

A Rails keretrendszer Gyakorlat

Kovács Gábor

2011. március 1.

A Rails keretrendszert egy virtuális környezetben ¹ futó Debian Linuxra ² telepítjük a gyakorlat folyamán. Ez a dokumentáció kiindulási pontként egy ilyen, grafikus felhasználói környezetet is tartalmazó előtelepített operációs rendszert feltételez.

A Rails környezetünk back-endje hosszú távon egy MySQL adatbázis kezelő lesz. Ezt az `apt-get install mysql-server` parancs konzolon történő kiadásával telepíthetjük, a telepítő többször rá fog kérdezni a root felhasználó jelszavára, azt tetszés szerinti értékre állíthatjuk, azonban jegyezzük meg, mert a Rails adatbázis konfigurációs fájljában ezt meg kell adnunk. A gyakorlatra telepített környezeten az egyszerűség kedvéért üres jelszót fogunk használni.

A Rails adatbázis adapterei fordításához szükségünk van egy C fordítóra és fejlesztői függvénykönyvtárakra, amelyeket az `apt-get install gcc` parancs konzolon való kiadásával telepíthetünk.

A Rails környezet telepítéséhez egy Ruby környezet szükséges, ami mellé kiegészítőként telepítjük az RI dokumentációs rendszert és a natív kiegészítések fordításához szükséges Ruby fejlesztői függvénykönyvtárakat. Ezt a következő paranccsal telepíthetjük rendszergazdaként:

```
kovacsg@debian:~$ sudo bash
[sudo] password for kovacsg:
root@debian:~/home/kovacsg# apt-get install ruby1.9.1
  ri1.9.1 ruby1.9.1-dev
Reading package lists ... Done
```

¹Oracle VirtualBox (<http://www.virtualbox.org>) az otthonra javasolt virtualizációs eszköz, viszont használható a vmware, illetve a parallels is

²Mivel az Ubuntu Linux csomagkezelője megegyezik a Debian Linux disztribúció csomagkezelőjével, a dokumentumban felsorolt csomagnevek egy az egyben átvehetők. Más disztribúciók esetén a csomagok nevei különbözhetnek.

```

Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libruby1.9.1 libyaml-0-2
Suggested packages:
  ruby1.9.1-examples graphviz
The following NEW packages will be installed:
  libruby1.9.1 libyaml-0-2 ri1.9.1 ruby1.9.1 ruby1.9.1-dev
0 upgraded, 5 newly installed, 0 to remove and 333 not
upgraded.
Need to get 0 B/9,270 kB of archives.
After this operation, 64.9 MB of additional disk space
will be used.
Do you want to continue [Y/n]?
Selecting previously deselected package libyaml-0-2.
(Reading database ... 71754 files and directories
currently installed.)
Unpacking libyaml-0-2 (from .../libyaml-0-2_0.1.3-1
_amd64.deb) ...
Selecting previously deselected package libruby1.9.1.
Unpacking libruby1.9.1 (from .../libruby1.9.1_1.9.2.0-2
_amd64.deb) ...
Selecting previously deselected package ruby1.9.1.
Unpacking ruby1.9.1 (from .../ruby1.9.1_1.9.2.0-2_amd64
.deb) ...
Selecting previously deselected package ri1.9.1.
Unpacking ri1.9.1 (from .../ri1.9.1_1.9.2.0-2_all.deb)
...
Selecting previously deselected package ruby1.9.1-dev.
Unpacking ruby1.9.1-dev (from .../ruby1.9.1-dev_1
.9.2.0-2_amd64.deb) ...
Processing triggers for man-db ...
Setting up libyaml-0-2 (0.1.3-1) ...
Setting up libruby1.9.1 (1.9.2.0-2) ...
Setting up ruby1.9.1 (1.9.2.0-2) ...
Setting up ri1.9.1 (1.9.2.0-2) ...
Setting up ruby1.9.1-dev (1.9.2.0-2) ...

```

Ezután ellenőrizzük, hogy elérhető-e a ruby értelmező és a gem Ruby csomagkezelő.

```
root@debian:~/home/kovacs# ruby --version
ruby 1.9.2p0 (2010-08-18 revision 29036) [x86_64-linux]
root@debian:~/home/kovacs# gem --version
1.3.7
```

A következő lépés a Rails keretrendszer telepítése, amit a `gem` Ruby csomagkezelővel teszünk meg. Ez egyben feltelepíti a legfontosabb kiegészítőket is, illetve a hozzájuk kapcsolódó dokumentációt.

```
root@debian:~/home/kovacs# gem install rails
Successfully installed activesupport-3.0.5
Successfully installed builder-2.1.2
Successfully installed i18n-0.5.0
Successfully installed activemodel-3.0.5
Successfully installed rack-1.2.1
Successfully installed rack-test-0.5.7
Successfully installed rack-mount-0.6.13
Successfully installed tzinfo-0.3.24
Successfully installed abstract-1.0.0
Successfully installed erubis-2.6.6
Successfully installed actionpack-3.0.5
Successfully installed arel-2.0.9
Successfully installed activerecord-3.0.5
Successfully installed activerecord-3.0.5
Successfully installed mime-types-1.16
Successfully installed polyglot-0.3.1
Successfully installed treetop-1.4.9
Successfully installed mail-2.2.15
Successfully installed actionmailer-3.0.5
Successfully installed rake-0.8.7
Successfully installed thor-0.14.6
Successfully installed railties-3.0.5
Successfully installed bundler-1.0.10
Successfully installed rails-3.0.5
24 gems installed
Installing ri documentation for activesupport-3.0.5...
Installing ri documentation for builder-2.1.2...
Installing ri documentation for i18n-0.5.0...
Installing ri documentation for activemodel-3.0.5...
Installing ri documentation for rack-1.2.1...
Installing ri documentation for rack-test-0.5.7...
```

```
Installing ri documentation for rack-mount-0.6.13...
Installing ri documentation for tzinfo-0.3.24...
Installing ri documentation for abstract-1.0.0...
Installing ri documentation for erubis-2.6.6...
Installing ri documentation for actionpack-3.0.5...
Installing ri documentation for arel-2.0.9...
Installing ri documentation for activerecord-3.0.5...
Installing ri documentation for activeresource-3.0.5...
Installing ri documentation for mime-types-1.16...
Installing ri documentation for polyglot-0.3.1...
Installing ri documentation for treetop-1.4.9...
Installing ri documentation for mail-2.2.15...
Installing ri documentation for actionmailer-3.0.5...
Installing ri documentation for rake-0.8.7...
Installing ri documentation for thor-0.14.6...
Installing ri documentation for railties-3.0.5...
Installing ri documentation for bundler-1.0.10...
Installing ri documentation for rails-3.0.5...
Installing RDoc documentation for activesupport
-3.0.5...
Installing RDoc documentation for builder-2.1.2...
Installing RDoc documentation for i18n-0.5.0...
Installing RDoc documentation for activemodel-3.0.5...
Installing RDoc documentation for rack-1.2.1...
Installing RDoc documentation for rack-test-0.5.7...
Installing RDoc documentation for rack-mount-0.6.13...
Installing RDoc documentation for tzinfo-0.3.24...
Installing RDoc documentation for abstract-1.0.0...
Installing RDoc documentation for erubis-2.6.6...
Installing RDoc documentation for actionpack-3.0.5...
Installing RDoc documentation for arel-2.0.9...
Installing RDoc documentation for activerecord-3.0.5...
Installing RDoc documentation for activeresource
-3.0.5...
Installing RDoc documentation for mime-types-1.16...
Installing RDoc documentation for polyglot-0.3.1...
Installing RDoc documentation for treetop-1.4.9...
Installing RDoc documentation for mail-2.2.15...
Installing RDoc documentation for actionmailer-3.0.5...
Installing RDoc documentation for rake-0.8.7...
Installing RDoc documentation for thor-0.14.6...
```

```
Installing RDoc documentation for railties -3.0.5...
Installing RDoc documentation for bundler -1.0.10...
Installing RDoc documentation for rails -3.0.5...
```

A következő lépésben az adatbáziskezelő Ruby illesztőjét telepítjük. Az adatbázis adapterek natív kiterjesztései fordításához két további fejlesztői függvénykönyvtárat kell telepítenünk az operációs rendszer csomagkezelőjével, a `libsqlite3-dev` és a `libmysqlclient-dev` nevű csomagot ³:

```
root@debian:~/home/kovacs# apt-get install libsqlite3-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  sqlite3-doc
The following NEW packages will be installed:
  libsqlite3-dev
0 upgraded, 1 newly installed, 0 to remove and 333 not
upgraded.
Need to get 0 B/508 kB of archives.
After this operation, 1,253 kB of additional disk space
will be used.
Selecting previously deselected package libsqlite3-dev.
(Reading database ... 83400 files and directories
currently installed.)
Unpacking libsqlite3-dev (from .../libsqlite3-dev_3
.7.4-2_amd64.deb) ...
Setting up libsqlite3-dev (3.7.4-2) ...
root@debian:~/home/kovacs# apt-get install
libmysqlclient-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libmysqlclient16 mysql-common zlib1g-dev
The following NEW packages will be installed:
  libmysqlclient-dev zlib1g-dev
The following packages will be upgraded:
```

³Más Linux disztribúciókon ezek neve eltérhet. Windows-on elég az előre lefordított DLL-t elhelyezni az elérési úton

```

libmysqlclient16 mysql-common
2 upgraded, 2 newly installed, 0 to remove and 331 not
  upgraded.
Need to get 5,499 kB of archives.
After this operation, 10.3 MB of additional disk space
  will be used.
Do you want to continue [Y/n]?
Get:1 http://ftp.hu.debian.org/debian/ sid/main mysql-
  common all 5.1.55-1 [90.2 kB]
Get:2 http://ftp.hu.debian.org/debian/ sid/main
  libmysqlclient16 amd64 5.1.55-1 [1,994 kB]
Get:3 http://ftp.hu.debian.org/debian/ sid/main zlib1g-
  dev amd64 1:1.2.3.4.dfsg-3 [192 kB]
Get:4 http://ftp.hu.debian.org/debian/ sid/main
  libmysqlclient-dev amd64 5.1.55-1 [3,223 kB]
Fetched 5,499 kB in 2s (2,543 kB/s)
Reading changelogs... Done
(Reading database ... 83412 files and directories
  currently installed.)
Preparing to replace mysql-common 5.1.49-3 (using .../
  mysql-common_5.1.55-1_all.deb) ...
Unpacking replacement mysql-common ...
Preparing to replace libmysqlclient16 5.1.49-3 (using
  .../libmysqlclient16_5.1.55-1_amd64.deb) ...
Unpacking replacement libmysqlclient16 ...
Selecting previously deselected package zlib1g-dev.
Unpacking zlib1g-dev (from .../zlib1g-dev_1%3a1.2.3.4.
  dfsg-3_amd64.deb) ...
Selecting previously deselected package libmysqlclient-
  dev.
Unpacking libmysqlclient-dev (from .../libmysqlclient-
  dev_5.1.55-1_amd64.deb) ...
Processing triggers for man-db ...
Setting up mysql-common (5.1.55-1) ...
Setting up libmysqlclient16 (5.1.55-1) ...
Setting up zlib1g-dev (1:1.2.3.4.dfsg-3) ...
Setting up libmysqlclient-dev (5.1.55-1) ...

```

Ezután telepíthetjük először az SQLite, majd a MySQL adapterét:

```

root@debian:~/home/kovacs# gem install sqlite3
Building native extensions. This could take a while...

```

```

Successfully installed sqlite3-ruby-1.3.3
1 gem installed
Installing ri documentation for sqlite3-ruby-1.3.3...
Installing RDoc documentation for sqlite3-ruby-1.3.3...
root@vb:/home/kovacsg# gem install mysql2
Building native extensions. This could take a while...
Successfully installed mysql2-0.2.6
1 gem installed
Installing ri documentation for mysql2-0.2.6...
Enclosing class/module 'mMysql2' for class Client not
  known
Enclosing class/module 'mMysql2' for class Result not
  known
Installing RDoc documentation for mysql2-0.2.6...
Enclosing class/module 'mMysql2' for class Client not
  known
Enclosing class/module 'mMysql2' for class Result not
  known

```

A Rails környezetünket létrehoztuk, a következő lépés annak ellenőrzése, hogy a rails parancs elérhető-e. Ha nem, akkor adjuk hozzá a PATH környezeti változóhoz, majd ismételjük meg az ellenőrzést.

```

kovacsg@debian:~$ rails -v
bash: rails: command not found
kovacsg@debian:~$ export PATH=$PATH:/var/lib/gems
  /1.9.1/bin
kovacsg@debian:~$ rails -v
Rails 3.0.5

```

A `-d` kapcsoló különös jelentőséggel bír számunkra, ezzel adhatjuk meg a használni kívánt adatbáziskezelő típusát, ami lehet `mysql`, `oracle`, `postgresql`, `sqlite3`, `frontbase` vagy `ibm_db`. A gyakorlatok keretében ezek közül a `sqlite`-ot mint alapértelmezett adatbáziskezelőt és a `mysql`-t fogjuk használni.

```

kovacsg@debian:~$ rails
Usage:
  rails new APP_PATH [options]

Options:
  -r, [--ruby=PATH]          # Path to the Ruby binary
                             of your choice

```

```

# Default: /usr/bin/ruby1
# .9.1
-d, [--database=DATABASE] # Preconfigure for
selected database (options: mysql/oracle/
postgresql/sqlite3/frontbase/ibm_db)
# Default: sqlite3
-b, [--builder=BUILDER] # Path to an application
builder (can be a filesystem path or URL)
-m, [--template=TEMPLATE] # Path to an application
template (can be a filesystem path or URL)
[--dev] # Setup the application
with Gemfile pointing to your Rails checkout
[--edge] # Setup the application
with Gemfile pointing to Rails repository
[--skip-gemfile] # Don't create a Gemfile
-O, [--skip-active-record] # Skip Active Record
files
-T, [--skip-test-unit] # Skip Test::Unit files
-J, [--skip-prototype] # Skip Prototype files
-G, [--skip-git] # Skip Git ignores and
keeps

Runtime options:
-f, [--force] # Overwrite files that already exist
-p, [--pretend] # Run but do not make any changes
-q, [--quiet] # Suppress status output
-s, [--skip] # Skip files that already exist

Rails options:
-v, [--version] # Show Rails version number and quit
-h, [--help] # Show this help message and quit

Description:
The 'rails new' command creates a new Rails
application with a default
directory structure and configuration at the path
you specify.

Example:
rails new ~/Code/Ruby/weblog

```


This generates a skeletal Rails installation in `~/Code/Ruby/weblog`.
See the README in the newly created application to get going.

A Rails keretrendszerben a `rails` parancs az az univerzális eszköz mellyel többek között új komponenseket hozhatunk létre és elindíthatjuk a beépített webservert. Egy új Rails alkalmazást a `rails` parancsnak `new` opciót megadva hozhatunk létre, a második argumentum az alkalmazás neve. A paracsot kiadva a konzolon láthatjuk az automatikusan generált fájlokat. Az egyes könyvtárak értelmezéséről előadáson esett szó, e gyakorlat keretében megnézzük az egyes fájlok szerepét és tartalmát.

```
kovacsg@debian:~$ rails new gyakorlat
create
create  README
create  Rakefile
create  config.ru
create  .gitignore
create  Gemfile
create  app
create  app/controllers/application_controller.rb
create  app/helpers/application_helper.rb
create  app/mailers
create  app/models
create  app/views/layouts/application.html.erb
create  config
create  config/routes.rb
create  config/application.rb
create  config/environment.rb
create  config/environments
create  config/environments/development.rb
create  config/environments/production.rb
create  config/environments/test.rb
create  config/initializers
create  config/initializers/backtrace_silencers.
      rb
create  config/initializers/inflections.rb
create  config/initializers/mime_types.rb
create  config/initializers/secret_token.rb
create  config/initializers/session_store.rb
create  config/locales
```

```
create config/locales/en.yml
create config/boot.rb
create config/database.yml
create db
create db/seeds.rb
create doc
create doc/README_FOR_APP
create lib
create lib/tasks
create lib/tasks/.gitkeep
create log
create log/server.log
create log/production.log
create log/development.log
create log/test.log
create public
create public/404.html
create public/422.html
create public/500.html
create public/favicon.ico
create public/index.html
create public/robots.txt
create public/images
create public/images/rails.png
create public/stylesheets
create public/stylesheets/.gitkeep
create public/javascripts
create public/javascripts/application.js
create public/javascripts/controls.js
create public/javascripts/dragdrop.js
create public/javascripts/effects.js
create public/javascripts/prototype.js
create public/javascripts/rails.js
create script
create script/rails
create test
create test/fixtures
create test/functional
create test/integration
create test/performance/browsing_test.rb
create test/test_helper.rb
```

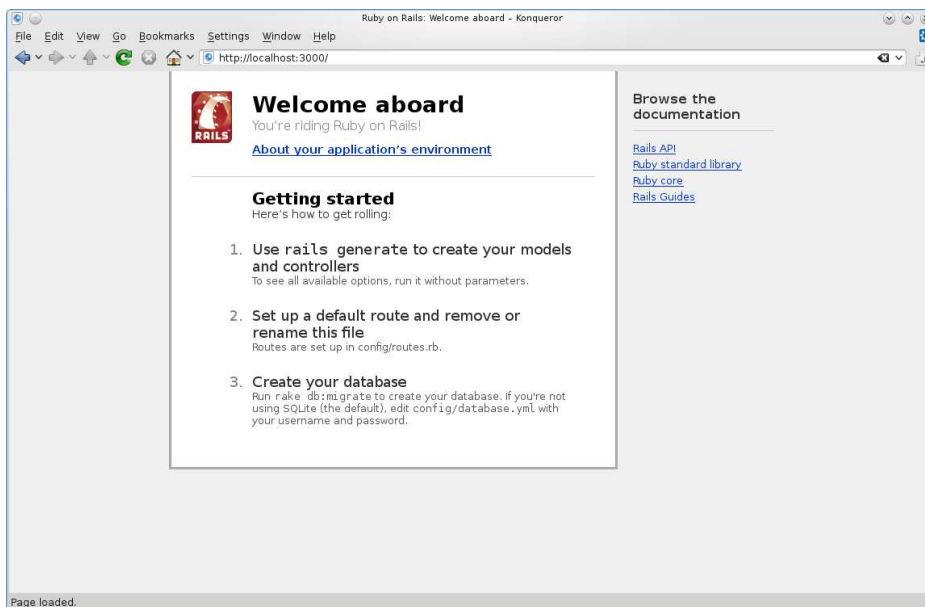
```
create test/unit
create tmp
create tmp/sessions
create tmp/sockets
create tmp/cache
create tmp/pids
create vendor/plugins
create vendor/plugins/.gitkeep
```

Az `app` könyvtár fogja tartalmazni az általunk készített Ruby és beágyazott Ruby kódot tartalmazó HTML forrásokat, amelyeket a MVC minta alapján struktúrál a Rails, amelyeket a `test` könyvtárban elhelyezett teszt osztályokkal ellenőrizzük. A `config` könyvtár a Rails alkalmazásunk konfigurációs beállításait tartalmazza. A `db` könyvtár az aktuális adatbázis sémát, az összes eddig adatbázis séma migrációt és `sqlite` adatbáziskezelő esetén szeriazált formában magát az adatbázis tartalmazza. A `doc` a dokumentációk gyűjtőhelye, a `lib` és `vendor` könyvtárak mások által készített Ruby, illetve Rails függvénykönyvtárakat tartalmazhatnak. A `public` könyvtár a beépített webservert területe, az összes ott bekövetkezett esemény a `log` könyvtárban található az aktuális Rails környezetnek megfelelő állományban kerülnek naplózásra, a webservert `tmp` könyvtárban helyezheti el az átmeneti fájljait, mint például `session` azonosítókat, sütiket. A `script` könyvtár elérhetővé teszi számunkra a `rails` parancsot. A `Gemfile` és `Rakefile` a Rails alkalmazásunk által használt Ruby függvénykönyvtárakat specifikálja, amelyek vagy a telepített Ruby és Rails környezeten vagy a már említett `lib` és `vendor` könyvtárakban érhetők el.

A létrehozott `gyakorlat` nevű Rails alkalmazás könyvtárába belépve a konzolon kiadott `bundle install` paranccsal telepíthetjük az összes függőséget, az esetlegesen hiányzó vagy nem megfelelő verziójú Ruby függvénykönyvtárat.

A beépített webservert a `rails server` opciójával indíthatjuk el a Rails alkalmazásunk gyökérkönyvtárából. Ha a szerveret háttéralkalmazásként kívánjuk futtatni, akkor a `-d` kapcsolót is hozzá kell fűznünk a parancsához. Ezután a szerver elérhető a `http://localhost:3000` webcímen, lásd 1. ábra.

```
kovacsg@debian:/home/kovacsg/gyakorlat# rails server
=> Booting WEBrick
=> Rails 3.0.5 application starting in development on
    http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
[2011-03-03 10:45:13] INFO WEBrick 1.3.1
```



1. ábra. A beágyazott webszerver elérése

```
[2011-03-03 10:45:13] INFO  ruby 1.9.2 (2010-08-18) [
  x86_64-linux ]
[2011-03-03 10:45:13] INFO  WEBrick::HTTPServer#start:
  pid=17130 port=3000
^C[2011-03-03 10:45:42] INFO  going to shutdown ...
[2011-03-03 10:45:42] INFO  WEBrick::HTTPServer#start
  done.
Exiting
kovacsg@debian:/home/kovacsg/gyakorlat# rails server -d
=> Booting WEBrick
=> Rails 3.0.5 application starting in development on
  http://0.0.0.0:3000
```

Két fontos konfigurációs fájl érdemel most említést: a adatbázis hozzáférés konfigurációjára szolgáló `database.yml` és az URL leképezéseket definiáló `routes.rb`.

Mivel a Rails alkalmazás létrehozásakor nem adtunk meg, hogy melyik adatbáziskezelőt kívánjuk használni, ezért az alapértelmezett SQLite konfigurációs paraméterei jelennek meg a `database.yml`-ben mindhárom definiált környezethez, vagyis a fejlesztési (`development`), tesztelési (`test`) és éles (`production`). Az SQLite az adatait fájlokban tárolja, amelyek a Rails alkalmazás db könyvtárában találhatóak, és a környezet nevével azonosítottak

e fájl szerint.

```
# SQLite version 3.x
# gem install sqlite3-ruby (not necessary on OS X
# Leopard)
development:
  adapter: sqlite3
  database: db/development.sqlite3
  pool: 5
  timeout: 5000

# Warning: The database defined as "test" will be
# erased and
# re-generated from your development database when you
# run "rake".
# Do not set this db to the same as development or
# production.
test:
  adapter: sqlite3
  database: db/test.sqlite3
  pool: 5
  timeout: 5000

production:
  adapter: sqlite3
  database: db/production.sqlite3
  pool: 5
  timeout: 5000
```

A másik leírónk azt adja meg, hogy milyen struktúrájú legyen az URL, amivel elérjük a Rails alkalmazásunk egyes funkcióit. Az alábbi kódrészlet a legáltalánosabb beállítást tartalmazza. A webservert IP címe után a Controller osztály neve (:controller), majd a Controller osztály egy metódusa (:action), majd egy adatbázis azonosító (:id), és végül formázási útmutató következik. Az utolsó három megadása opcionális.

```
Feladat::Application.routes.draw do
  match ':controller(/:action(/:id(.:format)))'
end
```

Nézzük meg, hogy miként tudunk dinamikus tartalmat létrehozni Rails-szel. A példák a [1] könyvből valók.

Hozzunk létre egy új controllert a rails parancs `generate` opciójával. A második argumentum (`controller`) azt mondja meg, hogy egy új controllert hozunk létre, a harmadik a controller nevét. A negyedik és minden további paraméter a controllerben definiál akciókat. E parancs négy Ruby forrásfájlt és egy könyvárat hoz létre az akcióknak megfelelő weboldalak, view-k számára. A controller nevének megfelelő controller osztályt (`say_controller.rb`), helper osztályt, illetve ezek funkcionális és egységtesztjéhez használható osztályokat.

```
kovacs@debian:~/gyakorlat/config$ rails generate
controller say
  create  app/controllers/say_controller.rb
  invoke erb
  create  app/views/say
  invoke test_unit
  create  test/functional/say_controller_test.rb
  invoke helper
  create  app/helpers/say_helper.rb
  invoke test_unit
  create  test/unit/helpers/say_helper_test.rb
```

Bármilyen tartalom megjelenítéséhez a `routes.rb` alapján a view könyvtárban kell elhelyeznünk az akciónak megfelelő néven egy beágyazott Ruby kódot tartalmazó HTML fájlt (`.rhtml` vagy `.html.erb`).

Az alkalmazás keretét az `app/views/layouts/application.html.erb` fájl definiálja, amely a HTML dokumentum törzs helyén egy beágyazott `yield` parancsot tartalmaz, amely átadja a vezérlést az akció HTML-ének, ami jelen esetben `hello` lesz.

Ezalapján nézzük meg a szokásos Hello, world alkalmazást ezúttal Railsben. Az `app/views/say` könyvtárban létrehozunk egy `hello.html.erb` nevű view-t, amely a `say` controller `hello` akciójához kötődik. Az eredményt a `http://localhost:3000/say/hello` linken ellenőrizhetjük.

```
<h1>Hello , world!</h1>
```

Ez dinamikussá tehetjük az aktuális idő kiírásával.

```
<h1>Hello , world!</h1>
<%= Time.now %>
```

Mivel a View-ba nem illik logikát rakni, csak a megjelenítendő értéket, ezért áttesszük az idő lekérdezését a Controllerbe, annak is az akciónak megfelelő metódusába, a `hello`-ba

```

class SayController < ApplicationController
  def hello
    @time=Time.now
  end
end

```

a View-ban hivatkozhatunk a Controller példányváltozóira.

```

<h1>Hello , world!</h1>
<%= @time %>

```

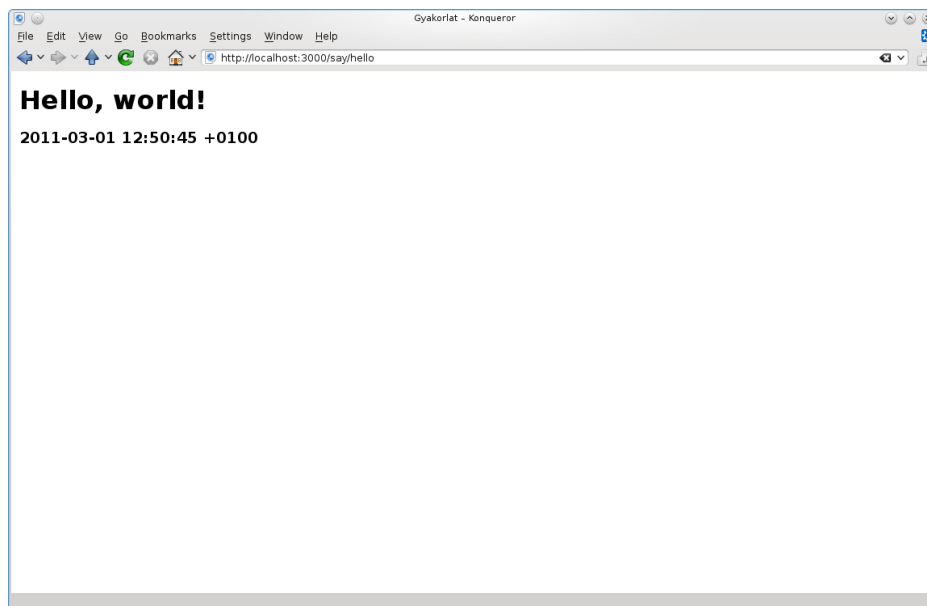
Az oldalak akciói átadhatják egymásnak a vezérlést, ehhez linkeket definiálunk a `link_to` metódussal. Az alábbi példában a link neve "Hello, world" lesz, a kapcsolódó akció pedig `helloworld` névre hallgat. Ez utóbbihoz definiálnunk kell egy új metódust a Controller osztályban, és egy a View-t megvalósító HTML oldalt.

```

<h1><%= link_to "Hello , _world" , :action => "helloworld"
%></h1>
<%= @time %>

```

A controllernek és akciónak megfelelő megjelenített oldalt a 2. ábra mutatja.



2. ábra. A hello akció megjelenítve

Az oldal forrását megnézve felismerjük benne a layout által nyújtott keretet és a View beágyazott kódját.

```
<!DOCTYPE html>
<html>
<head>
  <title>Gyakorlat</title>

  <script src="/javascripts/prototype.js?1298978910" type="text/javascript">
  </script>
<script src="/javascripts/effects.js?1298978910" type="text/javascript"></
  script>
<script src="/javascripts/dragdrop.js?1298978910" type="text/javascript"></
  script>
<script src="/javascripts/controls.js?1298978910" type="text/javascript"></
  script>
<script src="/javascripts/rails.js?1298978910" type="text/javascript"></
  script>
<script src="/javascripts/application.js?1298978910" type="text/javascript">
  </script>
  <meta name="csrf-param" content="authenticity_token"/>
<meta name="csrf-token" content="VJ6NL7G9b88lsQG6hlCIGVFM8pJSFUmKNSCVFyz/y4
  ="/>
</head>
<body>

<h1>Hello, world!</h1>
<h3>2011-03-01 12:50:45 +0100</h3>

</body>
</html>
```

A Rails MVC filozófiájának harmadik eleme a modell, amelyet szintén a rails parancs generate opciójával hozhatunk létre. A harmadik argumentum a modell osztlály neve, amely a konvenció alapján egy egyes számban megadó és a szavakat _ szimbólummal összefűző string. Ennek többes számú változatával jön létre az az adatbázisban egy tábla. A parancs kiadása négy fájlt hoz létre: egy adatbázis migrációs Ruby szkriptet, egy ActiveRecord::Base leszármazottat a modell osztályok közé, egy egységteszteket tartalmazó osztályt és egy tesztadatokat tartalmazó YAML fájlt.

```
kovacs@debian:~/gyakorlat$ rails generate model task
  invoke  active_record
  create  db/migrate/20110301115851_create_tasks.
         rb
  create  app/models/task.rb
  invoke  test_unit
  create  test/unit/task_test.rb
  create  test/fixtures/tasks.yml
```

Az adatbázis migrációs szkriptben az adatmodell változtatásait adjuk meg. A modell létrehozása után egy olyan táblát hozna létre az alapértelme-

zett szkript, amelyben egy azonosító és két időpecsét oszlop szerepelne. A feladatunk az, hogy ezt kiegészítsük az általunk tárolni kívánt adatokkal. A tábla neve `tasks`, ehhez hozzáadunk egy egész szám típusú, egy string típusú és egy időpont típusú attribútumot rendre a `number`, `url` és `deadline` azonosítókkal. Mivel a struktúrát most hozzuk létre, csak a fel irányú migrációt definiáljuk, az ellentétes irányú migráció a struktúra törlését jelenti.

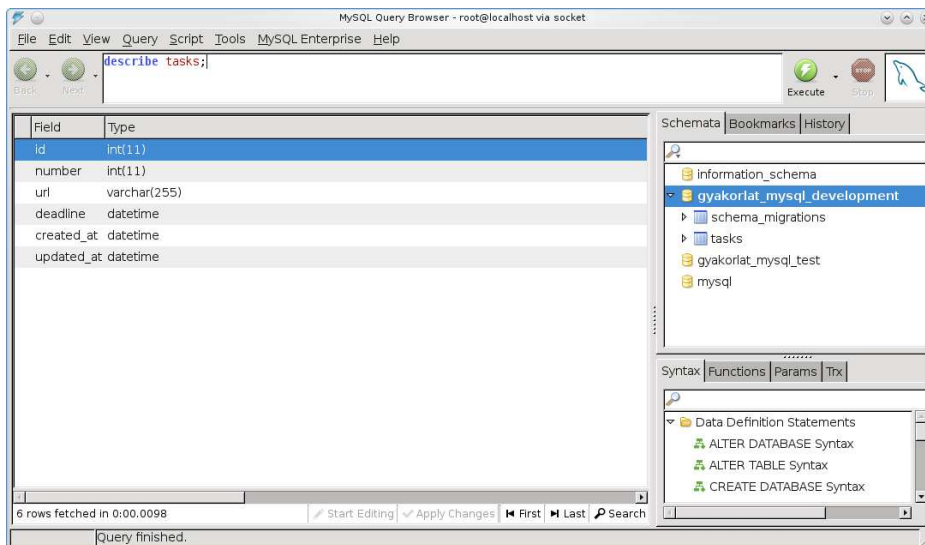
```
class CreateTasks < ActiveRecord::Migration
  def self.up
    create_table :tasks do |t|
      t.integer :number
      t.string :url
      t.datetime :deadline
      t.timestamps
    end
  end

  def self.down
    drop_table :tasks
  end
end
```

Ha a Rails alkalmazás létrehozásak megadtuk volna a `mysql` adatbázis opciót, akkor először létre kell hoznunk a megfelelő táblát, majd utána elvégezhajük a tábla struktúrájának módosítását. Ezekben a `rake` parancs nyújt segítséget. A `rake db:create` létrehozza a táblákat, a `rake db:migrate` módotítja a táblák struktúráját.

```
kovacsg@debian:~/gyakorlat_mysql/db/migrate$ rake db:
  create
(in /home/kovacsg/gyakorlat_mysql)
kovacsg@debian:~/gyakorlat_mysql/db/migrate$ rake db:
  migrate
(in /home/kovacsg/gyakorlat_mysql)
== CreateTasks: migrating
-----
-- create_table(:tasks)
--> 0.0027 s
== CreateTasks: migrated (0.0028 s)
-----
```

A gyakorlaton a `mysql` illesztővel létrehozott Rails alkalmazás a migráció után a 3. ábrán látható táblát hozta létre.



3. ábra. A migráció által létrehozott tábla

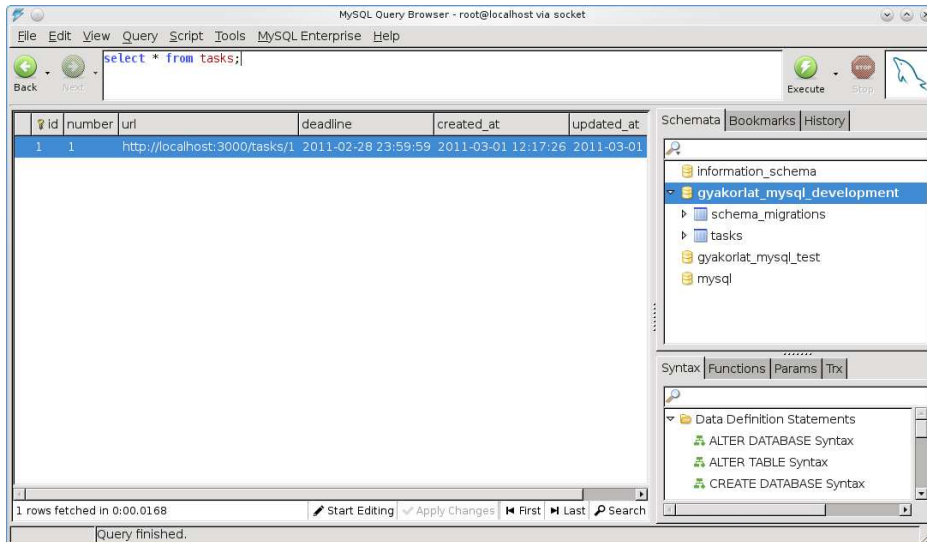
Az adatokon végzett manipulációt a rails console paranccsal előhívható interaktív Ruby konzolon (irb) illusztráljuk. A Task osztály konstruktorának argumentumai között a tábla attribútumaira a migrációs szkriptben megadott szimbólumokkal hivatkozhatunk, és rendelhetünk hozzá értéket. A nem inicializált attribútumok null értéket vesznek fel.

```

kovacsg@debian:~/gyakorlat_mysql/db/migrate$ rails
console
Loading development environment (Rails 3.0.5)
irb(main):001:0> t = Task.new
=> #<Task id: nil, number: nil, url: nil, deadline: nil,
      , created_at: nil, updated_at: nil>
irb(main):002:0> t.number=1
=> 1
irb(main):003:0> t.url='http://localhost:3000/tasks/1'
=> "http://localhost:3000/tasks/1"
irb(main):004:0> t
=> #<Task id: nil, number: 1, url: "http://localhost:3000/tasks/1",
      deadline: nil, created_at: nil,
      updated_at: nil>
irb(main):005:0> t.save
=> true
irb(main):006:0>

```

A `save` metódussal elmentettük a létrehozott példányt az adatbázisba, amelyet az adatbáziskezelőn ellenőrizhetünk, amit a 4. ábra is mutat.



4. ábra. Az elmentett rekord

Végül meggyőződhetünk arról, hogy az adatbázisban módosított érték (a `deadline` attribútum értéke) a modellben is módosul.

```
kovacsg@debian:~/gyakorlat_mysql/db/migrate$ rails console
Loading development environment (Rails 3.0.5)
irb(main):001:0> t = Task.find :first
=> #<Task id: 1, number: 1, url: "http://localhost:3000/tasks/1", deadline: "2011-02-28 23:59:59", created_at: "2011-03-01 12:17:26", updated_at: "2011-03-01 12:17:26">
irb(main):002:0> t.deadline
=> Mon, 28 Feb 2011 23:59:59 UTC +00:00
```

Hivatkozások

- [1] David Heinemeier Hansson et al. Sam Ruby, Dave Thomas. *Agile Web Development with Rails*, volume Third Edition. The Pragmatic Bookshelf, 2009 Mar.