

Rails routing, kontrollerek

Gyakorlat

Kovács Gábor

2015. november 10.

A gyakorlat során az előző gyakorlatokon megkezdett feladat megoldását folytatjuk. Legelőször az útvonalakat tartalmazó `config/routes.rb` fájlban állítsuk át a webservert a gyökerét a `hello` nézetre.

```
root :to => "say#hello"
```

Az előző alkalommal a teszt kitöltésének mevalósításánál hagytuk abba. A tesztet kitöltő form során tovább kell adnunk az épp kitöltött teszt azonosítóját és az épp megválaszolt kérdés indexét. Legyen e két paraméter neve `tid` és `sid`. A tesztkérdést megjelenítő útvonal a `words` kontrollerek `test` metódusa lesz, az eseménykezelő pedig a `solve` akció, rendeljünk mindkettőhöz útvonal helpereket is!

```
match 'test(/:tid(/:sid))', to: 'words#test', as: 'test', via: [:get, :post]
post 'test/:tid/solve/:sid', to: 'words#solve', as: 'solve'
```

A `test` akciónak két működési módja van. Ha a felhasználó megadta a folytatni kívánt tesztet, akkor előkeressük az, ha nem, akkor újat generálunk. A következő kérdés tömbjének indexét a megválaszolt kérdések tömbjének mérete adja.

```
class WordsController < ApplicationController
  def test
    if params[:tid].nil?
      @test = Test.new
      @test.user = User.find session[:user]
      @test.save
    else
      @test = Test.find params[:tid]
    end
  end
end
```

```

    @i = JSON.parse(@test.solution).size

    @word = Word.find(JSON.parse(@test.questions)[@i])
    @word.word = ''
  end
end

```

A teszt modellünkben javítsuk ki a `create_seq` függvényünket, hogy az csak új teszt létrehozásakor generáljon véletlen kérdéssorozatot.

```

class Test < ActiveRecord::Base
  def create_seq
    if new_record?
      self.questions = Word.all.sample(20).collect do |x| x.id
      end.to_json
      self.solution = [].to_json
    end
  end
end

```

Amikor a tesztet kitöltő felhasználó megadja a leírásnak megfelelő szót, akkor nekünk annak helyességét ellenőriznünk kell a `solve` akcióban. Ehhez először előkeressük az adatbázisból a tesztet a `tid` paraméter alapján, és az aktális szót a `sid` paraméter alapján. Ha a keresés nem üres eredményhalmazt ad, akkor a teszt megfejtéseinek listájába beszurunk egy `true` értéket, ellenkező esetben pedig egy `false` értéket. Ha a teszt végére értünk, akkor megjelenítjük a teszt összesítő képernyőjét. Fontos, hogy a függvény törzsében kiadott `render` után visszatérjünk a függvényből, különben a válasz összeállítása folytatódna a további sorokon. Végül, tehát, ha még nem értünk a teszt végére, előkeressük a következő szót, és megjelenítjük a `test` nézetet.

```

class WordsController < ApplicationController
  def solve
    @test = Test.find(params[:tid])
    arr = JSON.parse(@test.solution)
    @word = Word.find(params[:sid])
    solution = params[:word][:word]
    if Word.where('word_LIKE_?_AND_description_=?',
      "%#{solution.strip}%", @word.description).size > 0
      arr << true
      @test.solution = arr.to_json
    else
      arr << false
      @test.solution = arr.to_json
    end
    @test.save if arr.size <= 20
  end
end

```

```

arr = JSON.parse(@test.solution)
if arr.size >= 20
  @hits = arr.count(true)
  render template: 'words/results'
  return
end
@i = arr.size

@word = Word.find(JSON.parse(@test.questions)[@i])
@word.word = ''
render :test
end
end

```

Vegyünk fel egy profil oldalt, ahol a felhasználó megtekintheti a saját statisztikáit. Ennek az útvonala legyen paraméterezzhető a felhasználó azonosítójával.

```
get 'users/show/:id', to: 'users#show', as: 'show_user'
```

A `show` akció a felhasználók kontrollerében összeszámolja az összes és a sikeres válaszokat.

```

class UsersController < ApplicationController
  def show
    @sum = 0
    @ans = 0
    @user.tests.each do |t|
      @sum += JSON.parse(t.solution).count if t.solution
      @ans += JSON.parse(t.solution).count(true) if t.solution
    end
  end
end
end

```

A `show` nézetben jelenítsük meg az aktuális felhasználó profil adatait.

```

<h1>User profile</h1>
<%= t('activerecord/attributes/user/username') %>: <%= @user.
  username %> <br/>
Email: <%= @user.email %> <br/>
Birthdate: <%= l @user.birthdate, format: :datetime %> <br/>
Statistics: <%= @ans %> correct out of <%= @sum %>, <%= @ans/@sum
  .to_f %>% <br/>

<%= link_to "Back", :back %>

```

Módosítsuk a menüt, hogy az tartalmazza a bejelentkezett felhasználó számára a profil oldalt, és a linkek jó helyre mutassanak, és specializáljuk az aktuális felhasználónak megfelelően. A menü nem csak a felhasználók

kontrollerben érhető el, de jelenleg csak ott keressük elő a `@user` példányváltozót. Mozgassuk át ezt a keresést alkalmazásszintre, hogy az az összes kontrollerben elérhető legyen, ez egyúttal feleslegessé teszi a `UserController` `find_user` szűrőjét.

```
class ApplicationController < ActionController::Base
  before_filter :load_user

  private
  def load_user
    if session[:user]
      @user = User.find session[:user]
    end
  end
end
```

A módosítások során a menüben felhasználót név szerint üdvözljük, és megjelenik a profil link is.

```
<%= flash[:notice] %><br/>
<%= if logged_in? %>
Welcome, <%= @user.username %>!<br/>
  <%= link_to "Register_new_words", '/words/new' %><br/>
  <%= link_to "Take_test", '/test' %><br/>
  <%= link_to "Profile", show_user_path(@user.id) %><br/>
  <%= link_to "Logout", '/sessions/destroy'%>
<%= else %>
Welcome, guest!
<fieldset>
  <legend>Login</legend>
  <%= form_tag '/sessions/create', :method => :post do%>
    <%= label_tag :username, 'Username' %><br/>
    <%= text_field_tag :username, '', :size => 16 %><br/>
    <%= label_tag :password, "Password" %><br/>
    <%= password_field_tag :password, '', :size => 16 %><br/>
    <%= submit_tag 'Login' %>
  <%= end %>
</fieldset>
<%= link_to "Register", '/users/new' %><br/>
<%= link_to "Forgotten_password", '/users/forgotten' %>
<%= end %>
```

Bővítsük ki lokalizációnkat! A modellek lokalizációja során megadhatjuk a megjelenítendő hibaüzenetek helyi szövegét, a modellek nevének és a modellek attribútumainak fordítását az alábbi módon.

```
en :
```

```
activerecord:
  error:
    models:
      user:
        attributes:
          username:
            blank: "Username_is_blank"
    models:
      user: "Felhasznalok"
    attributes:
      user:
        username: "Felhaszalonev"
```