Intro to Web Design and Development, Class 11

More ActiveRecord

Schedule

Part 1 - Homework Questions, ActiveRecord Review

Homework questions?

1. To use ActiveRecord and by extension a database in your project, add these lines to your Gemfile:

gem 'activerecord'
gem 'sinatra-activerecord'
gem 'sqlite3'
gem 'rake'

Then run **bundle** from the terminal.

2. Database setup

To setup/connect your database, add the following to your main **app.rb** file after your **require 'sinatra'** line:
require 'sinatra/activerecord'
set :database, "sqlite3:///<databasename>.**sqlite3"**

2. Rakefile

The Rakefile is a new file in your project, without an extension (similar to the Gemfile in this way)
-It contains tasks for your app to complete, run using the **rake** terminal command, such as:
    -rake db:migrate
    -rake db:rollback
and any custom commands you define in your Rakefile
-To create your Rakefile, make a new document in Sublime Text 2 and save it in your project directory as **Rakefile**. Add the following lines to it:

require 'sinatra/activerecord/rake'
require './your_app'

3. Migrations

Migration files are used to create new tables in your database on a migration (up) and to drop them on rollback (down).

To generate a new migration, run:
rake db:create_migration NAME=migration_name

To edit your migration, navigate to the /db/migrate folder. Migration files are automatically named after being created in the format:
<date><randomhash>_<migrationname>.rb

Inside of your migration, you'll want code similar to the following:

```
class CreatePosts < ActiveRecord::Migration

  def up
        create_table :posts do |t|
              t.string :title
              t.datetime :created_at
              t.text :text
              t.integer :user_id
        end
  end

  def down
        drop_table :posts
  end
end
```

4. Running migrations

To run your migration, execute **rake db:migrate** in the terminal. To rollback by one migration, execute **rake db:rollback.** Use the **STEP=<**steps to go back> option to back a certain number of steps instead of one with your rollback, i.e. (**rake db:rollback STEP=5).** To rollback and then migrate your last migration in one step, use **rake db:migrate:redo**. To run a specific migration, use **rake db:migrate:up VERSION=** <date><randomhash> (i.e. **rake db:migrate:up VERSION=20130227014916**). You can also migrate down by specifying :down instead of :up in this command.

If ActiveRecord believes the migration has already been run, it will not be run again.

5. Models

Once your migrations are run, you should have some database tables setup in your database. In order to access/work with them in your application, you'll need to create some models. It would be wise to put these in a separate file, then include them in your main app. Create a file called **models.rb** in your project directory, then require it in your main app underneath your database requires using **require './models'**.

An extremely basic models file only needs the following to access the users model:

class User < ActiveRecord::Base
end

Once your basic model is implemented, test it all out by firing up the Ruby console using **irb** in the terminal. Then type **User**, or **User.first**.

6. Basic model interactions

Lookup up a user by ID:

User.find(5)

Lookup a user by a parameter in the users table:

User.where("bananas = ?", "18")

Get all the users

User.all

Create a new user

A
User.create

B
user = User.new
user.save

7. Declaring associations

You'll want your models to associate with each other for easy access. To create a one-to-many relationship between a user and their posts, do the following:

-Create a new migration to add the appropriate reference to the **posts** table by running **rake**

db:create_migration NAME=modify_post_relationships

```
def up
  change_table :posts do |t|
        t.references :users
  end
end

def down
   remove_column :posts, :user_id
end
```

-Your models.rb file should have the following lines added:

```
class User < ActiveRecord::Base

  has_many :posts
end

class Post < ActiveRecord::Base

  belongs_to :user
end
```

Homework

Goals

1. Learn ActiveRecord
2. Integrate ActiveRecord into your Sinatra project

Assignment

1. Pick two of your favorite websites and draw out diagrams illustrating what their database tables and relationships look like.

Recommended Activity

sinatra-activerecord gem instructions
https://github.com/janko-m/sinatra-activerecord