

A Rails keretrendszer Gyakorlat

Kovács Gábor

2014. október 1.

1. Ruby telepítése, Gemek

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 és néhány további komponens fordításához szükségünk van C és C++ fordítóra, valamint a fejlesztői függvénykönyvtárakra, amelyeket az `apt-get install gcc` és `apt-get install g++` parancsok 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. A gyakorlaton a Ruby stabil, 1.9.3-as verzióját használjuk, ami előfeltétele a Rails 4-es változatának telepítésének. Ezt a következő paranccsal telepíthetjük rendszergazdaként:

```
kovacsg@debian:~$ sudo bash
```

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.

```

[sudo] password for kovacs:
root@debian:/home/kovacs# su -
root@debian:~# ruby
-su: ruby: command not found
root@debian:~# irb
-su: irb: command not found
root@debian:~# gem
-su: gem: command not found
root@debian:~# apt-get install ruby1.9.3
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libruby1.9.1 libruby1.9.1-dbg libtcltk-ruby1.9.1 libyaml-0-2 ri1.9.1 ruby1
  .9.1 ruby1.9.1-dev ruby1.9.1-examples ruby1.9.1-full tcl8.5 tk8.5
Suggested packages:
  graphviz ruby-switch tcl-tclreadline
The following NEW packages will be installed:
  libruby1.9.1 libruby1.9.1-dbg libtcltk-ruby1.9.1 libyaml-0-2 ri1.9.1 ruby1
  .9.1 ruby1.9.1-dev ruby1.9.1-examples ruby1.9.1-full ruby1.9.3 tcl8.5
  tk8.5
0 upgraded, 12 newly installed, 0 to remove and 319 not upgraded.
Need to get 0 B/18.2 MB of archives.
After this operation, 49.3 MB of additional disk space will be used.
Do you want to continue [Y/n]?
Selecting previously unselected package libyaml-0-2.
(Reading database ... 112679 files and directories currently installed.)
Unpacking libyaml-0-2 (from .../libyaml-0-2_0.1.4-2+deb7u4_amd64.deb) ...
Selecting previously unselected package libruby1.9.1.
Unpacking libruby1.9.1 (from .../libruby1.9.1_1.9.3.194-8.1+deb7u2_amd64.deb
) ...
Selecting previously unselected package tcl8.5.
Unpacking tcl8.5 (from .../tcl8.5_8.5.11-2_amd64.deb) ...
Selecting previously unselected package tk8.5.
Unpacking tk8.5 (from .../tk8.5_8.5.11-2_amd64.deb) ...
Selecting previously unselected package libtcltk-ruby1.9.1.
Unpacking libtcltk-ruby1.9.1 (from .../libtcltk-ruby1.9.1_1.9.3.194-8.1+
deb7u2_amd64.deb) ...
Selecting previously unselected package ruby1.9.1.
Unpacking ruby1.9.1 (from .../ruby1.9.1_1.9.3.194-8.1+deb7u2_amd64.deb) ...
Selecting previously unselected package ri1.9.1.
Unpacking ri1.9.1 (from .../ri1.9.1_1.9.3.194-8.1+deb7u2_all.deb) ...
Selecting previously unselected package ruby1.9.1-dev.
Unpacking ruby1.9.1-dev (from .../ruby1.9.1-dev_1.9.3.194-8.1+deb7u2_amd64.
deb) ...
Selecting previously unselected package ruby1.9.1-examples.
Unpacking ruby1.9.1-examples (from .../ruby1.9.1-examples_1.9.3.194-8.1+
deb7u2_all.deb) ...
Selecting previously unselected package libruby1.9.1-dbg.
Unpacking libruby1.9.1-dbg (from .../libruby1.9.1-dbg_1.9.3.194-8.1+
deb7u2_amd64.deb) ...
Selecting previously unselected package ruby1.9.1-full.
Unpacking ruby1.9.1-full (from .../ruby1.9.1-full_1.9.3.194-8.1+deb7u2_all.
deb) ...
Selecting previously unselected package ruby1.9.3.
Unpacking ruby1.9.3 (from .../ruby1.9.3_1.9.3.194-8.1+deb7u2_all.deb) ...
Processing triggers for man-db ...
Setting up libyaml-0-2 (0.1.4-2+deb7u4) ...
Setting up libruby1.9.1 (1.9.3.194-8.1+deb7u2) ...
Setting up tcl8.5 (8.5.11-2) ...
update-alternatives: using /usr/bin/tclsh8.5 to provide /usr/bin/tclsh (
tclsh) in auto mode.

```

```

Setting up tk8.5 (8.5.11-2) ...
update-alternatives: using /usr/bin/wish8.5 to provide /usr/bin/wish (wish)
  in auto mode.
Setting up libtcltk-ruby1.9.1 (1.9.3.194-8.1+deb7u2) ...
Setting up ruby1.9.1 (1.9.3.194-8.1+deb7u2) ...
update-alternatives: using /usr/bin/gem1.9.1 to provide /usr/bin/gem (gem)
  in auto mode.
update-alternatives: using /usr/bin/ruby1.9.1 to provide /usr/bin/ruby (ruby)
  in auto mode.
Setting up ri1.9.1 (1.9.3.194-8.1+deb7u2) ...
Setting up ruby1.9.1-dev (1.9.3.194-8.1+deb7u2) ...
Setting up ruby1.9.1-examples (1.9.3.194-8.1+deb7u2) ...
Setting up libruby1.9.1-dbg (1.9.3.194-8.1+deb7u2) ...
Setting up ruby1.9.1-full (1.9.3.194-8.1+deb7u2) ...
Setting up ruby1.9.3 (1.9.3.194-8.1+deb7u2) ...

```

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

```

root@debian:~# ruby --version
ruby 1.9.3p194 (2012-04-20 revision 35410) [x86_64-linux]
root@debian:~# gem --version
1.8.23
root@debian:~# gem list

*** LOCAL GEMS ***

```

A következő lépés a Ruby dokumentációgeneráló függvénykönyvtárának telepítése, amit a gem Ruby csomagkezelővel teszünk meg. A lépést kihagyva a Rails keretrendszer dokumentációjának telepítése sikertelen lenne, ami mindazonáltal nem nagy tragédia lévén annak telepítését amúgy is előszere-ttel kikapcsoljuk a gem --no-rdoc --no-ri kapcsolóival. A dokumentum-ban ezeket nem használjuk, teljes telepítést végzünk.

```

root@debian:/home/kovacs# gem install rdoc
root@debian:~# gem install rdoc
Fetching: json-1.8.1.gem (100%)
Building native extensions. This could take a while...
Fetching: rdoc-4.1.2.gem (100%)
Depending on your version of ruby, you may need to install ruby rdoc/ri data
:

<= 1.8.6 : unsupported
= 1.8.7 : gem install rdoc-data; rdoc-data --install
= 1.9.1 : gem install rdoc-data; rdoc-data --install
>= 1.9.2 : nothing to do! Yay!
Successfully installed json-1.8.1
Successfully installed rdoc-4.1.2
2 gems installed
Installing ri documentation for json-1.8.1...
Installing ri documentation for rdoc-4.1.2...
Installing RDoc documentation for json-1.8.1...
Installing RDoc documentation for rdoc-4.1.2...

```

Ezután telepítjük a Rails keretrendszert. A gyakorlaton fordítva tettük meg, ezért az imént említett hibára ráfutottunk.

```

root@debian:~# gem install rails
Fetching: i18n-0.6.11.gem (100%)
Fetching: thread_safe-0.3.4.gem (100%)
Fetching: tzinfo-1.2.2.gem (100%)
Fetching: minitest-5.4.2.gem (100%)
Fetching: activesupport-4.1.6.gem (100%)
Fetching: rack-1.5.2.gem (100%)
Fetching: rack-test-0.6.2.gem (100%)
Fetching: builder-3.2.2.gem (100%)
Fetching: erubis-2.7.0.gem (100%)
Fetching: actionview-4.1.6.gem (100%)
Fetching: actionpack-4.1.6.gem (100%)
Fetching: activemodel-4.1.6.gem (100%)
Fetching: arel-5.0.1.20140414130214.gem (100%)
Fetching: activerecord-4.1.6.gem (100%)
Fetching: mime-types-2.3.gem (100%)
Fetching: mail-2.6.1.gem (100%)
Fetching: actionmailer-4.1.6.gem (100%)
Fetching: rake-10.3.2.gem (100%)
Fetching: thor-0.19.1.gem (100%)
Fetching: railties-4.1.6.gem (100%)
Fetching: bundler-1.7.3.gem (100%)
Fetching: hike-1.2.3.gem (100%)
Fetching: multi_json-1.10.1.gem (100%)
Fetching: tilt-1.4.1.gem (100%)
Fetching: sprockets-2.12.2.gem (100%)
Fetching: sprockets-rails-2.1.4.gem (100%)
Fetching: rails-4.1.6.gem (100%)
Successfully installed i18n-0.6.11
Successfully installed thread_safe-0.3.4
Successfully installed tzinfo-1.2.2
Successfully installed minitest-5.4.2
Successfully installed activesupport-4.1.6
Successfully installed rack-1.5.2
Successfully installed rack-test-0.6.2
Successfully installed builder-3.2.2
Successfully installed erubis-2.7.0
Successfully installed actionview-4.1.6
Successfully installed actionpack-4.1.6
Successfully installed activemodel-4.1.6
Successfully installed arel-5.0.1.20140414130214
Successfully installed activerecord-4.1.6
Successfully installed mime-types-2.3
Successfully installed mail-2.6.1
Successfully installed actionmailer-4.1.6
Successfully installed rake-10.3.2
Successfully installed thor-0.19.1
Successfully installed railties-4.1.6
Successfully installed bundler-1.7.3
Successfully installed hike-1.2.3
Successfully installed multi_json-1.10.1
Successfully installed tilt-1.4.1
Successfully installed sprockets-2.12.2
Successfully installed sprockets-rails-2.1.4
Successfully installed rails-4.1.6
27 gems installed
Installing ri documentation for i18n-0.6.11...
Installing ri documentation for thread_safe-0.3.4...
Installing ri documentation for tzinfo-1.2.2...
Installing ri documentation for minitest-5.4.2...
Installing ri documentation for activesupport-4.1.6...
Installing ri documentation for rack-1.5.2...

```

```
Installing ri documentation for rack-test -0.6.2...
Installing ri documentation for builder -3.2.2...
Installing ri documentation for erubis -2.7.0...
Installing ri documentation for actionview -4.1.6...
Installing ri documentation for actionpack -4.1.6...
Installing ri documentation for activemodel -4.1.6...
Installing ri documentation for arel -5.0.1.20140414130214...
Installing ri documentation for activerecord -4.1.6...
Installing ri documentation for mime-types -2.3...
Installing ri documentation for mail -2.6.1...
Installing ri documentation for actionmailer -4.1.6...
Installing ri documentation for rake -10.3.2...
Installing ri documentation for thor -0.19.1...
Installing ri documentation for railties -4.1.6...
Installing ri documentation for bundler -1.7.3...
Installing ri documentation for hike -1.2.3...
Installing ri documentation for multi_json -1.10.1...
Installing ri documentation for tilt -1.4.1...
Installing ri documentation for sprockets -2.12.2...
Installing ri documentation for sprockets-rails -2.1.4...
Installing ri documentation for rails -4.1.6...
Installing RDoc documentation for i18n -0.6.11...
Installing RDoc documentation for thread_safe -0.3.4...
Installing RDoc documentation for tzinfo -1.2.2...
Installing RDoc documentation for minitest -5.4.2...
Installing RDoc documentation for activesupport -4.1.6...
Installing RDoc documentation for rack -1.5.2...
Installing RDoc documentation for rack-test -0.6.2...
Installing RDoc documentation for builder -3.2.2...
Installing RDoc documentation for erubis -2.7.0...
Installing RDoc documentation for actionview -4.1.6...
Installing RDoc documentation for actionpack -4.1.6...
Installing RDoc documentation for activemodel -4.1.6...
Installing RDoc documentation for arel -5.0.1.20140414130214...
Installing RDoc documentation for activerecord -4.1.6...
Installing RDoc documentation for mime-types -2.3...
Installing RDoc documentation for mail -2.6.1...
Installing RDoc documentation for actionmailer -4.1.6...
Installing RDoc documentation for rake -10.3.2...
Installing RDoc documentation for thor -0.19.1...
Installing RDoc documentation for railties -4.1.6...
Installing RDoc documentation for bundler -1.7.3...
Installing RDoc documentation for hike -1.2.3...
Installing RDoc documentation for multi_json -1.10.1...
Installing RDoc documentation for tilt -1.4.1...
Installing RDoc documentation for sprockets -2.12.2...
Installing RDoc documentation for sprockets-rails -2.1.4...
Installing RDoc documentation for rails -4.1.6...
```

A Rails rendszerünk konfigurációja a következő lesz a félév során. Kétféle webszervert használunk, a fejlesztéshez a beágyazott Webricket, míg az éles rendszerhez az Apache2-t. Adatbáziskezelőből szintén kétfélét nézünk meg, a beágyazott SQLite3-at és a kliens-szerver alapú MySQL-t. A következőkben ezek illesztéséhez szükséges adapterek fordítását lehetővé tevő C++ és Ruby függvénykönyvtárakat telepítjük.

Railshez egy opcionális, ám az éles rendszerek szempontjából annál hasznosabb komponens az Apache webszerverrel való integrációt lehetővé tevő

Passenger plugin.

```
root@debian:~# gem install passenger
Fetching: daemon_controller-1.2.0.gem (100%)
Fetching: passenger-4.0.52.gem (100%)
Building native extensions. This could take a while...
Successfully installed daemon_controller-1.2.0
Successfully installed passenger-4.0.52
2 gems installed
Installing ri documentation for daemon_controller-1.2.0...
Installing ri documentation for passenger-4.0.52...
Installing RDoc documentation for daemon_controller-1.2.0...
Installing RDoc documentation for passenger-4.0.52...
```

Ezután telepíthetjük először az SQLite, majd a MySQL adapterének Ruby API-ját. Ez két lépésből áll, először a operációs rendszertől függő fájlok fordításához szükséges forrásokat (C és C++ headereket) telepítjük.

```
root@debian:~# apt-get install libsqlite3-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libpq-dev
Use 'apt-get autoremove' to remove them.
Suggested packages:
  sqlite3-doc
The following NEW packages will be installed:
  libsqlite3-dev
0 upgraded, 1 newly installed, 0 to remove and 319 not upgraded.
Need to get 0 B/578 kB of archives.
After this operation, 1,382 kB of additional disk space will be used.
Selecting previously unselected package libsqlite3-dev.
(Reading database ... 129571 files and directories currently installed.)
Unpacking libsqlite3-dev (from .../libsqlite3-dev_3.7.13-1+deb7u1_amd64.deb)
...
Setting up libsqlite3-dev (3.7.13-1+deb7u1) ...
root@debian:~# apt-get install libmysqlclient-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libpq-dev
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  libmysqlclient18
The following NEW packages will be installed:
  libmysqlclient-dev libmysqlclient18
0 upgraded, 2 newly installed, 0 to remove and 319 not upgraded.
Need to get 0 B/1,636 kB of archives.
After this operation, 9,201 kB of additional disk space will be used.
Do you want to continue [Y/n]?
Selecting previously unselected package libmysqlclient18.
(Reading database ... 129582 files and directories currently installed.)
Unpacking libmysqlclient18 (from .../libmysqlclient18_5.5.37-0+wheezy1_amd64
.deb) ...
Selecting previously unselected package libmysqlclient-dev.
Unpacking libmysqlclient-dev (from .../libmysqlclient-dev_5.5.37-0+
wheezy1_amd64.deb) ...
Processing triggers for man-db ...
Setting up libmysqlclient18 (5.5.37-0+wheezy1) ...
```

```
Setting up libmysqlclient-dev (5.5.37-0+wheezy1) ...
```

Ezután telepíthetjük magukat az adatbáziskezelő-adaptereket, a `sqlite3`-t és a `mysql2`-t.

```
root@debian:~# gem install sqlite3
Fetching: sqlite3-1.3.9.gem (100%)
Building native extensions. This could take a while...
Successfully installed sqlite3-1.3.9
1 gem installed
Installing ri documentation for sqlite3-1.3.9...
Installing RDoc documentation for sqlite3-1.3.9...
root@debian:~# gem install mysql2
Fetching: mysql2-0.3.16.gem (100%)
Building native extensions. This could take a while...
Successfully installed mysql2-0.3.16
1 gem installed
Installing ri documentation for mysql2-0.3.16...
Installing RDoc documentation for mysql2-0.3.16...
```

A Rails 3.1-es verziója változást hozott a JavaScript kezelésben, amely két további gem telepítését teszi szükségessé. Az `execjs` JavaScriptek Ruby-ba ágyzását teszi lehetővé, a `therubyracer` pedig egy szerver oldali JavaScript interpreter. Ahogy gyakorlaton láttuk a `turbolinks` API nélkül a Rails már nem hajlandó elindulni, noha az opcionális.

```
root@debian:~# gem install execjs
Fetching: execjs-2.2.1.gem (100%)
Successfully installed execjs-2.2.1
1 gem installed
Installing ri documentation for execjs-2.2.1...
Installing RDoc documentation for execjs-2.2.1...
root@debian:~# gem install therubyracer
Fetching: ref-1.0.5.gem (100%)
Fetching: libv8-3.16.14.7-x86_64-linux.gem (100%)
Fetching: therubyracer-0.12.1.gem (100%)
Building native extensions. This could take a while...
Successfully installed ref-1.0.5
Successfully installed libv8-3.16.14.7-x86_64-linux
Successfully installed therubyracer-0.12.1
3 gems installed
Installing ri documentation for ref-1.0.5...
Installing ri documentation for libv8-3.16.14.7-x86_64-linux...
Installing ri documentation for therubyracer-0.12.1...
Installing RDoc documentation for ref-1.0.5...
Installing RDoc documentation for libv8-3.16.14.7-x86_64-linux...
Installing RDoc documentation for therubyracer-0.12.1...
root@debian:~# gem install jquery-rails
Fetching: jquery-rails-3.1.2.gem (100%)
Successfully installed jquery-rails-3.1.2
1 gem installed
Installing ri documentation for jquery-rails-3.1.2...
Installing RDoc documentation for jquery-rails-3.1.2...
root@debian:~# gem install turbolinks
Fetching: coffee-script-source-1.8.0.gem (100%)
Fetching: coffee-script-2.3.0.gem (100%)
Fetching: coffee-rails-4.0.1.gem (100%)
Fetching: turbolinks-2.3.0.gem (100%)
Successfully installed coffee-script-source-1.8.0
Successfully installed coffee-script-2.3.0
```

```
Successfully installed coffee-rails-4.0.1
Successfully installed turbolinks-2.3.0
4 gems installed
Installing ri documentation for coffee-script-source-1.8.0...
Installing ri documentation for coffee-script-2.3.0...
Installing ri documentation for coffee-rails-4.0.1...
Installing ri documentation for turbolinks-2.3.0...
Installing RDoc documentation for coffee-script-source-1.8.0...
Installing RDoc documentation for coffee-script-2.3.0...
Installing RDoc documentation for coffee-rails-4.0.1...
Installing RDoc documentation for turbolinks-2.3.0...
```

A Rails rendszerünk ezután az alábbi Ruby függvénykönyvtárból áll:

```
root@debian:~# gem list

*** LOCAL GEMS ***

actionmailer (4.1.6)
actionpack (4.1.6)
actionview (4.1.6)
activemodel (4.1.6)
activerecord (4.1.6)
activesupport (4.1.6)
arel (5.0.1.20140414130214)
builder (3.2.2)
bundler (1.7.3)
coffee-rails (4.0.1)
coffee-script (2.3.0)
coffee-script-source (1.8.0)
daemon_controller (1.2.0)
erubis (2.7.0)
execjs (2.2.1)
hike (1.2.3)
i18n (0.6.11)
jquery-rails (3.1.2)
json (1.8.1)
libv8 (3.16.14.7 x86_64-linux)
mail (2.6.1)
mime-types (2.3)
minitest (5.4.2)
multi_json (1.10.1)
mysql2 (0.3.16)
passenger (4.0.52)
rack (1.5.2)
rack-test (0.6.2)
rails (4.1.6)
railties (4.1.6)
rake (10.3.2)
rdoc (4.1.2)
ref (1.0.5)
sprockets (2.12.2)
sprockets-rails (2.1.4)
sqlite3 (1.3.9)
therubyracer (0.12.1)
thor (0.19.1)
thread_safe (0.3.4)
tilt (1.4.1)
turbolinks (2.3.0)
```


2. Rails alkalmazás létrehozása

A Rails használatához elérhetővé kell tennünk a rails szkriptet, amely Debian/Ubuntu Linux és 1.9.3-es ruby esetén a /var/lib/gems/1.9.1/bin/ könyvtárba kerül.

```
kovacs@rails:~$ export PATH=$PATH:/var/lib/gems/1.9.1/bin/
```

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 sqlite3-ot mint alapértelmezett adatbáziskezelőt és a mysql-t fogjuk használni. A -B kapcsoló mellőzi a Rails alkalmazásunk Ruby függőségeinek feloldását, amelyet azonban később pótolnunk kell. Javascript API-nak a JQuery tökéletesen meg fog felelni számunkra, így azt nem módosítjuk.

```
kovacs@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
  -m, [--template=TEMPLATE]        # Path to some
                                   application template (can be a filesystem path or URL)
  --skip-gemfile, [--no-skip-gemfile] # Don't create a
                                   Gemfile
  -B, [--skip-bundle], [--no-skip-bundle] # Don't run bundle
                                   install
  -G, [--skip-git], [--no-skip-git]     # Skip .gitignore
                                   file
  --skip-keeps, [--no-skip-keeps]       # Skip source
                                   control .keep files
  -O, [--skip-active-record], [--no-skip-active-record] # Skip Active
                                   Record files
  -V, [--skip-action-view], [--no-skip-action-view]     # Skip Action View
                                   files
  -S, [--skip-sprockets], [--no-skip-sprockets]         # Skip Sprockets
                                   files
  --skip-spring, [--no-skip-spring]                   # Don't install
                                   Spring application preloader
  -d, [--database=DATABASE]                          # Preconfigure for
                                   selected database (options: mysql/oracle/postgresql/sqlite3/frontbase/
                                   ibm_db/sqlserver/jdbcmysql/jdbcsqlite3/jdbcpostgresql/jdbc)
                                   # Default: sqlite3
  -j, [--javascript=JAVASCRIPT]                      # Preconfigure for
                                   selected JavaScript library
                                   # Default: jquery
  -J, [--skip-javascript], [--no-skip-javascript]    # Skip JavaScript
                                   files
  --dev, [--no-dev]                                   # Setup the
                                   application with Gemfile pointing to your Rails checkout
  --edge, [--no-edge]                                 # Setup the
                                   application with Gemfile pointing to Rails repository
```

```

-T, [--skip-test-unit], [--no-skip-test-unit] # Skip Test::Unit
files
  [--rc=RC] # Path to file
             containing extra configuration options for rails command
  [--no-rc], [--no-no-rc] # Skip loading of
                           extra configuration options from .railsrc file

Runtime options:
-f, [--force] # Overwrite files that already exist
-p, [--pretend], [--no-pretend] # Run but do not make any changes
-q, [--quiet], [--no-quiet] # Suppress status output
-s, [--skip], [--no-skip] # Skip files that already exist

Rails options:
-h, [--help], [--no-help] # Show this help message and quit
-v, [--version], [--no-version] # Show Rails version number and quit

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

You can specify extra command-line arguments to be used every time
'rails new' runs in the .railsrc configuration file in your home
directory.

Note that the arguments specified in the .railsrc file don't affect the
defaults values shown above in this help message.

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** parancsnek **new** opciót megadva hozhatunk létre, a második argumentum az alkalmazás neve. A parancsot 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.

A parancs végén automatikusan lefutna a Rails keretrendszer egy másik parancsa, a **bundle**, amely az **install** opció hatására összeszedi a gemek közül azokat, amelyekre az alkalmazásunknak szüksége lesz, azonban ezt a **-B** kapcsolóval letiltjuk, mert testre akarjuk szabni a keretrendszerünket.

A konzolon kiadott **bundle install** paranccsal később telepíthetjük az összes az esetlegesen hiányzó vagy nem megfelelő verziójú Ruby függvénykönyvtárat, a **bundle update** paranccsal pedig az éppen használni kívánt verziót tudunk betölteni az egyes gemekből. A **bundle** parancshoz a **--path** kapcsolót és egy fájlrendszeri útvonalat hozzáfűzve elérhetjük, hogy garantáltan mindig a saját Ruby környezetünkkel dolgozhassunk, és ne ütközzünk az időközben bekövetkező rendszerfrissítések okozta inkompatibilitás áldozata-

tául.

```
kovacs@debian:~$ rails new gyakorlat -B
create
create  README.rdoc
create  Rakefile
create  config.ru
create  .gitignore
create  Gemfile
create  app
create  app/assets/javascripts/application.js
create  app/assets/stylesheets/application.css
create  app/controllers/application_controller.rb
create  app/helpers/application_helper.rb
create  app/views/layouts/application.html.erb
create  app/assets/images/.keep
create  app/mailers/.keep
create  app/models/.keep
create  app/controllers/concerns/.keep
create  app/models/concerns/.keep
create  bin
create  bin/bundle
create  bin/rails
create  bin/rake
create  config
create  config/routes.rb
create  config/application.rb
create  config/environment.rb
create  config/secrets.yml
create  config/environments
create  config/environments/development.rb
create  config/environments/production.rb
create  config/environments/test.rb
create  config/initializers
create  config/initializers/assets.rb
create  config/initializers/backtrace_silencers.rb
create  config/initializers/cookies_serializer.rb
create  config/initializers/filter_parameter_logging.rb
create  config/initializers/inflections.rb
create  config/initializers/mime_types.rb
create  config/initializers/session_store.rb
create  config/initializers/wrap_parameters.rb
create  config/locales
create  config/locales/en.yml
create  config/boot.rb
create  config/database.yml
create  db
create  db/seeds.rb
create  lib
create  lib/tasks
create  lib/tasks/.keep
create  lib/assets
create  lib/assets/.keep
create  log
create  log/.keep
create  public
create  public/404.html
create  public/422.html
create  public/500.html
create  public/favicon.ico
create  public/robots.txt
create  test/fixtures
```

```

create test/fixtures/.keep
create test/controllers
create test/controllers/.keep
create test/mailers
create test/mailers/.keep
create test/models
create test/models/.keep
create test/helpers
create test/helpers/.keep
create test/integration
create test/integration/.keep
create test/test_helper.rb
create tmp/cache
create tmp/cache/assets
create vendor/assets/javascripts
create vendor/assets/javascripts/.keep
create vendor/assets/stylesheets
create vendor/assets/stylesheets/.keep
kovacs@debian:~$ cd gyakorlat/
kovacs@debian:~/gyakorlat$ ls
app  config  db      Gemfile.lock  log      Rakefile  test  vendor
bin  config.ru  Gemfile  lib           public  README.rdoc  tmp

```

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 `bin` könyvtár elérhetővé teszi számunkra a `rails`, a `rake` és a `bundle` parancsokat, amelyeket a Rails alkalmazásunk menedzsmentjét fogjuk megvalósítani. 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 szerIALIZED 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 `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 érhetőek el.

Az első dolgunk a `Gemfile` módosítása. Először kikommentezzük az első sort, hogy a Rails a Ruby függvénykönyvtárakat a helyi, frissen telepített helyen keresse, és ne a megadott távoli szerveren. Ez azért szükséges, hogy a rendszer egy rögzített verziójú keretrendszer mellett tudhassuk fejleszteni, és az ne frissüljön a `Gemfile` minden módosításakor. Egy esetleges frissítés fejlesztés közben elronthatja a teljes addigi munkánkat!

Praktikusan a `rails`, az adatbáziskezelő adaptere és a `jquery-rails-en`

kívül egyelőre másra nem lesz szükségünk. A 3.1-es verzió JavaScript változásai miatt meg kell adnunk még a `therubyracer`-t mint interpretert, különben a Rails rendszerünk működésképtelen marad. Ezt a megfelelő sor elejéről a komment szimbólum eltávolításával tehetjük meg. A másik módosítás, amit elvégzünk, a Ruby függvénykönyvtárak hálózati elérésének letiltása, amit az első sor elé helyezett komment szimbólummal érünk el. A módosítás után futtatnunk kell egy `bundle install` vagy `bundle update` parancsot.

A `rails` parancs gyakran használt argumentuma a `console`, amivel az első gyakorlatról megismert interaktív Ruby értelmezőt indíthatunk, melyben a Rails alkalmazásunk környezeti beállításai inicializálásra kerültek.

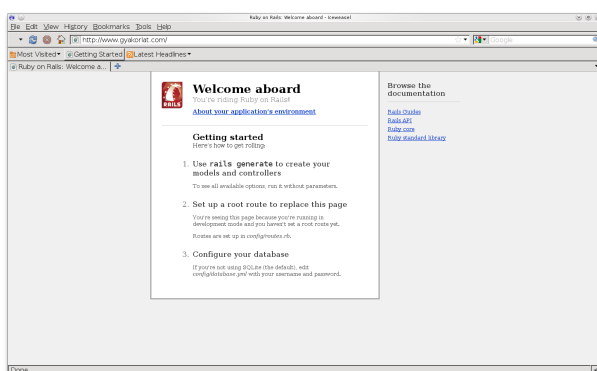
```
kovacsg@debian:~/gyakorlat$ bundle install
Resolving dependencies...
Using rake 10.3.2
Using i18n 0.6.11
Using json 1.8.1
Using minitest 5.4.2
Using thread_safe 0.3.4
Using tzinfo 1.2.2
Using activesupport 4.1.6
Using builder 3.2.2
Using erubis 2.7.0
Using actionview 4.1.6
Using rack 1.5.2
Using rack-test 0.6.2
Using actionpack 4.1.6
Using mime-types 2.3
Using mail 2.6.1
Using actionmailer 4.1.6
Using activemodel 4.1.6
Using arel 5.0.1.20140414130214
Using activerecord 4.1.6
Using bundler 1.7.3
Using hike 1.2.3
Using thor 0.19.1
Using railties 4.1.6
Using jquery-rails 3.1.2
Using libv8 3.16.14.7
Using multi_json 1.10.1
Using mysql2 0.3.16
Using tilt 1.4.1
Using sprockets 2.12.2
Using sprockets-rails 2.1.4
Using rails 4.1.6
Using ref 1.0.5
Using therubyracer 0.12.1
Your bundle is complete!
Use 'bundle show [gemname]' to see where a bundled gem is installed.
```

```
kovacsg@debian:~/gyakorlat$ rails console
Loading development environment (Rails 4.1.6)
irb(main):001:0 >
```

3. Web- és adatbázisszerver kapcsolat beállítása

A Rails keretrendszer webszerverek számára állít elő dinamikus tartalmat adatbázisbeli adatok alapján. A gyakorlat során kétféle webszervert és adatbáziskezelőt vizsgáltunk meg.

A beépített webszervert a `rails server` opciójával indíthatjuk el a Rails alkalmazásunk tetszőleges könyvtárából. Ha a szervert háttéralkalmazásként kívánjuk futtatni, akkor a `-d` kapcsolót is hozzá kell fűznünk a parancssorhoz. Ezután a szervert elérhetjük a `http://localhost:3000` webcímen, lásd 1. ábra.



1. ábra. A beágyazott webszerver elérése

```
kovacs@debian:~/gyakorlat$ rails server
=> Booting WEBrick
=> Rails 4.1.6 application starting in development on http://0.0.0.0:3000
=> Run 'rails server -h' for more startup options
=> Notice: server is listening on all interfaces (0.0.0.0). Consider using
    127.0.0.1 (--binding option)
=> Ctrl-C to shutdown server
[2014-10-01 12:56:31] INFO WEBrick 1.3.1
[2014-10-01 12:56:31] INFO ruby 1.9.3 (2012-04-20) [x86_64-linux]
[2014-10-01 12:56:31] INFO WEBrick::HTTPServer#start: pid=4088 port=3000
```

A Rails a már említett Passenger plugin segítségével illeszhető Apache2 webszerverhez. Ehhez az alábbi kódrészletben mutatott parancsot kell kiadnunk. A parancs lefordítja, majd telepíti az Apache webszerver Rails modulját, a konzolra kiírja az Apache2 konfigurációs állományába beírandó Rails specifikus három sort, és a Rails alkalmazásunkhoz való hozzáféréshez szükséges VirtualHost beállításokat. A Passenger alapértelmezés szerint a Rails éles környezetével működik együtt, ha ezt módosítani szeretnénk, akkor a `RailsEnv development` sort kell elhelyeznünk az Apache konfigurációs állományába a fejlesztői környezet eléréséhez.

```
passenger-install-apache2-module
```

A Rails kényelmesebb hozzáférése végett módosíthatjuk a helyi gépen doménnévtáblát (Windowson `C:\windows\system32\drivers\etc\hosts`, Linuxon `/etc/hosts`) a következő bejegyzés hozzáadva, amely minden, a `gyakorlat.com`, illetve `www.gyakorlat.com` címre küldött kérést a helyi gépen kezeltet le.

```
127.0.0.1      gyakorlat.com www.gyakorlat.com
```

Az Apache2 modul elérhetővé tételére az Apache2 konfigurációs könyvtárban létre kell hoznunk a modul beállításait. Szerencsére a telepítő parancs a helyes beállításokat kiírja a konzolra. A fájl a `/etc/apache2/mods-enabled/rails.load` néven hozzuk létre.

Az Apache2 virtuális hoszt beállításait szintén konfigurációs állományok között kell elhelyeznünk például `/etc/apache2/sites-available/gyakorlat.conf` néven. Ennek tartalma a telepítő parancs alapján a következő lehet:

```
RackEnv development
<VirtualHost *:80>

    ServerName www.gyakorlat.com
    ServerAdmin admin@gyakorlat.com
    DocumentRoot /home/kovacs/gyakorlat/public
    ServerSignature On

    CustomLog /var/log/apache/gyakorlat_access.log combined
    ErrorLog /var/log/apache/gyakorlat_error.log
    LogLevel info

    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /home/kovacs/gyakorlat/public/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Order allow,deny
        Allow from all
        Options -MultiViews
    </Directory>
```

Mivel módosítottuk az Apache2 beállításait, újra kell töltenünk annak konfigurációs állományait.

```
root@debian:~# /etc/init.d/apache2 reload
[ ok ] Reloading web server config: apache2.
```

Ezután egy böngészőbe beírva a `www.gyakorlat.com` címet a Rails alkalmazásunk fog megjelenni! A Rails alkalmazás módosítása után az Apache-beli frissítése a `tmp/restart.txt` fájl időpecsétjének módosításával lehetséges (mentés, létrehozás, `touch` stb.)

Mivel a Rails alkalmazás létrehozásakor nem adtunk meg, hogy melyik adatbáziskezelőt kívánjuk használni, ezért az alapértelmezett SQLite kon-

figurációs paramétereit jelennek meg a `database.yml`-ben mindhárom definiált környezethez, vagyis a fejlesztési (`development`), tesztelési (`test`) és éles (`production`). Az egyes környezetek adapter opciója határozza meg az adatbáziskezelő típusát és a használható paraméterkészletet. 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
#
# Ensure the SQLite 3 gem is defined in your Gemfile
# gem 'sqlite3'
#
default: &default
  adapter: sqlite3
  pool: 5
  timeout: 5000

development:
  <<: *default
  database: db/development.sqlite3

# 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:
  <<: *default
  database: db/test.sqlite3

production:
  <<: *default
  database: db/production.sqlite3
```

MySQL esetén (`rails new gyakorlat -d mysql`) a `mysql2` adaptert használjuk. A különbség a `Gemfile`-ban és a `database.yml`-ben jelentkezik. Fontos opció az alapértelmezett karakterkódolás megadása, ami, ha lehet hagyjunk változatlanul `utf8` értéken. A `database` opció a séma nevét tartalmazza értéként. A felhasználónév és jelszó megadása kötelező. A kapcsolat lehet `socket` vagy `host` és `port` alapon megadott.

Távoli gépen található MySQL esetén szükséges a Rails felhasználó hozzáférési jogosultságainak beállítása, amelyet a MySQL `mysql` adatbázisának `user` táblájában kell megtennünk. A Rails szerverének IP címére engedélyeznünk kell a Rails adatbázis-konfigurációs állományában megadott felhasználónévvel és jelszóval azonosított felhasználó számára a `select`, `update`, `insert`, `delete`, `create`, `alter`, `drop` és `index` jogosultságokat.

Az adatbáziskapcsolatnak a szerver elindítása előtt léteznie kell. Ez SQLite esetén nem probléma, hiszen a fejlesztői adatbázisfájl automatikusan létrejön, MySQL esetén viszont a sémákat explicite létre kell hoznunk. Ebben a `rake` parancs lesz segítségünkre, amelyet a Rails keretrendszerhez kapcsolódó műveletekre használunk. A műveletek listája alább látható.


```

kovacs@debian:~/gyakorlat$ rake -T
rake about # List versions of all Rails
            frameworks and the environment
rake assets:clean[keep] # Remove old compiled assets
rake assets:clobber # Remove compiled assets
rake assets:environment # Load asset compile environment
rake assets:precompile # Compile all the assets named in
            config/assets.precompile
rake cache_digests:dependencies # Lookup first-level dependencies
            for TEMPLATE (like messages/show or comments/_comment.html)
rake cache_digests:nested_dependencies # Lookup nested dependencies for
            TEMPLATE (like messages/show or comments/_comment.html)
rake db:create # Creates the database from
            DATABASE_URL or config/database.yml for the current RAILS_ENV (use db:
            create:all to create all databases in the config)
rake db:drop # Drops the database from
            DATABASE_URL or config/database.yml for the current RAILS_ENV (use db:
            drop:all to drop all databases in the config)
rake db:fixtures:load # Load fixtures into the current
            environment's database
rake db:migrate # Migrate the database (options:
            VERSION=x, VERBOSE=false, SCOPE=blog)
rake db:migrate:status # Display status of migrations
rake db:rollback # Rolls the schema back to the
            previous version (specify steps w/ STEP=n)
rake db:schema:cache:clear # Clear a db/schema_cache.dump file
rake db:schema:cache:dump # Create a db/schema_cache.dump file
rake db:schema:dump # Create a db/schema.rb file that is
            portable against any DB supported by AR
rake db:schema:load # Load a schema.rb file into the
            database
rake db:seed # Load the seed data from db/seeds.
            rb
rake db:setup # Create the database, load the
            schema, and initialize with the seed data (use db:reset to also drop the
            database first)
rake db:structure:dump # Dump the database structure to db/
            structure.sql
rake db:version # Retrieves the current schema
            version number
rake doc:app # Generate docs for the app — also
            available doc:rails, doc:guides (options: TEMPLATE=/rdoc-template.rb,
            TITLE="Custom Title")
rake log:clear # Truncates all *.log files in log/
            to zero bytes (specify which logs with LOGS=test,development)
rake middleware # Prints out your Rack middleware
            stack
rake notes # Enumerate all annotations (use
            notes:optimize, :fixme, :todo for focus)
rake notes:custom # Enumerate a custom annotation,
            specify with ANNOTATION=CUSTOM
rake rails:template # Applies the template supplied by
            LOCATION=(/path/to/template) or URL
rake rails:update # Update configs and some other
            initially generated files (or use just update:configs or update:bin)
rake routes # Print out all defined routes in
            match order, with names
rake secret # Generate a cryptographically
            secure secret key (this is typically used to generate a secret for
            cookie sessions)
rake stats # Report code statistics (KLOCs, etc
            ) from the application

```

```

rake test # Runs test:units, test:functionals,
          test:generators, test:integration together
rake test:all # Run tests quickly by merging all
              types and not resetting db
rake test:all:db # Run tests quickly, but also reset
                 db
rake time:zones:all # Displays all time zones, also
                   available: time:zones:us, time:zones:local — filter with OFFSET
                   parameter, e.g., OFFSET=-6
rake tmp:clear # Clear session, cache, and socket
               files from tmp/ (narrow w/ tmp:sessions:clear, tmp:cache:clear, tmp:
               sockets:clear)
rake tmp:create # Creates tmp directories for
                sessions, cache, sockets, and pids

```

Az adatbázis létrehozása következő paranccsal történhet meg, amely a development és a test környezetekhez hozza létre egy üres sémát.

```

bash-3.2$ rake db:create
(in /home/kovacs/gyakorlat)

```

4. Bevezetés a Rails használatába

Az adatbáziskapcsolat-leíró mellett a másik fontos konfigurációs fájlunk a `routes.rb`. Ez azt adja meg, hogy milyen struktúrájú legyen az URL, amivel elérjük a Rails alkalmazásunk egyes funkcióit. A Rails kontrollerek létrehozásakor automatikusan írja ezt a fájl, de megadhatunk egy általános mintát is a HTTP kérések URI-ainak útvonal részére, 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, például `.html` vagy `.xml`. Az utolsó három megadása opcionális.

```

Gyakorlat::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 kontrollert a `rails` parancs `generate` opciójával. A második argumentum (`controller`) azt mondja meg, hogy egy új kontrollert 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/db$ rails generate -h
Usage: rails generate GENERATOR [args] [options]

General options:
  -h, [--help]           # Print generator's options and usage
  -p, [--pretend]        # Run but do not make any changes
  -f, [--force]          # Overwrite files that already exist
  -s, [--skip]           # Skip files that already exist
  -q, [--quiet]          # Suppress status output

Please choose a generator below.

Rails:
  assets
  controller
  generator
  helper
  integration_test
  mailer
  migration
  model
  resource
  scaffold
  scaffold_controller
  task

Jquery:
  jquery:install

Js:
  js:assets

TestUnit:
  test_unit:generator
  test_unit:plugin
kovacs@debian:~/gyakorlat/db$ rails generate controller say hello
create  app/controllers/say_controller.rb
route   get 'say/hello'
invoke  erb
create  app/views/say
create  app/views/say/hello.html.erb
invoke  test_unit
create  test/controllers/say_controller_test.rb
invoke  helper
create  app/helpers/say_helper.rb
invoke  test_unit
create  test/helpers/say_helper_test.rb
invoke  assets
invoke  js
create  app/assets/javascripts/say.js
invoke  css
create  app/assets/stylesheets/say.css

```

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.

Ez alapján nézzük meg a szokásos Hello, world alkalmazást ezúttal Rails-ben. 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.

```
<%= Time.now %>
```

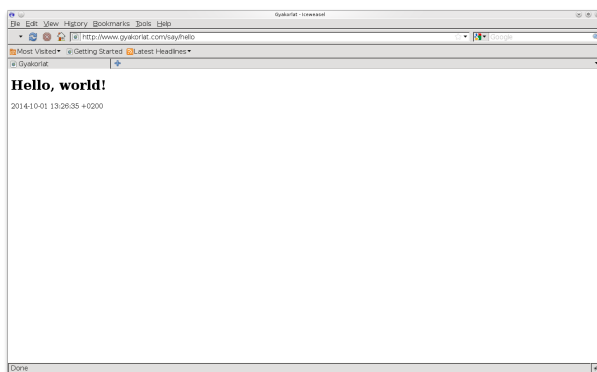
Mivel a nézetbe nem illik logikát rakni, csak a megjelenítendő értéket, ezért áttesszük az idő lekérdezését a controllerbe, annak is az akciónak megfelelő metódusába, a `hello`-ba

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

A nézetben pedig csak hivatkozunk a Controller példányváltozóra.

```
<%= @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>
</head>
<body>

<h1>Hello , world!</h1>
<p>2014-10-01 13:26:35 +0200</p>

</body>
</html>

```

A Rails MVC filozófiájának harmadik eleme a modell, amelyet szintén a `rails generate` opciójával hozhatunk létre. A harmadik argumentum a modell oszlály 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/app/views/layouts$ rails generate model User
username:string password:string email:string
  invoke  active_record
  create  db/migrate/20141001111344_create_users.rb
  create  app/models/user.rb
  invoke  test_unit
  create  test/models/user_test.rb
  create  test/fixtures/users.yml

```

Az adatbázis-migrációs szkriptben az adatmodell változtatásait adjuk meg. Az előző modellgeneráló szkript egy olyan táblát hozna létre, amelyben egy azonosító és két időpecsét attribútum mellett egy `username`, egy `password` és egy `email` azonosítójú string típusú attribútum szerepelne. Jelző attribútumot elfelejtettünk megadni, de ez nem fog problémát okozni, mert a sémánk verziókezelt.

```

class CreateUsers < ActiveRecord::Migration
  def change
    create_table :users do |t|
      t.string :username
      t.string :password
      t.string :email

      t.timestamps
    end
  end
end

```

Ezután elvégezhetjük a tábla struktúrájának módosítását, amiben a `rake` parancs nyújt segítséget. A `rake db:create` létrehozza a táblákat, ha még nem tettük volna meg, a `rake db:migrate` módosítja a séma struktúráját.

```

kovacs@debian:~/gyakorlat$ rake db:create
(in /home/kovacs/gyakorlat)

```

```

kovacsg@debian:~/gyakorlat$
kovacsg@debian:~/gyakorlat$ rake db:migrate
(in /home/kovacsg/gyakorlat)
== 20141001111344 CreateUsers: migrating

```

Az adatbáziskezelővel a `gyakorlat_development` adatbázist kiválasztva ezután ellenőrizhetjük, hogy a táblánk valóban létrejött. Alább ennek ellenőrzése látható a két adatbáziskezelő konzolján.

```

kovacsg@debian:~/gyakorlat/db$ sqlite3 development.sqlite3
SQLite version 3.7.13 2012-06-11 02:05:22
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .schema
CREATE TABLE "schema_migrations" ("version" varchar(255) NOT NULL);
CREATE TABLE "users" ("id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, "
  username" varchar(255), "password" varchar(255), "email" varchar(255), "
  created_at" datetime, "updated_at" datetime);
CREATE UNIQUE INDEX "unique_schema_migrations" ON "schema_migrations" ("
  version");
sqlite>
root@debian:~# mysql -u root
mysql> use gyakorlat_development;
Database changed
mysql> show tables;
+-----+
| Tables_in_gyakorlat_development |
+-----+
| schema_migrations |
| users |
+-----+
2 rows in set (0.00 sec)

mysql> describe users;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| username | varchar(255) | YES | | NULL | |
| password | varchar(255) | YES | | NULL | |
| email | varchar(255) | YES | | NULL | |
| created_at | datetime | YES | | NULL | |
| updated_at | datetime | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

Hivatkozások

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