

A Rails keretrendszer Gyakorlat

Kovács Gábor

2013. október 2.

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 gyakorlaton használt virtuális gép Linux disztribúciójában a gyakorlat időpontjában nem érhető el csomag formájában a Ruby legújabb, 1.9.3-as verziója, így az 1.9.2-es verziót használjuk helyette. Ennek következménye, hogy a Rails környezetből sem fogjuk tudni a legfrissebb, 4-es változatot.

A Rails környezet telepítéséhez egy Ruby környezet szükséges, ami mellé kiegészítőként telepítjük az RI dokumentációs rendszert és a natív kiegészí-

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.

tések fordításához szükséges Ruby fejlesztői függvénykönyvtárakat. Ezt a következő paranccsal telepíthetjük rendszergazdaként:

```
kovacsg@debian:~$ sudo bash
[sudo] password for kovacsg:
root@debian:/home/kovacsg# su -
root@debian:~# apt-get install ruby1.9.1 ruby1.9.1-dev ri1.9.1
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libreadline5 libruby1.9.1 libyaml-0-2
Suggested packages:
  ruby1.9.1-examples graphviz
The following NEW packages will be installed:
  libreadline5 libruby1.9.1 libyaml-0-2 ri1.9.1 ruby1.9.1 ruby1.9.1-dev
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 9,416 kB of archives.
After this operation, 65.3 MB of additional disk space will be used.
Do you want to continue [Y/n]?
Get:1 http://ftp.hu.debian.org/debian/ stable/main libreadline5 amd64 5.2-7
[146 kB]
Get:2 http://ftp.hu.debian.org/debian/ stable/main libyaml-0-2 amd64 0.1.3-1
[54.6 kB]
Get:3 http://ftp.hu.debian.org/debian/ stable/main libruby1.9.1 amd64
1.9.2.0-2 [4,517 kB]
Get:4 http://ftp.hu.debian.org/debian/ stable/main ruby1.9.1 amd64 1.9.2.0-2
[754 kB]
Get:5 http://ftp.hu.debian.org/debian/ stable/main ri1.9.1 all 1.9.2.0-2
[2,093 kB]
Get:6 http://ftp.hu.debian.org/debian/ stable/main ruby1.9.1-dev amd64
1.9.2.0-2 [1,852 kB]
Fetched 9,416 kB in 27s (339 kB/s)
Selecting previously unselected package libreadline5.
(Reading database ... 108841 files and directories currently installed.)
Unpacking libreadline5 (from .../libreadline5_5.2-7_amd64.deb) ...
Selecting previously unselected package libyaml-0-2.
Unpacking libyaml-0-2 (from .../libyaml-0-2_0.1.3-1_amd64.deb) ...
Selecting previously unselected package libruby1.9.1.
Unpacking libruby1.9.1 (from .../libruby1.9.1_1.9.2.0-2_amd64.deb) ...
Selecting previously unselected package ruby1.9.1.
Unpacking ruby1.9.1 (from .../ruby1.9.1_1.9.2.0-2_amd64.deb) ...
Selecting previously unselected package ri1.9.1.
Unpacking ri1.9.1 (from .../ri1.9.1_1.9.2.0-2_all.deb) ...
Selecting previously unselected 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 libreadline5 (5.2-7) ...
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:~# ruby --version
ruby 1.9.2p0 (2010-08-18 revision 29036) [x86_64-linux]
root@debian:~# gem --version
1.3.7
```

```
root@debian:~# gem list
```

```
*** LOCAL GEMS ***
```

A következő lépés a Rails keretrendszer telepítése, amit a `gem` Ruby csomagkezelővel teszünk meg. A gyakorlat napján aktuális verzió összeakadt a `rack` csomag telepítendő verziójával, így annak korábbi változatát külön telepítjük. A `rails` telepítése maga után vonja a legfontosabb kiegészítőket is, illetve a hozzájuk kapcsolódó dokumentációt.

```
root@debian:~# gem install rack -v 1.4.5
Successfully installed rack-1.4.5
1 gem installed
Installing ri documentation for rack-1.4.5...
Installing RDoc documentation for rack-1.4.5...
```

Mivel csak 1.9.2-es Ruby értelmezőnk van, a Rails 4 helyett a Rails 3.2-t telepítjük a `gem` csomagkezelővel.

```
root@debian:~# gem install rails -v 3.2.12
Building native extensions. This could take a while...
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 i18n-0.6.5
Successfully installed multi_json-1.8.0
Successfully installed activesupport-3.2.12
Successfully installed builder-3.0.4
Successfully installed activemodel-3.2.12
Successfully installed rack-cache-1.2
Successfully installed rack-test-0.6.2
Successfully installed journey-1.0.4
Successfully installed hike-1.2.3
Successfully installed tilt-1.4.1
Successfully installed sprockets-2.2.2
Successfully installed erubis-2.7.0
Successfully installed actionpack-3.2.12
Successfully installed arel-3.0.2
Successfully installed tzinfo-0.3.37
Successfully installed activerecord-3.2.12
Successfully installed activerecord-3.2.12
Successfully installed mime-types-1.25
Successfully installed polyglot-0.3.3
Successfully installed treetop-1.4.15
Successfully installed mail-2.4.4
Successfully installed actionmailer-3.2.12
Successfully installed rake-10.1.0
Successfully installed rack-ssl-1.3.3
Successfully installed thor-0.18.1
Successfully installed json-1.8.0
Successfully installed rdoc-3.12.2
Successfully installed railties-3.2.12
Successfully installed bundler-1.3.5
Successfully installed rails-3.2.12
30 gems installed
Installing ri documentation for i18n-0.6.5...
```

```
Installing ri documentation for multi_json-1.8.0...
Installing ri documentation for activesupport-3.2.12...
Installing ri documentation for builder-3.0.4...
Installing ri documentation for activemodel-3.2.12...
Installing ri documentation for rack-cache-1.2...
Installing ri documentation for rack-test-0.6.2...
Installing ri documentation for journey-1.0.4...
Installing ri documentation for hike-1.2.3...
Installing ri documentation for tilt-1.4.1...
Installing ri documentation for sprockets-2.2.2...
Installing ri documentation for erubis-2.7.0...
Installing ri documentation for actionpack-3.2.12...
Installing ri documentation for arel-3.0.2...
Installing ri documentation for tzinfo-0.3.37...
Installing ri documentation for activerecord-3.2.12...
Installing ri documentation for activeresource-3.2.12...
Installing ri documentation for mime-types-1.25...
Installing ri documentation for polyglot-0.3.3...
Installing ri documentation for treetop-1.4.15...
Installing ri documentation for mail-2.4.4...
Installing ri documentation for actionmailer-3.2.12...
Installing ri documentation for rake-10.1.0...
Installing ri documentation for rack-ssl-1.3.3...
Installing ri documentation for thor-0.18.1...
Installing ri documentation for json-1.8.0...
Installing ri documentation for rdoc-3.12.2...
Installing ri documentation for railties-3.2.12...
Installing ri documentation for bundler-1.3.5...
Installing ri documentation for rails-3.2.12...
Installing RDoc documentation for i18n-0.6.5...
Installing RDoc documentation for multi_json-1.8.0...
Installing RDoc documentation for activesupport-3.2.12...
Installing RDoc documentation for builder-3.0.4...
Installing RDoc documentation for activemodel-3.2.12...
Installing RDoc documentation for rack-cache-1.2...
Installing RDoc documentation for rack-test-0.6.2...
Installing RDoc documentation for journey-1.0.4...
Installing RDoc documentation for hike-1.2.3...
Installing RDoc documentation for tilt-1.4.1...
Installing RDoc documentation for sprockets-2.2.2...
Installing RDoc documentation for erubis-2.7.0...
Installing RDoc documentation for actionpack-3.2.12...
Installing RDoc documentation for arel-3.0.2...
Installing RDoc documentation for tzinfo-0.3.37...
Installing RDoc documentation for activerecord-3.2.12...
Installing RDoc documentation for activeresource-3.2.12...
Installing RDoc documentation for mime-types-1.25...
Installing RDoc documentation for polyglot-0.3.3...
Installing RDoc documentation for treetop-1.4.15...
Installing RDoc documentation for mail-2.4.4...
Installing RDoc documentation for actionmailer-3.2.12...
Installing RDoc documentation for rake-10.1.0...
Installing RDoc documentation for rack-ssl-1.3.3...
Installing RDoc documentation for thor-0.18.1...
Installing RDoc documentation for json-1.8.0...
Installing RDoc documentation for rdoc-3.12.2...
Installing RDoc documentation for railties-3.2.12...
Installing RDoc documentation for bundler-1.3.5...
Installing RDoc documentation for rails-3.2.12...
```

Railshez egy opcionális, ám az éles rendszerek szempontjából annál hasz-

nosabb komponens az Apache webserverral való integrációt lehetővé tevő Passenger plugin.

```
root@debian:~# gem install passenger
Building native extensions. This could take a while...
Successfully installed daemon_controller-1.1.5
Successfully installed passenger-4.0.19
2 gems installed
Installing ri documentation for daemon_controller-1.1.5...
Installing ri documentation for passenger-4.0.19...
Installing RDoc documentation for daemon_controller-1.1.5...
Installing RDoc documentation for passenger-4.0.19...
```

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 libmysqlclient-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libmysqlclient18 libsqlite3-0
Suggested packages:
  sqlite3-doc
The following NEW packages will be installed:
  libmysqlclient-dev libmysqlclient18 libsqlite3-dev
The following packages will be upgraded:
  libsqlite3-0
1 upgraded, 3 newly installed, 0 to remove and 791 not upgraded.
Need to get 2,672 kB of archives.
After this operation, 10.6 MB of additional disk space will be used.
Do you want to continue [Y/n]?
Get:1 http://ftp.hu.debian.org/debian/ stable/main libsqlite3-0 amd64
    3.7.13-1+deb7u1 [455 kB]
Get:2 http://ftp.hu.debian.org/debian/ stable/main libmysqlclient18 amd64
    5.5.31+dfsg-0+wheezy1 [687 kB]
Get:3 http://ftp.hu.debian.org/debian/ stable/main libmysqlclient-dev amd64
    5.5.31+dfsg-0+wheezy1 [952 kB]
Get:4 http://ftp.hu.debian.org/debian/ stable/main libsqlite3-dev amd64
    3.7.13-1+deb7u1 [578 kB]
Fetched 2,672 kB in 10s (244 kB/s)
Reading changelogs... Done
(Reading database ... 120299 files and directories currently installed.)
Preparing to replace libsqlite3-0 3.7.3-1 (using .../libsqlite3-0_3.7.13-1+
deb7u1_amd64.deb) ...
Unpacking replacement libsqlite3-0 ...
Selecting previously unselected package libmysqlclient18.
Unpacking libmysqlclient18 (from .../libmysqlclient18_5.5.31+dfsg-0+
wheezy1_amd64.deb) ...
Selecting previously unselected package libmysqlclient-dev.
Unpacking libmysqlclient-dev (from .../libmysqlclient-dev_5.5.31+dfsg-0+
wheezy1_amd64.deb) ...
Selecting previously unselected package libsqlite3-dev.
Unpacking libsqlite3-dev (from .../libsqlite3-dev_3.7.13-1+deb7u1_amd64.deb)
...
Processing triggers for man-db ...
Setting up libsqlite3-0 (3.7.13-1+deb7u1) ...
Setting up libmysqlclient18 (5.5.31+dfsg-0+wheezy1) ...
Setting up libmysqlclient-dev (5.5.31+dfsg-0+wheezy1) ...
```

```
Setting up libsqlite3-dev (3.7.13-1+deb7u1) ...
```

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

```
root@debian:~# gem install sqlite3
Building native extensions. This could take a while...
Successfully installed sqlite3-1.3.8
1 gem installed
Installing ri documentation for sqlite3-1.3.8...
Installing RDoc documentation for sqlite3-1.3.8...
root@debian:~# gem install mysql2
Building native extensions. This could take a while...
Successfully installed mysql2-0.3.13
1 gem installed
Installing ri documentation for mysql2-0.3.13...
Installing RDoc documentation for mysql2-0.3.13...
```

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 JavaScript interpreter.

```
root@debian:~# gem install execjs
Successfully installed execjs-2.0.2
1 gem installed
Installing ri documentation for execjs-2.0.2...
Installing RDoc documentation for execjs-2.0.2...
root@debian:~# gem install therubyracer
Building native extensions. This could take a while...
Successfully installed ref-1.0.5
Successfully installed libv8-3.16.14.3-x86_64-linux
Successfully installed therubyracer-0.12.0
3 gems installed
Installing ri documentation for ref-1.0.5...
Installing ri documentation for libv8-3.16.14.3-x86_64-linux...
Installing ri documentation for therubyracer-0.12.0...
Installing RDoc documentation for ref-1.0.5...
Installing RDoc documentation for libv8-3.16.14.3-x86_64-linux...
Installing RDoc documentation for therubyracer-0.12.0...
```

A Rails rendszerünk ezután az alábbi Ruby függvénykönyvtárakból áll

```
root@debian:~# gem list

*** LOCAL GEMS ***

actionmailer (3.2.12)
actionpack (3.2.12)
activemodel (3.2.12)
activerecord (3.2.12)
activeresource (3.2.12)
activesupport (3.2.12)
arel (3.0.2)
atomic (1.1.14)
builder (3.0.4)
bundler (1.3.5)
coffee-rails (3.2.2)
coffee-script (2.2.0)
coffee-script-source (1.6.3)
daemon_controller (1.1.5)
```

```
erubis (2.7.0)
execjs (2.0.2)
hike (1.2.3)
i18n (0.6.5)
journey (1.0.4)
jquery-rails (3.0.4)
json (1.8.0)
libv8 (3.16.14.3 x86_64-linux)
mail (2.4.4)
mime-types (1.25)
minitest (4.7.5)
multi_json (1.8.0)
mysql2 (0.3.13)
passenger (4.0.19)
polyglot (0.3.3)
rack (1.4.5)
rack-cache (1.2)
rack-ssl (1.3.3)
rack-test (0.6.2)
rails (3.2.12)
railties (3.2.12)
rake (10.1.0)
rdoc (3.12.2)
ref (1.0.5)
sass (3.2.11)
sass-rails (3.2.6)
sprockets (2.2.2)
sqlite3 (1.3.8)
therubyracer (0.12.0)
thor (0.18.1)
thread_safe (0.1.3)
tilt (1.4.1)
treetop (1.4.15)
tzinfo (0.3.37)
uglifier (2.2.1)
```

2. Rails alkalmazás létrehozása

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

```
kovacsg@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 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/ruby
-b, [--builder=BUILDER] # Path to a application builder (can be a
                        filesystem path or URL)
-m, [--template=TEMPLATE] # Path to an application template (can be a
                        filesystem path or URL)
  [--skip-gemfile] # Don't create a Gemfile
  [--skip-bundle] # Don't run bundle install
-G, [--skip-git] # Skip Git ignores and keeps
-O, [--skip-active-record] # Skip Active Record files
-S, [--skip-sprockets] # Skip Sprockets files
-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] # Skip JavaScript files
  [--dev] # Setup the application with Gemfile
  pointing to your Rails checkout
  [--edge] # Setup the application with Gemfile
  pointing to Rails repository
-T, [--skip-test-unit] # Skip Test::Unit files
  [--old-style-hash] # Force using old style hash (:foo => 'bar
  ') on Ruby >= 1.9

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:
-h, [--help] # Show this help message and quit
-v, [--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 webszerveret. Egy új Rails alkalmazást a rails parancsnak 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 lefut a Rails keretrendszer egy másik parancsa, a `bundle`, amely az `install` opció hatására összeszedi a gemek közül azokat, amelyekre az alkalmazásunknak szüksége lesz. 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.

```
kovacs@debian:~$ rails new gyakorlat_sqlite
create
create  README.rdoc
create  Rakefile
create  config.ru
create  .gitignore
create  Gemfile
create  app
create  app/assets/images/rails.png
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/mailers/.gitkeep
create  app/models/.gitkeep
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/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  doc
create  doc/README_FOR_APP
create  lib
create  lib/tasks
create  lib/tasks/.gitkeep
create  lib/assets
create  lib/assets/.gitkeep
create  log
create  log/.gitkeep
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 script
create script/rails
create test/fixtures
create test/fixtures/.gitkeep
create test/functional
create test/functional/.gitkeep
create test/integration
create test/integration/.gitkeep
create test/unit
create test/unit/.gitkeep
create test/performance/browsing_test.rb
create test/test_helper.rb
create tmp/cache
create tmp/cache/assets
create vendor/assets/javascripts
create vendor/assets/javascripts/.gitkeep
create vendor/assets/stylesheets
create vendor/assets/stylesheets/.gitkeep
create vendor/plugins
create vendor/plugins/.gitkeep
run bundle install
Fetching source index from https://rubygems.org/
Could not fetch specs from https://rubygems.org/
kovacs@debian:~/gyakorlat_sqlite$ bundle update
Resolving dependencies...
Using rake (10.1.0)
Using i18n (0.6.5)
Using multi_json (1.8.0)
Using activesupport (3.2.12)
Using builder (3.0.4)
Using activemodel (3.2.12)
Using erubis (2.7.0)
Using journey (1.0.4)
Using rack (1.4.5)
Using rack-cache (1.2)
Using rack-test (0.6.2)
Using hike (1.2.3)
Using tilt (1.4.1)
Using sprockets (2.2.2)
Using actionpack (3.2.12)
Using mime-types (1.25)
Using polyglot (0.3.3)
Using treetop (1.4.15)
Using mail (2.4.4)
Using actionmailer (3.2.12)
Using arel (3.0.2)
Using tzinfo (0.3.37)
Using activerecord (3.2.12)
Using activerecord (3.2.12)
Using bundler (1.3.5)
Using coffee-script-source (1.6.3)
Using execjs (2.0.2)
Using coffee-script (2.2.0)
Using rack-ssl (1.3.3)
Using json (1.8.0)
Using rdoc (3.12.2)
Using thor (0.18.1)

```

```
Using railties (3.2.12)
Using coffee-rails (3.2.2)
Using jquery-rails (3.0.4)
Using libv8 (3.16.14.3)
Using rails (3.2.12)
Using ref (1.0.5)
Using sass (3.2.11)
Using sass-rails (3.2.6)
Using sqlite3 (1.3.8)
Using therubyracer (0.12.0)
Using uglifier (2.2.1)
Your bundle is updated!
```

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 szerializált 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 3.1-es verzió JavaScript változásai miatt módosítanunk kell a `Gemfile` állományt megadva a `therubyracer`-t mint interpretert. Az állomány tetszőleges helyére a következő sort szúrjuk be:

```
gem 'execjs'
gem 'therubyracer'
```

3. Web- és adatbázisszerver kapcsolat beállítása

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

A beépített webservert 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 szerver elérhető a `http://localhost:3000` webcímen, lásd ??-
ábra.

```
kovacs@debian:~/gyakorlat$ rails server
=> Booting WEBrick
=> Rails 3.2.12 application starting in development on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
[2013-10-02 13:03:48] INFO WEBrick 1.3.1
[2013-10-02 13:03:48] INFO ruby 1.9.2 (2010-08-18) [x86_64-linux]
[2013-10-02 13:03:48] INFO WEBrick::HTTPServer#start: pid=20448 port=30000
```

A Rails a már említett Passenger plugin segítségével illeszhető Apache webservertől. Ehhez a következő parancsot kell kiadnunk. A parancs lefordítja, majd telepíti az Apache webservert Rails moduljára, 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 Apache virtuális hoszt beállításai ezek alapján a következő lehet:

```
<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 MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>
```

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 konfigurációs paraméterei 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'
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
```

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
(in /home/kovacs/gyakorlat)
rake about # List versions of all Rails frameworks and the
           environment
rake assets:clean # Remove compiled assets
rake assets:precompile # Compile all the assets named in config/assets.
precompile
rake db:create # Create the database from DATABASE_URL or config/
database.yml for t...
rake db:drop # Drops the database using DATABASE_URL or the
current Rails.env (us...
rake db:fixtures:load # Load fixtures into the current environment's
database
rake db:migrate # Migrate the database (options: VERSION=x, VERBOSE=
false)
rake db:migrate:status # Display status of migrations
rake db:rollback # Rolls the schema back to the previous version (
specify steps w/ ST...
rake db:schema:dump # Create a db/schema.rb file that can be portably
used against any D...
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...
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,...
rake log:clear # Truncates all *.log files in log/ to zero bytes
rake middleware # Prints out your Rack middleware stack
rake notes # Enumerate all annotations (use notes:optimize, :
fixme, :todo for f...
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 ju...
rake routes # Print out all defined routes in match order, with
names
rake secret # Generate a cryptographically secure secret key (
this is typically ...
rake stats # Report code statistics (KLOCs, etc) from the
application
rake test # Runs test:units, test:functionals, test:
integration together (also...
rake test:recent # Run tests for {:recent=>"test:prepare"} / Test
recent changes
rake test:single # Run tests for {:single=>"test:prepare"}
rake test:uncommitted # Run tests for {:uncommitted=>"test:prepare"} /
Test changes since ...
rake time:zones:all # Displays all time zones, also available: time:
```

```
zones:us, time:zones...
rake tmp:clear          # Clear session, cache, and socket files from tmp/ (
  narrow w/ tmp:se...
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. Az alábbi kódrészlet a legáltalánosabb beállítást tartalmazza. A webszerver 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.

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

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

Hozzunk létre egy új controllert a `rails 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 könyvát 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$ 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/functional/say_controller_test.rb
invoke  helper
create  app/helpers/say_helper.rb
invoke  test_unit
create  test/unit/helpers/say_helper_test.rb
```

```
invoke assets
invoke coffee
create app/assets/javascripts/say.js.coffee
invoke scss
create app/assets/stylesheets/say.css.scss
```

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.

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

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.

```
<p><%= @time %></p>
```

A controllernek és akciónak megfelelő megjelenített oldalt a `??`. ábra mutatja.

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

```
<!DOCTYPE html>
<html>
<head>
  <title>Gyakorlat</title>
  <link href="/assets/application.css?body=1" media="all" rel="stylesheet"
    type="text/css" />
  <link href="/assets/say.css?body=1" media="all" rel="stylesheet" type="text/
    css" />
  <script src="/assets/jquery.js?body=1" type="text/javascript"></script>
  <script src="/assets/jquery_ujs.js?body=1" type="text/javascript"></script>
```



```

<script src="/assets/say.js?body=1" type="text/javascript"></script>
<script src="/assets/application.js?body=1" type="text/javascript"></script>

  <meta content="authenticity_token" name="csrf-param" />
<meta content="IOYYdCvoEvugQc6Acl/6MR34d204fzzK9dnoshT6ZEM=" name="csrf-
  token" />
</head>
<body>

<h1>Hello , world!</h1>
<p>2013-10-02 13:22:46 +0200</p>

</body>
</html>

```

A Rails MVC filozófiájának harmadik eleme a modell, amelyet szintén a `rails generate` parancs `generate` opciójával hozhatunk létre. A harmadik argumentum a modell osztály 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 user username:string email:
string password:string account:integer bank_account:integer
  invoke  active_record
  create  db/migrate/20131002112642_create_users.rb
  create  app/models/user.rb
  invoke  test_unit
  create  test/unit/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 `neptun`, egy `passwd` és egy `email` azonosítójú string típusú attribútum szerepelne.

```

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

      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/db/migrate$ rake db:migrate
(in /home/kovacs/gyakorlat)
== CreateUsers: migrating

-----
-- create_table(:users)
--> 0.0066s
== CreateUsers: migrated (0.0073s)

-----
```

Az adatbáziskezelővel a `gyakorlat_development` adatbázist kiválasztva ezután ellenőrizhetjük, hogy a táblánk valóban létrejött.

```
kovacs@debian:~/gyakorlat$ mysql -u root
mysql> use gyakorlat_development;
Database changed
mysql> show tables;
Empty set (0.00 sec)
```

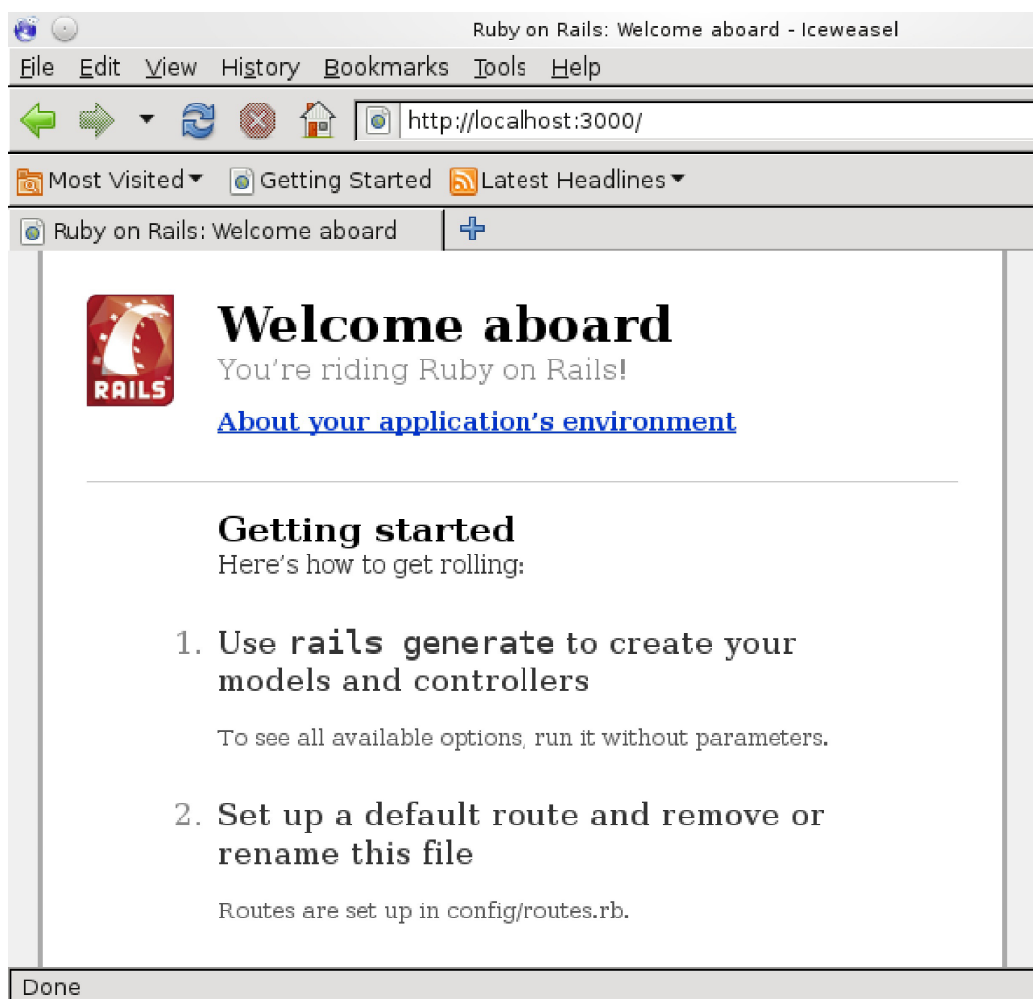
```
mysql> show tables;
```

```
+-----+
| Tables_in_gyakorlat_development |
+-----+
| schema_migrations                |
| users                             |
+-----+
```

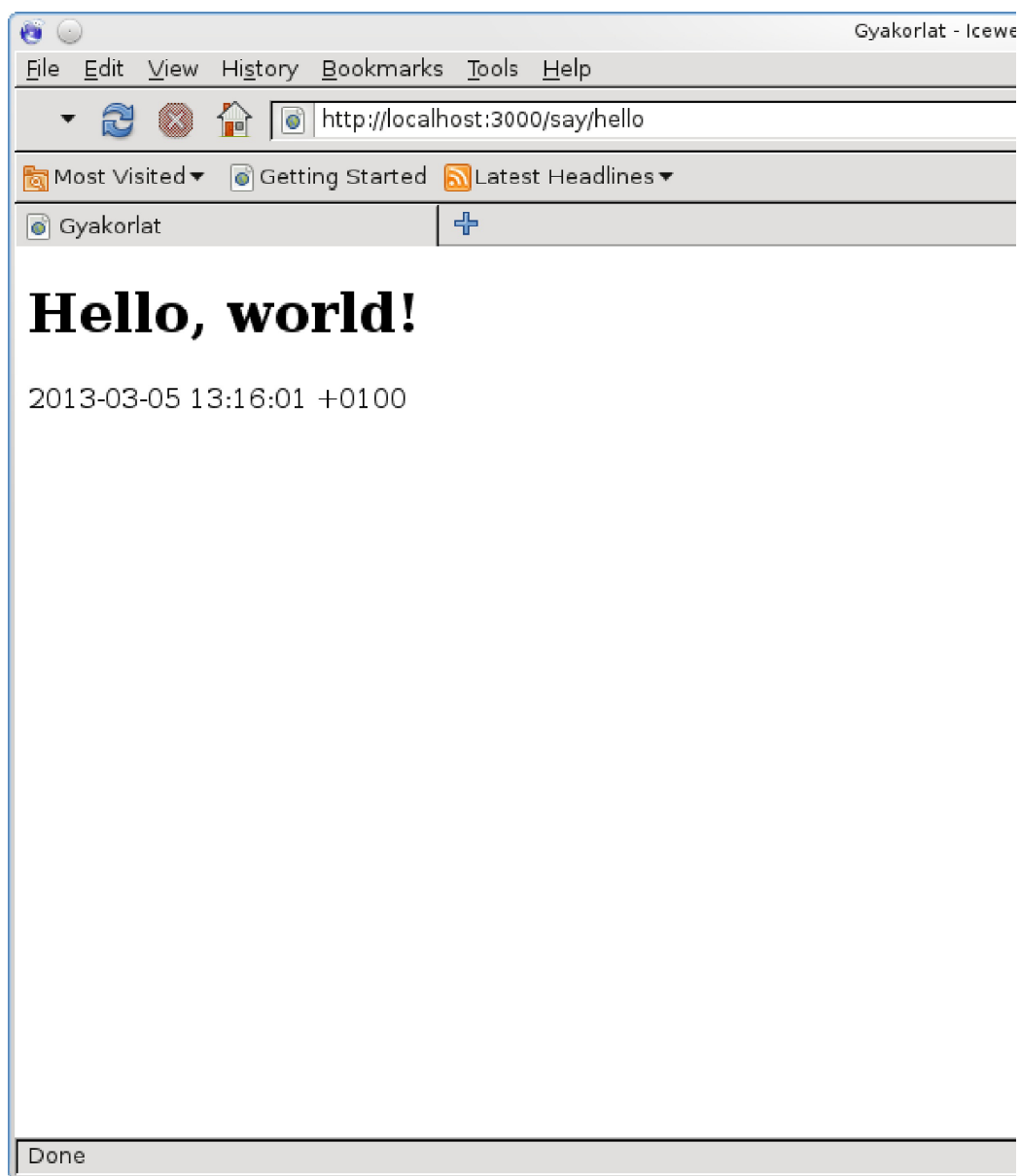
```
2 rows in set (0.00 sec)
```

```
mysql> describe users;
```

| Field | Type | Null | Key | Default | Extra |
|--------------|--------------|------|-----|---------|----------------|
| id | int(11) | NO | PRI | NULL | auto_increment |
| username | varchar(255) | YES | | NULL | |
| email | varchar(255) | YES | | NULL | |
| password | varchar(255) | YES | | NULL | |
| account | int(11) | YES | | NULL | |
| bank_account | int(11) | YES | | NULL | |
| created_at | datetime | NO | | NULL | |
| updated_at | datetime | NO | | NULL | |



1. ábra. A beágyazott webszerver elérése



2. ábra. A hello akció megjelenítve