software studio

our first Rails app: a tour

Daniel Jackson

For now, all rights reserved. Daniel Jackson, 2013.
let's make a Rails app!

create a new Rails app called “noteapp”

```shell
rails new noteapp
```

generate controllers, models & views for a resource called “Note” with a single string field called “content”

```shell
rails generate scaffold Note content:string
```

we now have code for a model class but no table in the relational database to back it; execute “migration” to update database schema

```shell
rake db:migrate
```
let's start the server

start the built-in WebBrick server

```
rails server
```

console reports

Rails 3.2.12 application starting in development on http://0.0.0.0:3000

in a browser, access the app at this URL
Welcome aboard
You're riding Ruby on Rails!

About your application's environment

Getting started
Here's how to get rolling:

1. Use rails generate to create your models and controllers
   To see all available options, run it without parameters.

2. Set up a default route and remove public/index.html
   Routes are set up in config/routes.rb.

3. Create your database
   Run rake db:create to create your database. If you're not using SQLite (the default), edit config/database.yml with your username and password.

Browse the documentation
Rails Guides
Rails API
Ruby core
Ruby standard library

home page
Listing notes

Content

New Note

http://0.0.0.0:3000/notes
creating a new resource

New note

Content

Create Note

Back

New note

Content

This is my first note

Create Note

Back

New note

Content: This is my first note

Create Note

Back

New note

Listing notes

Content

This is my first note Show Edit Destroy

New Note

watch the URL
let’s look at the code

parts we’ll look at
› routing: turning URL into method call
› model: class for datatypes
› schema: database structure for persistence
› controller: actions for responding to requests
› template: HTML file with embedded Ruby
Noteapp::Application.routes.draw do
  resources :notes
end

config/routes.rb

`resources` method generates whole set of URL/method bindings
model class

```
class Note < ActiveRecord::Base
  attr_accessible :content
end
```

app/models/note.rb

like Ruby accessor, but controls mass assignment
ActiveRecord::Schema.define(:version => 20130924155549) do
  create_table "notes", :force => true do |t|
    t.string "content"
    t.datetime "created_at", :null => false
    t.datetime "updated_at", :null => false
  end
end

`db/schema.rb` file is autogenerated from "migration"
index page template

<h1>Listing notes</h1>

<table>
  <tr>
    <th>Content</th>
    <th></th>
    <th></th>
    <th></th>
  </tr>

  <% @notes.each do |note| %>
  <tr>
    <td><%= note.content %></td>
    <td><%= link_to 'Show', note %></td>
    <td><%= link_to 'Edit', edit_note_path(note) %></td>
    <td><%= link_to 'Destroy', note, method: :delete, data: { confirm: 'Are you sure?' } %></td>
  </tr>
  <% end %>

</table>

<%= link_to 'New Note', new_note_path %>
controller actions for index & show

class NotesController < ApplicationController
  # GET /notes
  # GET /notes.json
  def index
    @notes = Note.all

    respond_to do |format|
      format.html # index.html.erb
      format.json { render json: @notes }
    end
  end

  # GET /notes/1
  # GET /notes/1.json
  def show
    @note = Note.find(params[:id])

    respond_to do |format|
      format.html # show.html.erb
      format.json { render json: @note }
    end
  end
...

app/controllers/notes_controller.rb
controller actions for new/create

```ruby
class NotesController < ApplicationController
  # GET /notes/new
  # GET /notes/new.json
  def new
    @note = Note.new

    respond_to do |format|
      format.html # new.html.erb
      format.json { render json: @note }
    end
  end

  # POST /notes
  # POST /notes.json
  def create
    @note = Note.new(params[:note])

    respond_to do |format|
      if @note.save
        format.html { redirect_to @note, notice: 'Note was successfully created.' }
        format.json { render json: @note, status: :created, location: @note }
      else
        format.html { render action: "new" }
        format.json { render json: @note.errors, status: :unprocessable_entity }
      end
    end
  end
...```
form partial & template

```erb
<% form_for(@note) do |f| %>
  <% if @note.errors.any? %>
    <div id="error_explanation">
      <h2><%= pluralize(@note.errors.count, "error") %> prohibited this note from being saved:</h2>
      <ul>
        <% @note.errors.full_messages.each do |msg| %>
          <li><%= msg %></li>
        <% end %>
      </ul>
    </div>
  <% end %>

  <div class="field">
    <%= f.label :content %><br />
    <%= f.text_field :content %>
  </div>

  <div class="actions">
    <%= f.submit %>
  </div>
<% end %>

<h1>New note</h1>

<% render 'form' %>

<%= link_to 'Back', notes_path %>
```

app/views/notes/_form.html.erb

app/views/notes/new.html.erb