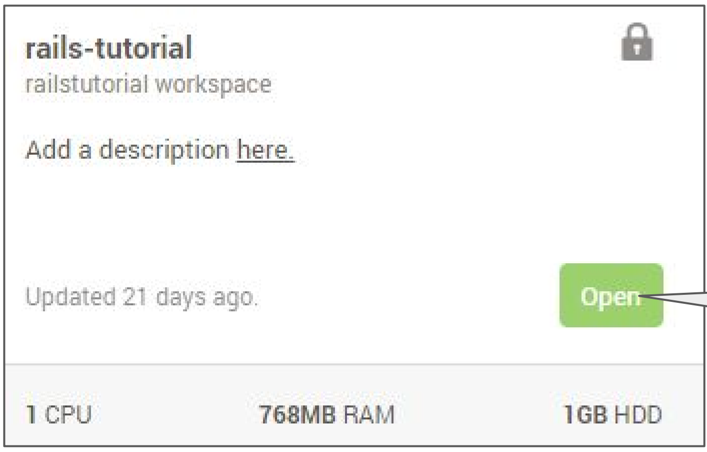
MY TOY RAILS APP

CREATE A NEW RAILS APP

Card 1 of 8
I'm Learning: **The Ruby on Rails Framework - *scaffold generators***

We're going to create a micro-blogging site - with users and posts! These instructions are based on Chapter 2 of Michael Hartl's excellent book: RUBY ON RAILS TUTORIAL https://www.railstutorial.org/book/

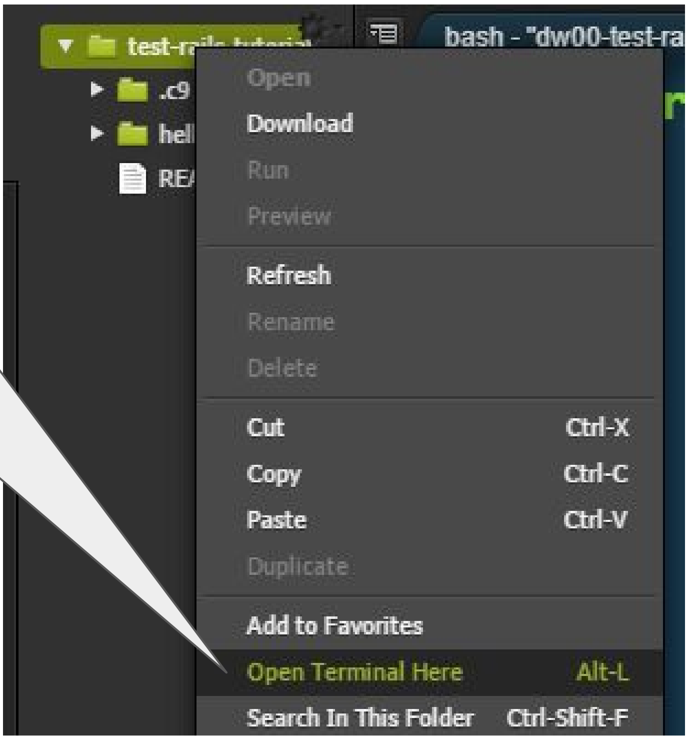
1 Logon to your account at c9.io. Open the workspace you created before:



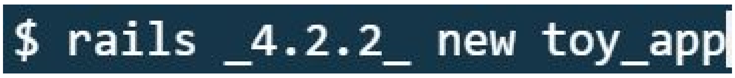
Click Open and wait until your workspace opens.

2 Open a terminal at the top-level of your workspace by right-clicking and choosing '***Open Terminal Here***'

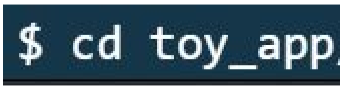
Ask your mentor about using only keyboard commands to do the same.



3



```
$ rails _4.2.2_ new toy_app
```



```
$ cd toy_app
```

Create a new rails project called '***toy_app***' and then go into the ***toy_app*** folder. All commands will be issued in this folder.

MY TOY RAILS APP

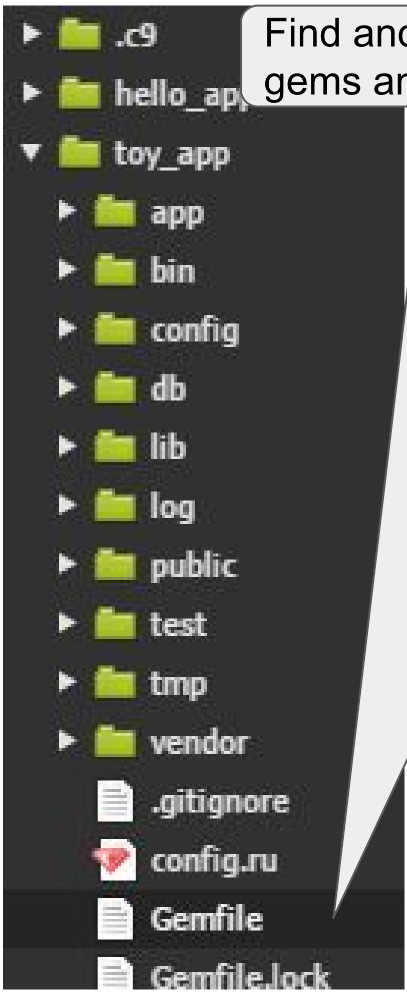
EDIT GEMFILE AND INSTALL GEMS

Card 2 of 8

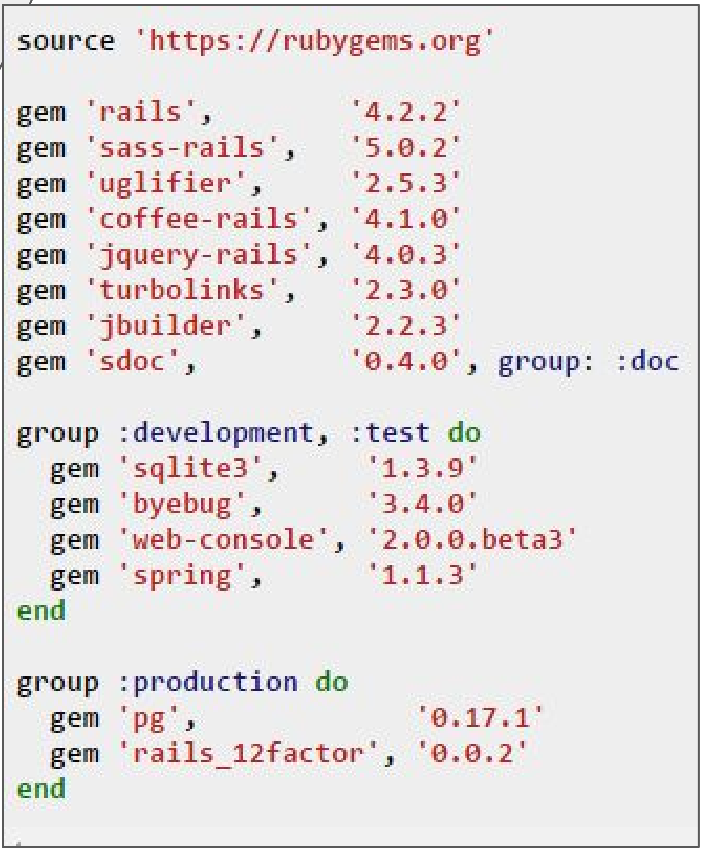
I'm Learning: **The Ruby on Rails Framework- *scaffold generators***

You need to open and edit the Gemfile for the ***toy_app***. You can delete everything in it and type the required gems, versions and environments. We only need the listed gems. Accurate typing is a must!

1



Find and open the Gemfile and make the changes to the gems and their versions. The file should save automatically.

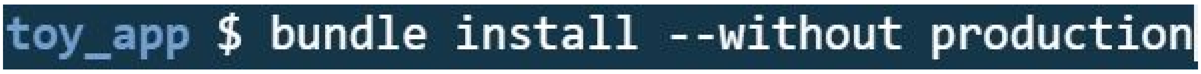


```
source 'https://rubygems.org'

gem 'rails',          '4.2.2'
gem 'sass-rails',     '5.0.2'
gem 'uglifier',       '2.5.3'
gem 'coffee-rails',   '4.1.0'
gem 'jquery-rails',   '4.0.3'
gem 'turbolinks',     '2.3.0'
gem 'jbuilder',       '2.2.3'
gem 'sdoc',           '0.4.0', group: :doc

group :development, :test do
  gem 'sqlite3',      '1.3.9'
  gem 'byebug',       '3.4.0'
  gem 'web-console',  '2.0.0.beta3'
  gem 'spring',       '1.1.3'
end

group :production do
  gem 'pg',             '0.17.1'
  gem 'rails_12factor', '0.0.2'
end
```

Make sure you have accurately typed the required gems and versions. Double-check and then check with a mentor. Ask a mentor how to know if a file is saved.

Now we can install all the gems listed above, using the following command:

2



```
toy_app $ bundle install --without production
```

Any errors here, check your Gemfile and reissue this command.

MY TOY RAILS APP

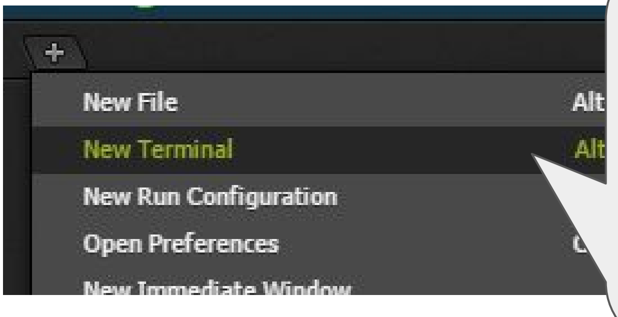
RUN A LOCAL RAILS SERVER

Card 3 of 8

I'm Learning: **The Ruby on Rails Framework- *scaffold generators***

1

If all went to plan, we can test our app by running a local rails server and viewing our app.

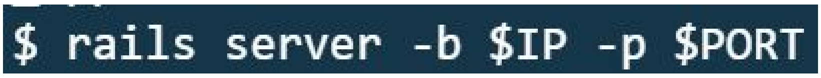


1. Click the +
2. Choose ***New Terminal***
3. Navigate to ***toy_app*** using ***cd toy_app***

We always have one terminal window for running the server and another for issuing all other commands.

2

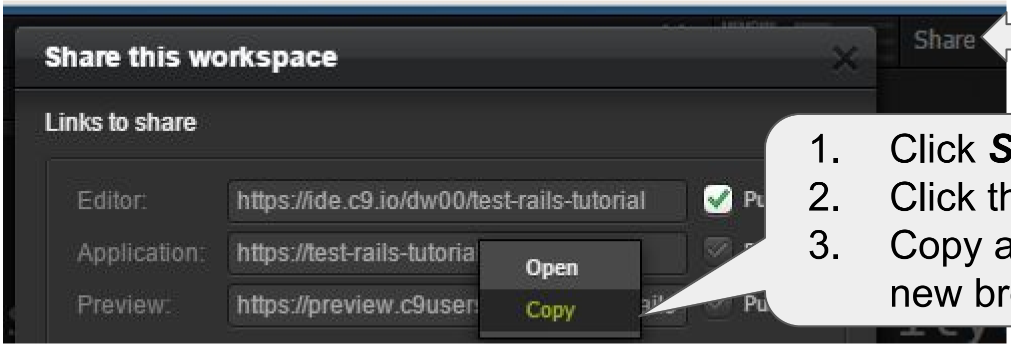
Now run a rails server in the new terminal window, using the command:



```
$ rails server -b $IP -p $PORT
```

3

Let's make sure everything is working as it should so far:



Links to share

Editor: https://ide.c9.io/dw00/test-rails-tutorial ☑ Pu

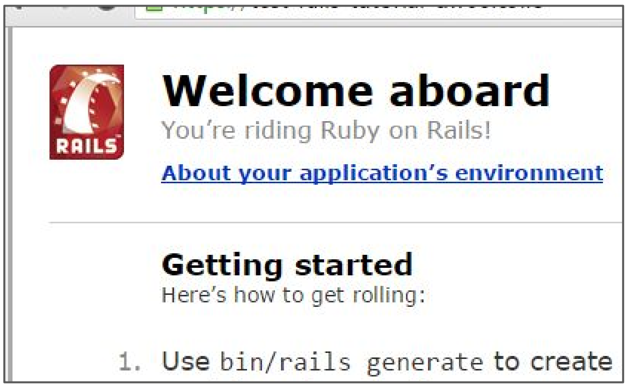
Application: https://test-rails-tutoria

Preview: https://preview.c9user Pu

1. Click ***Share***
2. Click the ***Application*** link
3. Copy and paste into a new browser tab

4

You should see the Ruby On Rails default page. If not, time to go back and check every step again!



Welcome aboard

You're riding Ruby on Rails!

About your application's environment

Getting started

Here's how to get rolling:

1. Use bin/rails generate to create

If you get stuck or are unsure about any of these, ask a mentor. A mentor can also show you shortcuts to correcting or reusing commands you have already typed.

MY TOY RAILS APP

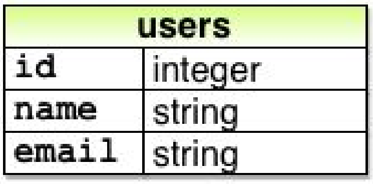
USERS

Card 4 of 8

I'm Learning: **The Ruby on Rails Framework- *scaffold generators***

1

Our micro-blog site will store information about the users and the posts they write. The users will have the following headings and data-types:



users	
id	integer
name	string
email	string

This shows how data will be stored in a database. Ask a mentor what this is and why it is needed.

2

Let's generate the files and code necessary to create, read, update and delete users:

```
$ rails generate scaffold User name:string email:string
```

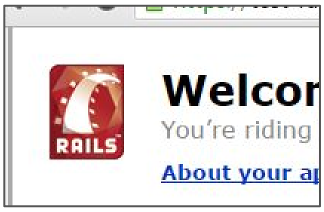
And then update the database:

```
$ bundle exec rake db:migrate
```

Run these commands in our bash terminal window - not the one for the server.

3

Let's investigate what the previous commands did to our app.

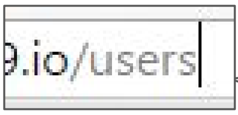


Welcor

You're riding

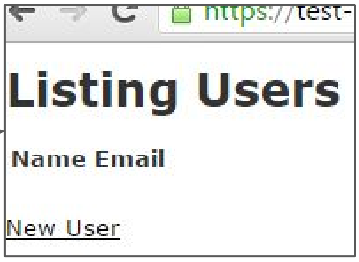
About your ap

If you refresh the browser tab for your app, it appears as before.



9.io/users

However, if you add ***/users*** to the end of the url, you get a page listing all users!



https://test-

Listing Users

Name Email

New User

4

Our micro-blogging app is beginning to take shape and with only a few commands, we can have a web-app with users!

Can you see the link between the users table in step 1, the rails command in step 2 and the information on our /users webpage?

MY TOY RAILS APP

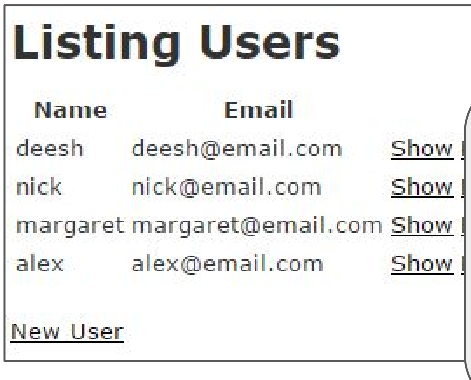
USERS AND POSTS

Card 5 of 8

I'm Learning: **The Ruby on Rails Framework- *scaffold generators***

1

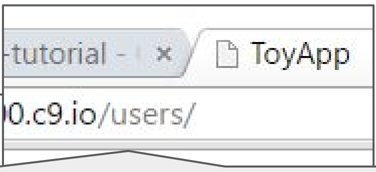
Let's make some users eg:



Listing Users

Name	Email	
deesh	deesh@email.com	Show
nick	nick@email.com	Show
margaret	margaret@email.com	Show
alex	alex@email.com	Show

New User

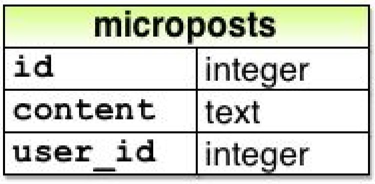


You can use the following urls to navigate to different pages related to users:

/users	- page shows all users
/users/1	- page shows user with id 1
/users/new	- page to make a new user
users/1/edit	- page to edit/update user1

2

Our microposts will each have the following headings:

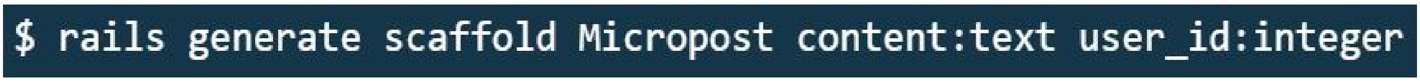


microposts	
id	integer
content	text
user_id	integer

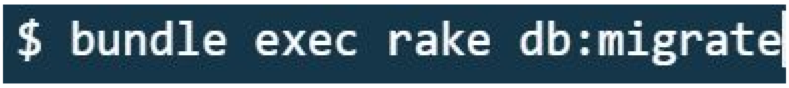
We use this data model to in our code to create all the necessary files and pages for posts.

3

Use the following commands to generate the code necessary for our microposts:

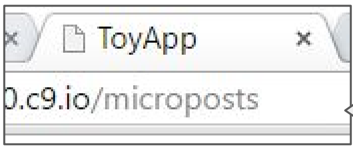


```
$ rails generate scaffold Micropost content:text user_id:integer
```

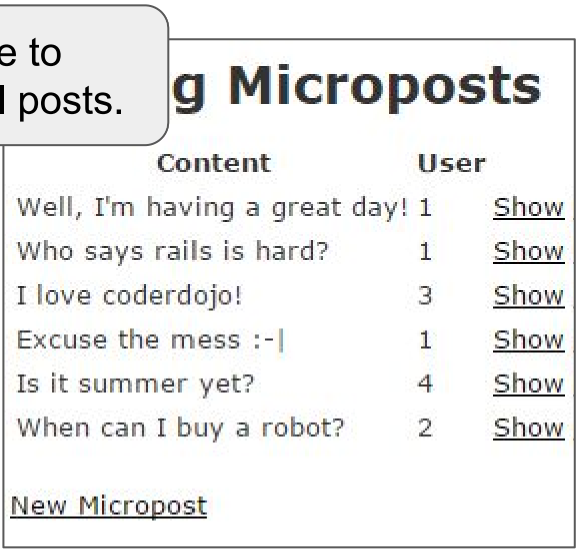


```
$ bundle exec rake db:migrate
```

4



You can now navigate to ***/microposts*** to list all posts.



g Microposts

Content	User	
Well, I'm having a great day!	1	Show
Who says rails is hard?	1	Show
I love coderdojo!	3	Show
Excuse the mess :-|	1	Show
Is it summer yet?	4	Show
When can I buy a robot?	2	Show

New Micropost

Try and work out what the number for ***User*** means. Check with a mentor whether this is correct.

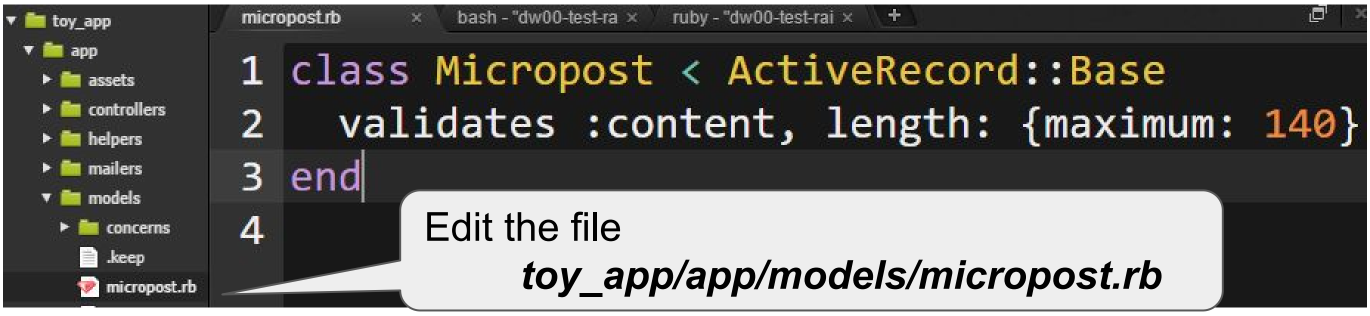
MY TOY RAILS APP

VALIDATION & RELATIONSHIPS

Card **6** of **8**
I'm Learning: **The Ruby on Rails Framework- *scaffold generators***

We are going to make some changes to the files that have been generated for our micro-blogging site. The first is to make sure that each post cannot be more that 140 characters (like Twitter). The second change will let us show all posts by a user, on one page.

1



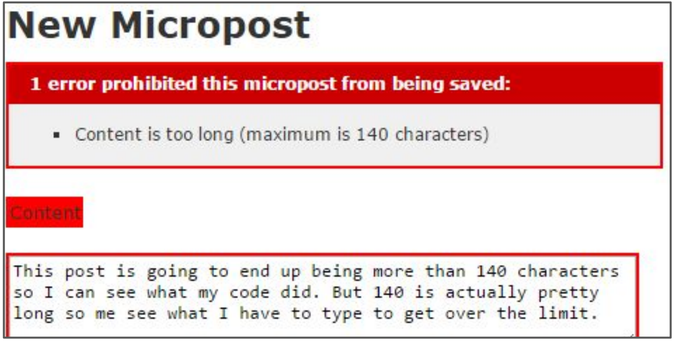
```ruby
class Micropost < ActiveRecord::Base
  validates :content, length: {maximum: 140}
end
```

Edit the file
toy_app/app/models/micropost.rb

2

When we try to make a post which is more than 140 characters long:

An error message appears and we cannot save our post unless we reduce the number of letters, including spaces!



New Micropost

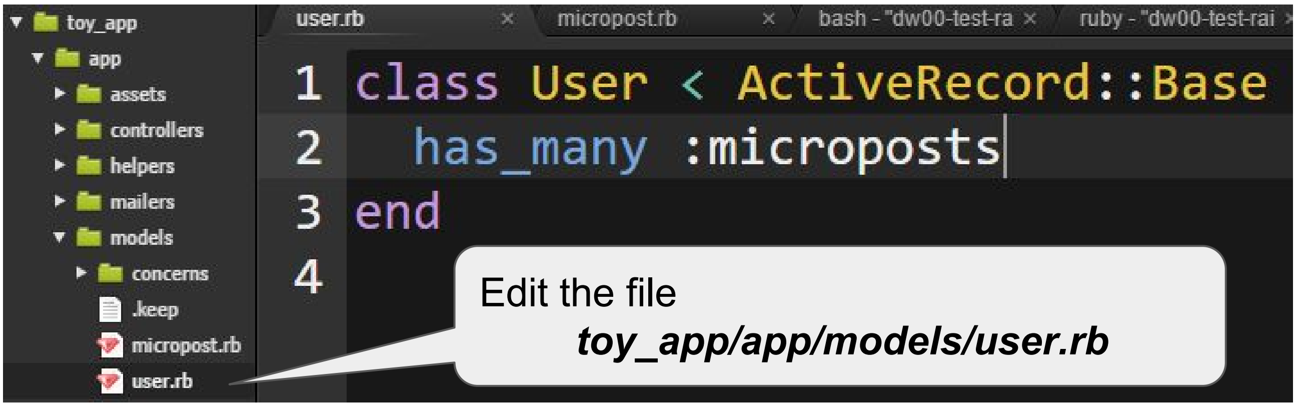
1 error prohibited this micropost from being saved:

- Content is too long (maximum is 140 characters)

Content

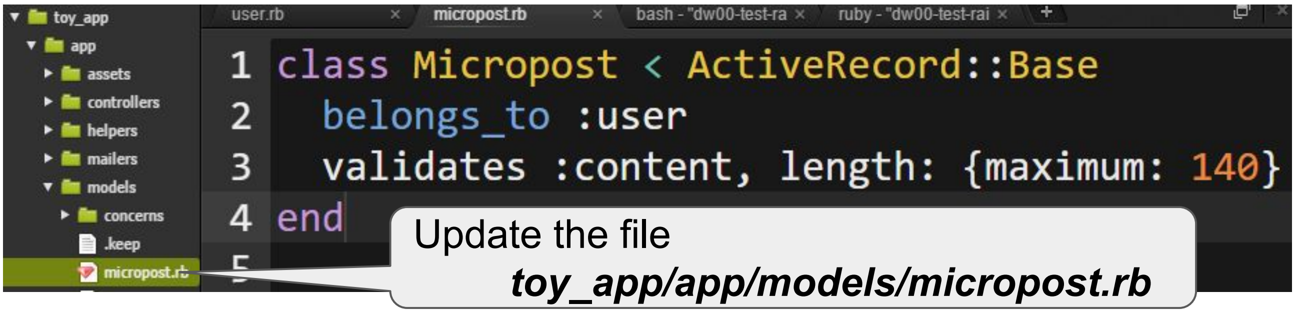
This post is going to end up being more than 140 characters so I can see what my code did. But 140 is actually pretty long so me see what I have to type to get over the limit.

3



```ruby
class User < ActiveRecord::Base
  has_many :microposts
end
```

Edit the file
toy_app/app/models/user.rb



```ruby
class Micropost < ActiveRecord::Base
  belongs_to :user
  validates :content, length: {maximum: 140}
end
```

Update the file
toy_app/app/models/micropost.rb

MY TOY RAILS APP

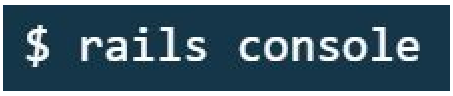
RAILS CONSOLE and DEPLOYING

Card 7 of 8

I'm Learning: **The Ruby on Rails Framework- *scaffold generators***

The changes we made to the ***toy_app/app/models/micropost.rb*** and ***toy_app/ap/models/user.rb*** files cannot be see in our running app at the moment. This change is behind the scenes and for us to see it's effect, we can use the rails console command.

1

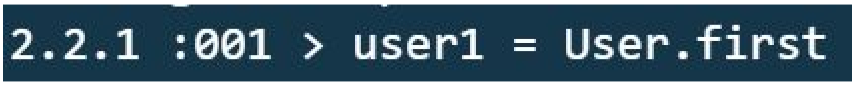


```
$ rails console
```

Start the console by typing this command

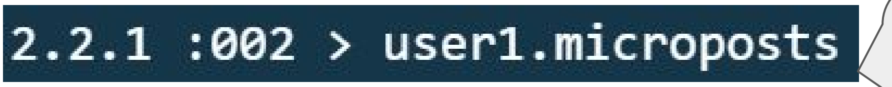
2

Let's get hold of our first user:



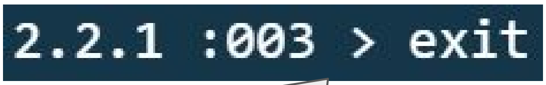
```
2.2.1 :001 > user1 = User.first
```

Here call our first user ***user1***



```
2.2.1 :002 > user1.microposts
```

Now we can get all the posts for ***user1*** - make sure ***user1*** has some posts otherwise they won't show.

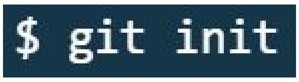


```
2.2.1 :003 > exit
```

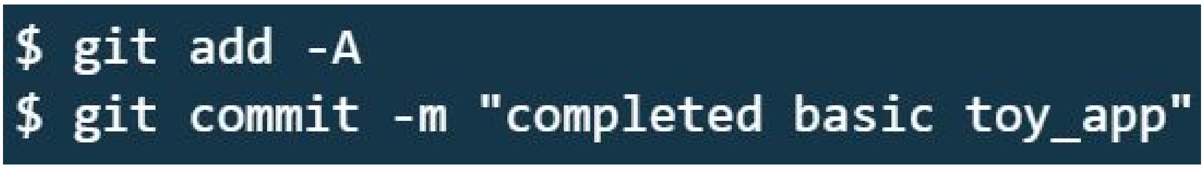
We can end the console session

3

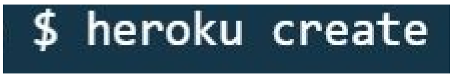
We are now ready to deploy to heroku using the following commands:



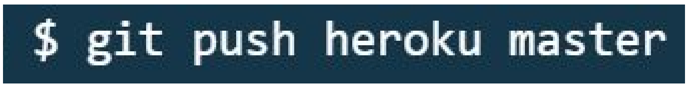
```
$ git init
```



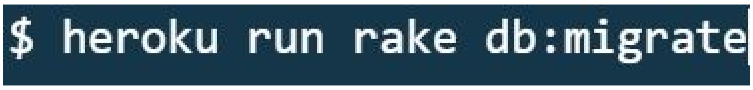
```
$ git add -A
$ git commit -m "completed basic toy_app"
```



```
 $ heroku create
```



```
 $ git push heroku master
```



```
$ heroku run rake db:migrate
```

Ask a mentor what these commands mean and whether they have to be typed in this order.

MY TOY RAILS APP

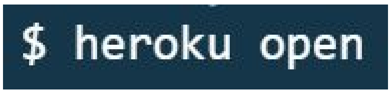
FINDING YOUR ONLINE APP

Card 8 of 8

I'm Learning: **The Ruby on Rails Framework- *scaffold generators***

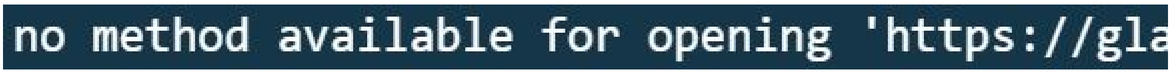
Unlike the first app you created, this one is slightly more difficult to find because we haven't created any of the usual pages, only those for users and their posts. Here are the steps to locate the app and also some other things you could try.

1



```
$ heroku open
```

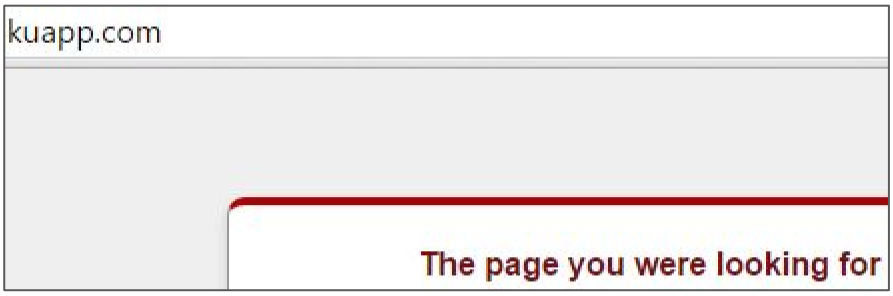
Use this to find the heroku url for our app



```
no method available for opening 'https://gla
```

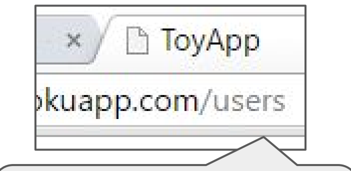
The url will be listed here - copy and paste into a browser tab - it may show an error.

2



kuapp.com

The page you were looking for



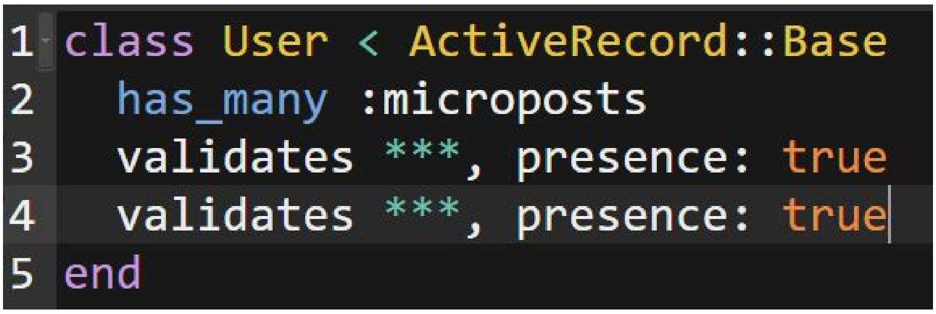
kuapp.com/users

add ***/users*** to the end of the url

You will have an empty database of users and posts and will have to add them here again if you want to see it working live.

3

As a challenge, look at the ***app/models/micropost.rb*** file and work out how to make sure a user cannot leave their name and email blank. What should go where we have ***



```
class User < ActiveRecord::Base
  has_many :microposts
  validates ***, presence: true
  validates ***, presence: true
end
```

Ask a mentor, how to stop your local rails server and leave your IDE in a tidy manner, ready for the next project!

How will you test that it works?
How will you deploy the changes to heroku?