4.2.7.1-1+deb9u3) ...
Setting up ruby-sprockets-rails (2.3.2-1) ...
Setting up ruby-jquery-rails (4.2.1-2) ...
Setting up ruby-coffee-rails (4.1.0-2) ...
Setting up ruby-rails (2:4.2.7.1-1+deb9u3) ...
Setting up ruby-sass-rails (5.0.6-1) ...
Setting up ruby-web-console (2.2.1-2) ...
Setting up ruby-turbolinks (2.5.3-3) ...
Processing triggers for libc-bin (2.24-11+deb9u4) ...
pi@raspberrypi:~ $
```

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

```
----[redacted]-----
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.
Use `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 them).
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)
 -> 0.005s
== 20200809223248 CreateUsers: migrated (0.0059s) =====
pi@raspberrypi:~/myExample $
```

12. Let's also initialize our server:

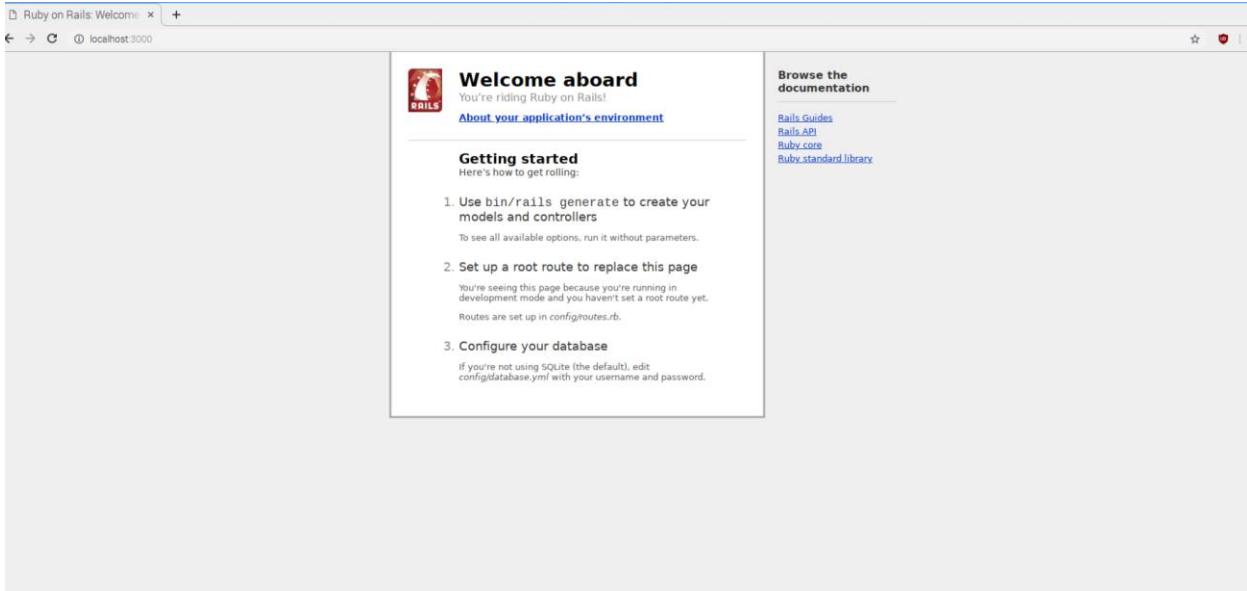
`rails server`

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 aboard.” 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.

The screenshot shows a web browser window titled 'MyExample'. The address bar displays 'localhost:3000/users'. The main content area is titled 'Listing Users' and contains two columns: 'Name' and 'Age'. Below this, there is a link labeled 'New User'.

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

New User

Name

Age

[Create User](#)

[Back](#)

User was successfully created.

Name: John

Age: 21

[Edit](#) | [Back](#)

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



Listing Users

Name Age

John 21 [Show](#) [Edit](#) [Destroy](#)

[New User](#)