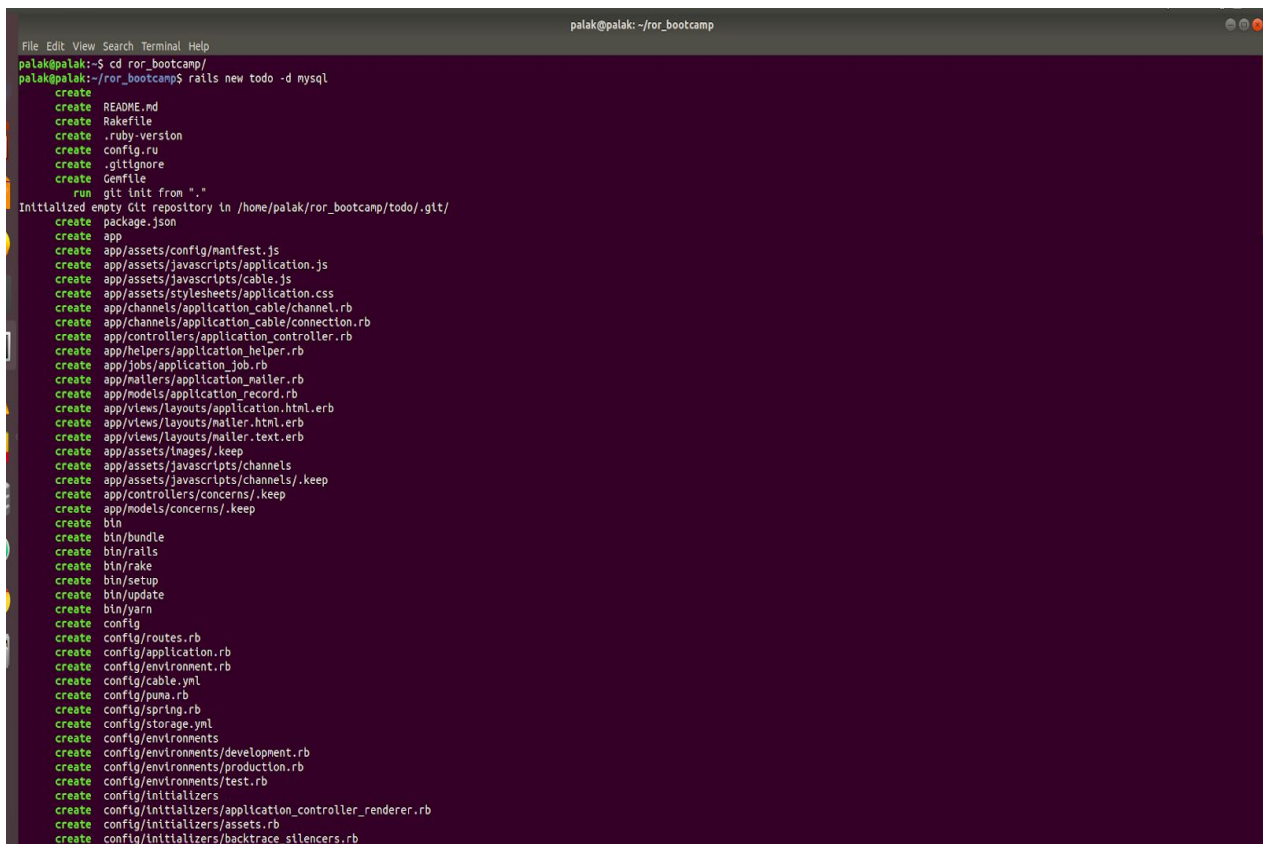


CREATING A SIMPLE TODO APP USING RAILS AND MYSQL

1. Create a rails application with database as mysql . Make sure mysql is already installed.

rails new todo -d mysql



```
palak@palak: ~/ror_bootcamp
palak@palak:~/ror_bootcamp$ rails new todo -d mysql
create
create  README.md
create  Rakefile
create  .ruby-version
create  config.ru
create  .gitignore
create  Gemfile
run    git init from "."
Initialized empty Git repository in /home/palak/ror_bootcamp/todo/.git/
create  package.json
create  app
create  app/assets/config/manifest.js
create  app/assets/javascripts/application.js
create  app/assets/javascripts/cable.js
create  app/assets/stylesheets/application.css
create  app/channels/application_cable/channel.rb
create  app/channels/application_cable/connection.rb
create  app/controllers/application_controller.rb
create  app/helpers/application_helper.rb
create  app/jobs/application_job.rb
create  app/mailers/application_mailer.rb
create  app/models/application_record.rb
create  app/views/layouts/application.html.erb
create  app/views/layouts/mailer.html.erb
create  app/views/layouts/mailer.text.erb
create  app/assets/images/.keep
create  app/assets/javascripts/channels/.keep
create  app/controllers/concerns/.keep
create  app/models/concerns/.keep
create  bin
create  bin/bundle
create  bin/rails
create  bin/rake
create  bin/setup
create  bin/update
create  bin/yarn
create  config
create  config/routes.rb
create  config/application.rb
create  config/environment.rb
create  config/cable.yml
create  config/puma.rb
create  config/spring.rb
create  config/storage.yml
create  config/environments
create  config/environments/development.rb
create  config/environments/production.rb
create  config/environments/test.rb
create  config/initializers
create  config/initializers/application_controller_renderer.rb
create  config/initializers/assets.rb
create  config/initializers/backtrace_silencers.rb
```

2. Go inside the newly created folder called todo and ls to see the folders and files which got created.

cd todo

ls

```
File Edit View Search Terminal Help
Installing thor 0.20.3
Fetching railties 5.2.2
Installing railties 5.2.2
Using coffee-rails 4.2.2
Using multi_json 1.13.1
Fetching jquery 2.0.0
Installing jquery 2.0.0
Using rb-fsevent 0.10.3
Using rb-inotify 0.9.10
Using ruby_dep 1.5.0
Using listen 3.1.5
Using mysql2 0.5.2
Using puma 3.12.0
Using sprockets 3.7.2
Using sprockets-rails 3.2.1
Fetching rails 5.2.2
Installing rails 5.2.2
Using rubyzip 1.2.2
Using sass-listen 4.0.0
Fetching sass 3.7.2
Installing sass 3.7.2
Fetching tilt 2.0.9
Installing tilt 2.0.9
Using sass-rails 5.0.7
Fetching selenium-webdriver 3.141.0
Installing selenium-webdriver 3.141.0
Using spring 2.0.2
Using spring-watcher-listen 2.0.1
Using turbolinks-source 5.2.0
Using turbolinks 5.2.0
Fetching uglifier 4.1.20
Installing uglifier 4.1.20
Using web-console 3.7.0
(Bundle complete) 18 Gemfile dependencies, 79 gems now installed.
Use 'bundle info [gemname]' to see where a bundled gem is installed.
Post-install message from sass:

Ruby Sass is deprecated and will be unmaintained as of 26 March 2019.

* If you use Sass as a command-line tool, we recommend using Dart Sass, the new
  primary implementation: https://sass-lang.com/install

* If you use Sass as a plug-in for a Ruby web framework, we recommend using the
  sassc gem: https://github.com/sass/sassc-ruby#readme

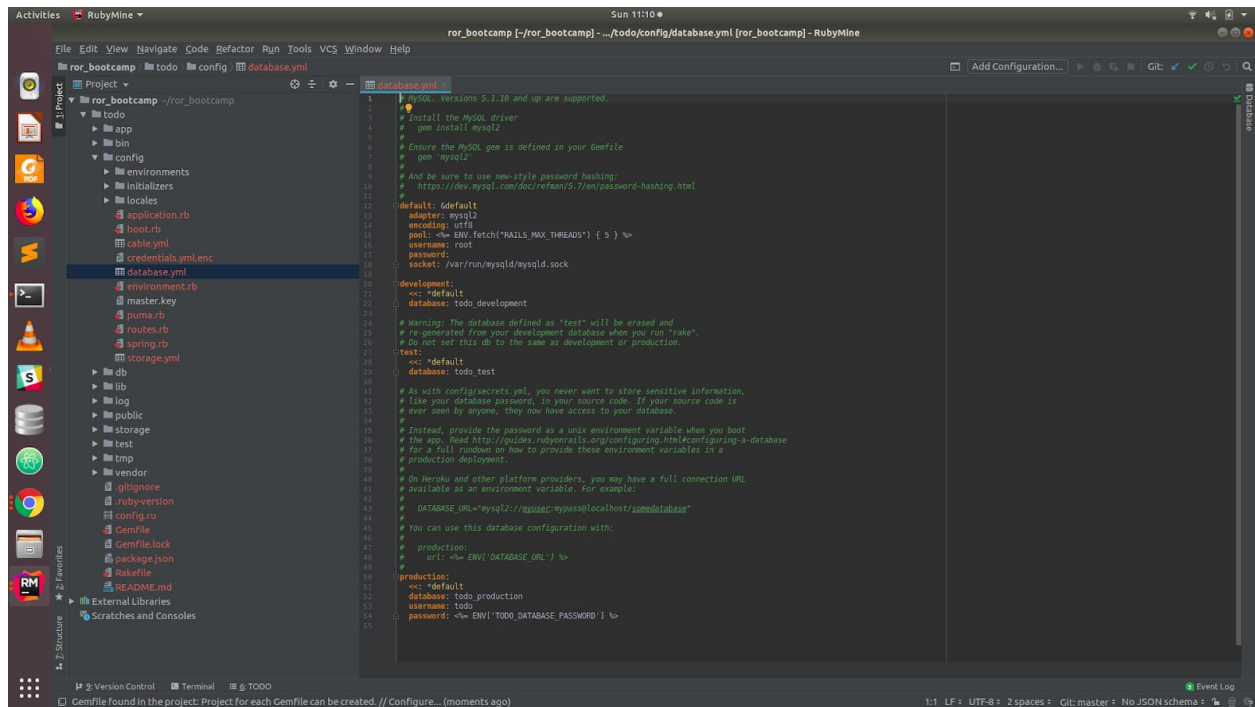
* For more details, please refer to the Sass blog:
  http://sass.logdown.com/posts/7081811

$ run bundle exec spring binstub --all
* bin/rake: spring inserted
* bin/rails: spring inserted
palak@palak:~/ror_bootcamp$ cd todo
palak@palak:~/ror_bootcamp/todo$ ls
app  bin  config  config.ru  db  Gemfile  Gemfile.lock  lib  log  package.json  public  Rakefile  README.md  storage  test  tmp  vendor
palak@palak:~/ror_bootcamp/todo$
```

3. Now open the folder **todo** in RubyMine or any other text editor you use.

Go to `todo/config/database.yml`

Open the file.



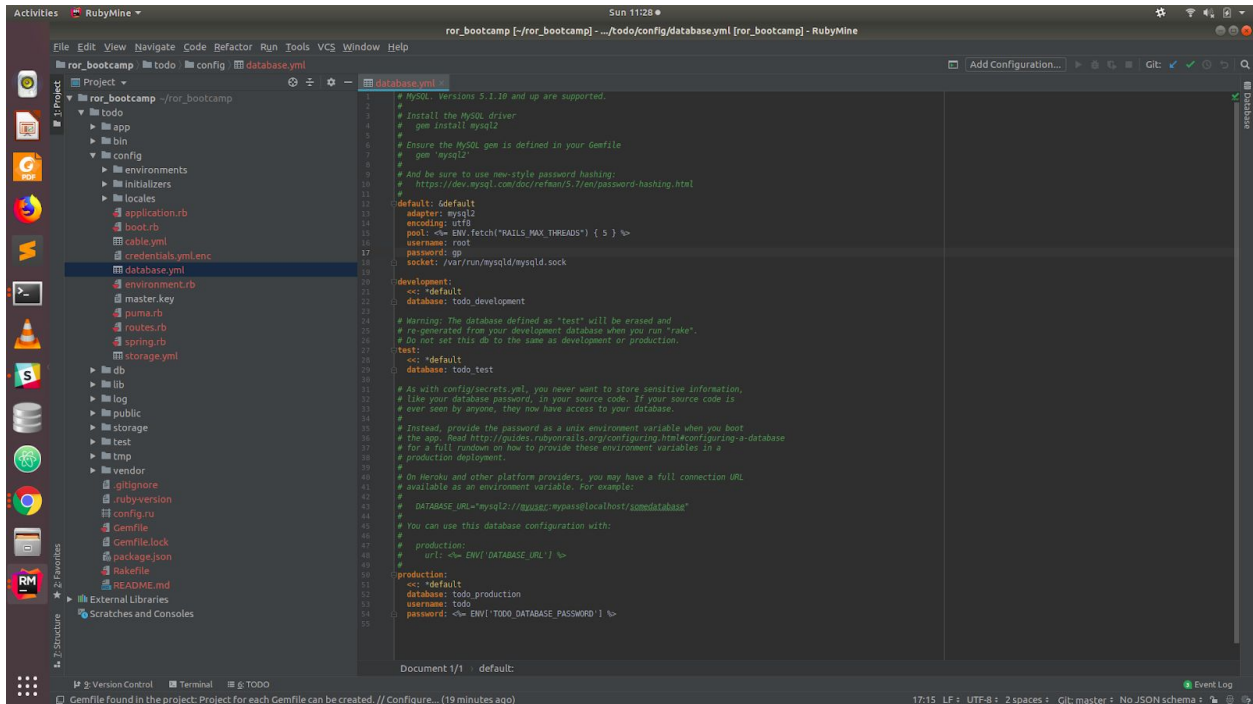
You can see that the adapter says `mysql2` which means that the database which we are using is `mysql`.

4. Fill the username and password field with the username and password you chose while installing mysql in your systems.

Eg. my username and password are

Username: root

Password: gp



```
1 # MySQL: Versions 5.1.10 and up are supported.
2
3 # Install the MySQL driver
4 # gem install mysql2
5
6 # Ensure the MySQL gem is defined in your Gemfile
7 # gem "mysql2"
8
9 # And be sure to use new-style password hashing:
10 # https://dev.mysql.com/doc/refman/5.7/en/password-hashing.html
11 #
12
13 default: <default>
14   adapter: mysql2
15   encoding: utf8
16   pool: <%= ENV.fetch("RAILS_MAX_THREADS") { 5 } %>
17   username: root
18   password: gp
19   socket: /var/run/mysqld/mysqld.sock
20
21 development:
22   <<: default
23   database: todo_development
24
25 # Warning: The database defined as "test" will be erased and
26 # re-generated from your development database when you run "rake".
27 # Do not set this db to the same as development or production.
28 test:
29   <<: default
30   database: todo_test
31
32 # As with config/secrets.yml, you never want to store sensitive information,
33 # like your database password, in your source code. If your source code is
34 # ever seen by anyone, they now have access to your database.
35 #
36 # Instead, provide the password as a unix environment variable when you boot
37 # the app. See http://guides.rubyonrails.org/configuring.html#configuring-a-database
38 # for a full rundown on how to provide these environment variables in a
39 # production deployment.
40 #
41 # On Heroku and other platform providers, you may have a full connection URL
42 # available as an environment variable. For example:
43 #
44 # DATABASE_URL="mysql2://username:password@localhost/somedatabase"
45 #
46 # You can use this database configuration with:
47 #
48 # production:
49 #   url: <%= ENV["DATABASE_URL"] %>
50 #
51 # production:
52   <<: default
53   database: todo_production
54   username: todo
55   password: <%= ENV["TODO_DATABASE_PASSWORD"] %>
```

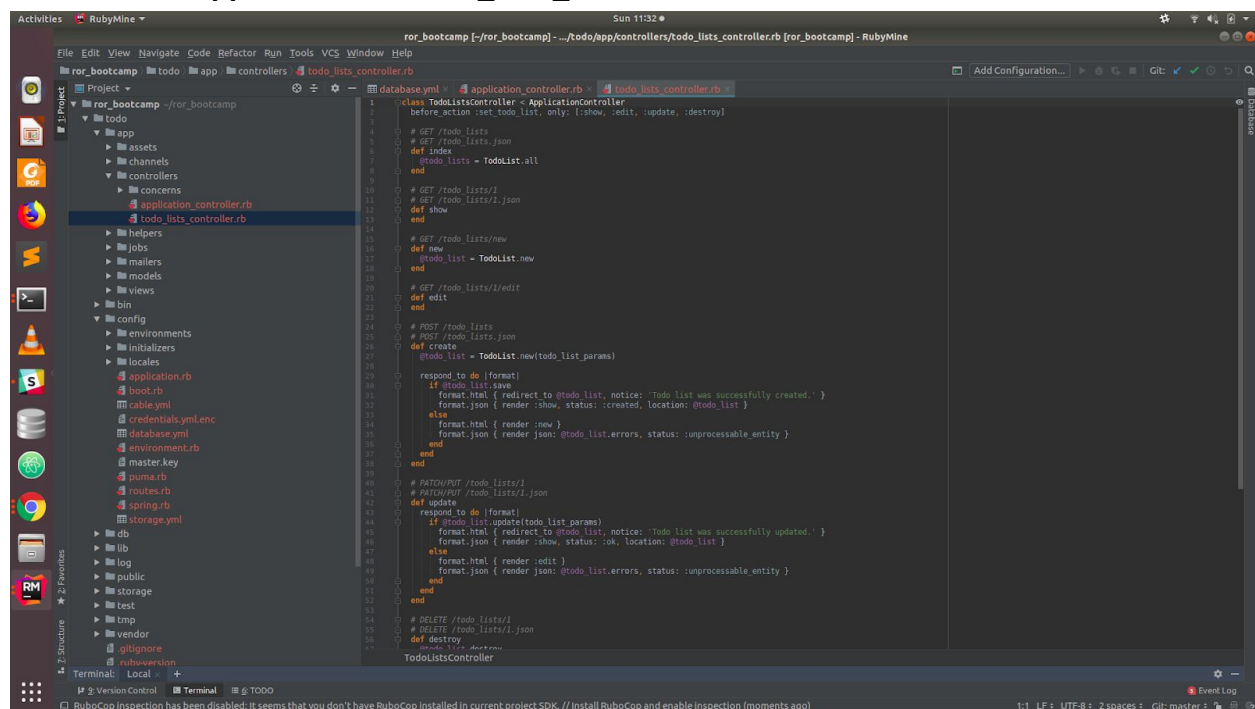
5. After that create a todo_list using scaffold

rails g scaffold todo_list title:string description:text

```
palak@palak:~/r/ror_bootcamp/todo$ rails g scaffold todo_list title:string description:text
Running via Spring preloader in process 10272
Invoke active_record
create db/migrate/20181209060122_create_todo_lists.rb
create app/models/todo_list.rb
invoke test_unit
create test/models/todo_list_test.rb
create test/fixtures/todo_lists.yml
invoke resource_route
route resources :todo_lists
invoke scaffold_controller
create app/controllers/todo_lists_controller.rb
invoke erb
create app/views/todo_lists
create app/views/todo_lists/index.html.erb
create app/views/todo_lists/edit.html.erb
create app/views/todo_lists/show.html.erb
create app/views/todo_lists/new.html.erb
create app/views/todo_lists/_form.html.erb
invoke test_unit
create test/controllers/todo_lists_controller_test.rb
create test/system/todo_lists_test.rb
invoke helper
create app/helpers/todo_lists_helper.rb
invoke test_unit
create app/views/todo_lists/index.json.jbuilder
create app/views/todo_lists/show.json.jbuilder
create app/views/todo_lists/_todo_list.json.jbuilder
invoke assets
create app/assets/javascripts/todo_lists.coffee
invoke scss
create app/assets/stylesheets/todo_lists.scss
invoke scss
create app/assets/stylesheets/scaffolds.scss
palak@palak:~/r/ror_bootcamp/todo$
```

This will create the controller, model, views, migration whatever is necessary. We will have a look at the things it generated for us one by one.

6. Go to todo/app/controllers/todo_lists_controller.rb



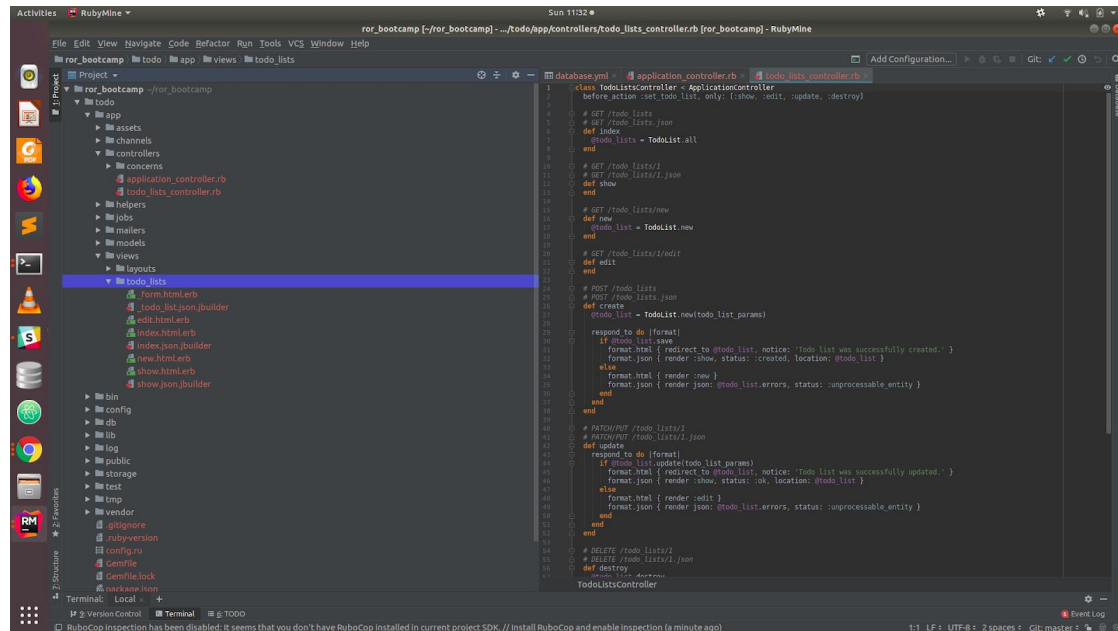
You can see the class `TodoListsController` which is inheriting from the class `ApplicationController` in the first line.

The class has all the functions defined which are called actions. They are defined with a keyword `def` and ended with `end`. Eg `index`, `show`, `new` etc.

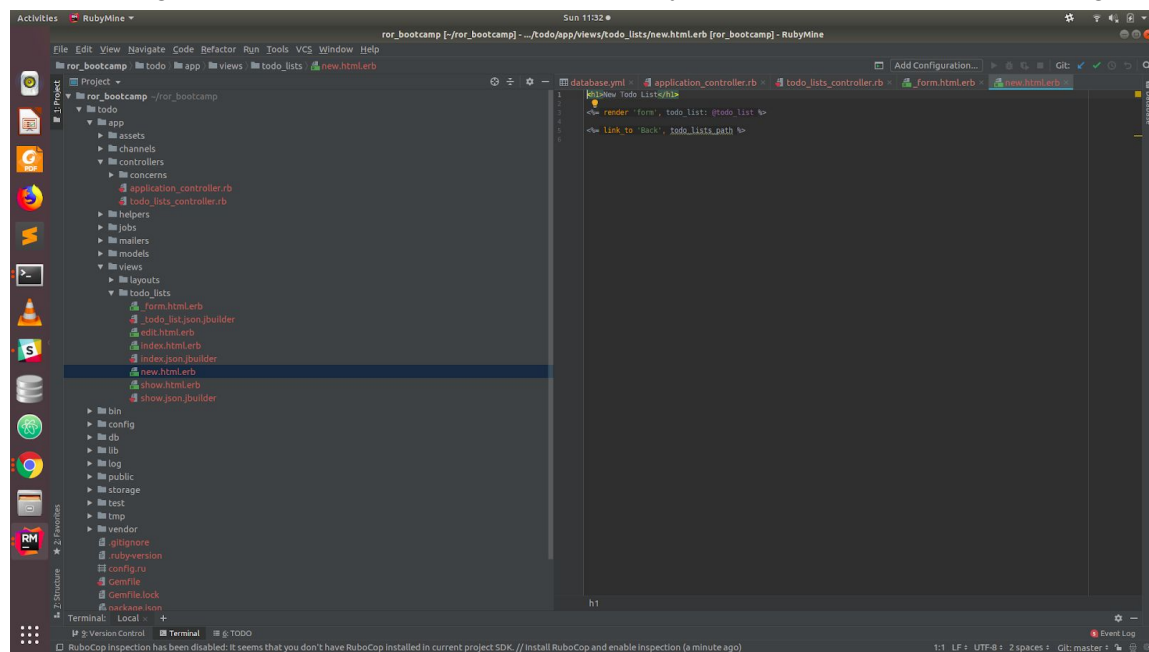
7. Go to todo/app/views/todo_lists

Here you can find lot of .html.erb files which take care of how your application appears. .erb refers to embedded ruby which means you can use ruby variables inside the html files but there is a proper syntax for that `<% %>` and `<%= %>`. The difference between these you can google.

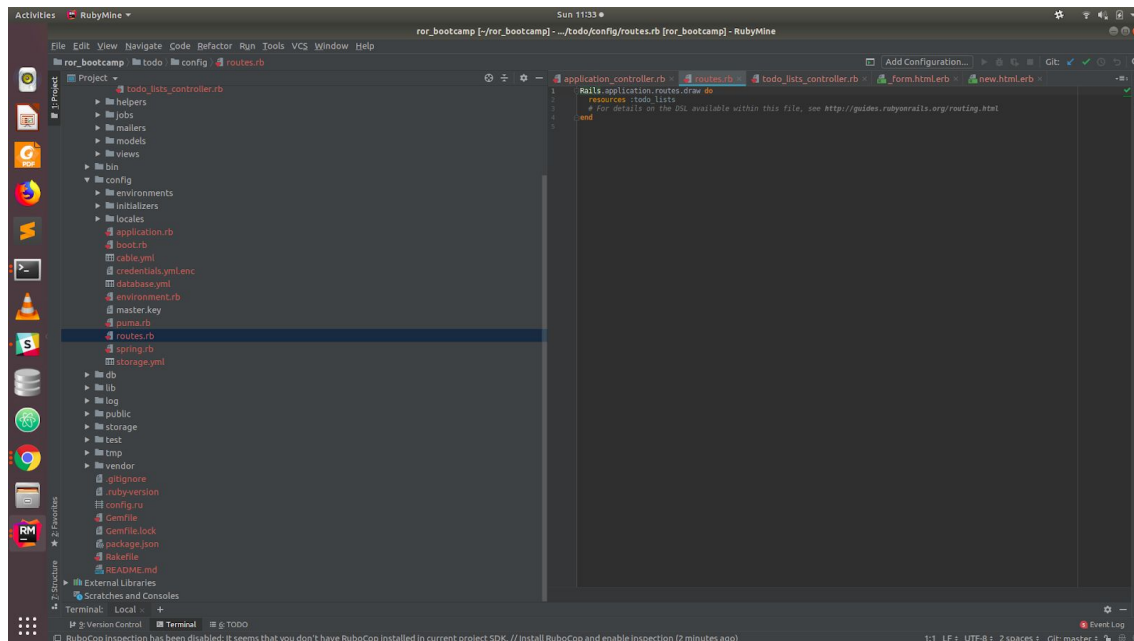
All the html.erb files are files are named same as the action. This is how rails know which html file to render when by finding the corresponding named html.erb file from the views.



8. You can go to each of these .html.erb files and try to understand what's happening there.



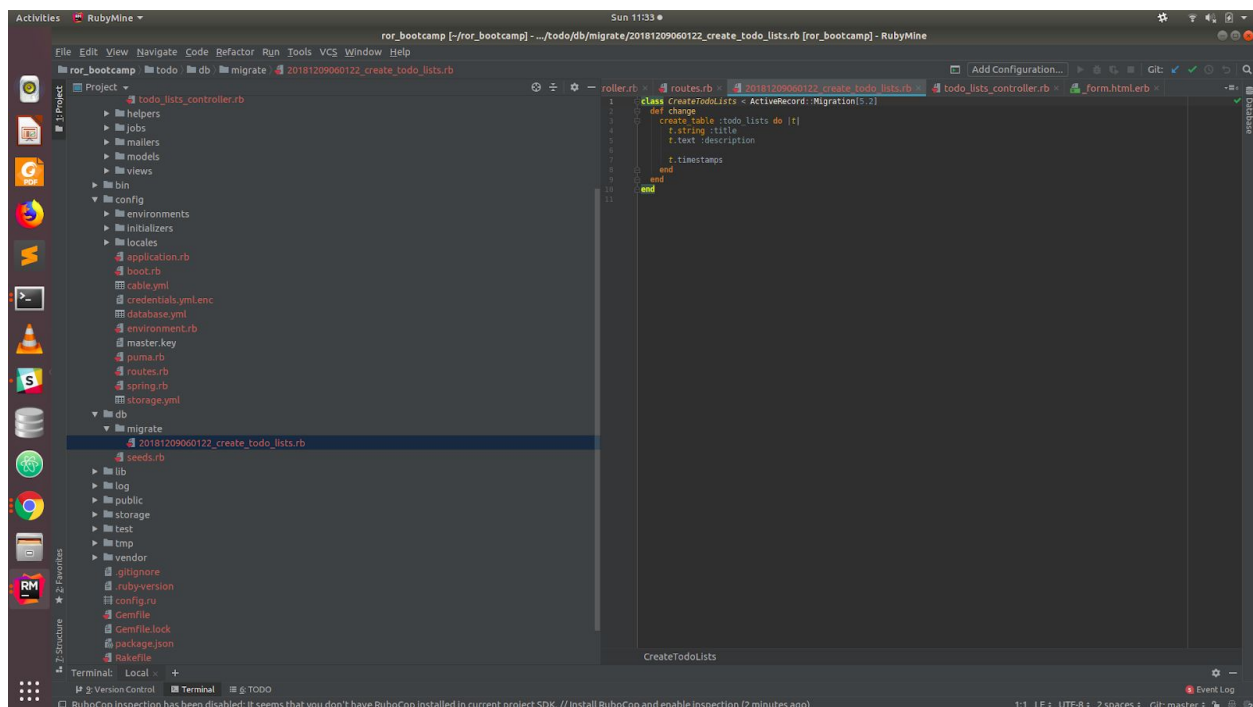
9. Go to `todo/config/routes.rb`



Here all the routes are defined.

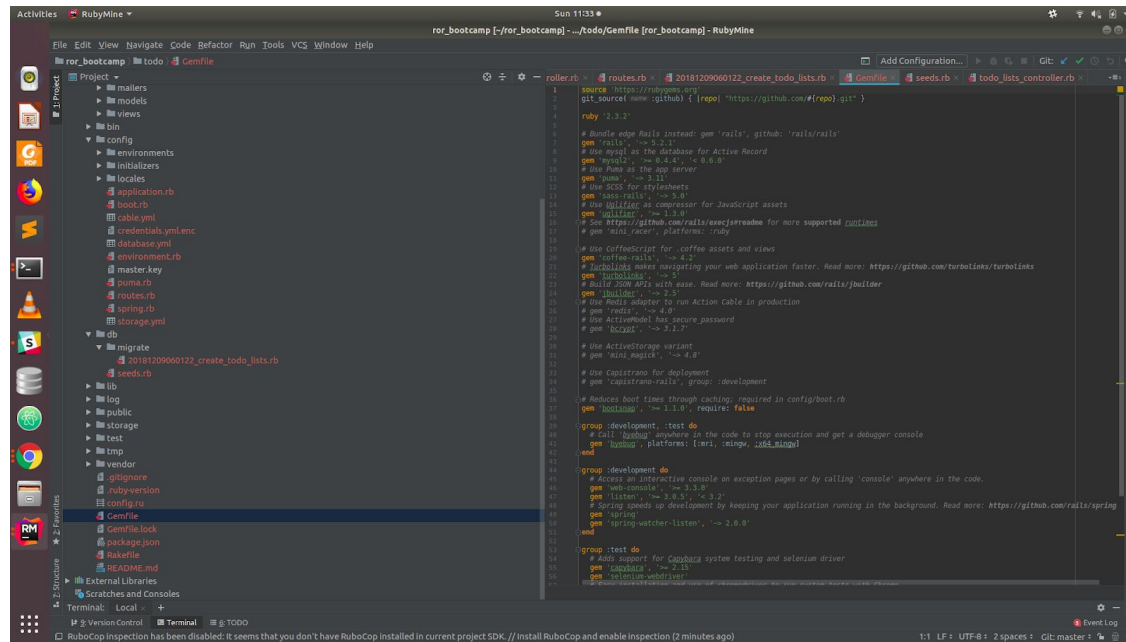
Resource routing allows you to quickly declare all of the common routes for a given resourceful controller. Instead of declaring separate routes for your index, show, new, edit, create, update and destroy actions, a resourceful route declares them in a single line of code. A resource route maps a number of related requests to actions in a single controller.

10. Go to `todo/db/migrate` to see the migrations create with timestamp. This got created when we did scaffold.



11. Go to todo/Gemfile

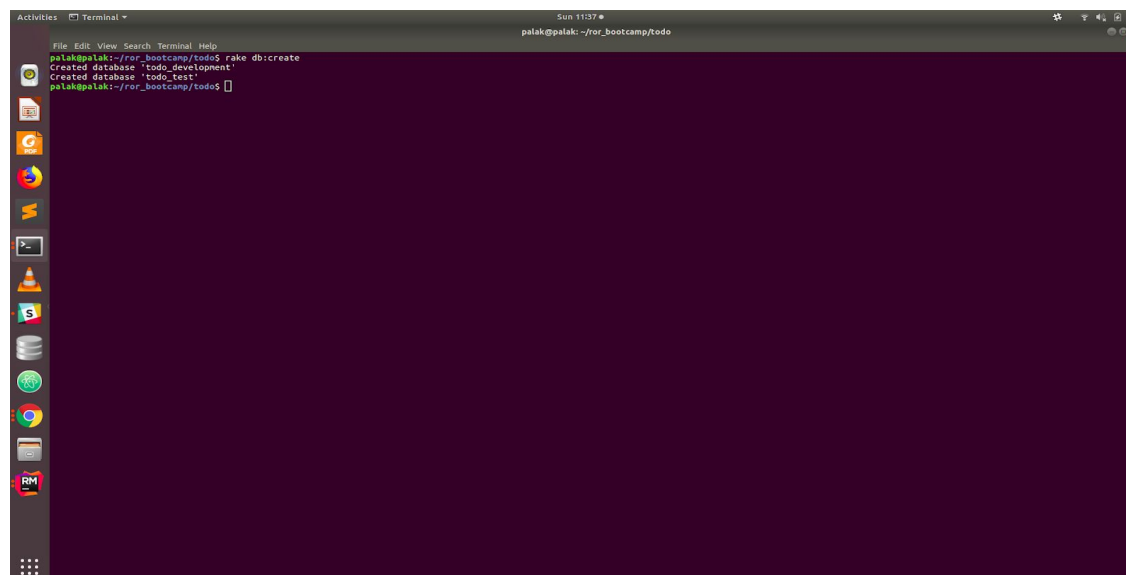
Here you can see all the gems currently installed. In case you want to add some new gems add it here and do bundle install to install them.



12. You need to create the mysql database in your local system. For that go to the terminal

```
cd todo
```

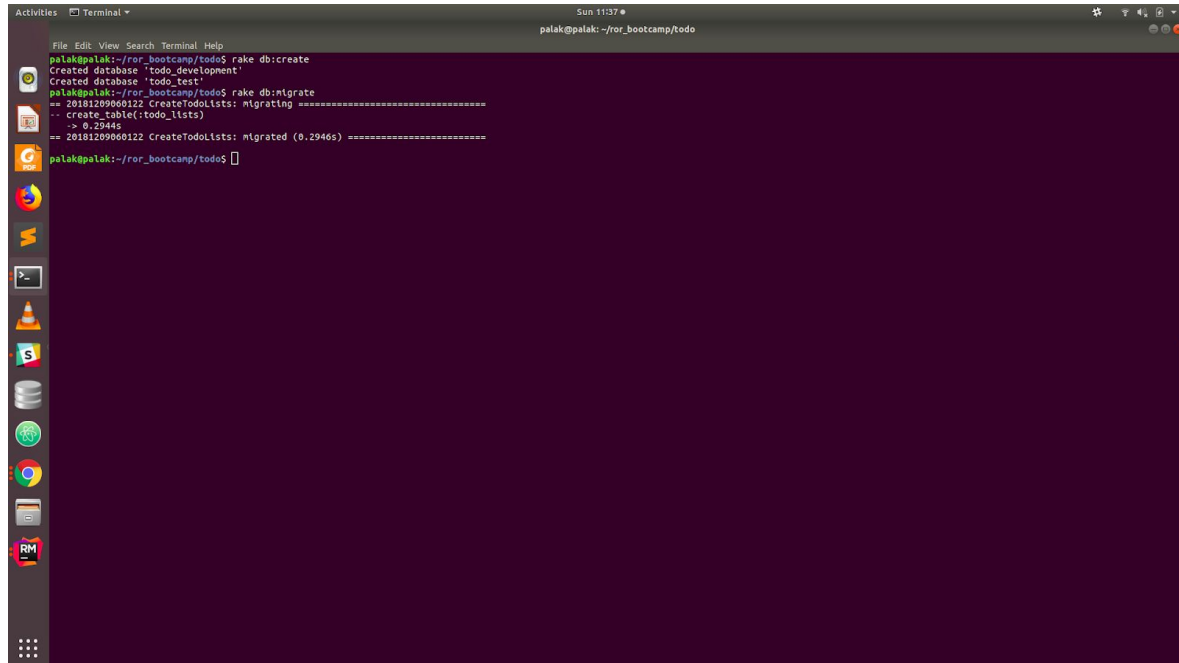
```
rake db:create
```



13. You need to make the tables actually appear in your local database. The migration created in the application won't reflect in your database until you run the following command

cd todo

rake db:migrate



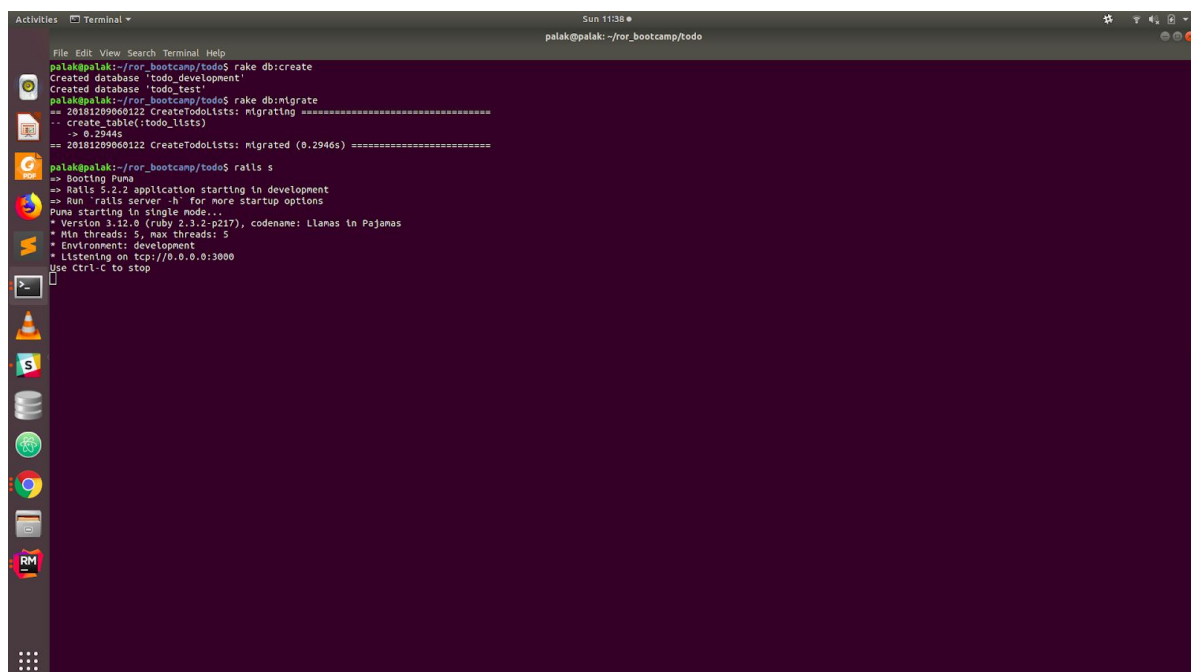
```
palak@palak:~/for_bootcamp/todo$ rake db:create
Created database 'todo_development'
palak@palak:~/for_bootcamp/todo$ rake db:migrate
== 20181209060122 CreateTodoLists: migrating =====
-- create_table(:todo_lists)
=> 0.2944s
== 20181209060122 CreateTodoLists: migrated (0.2946s) =====

palak@palak:~/for_bootcamp/todo$
```

14. To see your application running

cd todo

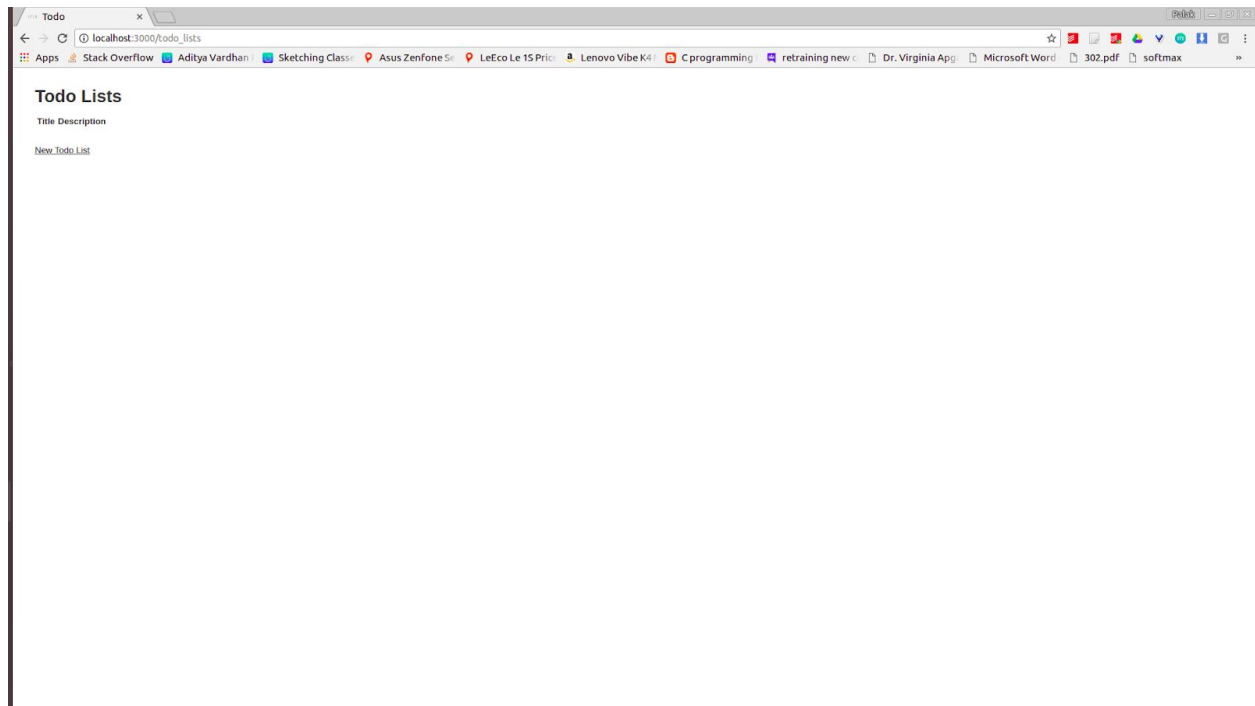
Rails server or rails s



```
palak@palak:~/for_bootcamp/todo$ rails s
=> Booting Puma
=> Rails 5.2.2 application starting in development
=> Run 'rails server -h' for more startup options
Puma starting in single mode...
* Version 3.12.0 (Ruby 2.3.2-p217), codename: Llanas in Pajamas
* Min threads: 5, max threads: 5
* Environment: development
* Listening on tcp://0.0.0.0:3000
Use Ctrl-C to stop
```

15. Open browser and go to

localhost:3000/todo_lists



14. You can also visit

http://localhost:3000/rails/info/routes

This will give you information about all the routes defined under the resources section which we saw in routes.rb file and the corresponding url and controller as well as the action of that controller associated with a particular url.