

Setting Up The Contents Of A Sinatra App

If you followed all the steps above, you should now have all the files you need to start building your sinatra app. The above will be needed every time, so it is best to get the space ready before you start adding the code.

STEP 1: WHERE TO BEGIN

It is best to start with setting up your database.

- In your terminal, cd into home directory (it is always good to start here) and create your database. It is a good idea to name your database the same as your home directory, will be easier to remember.

→ `hogwarts createdb <DATABASE NAME>`

e.g.

→ `hogwarts createdb hogwarts`

- In your sublime file, in your db directory click on the sql file you created. In our example it was called `hogwarts.sql`. In here, create your tables. Refer to previous examples for help (E.g. music library). You will have planned what to include in here earlier. The general rule is there should be a table for each model, THIS IS NOT ALWAYS THE CASE THOUGH. This is just a general rule.
- In your terminal, run your database to create the tables:

→ `hogwarts psql -d <DATABASE NAME> -f db/<NAME OF FILE>.sql`

CREATE TABLE

e.g.

→ `hogwarts psql -d hogwarts -f db/hogwarts.sql`

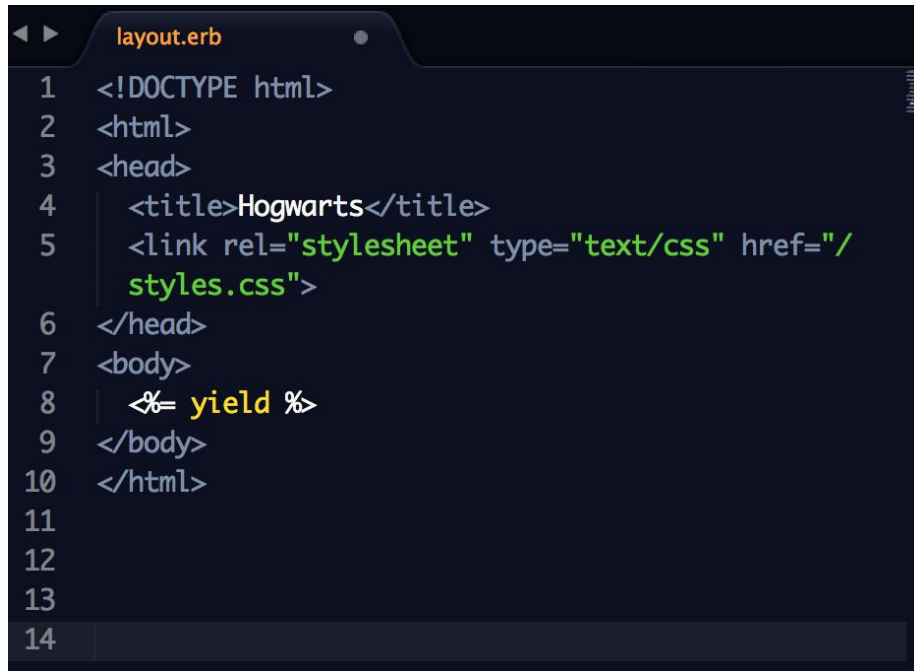
CREATE TABLE

STEP 2: SETUP YOUR SQL RUNNER

- Setup your sql runner, this should be the same every time **APART FROM THE DATABASE NAME**. Refer to your previous examples for help.

STEP 3: SETUP YOUR LAYOUT FILE

- In your views directory, go into your layout.erb. Type html and press the tab button. Your file should now be populated with the basic html page code.
- Add the link to your stylesheet, the name of your app and the yield erb. Example:



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Hogwarts</title>
5   <link rel="stylesheet" type="text/css" href="/
  styles.css">
6 </head>
7 <body>
8   <%= yield %>
9 </body>
10 </html>
11
12
13
14
```

STEP 4: START IN A CONTROLLER AND VIEWS

- Work through one at a time. Start with the table that has no external ids in it (foreign keys). This will be the simplest and you can build up your app from there. In our case it would be our students table so we will start in our students controller.
- Refer to previous examples that you have made in class to show what you should include in your controller. Remember to require everything (sinatra, sinatra/contrib/all, pry-byebug and any models you need).

HINT: with your requires, make sure it is requiring the file from the right level - this will be which file your models, etc, are in.

- Time to setup your restful routes. Refer to previous examples for help on the code. Start with your index route.
- Next move into your views folder, into your index.erb. Time to test that your index route works. Type something simple, such as “hello world”, into the index.erb file.

- **In your terminal, run your controller to start the server:**

→ `hogwarts` ruby controllers/student_controller.rb

- **In chrome, type in “localhost:4567/<CONTROLLER NAME>”, e.g. “localhost:4567/students”. If you see “hello world” on the page, then your route is working.**
- **Repeat this process with each of the routes in the controller and corresponding views. You should have seven restful routes by the time you are finished. Again, refer to previous examples for help.**

STEP 5: SPECS AND MODELS

Once you know all the routes are working by putting simple text in and testing the pages, you can move onto writing the functionality. What functionality you include will depend on what you decided on in your planning stage at the beginning. However, the basics usually still apply. If you have a table, then you should be able to save, edit, view and delete information from that table.

- **Start in your spec file and set up minitest and require the files you need. Define a setup method if you need it and then write your first test.**
- **Setup you class in the corresponding model and start to write the functionality that will pass the tests. Refer to your previous examples and your planning for help.**

STEP 6: PLUG YOUR MODELS INTO YOUR VIEWS

Once your tests are passing on your functionality in your models, you can then move to the corresponding views and start populating the views with the needed forms, information and erb tags.

- **Start with your index.rb. Check the code is appearing in the browser correctly.**
- **Work through the rest of your views in this order: new, show, edit, create, update, destroy.**

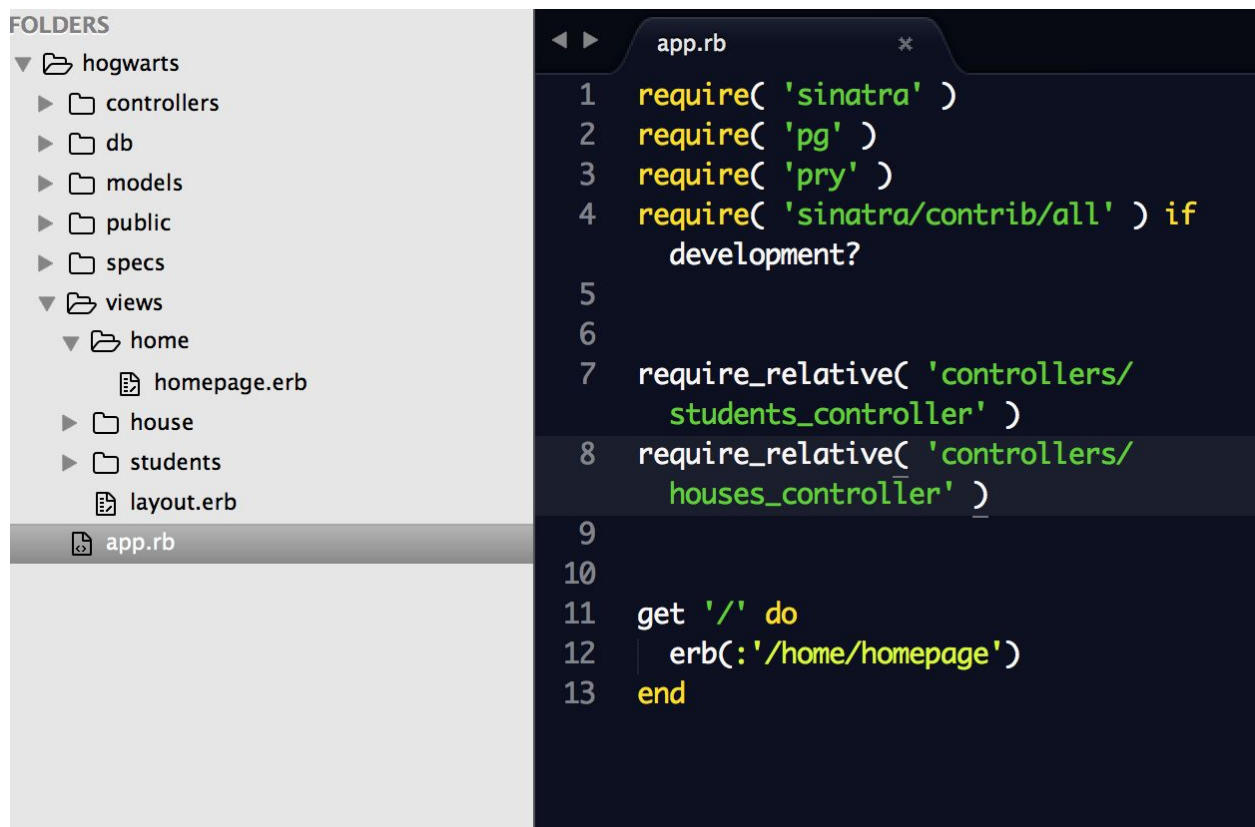
STEP 7: REPEAT STEPS 4-6 FOR EACH MODEL AND CONTROLLER THAT YOU HAVE UNTIL THEY ARE ALL COMPLETE.

You will need to make sure you stop and start your server again for each controller that you have.

STEP 8: MAKE AN APP FILE THAT RUNS ALL YOUR CONTROLLERS TOGETHER.

At the moment you have to run each controller individually to make them run. Now it is time to make a controller that runs all of your controllers. Yes, controller-ception. This is also a good point to create a homepage to welcome everyone to your website.

- In your terminal, cd into your views directory and make a home directory.
→ `views mkdir home`
- In your terminal, touch a homepage file.
→ `views touch homepage.erb`
- Cd into your main directory and touch an app.rb file.
→ `hogwarts touch app.rb`
- Include all of your controllers in the app.rb. Look at the example below to see what code you need to include:



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer, titled 'FOLDERS', shows a tree structure: 'hogwarts' (expanded) contains 'controllers', 'db', 'models', 'public', 'specs', 'views' (expanded), and 'app.rb'. Under 'views', there is a 'home' folder containing 'homepage.erb', and 'house' and 'students' folders. The code editor shows the content of 'app.rb' with line numbers 1 through 13. The code includes 'sinatra', 'pg', and 'pry', and uses 'require_relative' to include 'students_controller' and 'houses_controller'. It also has a 'get' request for '/' that renders the 'home/homepage' view.

```
1 require( 'sinatra' )
2 require( 'pg' )
3 require( 'pry' )
4 require( 'sinatra/contrib/all' ) if
  development?
5
6
7 require_relative( 'controllers/
  students_controller' )
8 require_relative( 'controllers/
  houses_controller' )
9
10
11 get '/' do
12   erb( '/home/homepage' )
13 end
```

- The get request at the bottom of the file redirects localhost:4567 to your homepage file.
- To run your app now, run the app.rb from your main directory in the terminal.
→ `hogwarts ruby app.rb`

STEP 9: DO CSS