Rails routing, kontrollerek
Gyakorlat

Kovács Gábor

2011. április 12.

Legelőször az útvonalakat tartalmazó `config/routes.rb` fájlban állítsuk át a webszerverünk gyölerét a bejelentkezési képernyő `new` oldalára.

```
root :to => "sessions#new"
```

A gyakorlat során az előző gyakorlatokon megkezdett feladat megoldását folytatjuk. Először módosítjuk a `Task` kontrollert és nézetet, átalakítjuk, hogy az a Rails form helpereket használja.

A `task_controller`-ben az előszűrő segítségével a session paraméterek alapján visszaállítjuk az aktuális felhasználót, és beállítjuk a feladat szerkesztésére vonatkozó jogosultságait.

```
  before_filter :check_session
  private
  def check_session
    @user = User.find session[:user]
    @perm = session[:user] && !@user.student
  end
```

Az új feladat létrehozása formot átalakítjuk modell alapúra, a `form_for` helper segítségével. A formot a `create` akció kezeli le, amelynek átadjuk az új feladat példány azonosítóját is. A sorszámot az `1..6` tartományból választjuk ki. Az URL attribútum egy maximum 60 karakter hosszú szövegbeviteli mezőben adható meg. A határidőt kiválasztó beviteli mezőket pedig a `select_datetime` helperrel hozzuk létre. A `edit` nézet ettől csak a címsorban és a formot kezelő akcióban különbözik, amely az `update` lesz.

```
<div>
<h1>Új feladat</h1>
<%= form_for :task, @task, :url => {:action => "create"
    , :id=>@task.id }, :method => :post do |f| %>
```

```
  <table class="editbox">
    <tr>
      <td>Sorszám</td>
      <td><%= f.select(:number, (1..6).to_a) %></td></
        tr>
    <tr>
      <td>URL</td>
      <td><%= f.text_field :url, :size => 60 %></td></
        tr>
    <tr>
      <td>Határidő</td>
      <td><%= select_datetime(@task.deadline,:prefix=>"
        task") %></td>
    </tr>
    <tr>
       <td><%= f.submit "Mentés" %></td><td></td>
    </tr>
  </table>
<% end %>
</div>
```

Az új feladat nézetet és a nézeten található form eseményét kezelő `create` akciót definiáljuk. A `create` inicializál egy új `Task` példányt a form paraméterek értékei alapján, majd, amennyiben a felhasználó rendelkezik a megfelelő jogosultságokkal, elmenti az adatbázisba. Végül a feladatok listája oldara navigálja át a felhasználót.

```
  def new
    @task = Task.new
  end

  def create
    @task = Task.new
    t = params[:task]
    @task.number = t[:number]
    @task.url = t[:url]
    deadline = DateTime.civil(t[:year].to_i, t[:month].
      to_i,
        t[:day].to_i, t[:hour].to_i, t[:minute].to_i)
    @task.deadline=deadline
    if @perm
      @task.save
```

```
    end
    redirect_to :action=>"index"
  end
```

Az összes feladat listázásához a kontroller `index` akciójában az adatbázisból lekérjük a feladatok halmazát.

```
  def index
    @tasks = Task.find :all
  end
```

Az `index` nézet a feladatok listáját jeleníti meg táblázatba rendezve. A felhasználó típusa (hallgató/oktató) szerint más más akciókat tesz lehetővé a táblázat. A hallgató feladatot adhat be, amely a `solutions` kontroller `new` akciójára viszi ált a felhasználót, a kérés `id` paramétere a feladat sorszáma. A feladat mindkét típusú felhasználó számára megtekinthető, a `Mutat` link a modell `url` attribútuma által reprezentált oldalra navigál. Ami a hallgató típusú felhasználó esetén a beadás akció volt, az oktató típusú felhasználó esetén a javítás, amely a `solutions` kontroller `show` akciójára navigál `id`-ként a feladat sorszámát megadva. A szerkesztés akció a `edit` nézetet aktiválja, a törlés akció pedig a `destroy` metódust hívja meg.

```
<h1>Feladatok listája</h1>

<table class="listbox">
  <tr>
    <th>Sorszám</th>
    <th>URL</th>
    <th>Határidő</th>
    <th></th>
    <th></th>
    <th></th>
    <th></th>
  </tr>
<% @tasks.each do |task| %>
  <tr>
    <td><%= task.number %></td>
    <td><%= task.url %> </td>
    <td><%= task.deadline %></td>
    <% if @user then if @user.student? %>
    <td><%= link_to "Bead", :controller => "solutions",
        :action => "new", :id=>task.number %></td>
    <td><%= link_to "Mutat", task.url %></td>
```

```
    <% else %>
    <td><%= link_to "Mutat", task.url %></td>
    <td><%= link_to "Javít", :controller=>"solutions",
        :action => "show", :task_id=>task.number %></td>
    <td><%= link_to "Szerkeszt", :action=>"edit", :id=>
        task.number %></td>
    <td><%= link_to "Töröl", :action => "destroy", :id
        => task.number %></td>
    <% end %>
    <% end %>
  </tr>
<%end %>
</table>
<% if @user then if !@user.student? %>
<div class="newbox">
  <%= link_to "Új feladat", :action=>"new" %>
</div>
<% end end %>
```

Egy az `id` kérés paraméterén kereszül átadott sorszámmal rendelkező feladat szerkesztéséhez a feladat példányát betöltjük az adatbázisból.

```
  def edit
    @task = Task.find_by_number params[:id]
  end
```

A szerkesztés nézet akcióját a `update` metódus kezeli le, amely szinte teljesen megegyezik a `create` metódussal. A különbség az, hogy itt nem egy új példány attribútumait állítjuk be, hanem egy az adatbázisból kikeresettét. A frissítés itt is a jogosultság ellenőrzése után történik csak meg, és a következő nézet itt is az `index`.

```
  def update
    t = params[:task]
    @task = Task.find_by_number t[:number]
    @task.number = t[:number]
    @task.url = t[:url]
    deadline = DateTime.civil(t[:year].to_i, t[:month].
        to_i,
      t[:day].to_i, t[:hour].to_i, t[:minute].to_i)
    @task.deadline=deadline
    if @perm
      @task.save
```

```
    end
    redirect_to :action=>"index"
  end
```

Egy feladat törlését a `destroy` metódus valósítja meg, amely a paraméterül kapott feladat sorszámnak megfelelő rekordok kikeresi az adatbázisból, majd törli azt. A következő nézet itt is a `index`.

```
  def destroy
    @task = Task.find_by_number params[:id]
    if @task
      @task.destroy
    end
    redirect_to :action=>"index"
  end
```

Mivel a feladatok nézeteinek akcióit átneveztük, az alkalmazás keretében módosítanunk kell a feladatok linket, hogy az a `show` helyett az `index` akcióra mutasson.

Hozzuk létre a megoldások kontrollerét három akcióval! A `new` akciót a hallgató típusú felhasználók használják feladat beadásakor. Ennek definíciójától a továbbiakban eltekintünk (lásd archívum).

```
rails generate controller solutions show new edit
```

Ezt a kontrollert is kezdjük a felhasználó objektum (`@user`) sessionből való kinyerésével:

```
  before_filter :init_user
  private
  def init_user
    @user = User.find session[:user]
  end
```

A megoldások listájának megjelenítése függ a felhasználó típusától. Hallgató típusú felhasználó esetén a felhasználó által beadott megoldásokat jelenítjük meg. Oktató típusú felhasználó esetén, ha az oktató egy adott feladatra kíváncsi, vagyis a kérés `task_id` paraméterrel rendelkezik, akkor a `task_id` által reprezentált feladat sorszámra született megoldásokat jelenítjük meg, ha nem definiált, akkor az összes megoldást összegyűjtjük.

```
  def show
    if @user
```

```
      if @user.student?
        @solutions=@user.solutions
      else
        if !params[:task_id].nil?
          task = Task.find_by_number params[:task_id]
          @solutions= task.solutions
        else
          @solutions = Array.new
          tasks = Task.find :all
          tasks.each do |task|
            task.solutions.each do |s|
              @solutions << s
            end
          end
        end
      end
    end
  end
```

A megoldások megjelenítésének megvalósítása előtt definiáljuk két helper metódust a `SolutionsHelper` modulban, amelyek a `late` és a `accepted` boolean attribútumokat transzformálják olvasható értékké.

```
module SolutionsHelper
  def accepted(a)
    if a
      "OK"
    else
      "!"
    end
  end

  def late(a)
    if a
      "x"
    else
      ""
    end
  end
end
```

A megoldások listájának megjelenítése nézetben szétválasztjuk az oktató,

illetve a hallgató típusú felhasználó számára szóló részeket. Az oktató számára a javítandó feladatok listája jelenik meg, a hallgató számára a beadott feladatok listája benne a feladat értékelésével, ha már van. Az oktató azon megoldások listáját látja a szerkesztés oldalra mutató linkekként, amelyre a megoldás még el nem fogadott állapotban van. A link követése esetén paraméterként átadódik a `edit` akciónak a feladat sorszáma és a felhasználó azonosítója.

```
<% if @user then if @user.student? %>
<h1>Beadott feladatok listája</h1>
<% else %>
<h1>Javítandó feladatok listája</h1>
<% if params[:task_id] %>
<h3><%= params[:task_id]%>. feladat</h3>
<% end %>
<% end end %>

<ul>
<% @solutions.each do |solution| %>
  <% neptun = solution.user.neptun %>
  <% if solution!=nil %>
  <% if @user then if @user.student? %>
<table border="1">
<tr><td>Elfogadva</td><td>Késés</td><td>Beadva</td><td>
   Megjegyzés</td></tr>
<% @solutions.each do |solution| %>
<tr><td><%= accepted solution.accepted? %></td><td><%=
   late solution.late? %></td><td><%= solution.
   updated_at.strftime "%Y.%m.%d %H:%M" %></td><td>
<%= solution.comment%></td></tr>
<% end %>
</table>
  <% else %>
  <% if !solution.accepted %>
  <p><%= link_to "#{neptun} által beadott #{solution.
      task.number}. feladat", :action=>"edit", :task_id
     =>solution.task.id, :user_id=>solution.user.id %><
     /p>
  <% end %>
  <% end end %>
  <% end %>
```

```
<% end %>
</ul>
```

A feladat szerkesztése nézet is két változattal rendelkezik a felhasználó típusától függően. A hallgató típus definíciójától most eltekintünk (lásd archívum). Oktató típusú felhasználó esetén a megoldás szerkesztése az értékelést jelenti. Feltételezzük, hogy a kéréssel paraméterként érkezik a feladat azonosítója és a kiválasztott megoldás beadójának azonosítója. A megoldás (@solution) e két kulcs alapján azonosítható az adatbázisban. Ha valamelyik paraméter hiányzik, visszairányítjuk a felhasználót a megoldások listája nézetre.

```
  def edit
    if params[:task_id].nil?
      redirect_to :action => "show"
    end
    if params[:user_id].nil?
      task = Task.find params[:task_id]
      redirect_to :action => "show", :task_id=>task.
          number
    end
    if @user
      if @user.student?
      else
        @solution=Solution.find_by_task_id_and_user_id
            params[:task_id], params[:user_id]
      end
    end
  end
```

A feladat javítása/módosítása nézetet is két megjelenésre bontjuk a felhasználó típusa alapján, most csak a javítás nézettel foglalkozunk. A nézet két részből áll. Egy a `download` akcióra mutató linkből, amelyen kereszül letölthető az aktuális megoldást, és egy formból, ahol lehetséges a feladathoz megjegyzés fűzése, és az elfogadottság, illetve a késés állapotának módosítása. A formot a `update` akció kezeli le, amelynek paraméterül átadjuk a feladat `id`-jét.

```
<% if @user then if @user.student? %>
<h1>Feladat módosítása</h1>
<% else %>
<h1>Feladat javítása</h1>
```

```
<%= link_to "A megoldás letöltése", :action => "
    download", :id => @solution.id %>
<%= form_for :solution, @solution, :url => {:action=>"
    update", :id=>@solution.id}, :method=>"post" do |f
    |%>
<table>
  <tr><td>Hallgató Neptun-kódja</td><td><%= @solution.
      user.neptun %></td></tr>
  <tr><td>A feladat beadási ideje</td><td><%= @solution
      .updated_at.strftime("%Y.%m.%d. %H:%M") %></td></
      tr>
  <tr><td>Késés</td><td><%= f.check_box :late, { :
      checked => (@solution.updated_at-@solution.task.
      deadline > 0)}%></td></tr>
  <tr><td>Elfogadva</td><td><%= f.check_box :accepted %
      ></td></tr>
  <tr><td>Megjegyzés</td><td><%= f.text_area :comment,
      :cols=>60, :rows=> 6%></td></tr>
  <tr><td><%= submit_tag "Mehet"%></td><td></td></tr>
</table>
<% end %>
<% end %>
<% end %>
```

A feladat értékelését a megoldás szerkesztése nézet formján keresztül teheti meg az oktató típusú felhasználó, annak eseményét a `update` metódus kezeli le. A metódul először a paraméterként kapott megoldás azonosító alapján kikeresi az adatbázisból a hivatkozott megoldás rekordot, majd frissíti annak attribútumait. Végül átirányítja a feljhasználót a megoldások listája nézetre.

```
  def update
    @solution = Solution.find params[:id]
    s = params[:solution]
    @solution.late = s[:late].to_i
    @solution.accepted = s[:accepted].to_i
    @solution.comment = s[:comment]
    @solution.save
    redirect_to :action=>"show",:task_id=>@solution.
        task.number
  end
```

Az értékelés során az oktató számára elérhetővé kell tenni a beadott megoldást. Ezt fájlként tesszük számára letölthetővé. Mivel a hallgatók általi feladatbeadás megvalósítása most nem célunk, a „beadott” megoldás fájl mindig ugyanaz lesz. A megoldást a `send_file` helperrel küldjük el a kliens böngészőnek.

```
def download
  file = "public/_index.html"
  send_file file
end
```