



ActionController és ActionView

Kovács Gábor

`kovacsg@tmit.bme.hu`

BME-TMIT

Action Pack

- ⑥ Feladata a HTTP kérések feldolgozása és a HTTP válaszok generálása
- ⑥ Két modul:
 - △ ActionController
 - △ ActionView

Routing – Alapkérdés

- ⑥ Kérdés: *Honnan tudja a Rails, hogy egy kérést melyik kontroller szolgáljon ki?*
 1. A Rails a kérés URL-be kódolja ezt az információt több más paraméterrel együtt. A konfiguráció a `config/routes.rb`-ben található hash-en keresztül történik, ahol a bejegyzések sorrendjében történik a mintaillesztés.
 2. A kontroller azonosítása után a Rails példányosítja a kontroller osztályát, majd meghívja annak `process` metódusát átadva annak a kérés paramétereit (`params` hash)
 3. A kontroller kiválasztja és meghívja az `action`-nek megfelelő akciót a kontroller osztályban, amely feldolgozza a kérést.

Routing – típusok

- ⑥ A Rails 3 kétféle routingot támogat
 - △ Az ún. resourceful utvonalak, ez az alapértelmezett, REST alapú útvonalakat generál

```
resources :tasks
```

- △ Kötött, vagyis nem resourceful útvonalak

```
match ':controller(/:action(/:id(.:format)))'
```

- ⑥ Most

- △ Csak kötöttről lesz szó, a resourceful a REST tárgyalásakor
- △ Útvonalak tesztelése később

Routing – kötött útvonalak 1

- ⑥ Kötött útvonalak: a HTTP kérés URL-jének közvetlen leképezése egy kontroller egy akciójára
 - △ A `:controller` és a `:action` megadása
 - △ Minden egyéb paraméter a `params` hash-be kerül
 - △ Például: a
`:controller(/:action(/:id(.:format)))`
 - △ Az `/users/show/1` jelentése: `users` kontroller, `show` akció, `params[:id]` értéke `1`

Routing – kötött útvonalak 2

- ⑥ Dinamikus útvonal részek, paraméterek:
 - △ Tetszőleges számú paraméter megadható
 - △ Például:
`:controller/:action/:user_id(/:task_id),`
`/users/show/1/2`
 - △ Paraméter az URL-be kódolva:
`/users/show/1?task_id=2`
- ⑥ Statikus útvonal részek:
 - △ Tetszőleges statikus útvonal szakasz megadható a leképezésben
 - △ Például: `:controller/:action/ut/ide/:id` és
`/users/show/ut/ide/1`

Routing – kötött útvonalak 3

6 Alapértelmezett útvonalak megadása

- △ A `:controller` és az `:action` elhagyása

- △ Például:

```
match 'users/:id' => 'users#show',
```

- △ Paraméterezés: `:defaults` hash elemei a `params` hash-be kerülnek

6 Útvonalak elnevezése, helperek

- △ Útvonal átnevezése

```
match 'index' => 'users#index', :as => :index
```

- △ Két helper metódus jön létre: `index_path` és `index_url`

- △ Ezek a kontrollerben átirányításra használhatók fel

Routing – kötött útvonalak 4

6 Megkötések, joker karakterek

- △ A `:constraints` opcióval korlátozható

△ Például

```
match 'users/:id' => 'users#show',  
:constraints {id=>/[a-zA-Z0-9]{6}/}
```

- △ A `*` karakter használható az útvonalakban, tetszőlegesen hosszú szakaszra illeszkedik

6 Átirányítás

- △ A `redirect` helperrel már az útvonalválasztáskor átirányítható egy kérés

- △ Például `match '/index' => redirect('/users')`

6 A gyökér

- △ `root :to => 'users#index'`

ActionController – A kontroller működése

- ⑥ Kérés feldolgozásának menete
 1. Az `:action`-nel megegyező nevű publikus példánymetódus keresése
 2. A `method_missing` metódus meghívása, ha van
 3. Az akcióval megegyező template keresése a nézetek között
 4. Egyébként hiba
- ⑥ Publikus akciók elfedése: `hide_action` metódus

ActionController – A kontroller környezete

- ⑥ `action_name`: az épp feldolgozott akció neve
- ⑥ `cookies`: egy hash a sütiket tárolására
- ⑥ `headers`: a válasz HTTP fejrész hash-e
- ⑥ `params`: a kérés paraméterek hash-e
- ⑥ `request`: a bejövő kérés objektuma
- ⑥ `response`: az előállítandó válasz objektuma
- ⑥ `session`: egy hash session adatok tárolására

ActionController – A bejövő kérés

- ⑥ A `request` objektum a következő attribútumokkal rendelkezik
 - △ `domain`: a kérés domén neve
 - △ `remote_ip`: a távoli gép IP címe
 - △ `env`: a böngésző által beállított HTTP fejrészek hash-e
 - △ `method`: a kérés HTTP parancsa, illetve `get?` stb. a HTTP parancs azonosítására

ActionController – Válasz a felhasználónak 1

- ⑥ A kontrollernek és az akciónak megfelelő nézet (az `app/views`-ből) megjelenítése a kontrollerben beállított információkkal. Ez a leggyakoribb.
- ⑥ A kontroller közvetlenül készíti el a választ a nézettel való együttműködés nélkül. Például hibaoldalak megjelenítése.
- ⑥ A kontroller nem csinál semmit. Például AJAX kérés kiszolgálásakor.
- ⑥ A kontroller egyéb adatot küld, például fájlt.

ActionController – Válasz a felhasználónak 2

- ⑥ Nézet megjelenítése: `render`
- ⑥ Ez az alapértelmezett hívás, ha az akció metódusa üres vagy nincs definiálva
- ⑥ Paraméterezhető:
 - △ Szöveg küldése a válaszban: `render(:text=>'Hello World')`
 - △ Beágyazott HTML küldése a válaszban:
`render(:inline=> %{\<h1>Hello World</h1>})`
 - △ Más akció által előállított válasz küldése: `render(:action=>'new')`
 - △ Más elérési úttal megadott nézet küldése:
`render(:file=>'app/views/solutions/new.html.erb')`
 - △ Más nézet küldése: `render(:template=>'solutions/new')`
 - △ Üres törzs küldése: `render(:nothing)`

ActionController – Válasz a felhasználónak

⑥ Fájl küldése `send_data`, `send_file`

```
send_data(pic, :type=>'image/png', :disposition=>'inline')  
send_file('/public/data/oweyoa_2.pdf', :type=>'application/pdf')
```

⑥ Állítható opciók:

- △ A `Content-type` a `:type` kulccsal
 - △ A böngésző viselkedése: megjelenítés (`inline`) vagy az alapértelmezett mentés (`attachment`)
 - △ Az elmentendő fájl neve: `:filename`
- ⑥ A `Content-type` és a többi HTTP válasz fejrész a `controller headers hash`-én keresztül is állítható

ActionController – Válasz a felhasználónak

- ⑥ Átirányítás: `redirect_to`
- ⑥ Hasonlít a `render :action-re`, azonban itt a kérés és a válasz URL különbözik
- ⑥ Paraméterek:
 - △ Átirányítás másik akcióra:
`redirect_to :action=>:show`
 - △ Átirányítás másik nézetre:
`redirect_to :file=>'public/404.html'`
 - △ Átirányítás az előző oldalra: `redirect_to :back`

ActionController – Sütik

- ⑥ A HTTP állapotmentes, a memóriát a szerver által a böngészőben elhelyezett sütik adják
- ⑥ A sütiket a `cookies` hash tárolja, amely kizárólag string típusú adatokat tartalmaz
- ⑥ Süti opciók
 - △ A `:value` hordozza a süti értékét
 - △ A `:domain` és a `:path` az elérési úttal korlátozza a süti érvényességi körét
 - △ Az `:expires` élettartamot rendel a sütihez
`:expires => 7.days.from_now`
 - △ A `:secure` HTTPS kérésekre korlátozza a sütit
- ⑥ A süti letiltható böngészőben

ActionController – Sessionök

- ⑥ `session` egy összetett adatstruktúrák tárolására alkalmas hash, amely kérések között állandó marad
- ⑥ A sessiont egy `_session_id` azonosítja, amit sütiben küld át a Rails a kliensnek
- ⑥ A session túléli a verziófrissítést!
- ⑥ Session törlése: a `reset_session` metódussal például kilépéskor
- ⑥ Session opciók: a session is korlátozható
 - :`session_domain-re`, `session_path-ra` és
 - :`session_secure-ra`

ActionController – Session-ök tárolása

- ⑥ Stratégiák a `session_store` beállítására
- ⑥ A `session` hash tárolható
 - △ szerializált formában a fájlrendszeren a `tmp` könyvtárban. Ez (`CHI::Session::PStore`) az alapértelmezés
 - △ ActiveRecord-ként (`:active_records_store`) az adatbázisban egy `sessions` táblában
 - △ a hálózaton több szerver között elosztva (`:drb_store`)
 - △ memóriában (`:memory_store`), ez nem éli túl az alkalmazás újraindítását
- ⑥ A session-ök takarítására a `session` sütihez kötött `:expires` opcióval gondoskodhatunk

ActionController – Flash

- ⑥ Átirányítás (`redirect_to`) használata esetén új HTTP kérés jön létre, a korábban beállított példányváltozók értékei elvesznek
- ⑥ A `flash` hash egy temporális tárhely a következő kérés számára
- ⑥ Leggyakoribb alkalmazása üzenetek küldése a következő oldalra
- ⑥ Például:

```
flash[:notice]='Sikeres bejelentkezés'
```

ActionController – Szűrők

- ⑥ Több akció során előforduló kódrészleteket privát metódusokkal és szűrőkkel valósíthatunk meg
- ⑥ Típusok: `before_filter`, `after_filter`, `around_filter`
- ⑥ Az előszűrő a példányváltozók inicializálására alkalmas
- ⑥ Az utószűrő a generált `response` módosítására, például tömörítésére alkalmas
- ⑥ Az `around` szűrő, ha tartalmaz `yield`-et, ami átadja a vezérlést a meghívott akciónak, akkor úgy viselkedik, mint egy elő- és utószűrő a `yield` előtti és utáni kódrészletet tekintve

ActionController – Gyorsítótár

- ⑥ Éles (production környezet) rendszerekben érdemes használni gyakori kérések gyorsabb kiszolgálására
- ⑥ Egész oldal (`cache_page`) vagy egy-egy akció (`cache_action`) cache-elhető
- ⑥ Mit érdemes? Olyan oldalakat, amiket a Rails alkalmazás kontrollál. Idő, session, és adatbázison kívüli adatokat tartalmazó oldalakat nem.
- ⑥ A gyorsítótár törölhető a `expire_page` és `expire_action` metódusokkal

ActionView – Oldal template-ek

- ⑥ Az oldal template-eket a Rails nézetek könyvtárában keresi
- ⑥ Milyen adatokhoz férnek hozzá a nézetek?
 - △ A kontroller példányváltozóit
 - △ A kontroller objektumait (a flash kivételével csak debug célra)
 - △ Az aktuális kontroller
 - △ Az elérési út `base_path`
- ⑥ Az oldal template-ek beágyazott Ruby kódot tartalmazó HTML, XML és JavaScript források

ActionView – Ruby kód beágyazása

- ⑥ Scriptletekkel
- ⑥ Ruby kifejezés beágyazása: `<%= Time.now %>`
- ⑥ Ruby kódrészlet beágyazása: `<% %>`

ActionView – Helperek

- ⑥ HTML escape: `<%= h(params[:user]) %>`
- ⑥ Kód helyett helper metódusok használata
 - △ A nézet elkülönítése a logikától
 - △ A beágyazott kód külön is tesztelhető
 - △ Az arculattervező csak a HTML kódot látja

ActionView – URL helperek

6 link_to: link létrehozása

```
<%= link_to user.neptun, :action=>"show", :id=>@user.id %>
<a href="/users/show/1">oweyoa</a>
```

6 button_to: navigációs gomb létrehozása, ami egy HTML form-ot hoz létre

```
<%= button_to "Töröl", { :action=>"destroy", :id=>@user.id},
  :confirm=>"Biztos?", :method=>"delete" %>
<form method="post" action="/users/destroy/1">
...
<input data-confirm="Biztos?" type="submit" value="Töröl" />
...
</form>
```

6 mail_to

```
<%= mail_to @user.email %>
<a href="mailto:kovacs@tmit.bme.hu">kovacs@tmit.bme.hu</a>
```

ActionView – Formázás helperek

- ⑥ Olvashatóbbá tesznek dátum, szám és szöveg típusú adatokat
- ⑥ Dátum: `ActionView::Helpers::DateHelper` modul
- ⑥ Szám: `ActionView::Helpers::NumberHelper` modul
- ⑥ Szöveg: `ActionView::Helpers::TextHelper` modul

Action View – Oldaltörés

- 6 A betördelt felhasználók és a töredékoldalak listáinak előállítása

```
def user_list
  @user_pages, @users = paginate(:users, :order => 'name' )
end
```

- 6 A betördelt felhasználók megjelenítése

```
<table>
<tr><th>Név</th></tr>
<% for user in @users %>
<tr><td><%= user.name %></td></tr>
<% end %>
</table>
```

- 6 A navigációs linkek megjelenítése

```
<hr><%= pagination_links(@user_pages) %><hr>
```

ActionView – Formok

users_controller.rb

```
def edit
  @user=User.find param[:id]
end
```

edit.html.erb

```
<% form_for :user, :url => { :action => "update", :id => @user.id } do |form| %>
<div>
  <%= form.label :neptun %><br />
  <%= form.text_field :neptun, :size => 35 %>
</div>
...
<% end %>
```

```
<form accept-charset="UTF-8" action="/users/update/1" method="post">
...
<div>
  <label for="user_neptun">Neptun</label><br />
  <input id="user_neptun" name="user[neptun]" size="35" type="text" value="oweyoa" />
</div>
...
</form>
```

```
@params={
  :id=>1,
  :user=>{
    neptun=>"oweyoa"
  }
}
```

users_controller.rb

```
def update
  @user=User.find param[:id]
  user.update_attributes(params[:user])
end
```

ActionView – Form helper

- ⑥ Form létrehozása: `form_for`
- ⑥ A formon belül a form iterátorának helpereivel adhatók hozzá a beviteli mezők

```
<% form_for :user,  
  :url => { :action => :create, :id => @user.id },  
  :html => { :class => "my_form" , :method => :put } do |form| %>
```

- ⑥ Ha a form nem kötődik egy modellhez sem

```
<% form_tag({ :action => :save }, { :class => "compact" }) do ...%>
```

ActionView – Form mezők

6 Szövegbeviteli mezők:

```
form.text_field :neptun, :size=>"6"  
form.hidden_field :secret, ...  
form.password_field :password, :maxlength=>"10"
```

6 Szövegdoboz:

```
form.text_area :text, :cols=>"10", :rows=>"10"
```

6 Választógomb:

```
form.radio_button :igaz, true, ...
```

6 Jelölőnégyzet:

```
form.check_box :igaz, ...
```

ActionView – Form mezők

⑥ Legördülő lista: a lista egy String tömb

```
form.select :valaszt, lista, ...
```

⑥ Dátum és Időpont választó mezők:

```
form.date_select :created_on, :order => [:year, :month, :day]
form.datetime_select :created_on
select_date(date=Date.today)
select_year(date)
select_minute(date=Time.now)
```

⑥ Nem modell mezők

Action View – Layout

- ⑥ Az alkalmazás elrendezését a `app/views/layouts/application.html.erb` adja meg
- ⑥ A kontrollere a `app/controllers/application_controller.rb`
- ⑥ Beágyazás `yield`-del vagy részleges template-ekkel lehetséges

```
<%= render(:partial => "welcome" , :object => @user) %>
```