Ruby on Rails 3

Week 6

Scaffolding

materials from the book, Agile Web Development with Rails
by Sam Ruby
How-To Setup ActiveScaffold with Rails 3.0 by Volker Hochstein
http://vhochstein.wordpress.com/2010/08/28/setup-activescaffold-rails-3/

What is Scaffolding?

- Scaffolding is a way to quickly put an Active Record class online by providing a series of standardized actions for listing, showing, creating, updating, and destroying objects of the class
- Useful for quick prototyping
- These standardized actions come with both controller logic and default templates that through introspection already know which fields to display and which input types to use

A Little Controversial

- DHH's blog in fifteen minutes was built on scaffolding
- Over time, however, it became less clear whether being used for production code, or intended to educate new Rails developers on best practices
- Eventually, scaffolding was supposed to be educational, illustrating the best practices around RESTful controllers and other Rails conventions

Scaffolding via Generator

- Create an application

```
$ rails new myStore -d mysql
```

- Create a scaffold for the "Product" model

```
$ cd myStore
$ rails generate scaffold Product
      title:string description:text
      image_url:string price:decimal
```

- The generator creates a bunch of files; the important one is the migration, namely db/migrate/20101206102532_create_products.rb

Refine the "Product" model

- Though, we have already told Rails about the basic data types for the "Product" model, we can refine the table

```
class CreateProducts < ActiveRecord::Migration
   def self.up
      create_table :products do |t|
          t.string :title
          t.text :description
          t.string :image_url
          t.decimal :price, :precision => 8, :scale => 2
          t.timestamps
      end
   end
   :
end
```

- Price now can have eight digits of significance, and two digits after the decimal point, e.g. 123456.78

Apply the Migration

- To get Rails to apply the migration to our development database, we use

```
$ rake db:migrate
```

- The products table is added to the database, defined by the development section of the database.yml file

See List of Products

- Let’s try our new application by starting the server

```
$ rails server
```

- We use the controller name in lowercase (i.e., products) to access the list of products

Modify some Form Field

- Let's change the number of lines in the description field of the product table in file myStore/app/views/products/_form.html.erb

```
<div class="field" >
  <%= f.label :description %><br />
  <%= f.text_area :description, :rows => 6 %>
</div>
```

Seed Data for Testing

- Rails has the ability to import seed data
- Try this by download the file from

  http://media.pragprog.com/titles/rails4/code/depot_b/db/seeds.rb
- To populate the products table with test data, simply run

```
$ rake db:seed
```

Refine a View Layout

- After we import the data, we need a couple of files for a better product display
  - http://media.pragprog.com/titles/rails4/code/depot_b/public/images
  - http://media.pragprog.com/titles/rails4/code/depot_b/public/stylesheets/depot.css
- Copy all of the jpeg images into public/images folder in the application
- Copy the depot.css file into public/stylesheets folder in the application

Application View Layout

- Rails keeps the file that is used to create a standard page environment for the entire application
- This file is application.html.erb, a Rails view layout, residing in the views/layouts directory

Inside Application Layout

- Here is an example:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Depot</title>
        <%= stylesheet_link_tag :all %>
        <%= javascript_include_tag :defaults %>
        <%= csrf_meta_tag %>
    </head>
    <body>
        <%= yield %>
    </body>
</html>
```

Load Stylesheets

- stylesheet_link_tag creates HTML <link> tag, loading stylesheets from public/stylesheets
- Specific stylesheet name can also be used here
- `<%= stylesheet_link_tag :all %>` is used to load all stylesheets inside public/stylesheets
- Brief info about Unobtrusive Javascript (UJS) can be read from http://www.themodestrubyist.com/2010/02/24/rails-3-ujs-and-csrf-meta-tags/

Modify index.html.erb Page

```
<div id="product_list" >
<h1>Listing products</h1>
<table>
   <% @products.each do |product| %>
     <tr class="<%= cycle('list_line_odd', 'list_line_even') %>" >
        <td>
         <%= image_tag(product.image_url, :class => 'list_image') %>
        </td>
        <td class="list_description" >
           <dl>
                 <dt><%= product.title %></dt>
              <dd><%= truncate(strip_tags(product.description),
                               :length => 80) %></dd>
           </dl>
        </td>
        :
```

Modify index.html.erb Page (2)

```
      :
      <td class="list_actions" >
        <%= link_to 'Show', product %><br/>
        <%= link_to 'Edit', edit_product_path(product)%><br/>
        <%= link_to 'Destroy', product,
            :confirm => 'Are you sure?', :method => :delete %>
      </td>
    </tr>
  <% end %>
</table>
</div>
<br />
<%= link_to 'New product', new_product_path %>
```

Explanations

- The rows in the listing have alternating background colors; the Rails helper method called *cycle* does this by setting the CSS class of each row to either *list_line_even* or *list_line_odd*
- The *truncate* helper displays just the first eighty characters of the description
- Before *truncate* was called, we called *strip_tags* to remove HTML tags from the description

Rollback Database

- you can experiment with rolling back the migration, so your schema will be transported back in time, and the products table will be disappeared

```
$ rake db:rollback
```

Play More with Active Scaffold

- Install ActiveScaffold to the application

```
$ rails plugin install
  https://github.com/vhochstein/active_scaffold.git
$ rails g active_scaffold_setup [prototype| jquery]
```

- If you are having troubles generating the setup, i.e.,

c:/Ruby192/lib/ruby/1.9.1/net/http.rb:677:in `connect': SSL_connect returned=1 errno=0 state=SSLv3 read server certificate B: certificate verify failed (OpenSSL::SSL::SSLError)

you should add the line,

```
OpenSSL::SSL::VERIFY_PEER = OpenSSL::SSL::VERIFY_NONE
```

into the config/application.rb at the location before the line, `require 'rails/all'`

Play More with Active Scaffold (2)

- Create 2 Active Scaffolds

```
$ rails g active_scaffold Team name:string  position:integer
$ rails g active_scaffold Player name:string
    injured:boolean salary:decimal date_of_birth:date
    team:references
```

- Data Migration

```
$ rake db:migrate
```

- Create one-to-many relation between the players and the team by editing app/model/team.rb with adding "has_many :players" into the Team class
- Then, test your Scaffolds

To develop Rails on Windows, you require the following components: ðŸ’Ž Ruby. ðŸ”¨ Build Tools (used for gem compilation). ðŸ“‹ RubyGems. ðŸš€ Rails. ðŸš¦ GIT. ðŸ’» IDE (Code Editor). These are all freely available, and are required for any OS Ruby install â†´. â€” ðŸ“ Note from the author (May 2020). As far as I am aware, this is the ONLY Ruby/Rails on Windows tutorial which doesnâ€™t require the installation of Bash. The bash dependency is around 215mb+ of unnecessary data, not worth it. This tutorial explains the most space-efficient way of installing a pre-compiled version of Ruby, which only requires around 5mb of data + your bundles. ruby-on-rails-3 Tutorial - This section provides an overview of what ruby-on-rails-3 is, and why a developer might want to use it.It should also mention any...Â It should also mention any large subjects within ruby-on-rails-3, and link out to the related topics. Since the Documentation for ruby-on-rails-3 is new, you may need to create initial versions of those related topics. Hello World in Rails. Say "Hello", Rails. To get Rails saying "Hello", you need to create at minimum a controller and a view. A controller's purpose is to receive specific requests for the application. Routing decides which controller receives which requests. Although Ruby 3.0 significantly decreased the size of JIT-ed code, it is still not ready for optimizing workloads like Rails, which often spend time on so many methods and therefore suffer from i-cache misses exacerbated by JIT. Stay tuned for Ruby 3.1 for further improvements on this issue. Concurrency / Parallel. Itâ€™s multi-core age today. Rails.application.config.action_controller.urlsafe_csrf_tokens = true. Scott Blum, Ã‰tienne BarriÃ©. Signed and encrypted cookies can now store false as their value when action_dispatch.use_cookies_with_metadata is enabled.Â Allow relative paths with trailing slashes to be passed to rails test. Eugene Kenny. Return a 405 Method Not Allowed response when a request uses an unknown HTTP method. Ruby on Rails is an open-source web development framework written in Ruby. Ruby on Rails follows the principle of convention over configuration, freeing you from having to re-invent things to stay productive. Use this tag only for Rails 3-specific questions, and also tag those questions [ruby-on-rails]. Learn moreâ€¦ Top users. Synonyms (2). 56,024 questions. Newest. Active.