Rails routing, kontrollerek
Gyakorlat

Kovács Gábor

2011. november 8.

Legelőször az útvonalakat tartalmazó `config/routes.rb` fájlban állítsuk át a webszerverünk gyökerét az idézetek index oldalára, ami nem más, mint a `quotes` kontroller `show` akciója és annak nézete. A webszerver könyvtárában, vagyis a `public`-ban nevezzük át az `index.html`-t!

```
root :to => "quotes#show"
```

A gyakorlat során az előző gyakorlatokon megkezdett feladat megoldását folytatjuk. Először kiegészítjük a `User` modell osztályunkat két további, indirekt kapcsolattal, és módosítjuk az egyik meglévő kapcsolatot. Megtehetjük, hogy egy másik modellre nem a modell nevével hivatkozunk, ekkor azonban nevesítenünk kell a modell osztályt és esetleg az idegen kulcsot. Ehhez nevezzük át a `:comments` relációt, és adjuk meg a hivatkozott modell osztály nevét a `:class_name` opción keresztül. Ez a reláció változatlanul a felhasználó által az oldalra beírt kommentek enumerációját adja vissza. A kommentek azonban egy másik reláción keresztül is elérhetők a felhasználó számára, a felhasználó által létrehozott idézetekhez is tartoznak kommentek. Jelöljük ezt a `:comments` relációval! Mivel ez egy indirekt kapcsolat, meg kell adnunk a `:through` opció után annak a kapcsolatnak a nevét, amin keresztül ezt használni akarjuk. Jelen esetben ez a `:quotes`, amely a `Quote` modell példányainak enumerációjára hivatkozik. Hasonlóan létrehozhatjuk a felhasználó idézeteit kommentelő felhasználók enumerációját. Ezt az imént létrehozott `:comments` kapcsolaton keresztül tesszük meg.

```
class User < ActiveRecord::Base
  has_many :quotes
  has_many :comments_made, :class_name => 'Comment'
  has_many :comments, :through => :quotes
  has_many :users, :through => :comments
end
```

Ellenőrizzük, hogy ez valóban működik-e! Először hozzunk létre e célból tesztadatokat! Felhasználónk már van négy, így azzal nem foglalkozunk most. Inicializáljunk egy forrás példányt, és mentsük el az adatbázisba (8-13. sorok). Hozzunk létre egy idézetet (5-6. sorok), inicializáljuk és mentsük el az adatbázisba (15-20. sorok). Végül rendeljünk az idézethez sok-sok kommentet (23-25. sorok).

```
irb(main):008:0> s = Source.new
=> #<Source id: nil, source: nil, subject: nil,
    released_at: nil, created_at: nil, updated_at: nil>
irb(main):009:0> s.source='Kovacs Gabor'
=> "Kovacs Gabor"
irb(main):010:0> s.subject = 'RoR'
=> "RoR"
irb(main):011:0> s.released_at = Time.now
=> 2011-11-08 12:24:54 +0100
irb(main):012:0> s.save
   (0.2ms)  BEGIN
  SQL (0.6ms)  INSERT INTO `sources` (`created_at`, `
      released_at`, `source`, `subject`, `updated_at`)
      VALUES ('2011-11-08 11:24:57', '2011-11-08
      11:24:54', 'Kovacs Gabor', 'RoR', '2011-11-08
      11:24:57')
   (1.4ms)  COMMIT
=> true
irb(main):013:0> Source.find :all
  Source Load (0.5ms)  SELECT `sources`.* FROM `sources
      `
=> [#<Source id: 1, source: "Kovacs Gabor", subject: "
    RoR", released_at: "2011-11-08 11:24:54", created_at
    : "2011-11-08 11:24:57", updated_at: "2011-11-08
    11:24:57">]
irb(main):005:0> q = Quote.find :all
  Quote Load (0.4ms)  SELECT `quotes`.* FROM `quotes`
=> []
irb(main):006:0> q = Quote.new
=> #<Quote id: nil, quote: nil, evaluation: nil, votes:
     nil, created_at: nil, updated_at: nil, user_id: nil
    , source_id: nil>
irb(main):015:0> q.user_id=1
=> 1
```

```
irb(main):016:0> q.source_id=1
=> 1
irb(main):017:0> q.quote="Valami nagyon-nagyon vicces"
=> "Valami nagyon-nagyon vicces"
irb(main):018:0> q.votes=1
=> 1
irb(main):019:0> q.evaluation=5
=> 5
irb(main):020:0> q.save
   (0.1ms)  BEGIN
  SQL (0.4ms)  INSERT INTO `quotes` (`created_at`, `
      evaluation`, `quote`, `source_id`, `updated_at`, `
      user_id`, `votes`) VALUES ('2011-11-08 11:26:07',
      5, 'Valami nagyon-nagyon vicces', 1, '2011-11-08
      11:26:07', 1, 1)
   (1.1ms)  COMMIT
=> true
irb(main):023:0> for i in 1..15
irb(main):024:1> c = Comment.create :comment => "Nagyon
   vicces", :user_id=>2, :quote_id=>1
irb(main):025:1> end
```

Most már kipróbálhatjuk, hogy működnek-e az új kapcsolataink. A kérések eredményeinek közlésétől terjedelmi okokból eltekintek.

```
irb(main):026:0> u = User.find 1
irb(main):027:0> u.quotes
irb(main):028:0> u.comments
irb(main):029:0> u.users
```

Az előző gyakorlat során létrehoztuk a `Source` modellt, amelyhez azonban jelenleg még nem tartozik kontroller és nézet. Pótoljuk ezt most az eddig megszokott menettől eltérő módon. Nevezzük át a modell létrehozásakor létrejött migrációt például `create_sources2`-re, töröljük a fájlrendszerről a `Source` modell osztályt, a `test/fixtures` könyvtárból a `sources.yml` állományt és a `test/unit` könyvtárból a `source_test.rb`-t!

Hozzunk létre egy új migrációt, amely a forrás létrehozását hivatott kompenzálni.

```
kovacsg@debian:~/gyakorlat/db/migrate$ rails generate
   migration DropSources
```

Fel irányban töröljük a `sources` táblát, le irányban pedig a korábbiakkal megegyező módon hozzuk létre.

```
class DropSources < ActiveRecord::Migration
  def up
    drop_table :sources
  end

  def down
    create_table :sources do |t|
        t.string :source, :limit=>40
        t.string :subject
        t.datetime :released_at
        t.timestamps
    end
  end
end
```

Hajtsuk végre a migrációt végleg megszabadulva a `Source` modell minden korábbi nyomától, majd egy `scaffold` paranccsal hozzuk létre újra a modellt és a hozzá kapcsolódó nézetet és kontrollert, végül újból migráljuk az adatbázist! Végül a fent már látott módon hozzunk létre egy új rekordot `sources` táblába!

```
kovacsg@debian:~/gyakorlat/db/migrate$ rake db:migrate
(in /home/kovacsg/gyakorlat)
==  DropSources: migrating
    ==========================================================
--  drop_table(:sources)
    -> 0.0033s
==  DropSources: migrated (0.0037s)
    =================================================
kovacsg@debian:~/gyakorlat/db/migrate$ rails generate
    scaffold Source source:string subject:string
    released_at:datetime
/var/lib/gems/1.9.1/gems/rack-1.3.4/lib/rack/backports/
    uri/common_192.rb:53: warning: already initialized
    constant WFKV_
       invoke  active_record
       create    db/migrate/20111108113426
           _create_sources.rb
       create    app/models/source.rb
```

```
      invoke    test_unit
      create      test/unit/source_test.rb
      create      test/fixtures/sources.yml
       route  resources :sources
      invoke  scaffold_controller
      create    app/controllers/sources_controller.rb
      invoke    erb
      create      app/views/sources
      create      app/views/sources/index.html.erb
      create      app/views/sources/edit.html.erb
      create      app/views/sources/show.html.erb
      create      app/views/sources/new.html.erb
      create      app/views/sources/_form.html.erb
      invoke    test_unit
      create      test/functional/
         sources_controller_test.rb
      invoke    helper
      create      app/helpers/sources_helper.rb
      invoke      test_unit
      create        test/unit/helpers/
         sources_helper_test.rb
      invoke  assets
      invoke    coffee
      create      app/assets/javascripts/sources.js.
         coffee
      invoke    scss
      create      app/assets/stylesheets/sources.css.
         scss
      invoke  scss
      create    app/assets/stylesheets/scaffolds.css.
         scss
kovacsg@debian:~/gyakorlat/db/migrate$ rake db:migrate
(in /home/kovacsg/gyakorlat)
==  CreateSources: migrating
    ============================================================
-- create_table(:sources)
   -> 0.0677s
==  CreateSources: migrated (0.0678s)
    ===========================================================
```

A következő nagyobb feladatunk az idézetek kontroller és nézet rendbe

tétele. A kettővel ezelőtti gyakorlaton a kontrollerbe drótozott adatokkal már létrehoztuk a `new` és a `show` akciókat, azonban azóta átalakítottuk az adatmodellünket, így azok megtekintése esetén hibát kapunk. Az idézetek modellről leválaszottuk a forrás modellt, ami mind a kontrollerben, mind a nézetben jelenleg még szerepel. Vessük el az eddig elkészült kódot!

Először javítsuk ki a `show` akciót, majd nézetét! Kezdetben jelenítsük meg az oldalunkon az összes idézetet!

```
class QuotesController < ApplicationController
  def show
    @quotes = Quote.find :all
  end
end
```

A nézetben írjuk ki az idézet forrásának tulajdonságait, vagyis a forrás nevét, a tárgy nevét, melynek keretein belül elhangzott és az elhangzás évét, melyet formázással nyerünk ki a `released_at` attribútumból! Ez alatt jelenítsük meg magát az idézetet, és alatta az értékelését! Végül egy linkkel tegyük elérhetővé az idézethez kapcsolódó kommenteket megjelenítő oldalt.

```
<div>
<h3>Funny quote</h3>
<% for q in @quotes %>
  <div class="quotebox">
  <b><%= q.source.source %> said in the
    <%= q.source.subject %> in year
    <%= q.source.released_at.strftime("%Y") %></b> <br
       />
    <%= q.quote %><br />
     Evaluation: <%= q.evaluation %> based on <%= q.
        votes %> votes.<br />
    <%= link_to "Comments", {:action => "comment", :id
       => q.id } %>
  </div>
<% end %>
</div>
```

Az idézetekhez fűzözz kommentek listázásának akcióját teljesen hasonló elvek alapján módosítjuk. A kontrollerben az `:id` paraméter alapján megszerezzük az aktuális idézet példányát és a hozzá kapcsolódó kommenteket.

```
class QuotesController < ApplicationController
  def comment
```

```
    @quote = Quote.find params[:id]
    @comments = @quote.comments
  end
end
```

A nézetben megjelenítjük az idézetet és az idézet forrásának néhány tulajdonságát, majd alatta az összes eddigi kommentet a kommentelő felhasználó azonosítójával és a komment létrehozásának formázott időpontjával.

```
<div>
<b><%= @quote.source.source %> said in subject
  <%= @quote.source.subject %> in year
  <%= @quote.source.released_at.strftime("%Y") %><br />
  <%= @quote.quote %></b><br />
<% for c in @comments %>
  <div class="commentbox">
    <%= c.user.username %> added comment on <%= c.
        updated_at.strftime("%d/%m/%Y %H:%M") %><br />
    <%= c.comment %><br />
  </div>
<% end %>
</div>
```

Kommentből azonban nagyon sok lehet, előfordulhat, hogy nem férnek el az oldal központi területén. Ezért a kommenteket oldalakra tördeljük, amihez a `will_paginate` Rails gemet használjuk. Új külső API használatához az első dolgunk a Gemfile módosítása. Megadjuk az alkalmazásunknak, hogy a megnevezett plugin legalább `3.0`-s verziójára van szükségünk.

```
gem 'will_paginate', '~> 3.0'
```

A következő lépés az alkalmazás frissítése, majd a webszerver újraindítása.

```
bundle install
```

A feladatunk a `comment` akcióban található `@quotes` enumeráció megfelelő tördelésének megvalósítása. A tördelést magát helyezzük át a modellbe egy új metódus segítségével, amely az aktulis modell példányra vonatkozó a paraméterben megadott oldalszámhoz tartozó kommentlistát adja vissza oldalanként öt találattal

```
class Quote < ActiveRecord::Base
  def get_comment_page(page)
    comments.paginate(:page=>page,:per_page=>5)
```

```
  end
end
```

A kontroller akcióban ezt a metódust hívjuk meg a `page` kérés paraméterrel.

```
class QuotesController < ApplicationController
  def comment
    @quote = Quote.find params[:id]
    @comments = @quote.get_comment_page params[:page]
  end
end
```

A nézetben hozzáadjuk a lapozó linkeket mondjuk az oldal aljára, amelyek beállítják a `page` paramétert, a tördelés a kommentek enumerációjának példányváltozója alapján történik.

```
<%= will_paginate @comments %>
```

Nem csak kommentből, hanem idézetből is lehet nagyon sok, ezért ugyanezt le kell játszanunk az idézetek listájára is. Hozzunk létre egy új idézetet egy másik felhasználó által, és legyen ennek az értékelése az előzőnél valamivel gyengébb!

```
irb(main):001:0> u = User.find 3
  User Load (0.6ms)  SELECT `users`.* FROM `users`
    WHERE `users`.`id` = 3 LIMIT 1
=> #<User id: 3, username: "Barki", encrypted_password:
    "60916
    d509e3299674f89c915fa662718ffa8d798b87ae422d32...",
    email: "barki@mail.bme.hu", created_at: "2011-10-25
    11:34:51", updated_at: "2011-10-25 11:46:37", salt:
    "CboQiN81EvA=">
irb(main):002:0> q = Quote.new
=> #<Quote id: nil, quote: nil, evaluation: nil, votes:
    nil, created_at: nil, updated_at: nil, user_id: nil
    , source_id: nil>
irb(main):003:0> q.user_id=3
=> 3
irb(main):004:0> q.source_id=1
=> 1
irb(main):005:0> q.evaluation=4
=> 4
irb(main):006:0> q.votes=1
```

```
=> 1
irb(main):007:0> q.quote = "Valami vicces"
=> "Valami vicces"
irb(main):008:0> q.save
   (0.1ms)  BEGIN
  SQL (0.3ms)  INSERT INTO `quotes` (`created_at`, `
      evaluation`, `quote`, `source_id`, `updated_at`, `
      user_id`, `votes`) VALUES ('2011-11-08 11:49:07',
      4, 'Valami vicces', 1, '2011-11-08 11:49:07', 3,
      1)
   (1.2ms)  COMMIT
=> true
```

Ez esetben is hozzuk létre a modellben a tördelést megvalósító metódust.

```
class Quote < ActiveRecord::Base
  def self.get_page(page)
    Quote.find(:all).paginate(:page=>page, :per_page
        =>3)
  end
end
```

Módosítsuk a kontoller `show` akcióját, hogy az ezt a metódust hívja meg.

```
class QuotesController < ApplicationController
  def show
    @quotes = Quote.get_page params[:page]
  end
end
```

A nézethez végül hozzáadjuk a tördelést megvalósító linkeket az oldal aljára.

```
<%= will_paginate @quotes %>
```

Az egyik használati esetünk szerint az oldalra látogató felhasználó megnézheti mind a legviccesebbnek tartott idézeteket, mind a legújabb idézeteket. A felhasználó egy link segítségével kiválaszthatja, hogy melyik elv szerint kívánja rendeztetni az idézeket. Az elv érvényessége a session. Helyezzük el a két linket a `show` nézet külső `div`-jében, a linkek mutassanak az aktuális kontroller `order` akciójára. Ha idő szerinti rendezést kívánunk, akkor legyen az `id` paraméter értéke nulla, ha értékelés szerinti rendezést kívánunk, akkor pedig egy.

```
<%= link_to "Most recent quotes", { :action=>"order", :
    id=>0} %>
<%= link_to "Most popular quotes", { :action=>"order",
    :id => 1}
 %%<br/>
```

A `order` akció feladata a `session` hash rendezésre vonatkozó elemének beállítása a kérés paramétere alapján. Majd az előző oldalra való visszatérét, lévén ehhez nem tartozik önálló nézet.

```
class QuotesController < ApplicationController
  def order
    if params[:id].to_i == 1
      session[:order]=true
    else
      session[:order]=false
    end
    redirect_to :back
  end
end
```

Most már tudjuk, hogy mi szerint kell rendeznünk, már csak a rendezés megvalósítása van hátra. A rendezést a modell osztály új metódusaiban tesszük meg. A `find ActiveRecord` metódus helyett az `order`-t használjuk, amely az argumentumban megadott attribútum szerint rendezve adja vissza az összes rekordot. A rendezési elvünk alapvetően a hozzáadási idő tekintetében csökkenő, amit a `order` session paraméter kiegészíthet a `show` akcióban használt `get_page` metódusban, amelynek így még egy paramétere lesz.

```
class Quote < ActiveRecord::Base
  def get_comment_page(page)
    comments.order("updated_at DESC").paginate(:page=>
        page,:per_page=>5)
  end

  def self.get_page(page,order)
    if order
      Quote.order("updated_at,evaluation DESC").
          paginate(:page=>page,:per_page=>3)
    else
      Quote.order("updated_at DESC").paginate(:page=>
          page,:per_page=>3)
```

```
    end
  end
end
```

A kontoller akcióban a paraméterszám most nem stimmel, javítsuk ki!

```
def show
  @quotes = Quote.get_page params[:page], session[:
    order]
end
```

Az új idézetet létrehozó akció és nézet is elavult a modellek átalakításával. Új idézetet a bejelentkezett felhasználók számára engedjünk létrehozni.

```
def new
  if logged_in?
    @quote=Quote.new
  else
   redirect_to '/quotes/show'
  end
end
```

Az új idézet létrehozása nézet formja a forrás nevét, magát az idézetet és egy nyomógombot kell, hogy tartalmazzon. Mivel a források halmaza nem zárt a források nevét tekintve, és a számuk nagyon nagy lehet az új források létrehozásakor a források listája nézeten kereszül kell lehetővé tennünk a kiválasztást. Ezt két linkkel oldjuk meg, amelyek közvetve a `sources` kontroller `new`, illetve `show` akcióira mutatnak az `add_source`, illetve a `list_sources` akciókon keresztül. Amikor először látogatunk az nézetre a forrás még nem tartalmaz értéket, ezért annak az ellenőrzését el kell végeznünk.

```
<fieldset>
  <legend>Submit new quote</legend>
  <%= form_for :quote, :url => {:controller => "quotes"
      , :action => "create"} do |form| %>
  <div>
    <%= label_tag "source" %>: <br />
    <% if @quote.source %>
      <%= @quote.source.source %>
    <% else %>
      <% for i in 1..10 do %>   <%end%>
    <% end %>
    <%= link_to "Available sources", {:action => "
      list_sources"} %>
```

```
    <%= link_to "Create a new source", {:action =>"
        add_source"} %>
  </div>
  <div>
    <%= form.label :quote %>: <br />
    <%= form.text_area :quote, :cols=>40, :rows=> 10 %>
  </div>
  <%= form.submit "Save" %>
      <% end %>
</fieldset>
```

Mind az `add_source`, mind a `list_sources` hívásakor a `adding_source` session paraméteren keresztül tudatjuk a `sources` kontrollerrel és nézettel, hogy session paraméteren keresztül juttassa vissza a kiválasztott forrás azonosítóját, és engedélyezzen a források kiválasztására vonatkoztó linkeket.

```
class QuotesController < ApplicationController
  def add_source
    session[:adding_source]
    redirect_to :controller=>"sources", :action=>"new"
  end
  def list_sources
    session[:adding_source]
    redirect_to :controller=>"sources"
  end
end
```

A források kontrollerében használjuk a következő helper metódust annak megállapítására, hogy engedélyezni kell-e a forráskiválasztáshoz kapcsolódó kódrészleteket.

```
module SourcesHelper
  def is_adding?
    session[:adding_source]
  end
end
```

A források kontrollerében egy új akciót veszünk fel, amely minden kiválasztás esetén beállítja a forrás session paramétert, törli a forráskiválasztás session paramétert, és visszairányítja a felhasználót az új idézet felvétele oldalra.

```
class SourcesController < ApplicationController
  def add_source
```

```
      session[:source]=params[:id]
      session[:adding_source]=nil
      redirect_to "/quotes/new"
    end
end
```

Ugyanitt az új forrást létrehozó `create` akcióban sikeres mentés esetén hajtsuk végre ugyanezt a metódust ha az `is_adding?` visszatérési értéke true, így kiválasztás esetén az új idézet létrehozása oldalra kerülünk a kijelölt forrással.

Az `index` nézet táblájához adjunk hozzá egy új oszlopot, ha az `is_adding?` feltétel teljesül benne egy linkkel, amely az `add_source` metódusra mutat.

```
<table>
  <tr>
    <th>Source</th>
    <th>Subject</th>
    <th>Released at</th>
    <th></th>
    <% if logged_in? %>
    <th></th>
    <th></th>
    <% if is_adding? %>
    <th></th>
    <% end %>
    <% end %>
  </tr>

<% @sources.each do |source| %>
  <tr>
    <td><%= source.source %></td>
    <td><%= source.subject %></td>
    <td><%= source.released_at %></td>
    <td><%= link_to 'Show', source %></td>
    <% if logged_in? %>
    <td><%= link_to 'Edit', edit_source_path(source)
       %></td>
    <td><%= link_to 'Destroy', source, confirm: 'Are␣
       you␣sure?', method: :delete %></td>
    <% if is_adding? %>
    <td><%= link_to 'Add␣source', {:action => '
       add_source', :id => source.id} %></td>
```

```
    <% end %>
    <% end %>
  </tr>
<% end %>
</table>
```

Így az új idézet bevite oldalra visszatérve mindkét esetben a `source` session paraméterből kinyerhetjük a forrás példányát a `new` akcióban.

```
class QuotesController < ApplicationController
  def new
    if session[:user]
      @quote = Quote.new
      if session[:source]
      @quote.source_id=session[:source]
      session[:source]=nil
    else
      redirect_to '/quotes/show'
    end
  end
end
```