

# A Rails keretrendszer Gyakorlat

Kovács Gábor

2017. február 28.

## 1. Ruby telepítése, Gemek

A Rails keretrendszert egy virtuális környezetben <sup>1</sup> futó Debian Linuxra <sup>2</sup> 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 környezetben 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 <sup>3</sup>, 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, 2.3-s verzióját használjuk <sup>4</sup>, amely előfeltétele a tavasszal

---

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

<sup>2</sup>Mivel 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.

<sup>3</sup>A Rails egyik függvénykönyvtára miatt szükségünk lesz még a `zlib1g` csomagra is.

<sup>4</sup>Az `apt-get install -t sid` parancsban az utóbbi kapcsoló a Debian legfrissebb csomagokat tartalmazó disztribúcióját választja ki, amelyben már elérhető a 2.3-as Ruby verzió. Ubuntu esetén ez elhagyandó.

megjelent Rails 5-ös változatának telepítésének. A C függvénykönyvtárhoz való illesztéshez szükségünk lesz a Ruby header fájlokra is, ezért telepítjük azok Linux csomagját is. Ezt a következő paranccsal telepíthetjük rendszergazdaként:

```
kovacs@debian:~$ ruby
bash: ruby: command not found
kovacs@debian:~$ sudo bash
[sudo] password for kovacs:
root@debian:/home/kovacs# su -
root@debian:~# apt-get install -t sid ruby ruby-dev ri
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libreadline7 libruby2.3 rake ruby2.3 ruby2.3-dev
The following NEW packages will be installed:
  libreadline7 libruby2.3 rake ri ruby ruby-dev ruby2.3 ruby2.3-dev
0 upgraded, 8 newly installed, 0 to remove and 1168 not upgraded.
Need to get 4,699 kB of archives.
After this operation, 19.8 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 libreadline7 amd64 7.0-2
[151 kB]
Get:2 http://ftp.hu.debian.org/debian/ sid/main libruby2.3 amd64 2.3.3-1
[3,104 kB]
Get:3 http://ftp.hu.debian.org/debian/ sid/main ruby2.3 amd64 2.3.3-1 [186
kB]
Get:4 http://ftp.hu.debian.org/debian/ sid/main ruby amd64 1:2.3.3 [10.8 kB]
Get:5 http://ftp.hu.debian.org/debian/ sid/main rake all 10.5.0-2 [49.4 kB]
Get:6 http://ftp.hu.debian.org/debian/ sid/main ri all 1:2.3.3 [9,476 B]
Get:7 http://ftp.hu.debian.org/debian/ sid/main ruby2.3-dev amd64 2.3.3-1
[1,179 kB]
Get:8 http://ftp.hu.debian.org/debian/ sid/main ruby-dev amd64 1:2.3.3
[9,574 B]
Fetched 4,699 kB in 0s (5,144 kB/s)
Selecting previously unselected package libreadline7:amd64.
(Reading database ... 162300 files and directories currently installed.)
Preparing to unpack .../libreadline7_7.0-2_amd64.deb ...
Unpacking libreadline7:amd64 (7.0-2) ...
Selecting previously unselected package libruby2.3:amd64.
Preparing to unpack .../libruby2.3_2.3.3-1_amd64.deb ...
Unpacking libruby2.3:amd64 (2.3.3-1) ...
Selecting previously unselected package ruby2.3.
Preparing to unpack .../ruby2.3_2.3.3-1_amd64.deb ...
Unpacking ruby2.3 (2.3.3-1) ...
Selecting previously unselected package ruby.
Preparing to unpack .../ruby_1%3a2.3.3_amd64.deb ...
Unpacking ruby (1:2.3.3) ...
Selecting previously unselected package rake.
Preparing to unpack .../archives/rake_10.5.0-2_all.deb ...
Unpacking rake (10.5.0-2) ...
Selecting previously unselected package ri.
Preparing to unpack .../archives/ri_1%3a2.3.3_all.deb ...
Unpacking ri (1:2.3.3) ...
Selecting previously unselected package ruby2.3-dev:amd64.
Preparing to unpack .../ruby2.3-dev_2.3.3-1_amd64.deb ...
Unpacking ruby2.3-dev:amd64 (2.3.3-1) ...
Selecting previously unselected package ruby-dev:amd64.
Preparing to unpack .../ruby-dev_1%3a2.3.3_amd64.deb ...
Unpacking ruby-dev:amd64 (1:2.3.3) ...
```

```

Processing triggers for libc-bin (2.19-18+deb8u7) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up libreadline7:amd64 (7.0-2) ...
Setting up libruby2.3:amd64 (2.3.3-1) ...
Setting up ruby2.3 (2.3.3-1) ...
Setting up ri (1:2.3.3) ...
Setting up ruby2.3-dev:amd64 (2.3.3-1) ...
Setting up ruby-dev:amd64 (1:2.3.3) ...
Setting up ruby (1:2.3.3) ...
Setting up rake (10.5.0-2) ...
Processing triggers for libc-bin (2.19-18+deb8u7) ...

```

Ezután ellenőrizzük, hogy elérhető-e a ruby értelmező és a gem Ruby csomagkezelő, és nézzük meg az előre telepített Ruby API-k listáját.

```

root@debian:~# ruby --version
ruby 2.3.3p222 (2016-11-21) [x86_64-linux-gnu]
root@debian:~# gem --version
2.5.2
root@debian:~# gem list

*** LOCAL GEMS ***

bigdecimal (1.2.8)
did_you_mean (1.0.0)
io-console (0.4.5)
json (1.8.3)
minitest (5.9.0)
net-telnet (0.1.1)
power_assert (0.2.7)
psych (2.1.0)
rake (10.5.0)
rdoc (4.2.1)
test-unit (3.1.7)

```

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. Először azonban hibára futottunk, mert az egyik szükséges C fejlesztői függvénykönyvtár nem volt elérhető.

```

root@debian:~# gem install rails --no-rdoc --no-ri
Fetching: i18n-0.8.1.gem (100%)
Successfully installed i18n-0.8.1
Fetching: thread_safe-0.3.6.gem (100%)
Successfully installed thread_safe-0.3.6
Fetching: tzinfo-1.2.2.gem (100%)
Successfully installed tzinfo-1.2.2
Fetching: concurrent-ruby-1.0.5.gem (100%)
Successfully installed concurrent-ruby-1.0.5
Fetching: activesupport-5.0.1.gem (100%)
Successfully installed activesupport-5.0.1
Fetching: rack-2.0.1.gem (100%)
Successfully installed rack-2.0.1
Fetching: rack-test-0.6.3.gem (100%)
Successfully installed rack-test-0.6.3

```

```

Fetching: mini_portile2-2.1.0.gem (100%)
Successfully installed mini_portile2-2.1.0
Fetching: nokogiri-1.7.0.1.gem (100%)
Building native extensions. This could take a while...
ERROR: Error installing rails:
      ERROR: Failed to build gem native extension.

      current directory: /var/lib/gems/2.3.0/gems/nokogiri-1.7.0.1/ext/
      nokogiri
/usr/bin/ruby2.3 -r ./siteconf20170227-4830-7e5e9w.rb extconf.rb
checking if the C compiler accepts ... yes
Building nokogiri using packaged libraries.
Using mini_portile version 2.1.0
checking for gzdopen() in -lz... no
zlib is missing; necessary for building libxml2
*** extconf.rb failed ***
Could not create Makefile due to some reason, probably lack of necessary
libraries and/or headers. Check the mkmf.log file for more details. You
may
need configuration options.

Provided configuration options:
  --with-opt-dir
  --without-opt-dir
  --with-opt-include
  --without-opt-include=${opt-dir}/include
  --with-opt-lib
  --without-opt-lib=${opt-dir}/lib
  --with-make-prog
  --without-make-prog
  --srcdir=.
  --curdir
  --ruby=/usr/bin/$(RUBY_BASE_NAME) 2.3
  --help
  --clean
  --use-system-libraries
  --enable-static
  --disable-static
  --with-zlib-dir
  --without-zlib-dir
  --with-zlib-include
  --without-zlib-include=${zlib-dir}/include
  --with-zlib-lib
  --without-zlib-lib=${zlib-dir}/lib
  --enable-cross-build
  --disable-cross-build

To see why this extension failed to compile, please check the mkmf.log which
can be found here:

  /var/lib/gems/2.3.0/extensions/x86_64-linux/2.3.0/nokogiri-1.7.0.1/mkmf.
  log

extconf failed, exit code 1

Gem files will remain installed in /var/lib/gems/2.3.0/gems/nokogiri-1.7.0.1
for inspection.
Results logged to /var/lib/gems/2.3.0/extensions/x86_64-linux/2.3.0/nokogiri
-1.7.0.1/gem_make.out

```

Az XML API telepítése során történt a hiba. A „Building native exten-

sions.” arra utal, hogy a Ruby függvénykönyvtár C függvénykönyvtárhoz történő illesztésekor történt a baleset, a hibaüzenetből látszik, hogy a zlib nevű csomag hiányzik, amit telepítünk:

```
root@debian:~# apt-get install zlib1g-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  zlib1g-dev
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 206 kB of archives.
After this operation, 462 kB of additional disk space will be used.
Get:1 http://ftp.hu.debian.org/debian/ stable/main zlib1g-dev amd64 1:1.2.8.dfsg-2+b1 [206 kB]
Fetched 206 kB in 0s (1,993 kB/s)
Selecting previously unselected package zlib1g-dev:amd64.
(Reading database ... 163436 files and directories currently installed.)
Preparing to unpack .../zlib1g-dev_1%3a1.2.8.dfsg-2+b1_amd64.deb ...
Unpacking zlib1g-dev:amd64 (1:1.2.8.dfsg-2+b1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up zlib1g-dev:amd64 (1:1.2.8.dfsg-2+b1) ...
```

Ezután folytathatjuk a Rails telepítését, amely sikeresen meg is történik:

```
root@debian:~# gem install rails --no-rdoc --no-ri
Building native extensions. This could take a while...
Successfully installed nokogiri-1.7.0.1
Fetching: loofah-2.0.3.gem (100%)
Successfully installed loofah-2.0.3
Fetching: rails-html-sanitizer-1.0.3.gem (100%)
Successfully installed rails-html-sanitizer-1.0.3
Fetching: rails-dom-testing-2.0.2.gem (100%)
Successfully installed rails-dom-testing-2.0.2
Fetching: builder-3.2.3.gem (100%)
Successfully installed builder-3.2.3
Fetching: erubis-2.7.0.gem (100%)
Successfully installed erubis-2.7.0
Fetching: actionview-5.0.1.gem (100%)
Successfully installed actionview-5.0.1
Fetching: actionpack-5.0.1.gem (100%)
Successfully installed actionpack-5.0.1
Fetching: activemodel-5.0.1.gem (100%)
Successfully installed activemodel-5.0.1
Fetching: arel-7.1.4.gem (100%)
Successfully installed arel-7.1.4
Fetching: activerecord-5.0.1.gem (100%)
Successfully installed activerecord-5.0.1
Fetching: globalid-0.3.7.gem (100%)
Successfully installed globalid-0.3.7
Fetching: activejob-5.0.1.gem (100%)
Successfully installed activejob-5.0.1
Fetching: mime-types-data-3.2016.0521.gem (100%)
Successfully installed mime-types-data-3.2016.0521
Fetching: mime-types-3.1.gem (100%)
Successfully installed mime-types-3.1
Fetching: mail-2.6.4.gem (100%)
Successfully installed mail-2.6.4
Fetching: actionmailer-5.0.1.gem (100%)
Successfully installed actionmailer-5.0.1
Fetching: nio4r-1.2.1.gem (100%)
```

```

Building native extensions. This could take a while...
Successfully installed nio4r-1.2.1
Fetching: websocket-extensions-0.1.2.gem (100%)
Successfully installed websocket-extensions-0.1.2
Fetching: websocket-driver-0.6.5.gem (100%)
Building native extensions. This could take a while...
Successfully installed websocket-driver-0.6.5
Fetching: actioncable-5.0.1.gem (100%)
Successfully installed actioncable-5.0.1
Fetching: thor-0.19.4.gem (100%)
Successfully installed thor-0.19.4
Fetching: method_source-0.8.2.gem (100%)
Successfully installed method_source-0.8.2
Fetching: railties-5.0.1.gem (100%)
Successfully installed railties-5.0.1
Fetching: bundler-1.14.5.gem (100%)
Successfully installed bundler-1.14.5
Fetching: sprockets-3.7.1.gem (100%)
Successfully installed sprockets-3.7.1
Fetching: sprockets-rails-3.2.0.gem (100%)
Successfully installed sprockets-rails-3.2.0
Fetching: rails-5.0.1.gem (100%)
Successfully installed rails-5.0.1
28 gems installed

```

A Rails rendszerünk konfigurációja a következő lesz a félév során. Kétféle webservert használunk, a fejlesztéshez a Puma-t, 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 webserverral való integrációt lehetővé tevő Passenger plugin.

```

root@debian:~# gem install passenger
Fetching: passenger-5.1.2.gem (100%)
Building native extensions. This could take a while...
Successfully installed passenger-5.1.2
Parsing documentation for passenger-5.1.2
Installing ri documentation for passenger-5.1.2
Done installing documentation for passenger after 198 seconds
1 gem installed

```

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
Suggested packages:
  sqlite3-doc
The following NEW packages will be installed:
  libsqlite3-dev

```

```

0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 538 kB of archives.
After this operation, 1,542 kB of additional disk space will be used.
Get:1 http://ftp.hu.debian.org/debian/ stable/main libsqlite3-dev amd64
    3.8.7.1-1+deb8u2 [538 kB]
Fetched 538 kB in 0s (1,527 kB/s)
Selecting previously unselected package libsqlite3-dev:amd64.
(Reading database ... 163466 files and directories currently installed.)
Preparing to unpack .../libsqlite3-dev_3.8.7.1-1+deb8u2_amd64.deb ...
Unpacking libsqlite3-dev:amd64 (3.8.7.1-1+deb8u2) ...
Setting up libsqlite3-dev:amd64 (3.8.7.1-1+deb8u2) ...

```

```

root@debian:~# apt-get install libmysqlclient-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  libmysqlclient-dev
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 948 kB of archives.
After this operation, 5,656 kB of additional disk space will be used.
Get:1 http://ftp.hu.debian.org/debian/ stable/main libmysqlclient-dev amd64
    5.5.53-0+deb8u1 [948 kB]
Fetched 948 kB in 0s (5,011 kB/s)
Selecting previously unselected package libmysqlclient-dev.
(Reading database ... 163477 files and directories currently installed.)
Preparing to unpack .../libmysqlclient-dev_5.5.53-0+deb8u1_amd64.deb ...
Unpacking libmysqlclient-dev (5.5.53-0+deb8u1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up libmysqlclient-dev (5.5.53-0+deb8u1) ...

```

Ezután telepíthetjük magukat az adatbáziskezelő-adaptereket, a `sqlite3`-t és a `mysql2`-t. A PostgreSQL adapterét `pg`-nek hívják, a gyakorlaton azt nem fogjuk használni.

```

root@debian:~# gem install sqlite3
Fetching: sqlite3-1.3.13.gem (100%)
Building native extensions. This could take a while...
Successfully installed sqlite3-1.3.13
Parsing documentation for sqlite3-1.3.13
Installing ri documentation for sqlite3-1.3.13
Done installing documentation for sqlite3 after 1 seconds
1 gem installed

```

```

root@debian:~# gem install mysql2
Fetching: mysql2-0.4.5.gem (100%)
Building native extensions. This could take a while...
Successfully installed mysql2-0.4.5
Parsing documentation for mysql2-0.4.5
Installing ri documentation for mysql2-0.4.5
Done installing documentation for mysql2 after 1 seconds
1 gem installed

```

Az alap Rails rendszerünk összeállt, menetközben szükségünk lesz további gemek telepítésére. Ilyen a `therubyracer`, ami egy szerver oldali JavaScript interpreter, vagy a `turbolinks` API nélkül a Rails már nem hajlandó elindulni, noha az opcionális.

```

root@debian:~# gem install execjs therubyracer
Fetching: execjs-2.7.0.gem (100%)
Successfully installed execjs-2.7.0
Parsing documentation for execjs-2.7.0
Installing ri documentation for execjs-2.7.0
Done installing documentation for execjs after 0 seconds
Fetching: ref-2.0.0.gem (100%)
Successfully installed ref-2.0.0
Fetching: libv8-3.16.14.17-x86_64-linux.gem (100%)
Successfully installed libv8-3.16.14.17-x86_64-linux
Fetching: therubyracer-0.12.3.gem (100%)
Building native extensions. This could take a while...
Successfully installed therubyracer-0.12.3
Parsing documentation for ref-2.0.0
Installing ri documentation for ref-2.0.0
Parsing documentation for libv8-3.16.14.17-x86_64-linux
Installing ri documentation for libv8-3.16.14.17-x86_64-linux
Parsing documentation for therubyracer-0.12.3
Installing ri documentation for therubyracer-0.12.3
Done installing documentation for ref, libv8, therubyracer after 3 seconds
4 gems installed

```

A telepítés után a következő Ruby függvénykönyvtáraink érhetőek el:

```

root@debian:~# gem list

*** LOCAL GEMS ***

actioncable (5.0.1)
actionmailer (5.0.1)
actionpack (5.0.1)
actionview (5.0.1)
activejob (5.0.1)
activemodel (5.0.1)
activerecord (5.0.1)
activesupport (5.0.1)
arel (7.1.4)
bigdecimal (1.2.8)
builder (3.2.3)
bundler (1.14.5)
concurrent-ruby (1.0.5)
did_you_mean (1.0.0)
erubis (2.7.0)
execjs (2.7.0)
globalid (0.3.7)
i18n (0.8.1)
io-console (0.4.5)
json (1.8.3)
libv8 (3.16.14.17 x86_64-linux)
loofah (2.0.3)
mail (2.6.4)
method_source (0.8.2)
mime-types (3.1)
mime-types-data (3.2016.0521)
mini_portile2 (2.1.0)
minitest (5.9.0)
mysql2 (0.4.5)
net-telnet (0.1.1)
nio4r (1.2.1)
nokogiri (1.7.0.1)
passenger (5.1.2)

```



```

power_assert (0.2.7)
psych (2.1.0)
rack (2.0.1)
rack-test (0.6.3)
rails (5.0.1)
rails-dom-testing (2.0.2)
rails-html-sanitizer (1.0.3)
railties (5.0.1)
rake (10.5.0)
rdoc (4.2.1)
ref (2.0.0)
sprockets (3.7.1)
sprockets-rails (3.2.0)
sqlite3 (1.3.13)
test-unit (3.1.7)
therubyracer (0.12.3)
thor (0.19.4)
thread_safe (0.3.6)
tzinfo (1.2.2)
websocket-driver (0.6.5)
websocket-extensions (0.1.2)

```

Az `action` és az `active` kezdetű függvénykönyvtárak adják a Rails keretrendszer magját.

## 2. Rails alkalmazás létrehozása

A Rails használatához a `rails` szkriptet használjuk immáron nem rendszer-gazdaként, hanem egyszerű felhasználóként. A parancs Debian/Ubuntu Linux és 2.3-as `ruby` esetén a `/usr/local/bin/` könyvtárba került.

A `-d` kapcsoló különös jelentőséggel bír számunkra, ezzel adhatjuk meg a használni kívánt adatbáziskezelő típusát. A gyakorlatok keretében ezek közül a `sqlite`-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.

```

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
  -m, [--template=TEMPLATE]        # Path to some
                                   application template (can be a filesystem path or URL)
  -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

```

```

    [--skip-gemfile], [--no-skip-gemfile] # Default: jquery
    Gemfile # Don't create a
-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
-M, [--skip-action-mailer], [--no-skip-action-mailer] # Skip Action
    Mailer files
-O, [--skip-active-record], [--no-skip-active-record] # Skip Active
    Record files
-P, [--skip-puma], [--no-skip-puma] # Skip Puma related
    files
-C, [--skip-action-cable], [--no-skip-action-cable] # Skip Action Cable
    files
-S, [--skip-sprockets], [--no-skip-sprockets] # Skip Sprockets
    files
    [--skip-spring], [--no-skip-spring] # Don't install
    Spring application preloader
    [--skip-listen], [--no-skip-listen] # Don't generate
    configuration that depends on the listen gem
-J, [--skip-javascript], [--no-skip-javascript] # Skip JavaScript
    files
    [--skip-turbolinks], [--no-skip-turbolinks] # Skip turbolinks
    gem
-T, [--skip-test], [--no-skip-test] # Skip test 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
    [--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
    [--api], [--no-api] # Preconfigure
    smaller stack for API only apps

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.

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 parancsot tetszőleges felhasználóként 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.

```
kovacs@debian:~$ rails new gyakorlat -B
create
create  README.md
create  Rakefile
create  config.ru
create  .gitignore
create  Gemfile
create  app
create  app/assets/config/manifest.js
create  app/assets/javascripts/application.js
create  app/assets/javascripts/cable.js
create  app/assets/stylesheets/application.css
create  app/channels/application_cable/channel.rb
create  app/channels/application_cable/connection.rb
create  app/controllers/application_controller.rb
create  app/helpers/application_helper.rb
create  app/jobs/application_job.rb
create  app/mailers/application_mailer.rb
create  app/models/application_record.rb
create  app/views/layouts/application.html.erb
create  app/views/layouts/mailer.html.erb
create  app/views/layouts/mailer.text.erb
create  app/assets/images/.keep
create  app/assets/javascripts/channels
create  app/assets/javascripts/channels/.keep
create  app/controllers/concerns/.keep
create  app/models/concerns/.keep
create  bin
create  bin/bundle
create  bin/rails
create  bin/rake
create  bin/setup
create  bin/update
create  config
create  config/routes.rb
create  config/application.rb
create  config/environment.rb
create  config/secrets.yml
create  config/cable.yml
create  config/puma.rb
create  config/spring.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/application_controller_renderer.rb
create  config/initializers/assets.rb
create  config/initializers/backtrace_silencers.rb
create  config/initializers/cookies_serializer.rb
create  config/initializers/cors.rb
create  config/initializers/filter_parameter_logging.rb
create  config/initializers/inflections.rb
create  config/initializers/mime_types.rb
create  config/initializers/new_framework_defaults.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/apple-touch-icon-precomposed.png
create  public/apple-touch-icon.png
create  public/favicon.ico
create  public/robots.txt
create  test/fixtures
create  test/fixtures/.keep
create  test/fixtures/files
create  test/fixtures/files/.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
create  tmp/.keep
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
remove config/initializers/cors.rb

```

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 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 `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 a többi, a fájlban szereplő függőségre nincs szükségünk. A webservert futtatásához szükségünk lesz szerver oldali JavaScript értelmezőre, ezért a fájlba felvesszük az alábbi gem függőséget, amit az elé írt kommentjellel eltávolításával tehetünk meg. A `Gemfile` minden további módosítása után futtatnunk kell majd a `bundle install` vagy `bundle update` parancsot.

```
gem 'therubyracer', platforms: :ruby
```

A konzolon kiadott `bundle install` paranccsal 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 áldozatául. Ha a Ruby csomagokat minden felhasználó számára elérhetővé akarjuk tenni, meg kell adnunk az adminisztrátor jelszavát.

```

kovacsg@debian:~/gyakorlat$ bin/bundle
The dependency tzinfo-data (>= 0) will be unused by any of the platforms
Bundler is installing for. Bundler is installing for ruby but the
dependency is only for x86-mingw32, x86-mswin32, x64-mingw32, java. To
add those platforms to the bundle, run 'bundle lock --add-platform x86-
mingw32 x86-mswin32 x64-mingw32 java'.
Resolving dependencies...
Using rake 12.0.0
Using concurrent-ruby 1.0.5
Using i18n 0.8.1
Using minitest 5.10.1
Using thread_safe 0.3.6
Using builder 3.2.3
Using erubis 2.7.0
Using mini_portile2 2.1.0
Using rack 2.0.1
Using nio4r 1.2.1
Using websocket-extensions 0.1.2
Using mime-types-data 3.2016.0521
Using arel 7.1.4
Using bundler 1.14.5
Using byebug 9.0.6
Using coffee-script-source 1.12.2
Using execjs 2.7.0
Using method_source 0.8.2
Using thor 0.19.4
Using debug_inspector 0.0.2
Using ffi 1.9.17
Using multi_json 1.12.1
Using rb-fsevent 0.9.8
Using puma 3.7.1
Using sass 3.4.23
Using tilt 2.0.6
Using sqlite3 1.3.13
Using turbolinks-source 5.0.0
Using tzinfo 1.2.2
Using nokogiri 1.7.0.1
Using rack-test 0.6.3
Using sprockets 3.7.1
Using websocket-driver 0.6.5
Using mime-types 3.1
Using coffee-script 2.4.1
Using uglifier 3.1.1
Using rb-inotify 0.9.8
Using turbolinks 5.0.1
Using activesupport 5.0.1
Using loofah 2.0.3
Using mail 2.6.4
Using listen 3.0.8
Using rails-dom-testing 2.0.2
Using globalid 0.3.7
Using activemodel 5.0.1
Using jbuilder 2.6.3
Using spring 2.0.1
Using rails-html-sanitizer 1.0.3
Using activejob 5.0.1
Using activerecord 5.0.1
Using spring-watcher-listen 2.0.1
Using actionview 5.0.1
Using actionpack 5.0.1
Using actioncable 5.0.1

```

```
Using actionmailer 5.0.1
Using railties 5.0.1
Using sprockets-rails 3.2.0
Using coffee-rails 4.2.1
Using jquery-rails 4.2.2
Using web-console 3.4.0
Using rails 5.0.1
Using sass-rails 5.0.6
Bundle complete! 15 Gemfile dependencies, 62 gems now installed.
Use 'bundle show [gemname]' to see where a bundled gem is installed.
```

A Rails rendszerünk használatához minden függőséget feloldottunk, ezután az alábbi Ruby függvénykönyvtárak kerültek telepítésre:

```
root@debian:/etc# gem list

*** LOCAL GEMS ***

actioncable (5.0.1)
actionmailer (5.0.1)
actionpack (5.0.1)
actionview (5.0.1)
activejob (5.0.1)
activemodel (5.0.1)
activerecord (5.0.1)
activesupport (5.0.1)
arel (7.1.4)
bigdecimal (1.2.8)
builder (3.2.3)
bundler (1.14.5)
byebug (9.0.6)
coffee-rails (4.2.1)
coffee-script (2.4.1)
coffee-script-source (1.12.2)
concurrent-ruby (1.0.5)
debug_inspector (0.0.2)
did_you_mean (1.0.0)
erubis (2.7.0)
execjs (2.7.0)
ffi (1.9.17)
globalid (0.3.7)
i18n (0.8.1)
io-console (0.4.5)
jbuilder (2.6.3)
jquery-rails (4.2.2)
json (1.8.3)
libv8 (3.16.14.17 x86_64-linux)
listen (3.0.8)
loofah (2.0.3)
mail (2.6.4)
method_source (0.8.2)
mime-types (3.1)
mime-types-data (3.2016.0521)
mini_portile2 (2.1.0)
minitest (5.10.1, 5.9.0)
multi_json (1.12.1)
mysql2 (0.4.5)
net-telnet (0.1.1)
nio4r (1.2.1)
nokogiri (1.7.0.1)
passenger (5.1.2)
```

```

power_assert (0.2.7)
psych (2.1.0)
puma (3.7.1)
rack (2.0.1)
rack-test (0.6.3)
rails (5.0.1)
rails-dom-testing (2.0.2)
rails-html-sanitizer (1.0.3)
railties (5.0.1)
rake (12.0.0, 10.5.0)
rb-fsevent (0.9.8)
rb-inotify (0.9.8)
rdoc (4.2.1)
ref (2.0.0)
sass (3.4.23)
sass-rails (5.0.6)
spring (2.0.1)
spring-watcher-listen (2.0.1)
sprockets (3.7.1)
sprockets-rails (3.2.0)
sqlite3 (1.3.13)
test-unit (3.1.7)
therubyracer (0.12.3)
thor (0.19.4)
thread_safe (0.3.6)
tilt (2.0.6)
turbolinks (5.0.1)
turbolinks-source (5.0.0)
tzinfo (1.2.2)
uglifyer (3.1.1)
web-console (3.4.0)
websocket-driver (0.6.5)
websocket-extensions (0.1.2)

```

A `rails` parancsot az újonnan létrehozott Rails alkalmazásunk bármely alkönyvtárából kiadva más üzenetet látunk. Az új alkalmazást létrehozó `new` mellett jelen van a kódgenerálásra alkalmas `generate` opció, a beágyazott webszervert indító `server` opció, a Ruby, illetve az adatbáziskonzolt indító `console`, illetve `dbconsole` opciók.

```

kovacs@debian:~/gyakorlat$ rails
Usage: rails COMMAND [ARGS]

The most common rails commands are:
generate      Generate new code (short-cut alias: "g")
console       Start the Rails console (short-cut alias: "c")
server        Start the Rails server (short-cut alias: "s")
test          Run tests (short-cut alias: "t")
dbconsole     Start a console for the database specified in config/database.
              yml
              (short-cut alias: "db")
new           Create a new Rails application. "rails_new_my_app" creates a
              new application called MyApp in "./my_app"

All commands can be run with -h (or --help) for more information.

In addition to those commands, there are:
destroy       Undo code generated with "generate" (
              short-cut alias: "d")

```



plugin new	Generates skeleton for developing a
Rails plugin	
runner	Run a piece of code in the application
environment (short-cut alias: "r")	
about	List versions of all Rails frameworks
and the environment	
app:template	Applies the template supplied by
LOCATION=(/path/to/template) or URL	
app:update	Update configs and some other initially
generated files (or use just update:configs or update:bin)	
assets:clean[keep]	Remove old compiled assets
assets:clobber	Remove compiled assets
assets:environment	Load asset compile environment
assets:precompile	Compile all the assets named in config.
assets.precompile	
cache_digests:dependencies	Lookup first-level dependencies for
TEMPLATE (like messages/show or comments/_comment.html)	
cache_digests:nested_dependencies	Lookup nested dependencies for TEMPLATE
(like messages/show or comments/_comment.html)	
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). Without RAILS_ENV or when	
RAILS_ENV is development, it defaults to creating the development and	
test databases	
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). Without RAILS_ENV or when RAILS_ENV is	
development, it defaults to dropping the development and test databases	
db:environment:set	Set the environment value for the
database	
db:fixtures:load	Loads fixtures into the current
environment's database	
db:migrate	Migrate the database (options: VERSION=
x, VERBOSE=false, SCOPE=blog)	
db:migrate:status	Display status of migrations
db:rollback	Rolls the schema back to the previous
version (specify steps w/ STEP=n)	
db:schema:cache:clear	Clears a db/schema_cache.dump file
db:schema:cache:dump	Creates a db/schema_cache.dump file
db:schema:dump	Creates a db/schema.rb file that is
portable against any DB supported by Active Record	
db:schema:load	Loads a schema.rb file into the
database	
db:seed	Loads the seed data from db/seeds.rb
db:setup	Creates the database, loads the schema,
and initializes with the seed data (use db:reset to also drop the	
database first)	
db:structure:dump	Dumps the database structure to db/
structure.sql	
db:structure:load	Recreates the databases from the
structure.sql file	
db:version	Retrieves the current schema version
number	
dev:cache	Toggle development mode caching on/off
initializers	Print out all defined initializers in
the order they are invoked by Rails	
log:clear	Truncates all/specified *.log files in
log/ to zero bytes (specify which logs with LOGS=test,development)	
middleware	Prints out your Rack middleware stack
notes	Enumerate all annotations (use notes:
optimize, :fixme, :todo for focus)	
notes:custom	Enumerate a custom annotation, specify

```

with ANNOTATION=CUSTOM
restart                Restart app by touching tmp/restart.txt
routes                Print out all defined routes in match
  order, with names
secret                Generate a cryptographically secure
  secret key (this is typically used to generate a secret for cookie
  sessions)
stats                Report code statistics (KLOCs, etc)
  from the application or engine
test                Runs all tests in test folder
test:db              Run tests quickly, but also reset db
time:zones[country_or_offset] List all time zones, list by two-letter
  country code ('rails time:zones[US]'), or list by UTC offset ('rails
  time:zones[-8]')
tmp:clear            Clear cache and socket files from tmp/
  (narrow w/ tmp:cache:clear, tmp:sockets:clear)
tmp:create           Creates tmp directories for cache,
  sockets, and pids

```

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

```

kovacs@debian:~/gyakorlat$ rails c
Loading development environment (Rails 5.0.0.1)
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.

Az alapértelmezett webszervert, a Puma-t a rails server vagy röviden s 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. Ha megnyitjuk az oldalt egy böngészőben, akkor annak eseményei megjelennek a konzolon, és bekerülnek a naplófájlba is. Itt azt látjuk, hogy HTTP GET művelet fért hozzá a "/" erőforráshoz, és, hogy a Rails ennek hatására milyen műveleteket végzett el. A kimeneten láthatjuk, hogy a webszerver a fejlesztői környezet konfigurációját használja.

```

kovacs@debian:~/gyakorlat$ rails s
=> Booting Puma
=> Rails 5.0.1 application starting in development on http://localhost:3000
=> Run 'rails server -h' for more startup options
Puma starting in single mode...
* Version 3.7.1 (ruby 2.3.3-p222), codename: Snowy Sagebrush
* Min threads: 5, max threads: 5
* Environment: development

```

```
* Listening on tcp://localhost:3000
Use Ctrl-C to stop
Started GET "/" for 127.0.0.1 at 2017-02-28 12:52:48 +0100
Processing by Rails::WelcomeController#index as HTML
  Parameters: {"internal"=>true}
  Rendering /var/lib/gems/2.3.0/gems/railties-5.0.1/lib/rails/templates/
rails/welcome/index.html.erb
  Rendered /var/lib/gems/2.3.0/gems/railties-5.0.1/lib/rails/templates/rails
/welcome/index.html.erb (3.9ms)
Completed 200 OK in 32ms (Views: 8.7ms | ActiveRecord: 0.0ms)
```

A Rails a már említett Passenger plugin segítségével illeszhető Apache2 webserververhez. Ehhez az alábbi kódrészletben mutatott parancsot kell kiadnunk. A parancs lefordítja, majd telepíti az Apache webserverver Rails (és egyéb további) 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. Az alábbi naplórészlet a Passenger plugin telepítését mutatja be, a C források fordítása nélkül.

```
root@debian:~# passenger-install-apache2-module
Welcome to the Phusion Passenger Apache 2 module installer , v5.1.2.

This installer will guide you through the entire installation process. It
shouldn't take more than 3 minutes in total.

Here's what you can expect from the installation process:

  1. The Apache 2 module will be installed for you.
  2. You'll learn how to configure Apache.
  3. You'll learn how to deploy a Ruby on Rails application.

Don't worry if anything goes wrong. This installer will advise you on how to
solve any problems.

Press Enter to continue, or Ctrl-C to abort.

-----

Which languages are you interested in?

Use <space> to select.
If the menu doesn't display correctly, press '!'.

  x Ruby
  > Python
  - Node.js
  - Meteor

-----

Which languages are you interested in?
```

```

Use <space> to select.
If the menu doesn't display correctly, press '!'

x Ruby
  Python
  Node.js
  > Meteor

```

---

```

Checking for required software ...

* Checking for C compiler ...
  Found: yes
  Location: /usr/bin/cc
* Checking for C++ compiler ...
  Found: yes
  Location: /usr/bin/c++
* Checking for Curl development headers with SSL support ...
  Found: yes
  curl-config location: /usr/bin/curl-config
  Header location: /usr/include/curl/curl.h
  Version: libcurl 7.38.0
  Usable: yes
  Supports SSL: yes
* Checking for Zlib development headers ...
  Found: yes
  Location: /usr/include/zlib.h
* Checking for Apache 2 ...
  Found: yes
  Location of httpd: /usr/sbin/apache2
  Apache version: 2.4.10
* Checking for Apache 2 development headers ...
  Found: yes
  Location of apxs2: /usr/bin/apxs2
* Checking for Rake (associated with /usr/bin/ruby2.3) ...
  Found: yes
  Location: /usr/bin/ruby2.3/usr/bin/rake
* Checking for OpenSSL support for Ruby ...
  Found: yes
* Checking for RubyGems ...
  Found: yes
* Checking for Ruby development headers ...
  Found: yes
  Location: /usr/include/ruby-2.3.0/ruby.h
* Checking for rack ...
  Found: yes
* Checking for Apache Portable Runtime (APR) development headers ...
  Found: yes
  Location: /usr/bin/apr-1-config
  Version: 1.5.1
* Checking for Apache Portable Runtime Utility (APU) development headers ...
  Found: yes
  Location: /usr/bin/apu-1-config
  Version: 1.5.4

```

---

```

Checking whether there are multiple Apache installations ...
Only a single installation detected. This is good.

```

```

Compiling_and_installing_Apache_2_module...

Almost_there!

Please_edit_your_Apache_configuration_file,_and_add_these_lines:

LoadModule_passenger_module_/var/lib/gems/2.3.0/gems/passenger-5.1.2/
  buildout/apache2/mod_passenger.so
<IfModule_mod_passenger.c>
  PassengerRoot_/var/lib/gems/2.3.0/gems/passenger-5.1.2
  PassengerDefaultRuby_/usr/bin/ruby2.3
</IfModule>

After_you_restart_Apache,_you_are_ready_to_deploy_any_number_of_web
applications_on_Apache,_with_a_minimum_amount_of_configuration!

Press_ENTER_when_you_are_done_editing.

Validating_installation...

[*]_Checking_whether_this_Passenger_install_is_in_PATH...x
[*]_Checking_whether_there_are_no_other_Passenger_installations...x
[*]_Checking_whether_Apache_is_installed...x
[*]_Checking_whether_the_Passenger_module_is_correctly_configured_in_Apache
...x

Everything_looks_good.:-)

```

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, a tartalmaz az alábbi konfigurációrészlethez hasonló.

```

LoadModule passenger_module /var/lib/gems/2.3.0/gems/passenger-5.1.2/
  buildout/apache2/mod_passenger.so
<IfModule mod_passenger.c>
  PassengerRoot /var/lib/gems/2.3.0/gems/passenger-5.1.2
  PassengerDefaultRuby /usr/bin/ruby2.3
</IfModule>

```

Az Apache2 virtuális hoszt beállításait szintén konfigurációs állományok között kell elhelyeznünk például /etc/apache2/sites-available/

gyakorlat.conf néven, majd az `a2ensite` paranccsal konzolon engedélyez-  
zük a gyakorlat oldalt<sup>5</sup>. Ennek tartalma a telepítő parancs alapján a követ-  
kező lehet:

```
RackEnv development

#<VirtualHost *:80>
<VirtualHost www.gyakorlat.com:80>

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

    CustomLog /var/log/apache2/gyakorlat_access.log combined
    ErrorLog /var/log/apache2/gyakorlat_error.log
    LogLevel info

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

</VirtualHost>
```

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 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.

<sup>5</sup>Alternatív megoldás: `ln -s /etc/apache2/sites-available/gyakorlat.conf /etc/apache2/sites-enabled`

```
# 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.

Konzolos adatbáziskapcsolatot a `rails db` paranccsal tudunk előhozni.

```
kovacsg@debian:~/gyakorlat$ rails db
SQLite version 3.8.7.1 2014-10-29 13:59:56
Enter ".help" for usage hints.
sqlite> .schema
sqlite>
```

Az adatbáziskapcsolatnak a szervert 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 Rails korábbi verziói esetén a `rake` parancs volt segítségünkre, amelyet a Rails keretrendszerhez kapcsolódó olyan műveletekre használtunk, mint az adatbázis menedzsmentje vagy a tesztvégrehajtás vezérlése. A Rails 5-

től ezek a műveletek a rails paranccsal is elvégezhetők, a rake műveletek mindazonáltal továbbra is használhatók. A műveletek listája alább látható.

```
kovacs@debian:~/gyakorlat$ bin/rake -T
rake about # List versions of all Rails
            frameworks and the environment
rake app:template # Applies the template supplied by
                  LOCATION=(/path/to/template) or URL
rake app:update # Update configs and some other
                initially generated files (or use just update:configs or update:bin)
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). Without RAILS_ENV or
               when RAILS_...
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). Without RAILS_ENV or when
             RAILS_ENV is...
rake db:environment:set # Set the environment value for the
                        database
rake db:fixtures:load # Loads 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 # Clears a db/schema_cache.dump file
rake db:schema:cache:dump # Creates a db/schema_cache.dump
                           file
rake db:schema:dump # Creates a db/schema.rb file that
                    is portable against any DB supported by Active Record
rake db:schema:load # Loads a schema.rb file into the
                   database
rake db:seed # Loads the seed data from db/seeds.
             rb
rake db:setup # Creates the database, loads the
              schema, and initializes with the seed data (use db:reset to also drop
              the database first)
rake db:structure:dump # Dumps the database structure to db
                       /structure.sql
rake db:structure:load # Recreates the databases from the
                       structure.sql file
rake db:version # Retrieves the current schema
                version number
rake dev:cache # Toggle development mode caching on
               /off
rake initializers # Print out all defined initializers
                  in the order they are invoked by Rails
rake log:clear # Truncates all/specified *.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 restart # Restart app by touching tmp/
  restart.txt
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 or engine
rake test # Runs all tests in test folder
rake test:db # Run tests quickly, but also reset
  db
rake time:zones[country_or_offset] # List all time zones, list by two-
  letter country code ('rails time:zones[US]'), or list by UTC offset ('
  rails time:zones[-8]')
rake tmp:clear # Clear cache and socket files from
  tmp/ (narrow w/ tmp:cache:clear, tmp:sockets:clear)
rake tmp:create # Creates tmp directories for cache,
  sockets, and pids

```

Töröljük a korábbi Rails alkalmazásunkat, amely az SQLite3-at használta, és hozzuk létre ugyanazt MySQL adatbáziskezelővel a `-d mysql` opció felhasználásával.

A MySQL adatbázis nem jön automatikusan létre az SQLite3-mal elmentésben. A létrehozása következő parancsokkal történhet meg. Először eltávolítjuk az összes azonos nevű adatbázist a `db:drop` opcióval, majd a `db:create` opciót használjuk, amely a development és a test környezetekhez hozza létre egy üres sémát. Az eredmény a Rails adatbázis konzolán ellenőrizzük.

```

kovacs@debian:~/gyakorlat/app/views/say$ rails db:drop
Dropped database 'gyakorlat_development'
Dropped database 'gyakorlat_test'
kovacs@debian:~/gyakorlat/app/views/say$ rails db
ERROR 1049 (42000): Unknown database 'gyakorlat_development'
kovacs@debian:~/gyakorlat/app/views/say$ rails db:create
Created database 'gyakorlat_development'
Created database 'gyakorlat_test'
kovacs@debian:~/gyakorlat/app/views/say$ rails db
mysql> show tables;
Empty set (0.00 sec)

mysql> Bye

```

## 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 webserverver 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 könyvvarat 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.

```
kovacsg@debian:~/gyakorlat$ rails generate
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.

Expected string default value for '--helper'; got true (boolean)
Expected string default value for '--assets'; got true (boolean)
Rails:
  assets
  channel
  controller
  generator
  helper
  integration_test
  jbuilder
  job
  mailer
  migration
```

```
model
resource
scaffold
scaffold_controller
task

Coffee:
  coffee: assets

Js:
  js: assets

TestUnit:
  test_unit: generator
  test_unit: plugin
```

```
kovacs@debian:~/gyakorlat$ rails generate controller say hello
Expected string default value for '--helper'; got true (boolean)
Expected string default value for '--assets'; got true (boolean)
   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
   invoke  assets
   invoke  coffee
   create  app/assets/javascripts/say.coffee
   invoke  scss
   create  app/assets/stylesheets/say.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 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.

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

Mivel a nézetbe nem illik logikát rakni, csak a megjelenítendő értéket,

ezért áttesszük az idő lekérdezését a kontrollerbe, 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 nézet többnyelvűsítését a `/config/locales/` könyvtárban lévő YAML fájlokkal érhetjük el. Definiáljuk a `hello` fordítását és egy időformátumot.

```
en:
  hello: "Hello, _world!"

  time:
    formats:
      default: "%Y.%m.%d_%H
      datetime: "%Y.%m.%d. %H:%M:%S"
```

A nézetünkben (`app/views/say/hello.html.erb`) pedig használjuk a fordítást végző `t` (vagy hosszabban `translate`) azonosítójú, és a lokalizációt végző `l` (vagy hosszabban `localize`) azonosítójú függvényt.

```
<h1><%= t :hello %></h1>
<p><%= l @time, format: :datetime %></p>
```

A Rails MVC filozófiájának harmadik eleme a modell, amelyet szintén a `rails` 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/app$ rails g model user name:string password:
string email:string
  invoke  active_record
  create  db/migrate/20170228122321_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.

```

class CreateUsers < ActiveRecord::Migration[5.0]
  def change
    create_table :users do |t|
      t.string :name
      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/db/migrate$ rake db:migrate
(in /home/kovacs/gyakorlat)
== 20170228122321 CreateUsers: migrating
-----
-- create_table(:users)
--> 0.0142s
== 20170228122321 CreateUsers: migrated (0.0144s)
-----

```

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

```

kovacs@debian:~/gyakorlat/db/migrate$ rails db
mysql> show tables;
+-----+
| Tables_in_gyakorlat_development |
+-----+
| ar_internal_metadata |
| schema_migrations |
| users |
+-----+
3 rows in set (0.00 sec)

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

mysql> Bye

```

Nézzük meg, miként tudunk az adatbázishoz Rubyból hozzáférni. Nyissunk egy konzolt, és hozzunk létre egy felhasználót (2. sor), állítsuk be a

tulajdonságait (3-5. sorok), mentjük el az adatbázisba (7. sor). Nézzük meg ezután, hogy létrejött-e a rekord az adatbázisban, vagyis a példány id attribútuma még mindig `nil`-e? Létrejött a rekord, az azonosítója 1, amit a modell osztály `find` osztálymetódusának átadva az adatbázisból előkereshetjük az objektumot.

```
kovacs@debian:~/gyakorlat/db/migrate$ rails c
Loading development environment (Rails 5.0.1)
irb(main):001:0> User.new
=> #<User id: nil, name: nil, password: nil, email: nil, created_at: nil,
updated_at: nil>
irb(main):002:0> u = User.new
=> #<User id: nil, name: nil, password: nil, email: nil, created_at: nil,
updated_at: nil>
irb(main):003:0> u.name = 'Kovacs_Gabor'
=> "Kovacs_Gabor"
irb(main):004:0> u.password = 'titok'
=> "titok"
irb(main):005:0> u.email = 'kovacs@tmit.bme.hu'
=> "kovacs@tmit.bme.hu"
irb(main):006:0> u.save
(0.3ms) BEGIN
SQL (0.4ms) INSERT INTO 'users' ('name', 'password', 'email', 'created_at',
'updated_at') VALUES ('Kovacs_Gabor', 'titok', 'kovacs@tmit.bme.hu',
'2017-02-28_12:28:16', '2017-02-28_12:28:16')
(4.2ms) COMMIT
=> true
irb(main):008:0> User.first
User Load (2.4ms) SELECT 'users'.* FROM 'users' ORDER BY 'users'.'id'
ASC LIMIT 1
=> #<User id: 1, name: "Kovacs Gabor", password: "titok", email: "
kovacs@tmit.bme.hu", created_at: "2017-02-28 12:28:16", updated_at:
"2017-02-28 12:28:16">
```

Ellenőrizzük az adatbáziskonzolon, hogy valóban megjelent-e ez adat az adatbázisban.

```
kovacs@debian:~/gyakorlat/db/migrate$ rails db
mysql> select * from users;
+----+-----+-----+-----+-----+
| id | name          | password | email          | created_at |
|   | updated_at    |         |                |            |
+----+-----+-----+-----+-----+
|  1 | Kovacs Gabor | titok    | kovacs@tmit.bme.hu | 2017-02-28 12:28:16 |
|   | 2017-02-28 12:28:16 |         |                |            |
+----+-----+-----+-----+-----+

1 row in set (0.00 sec)

mysql> Bye
```

Hozzuk létre a receptmegosztó portálunk receptjeinek összetevőire vonatkozó modellünket és a hozzá tartozó kontrollert egy paranccsal. Az `Ingredient` modellünkben legyenek string típusú `element`, egész típusú `amount`, és string típusú `unit` attribútumok.

```

kovacsg@debian:~/gyakorlat/db$ rails g scaffold ingredient element:string
amount:integer unit:string
  invoke  active_record
  create  db/migrate/20170228123223_create_ingredients.rb
  create  app/models/ingredient.rb
  invoke  test_unit
  create  test/models/ingredient_test.rb
  create  test/fixtures/ingredients.yml
  invoke  resource_route
  route   resources :ingredients
  invoke  scaffold_controller
  create  app/controllers/ingredients_controller.rb
  invoke  erb
  create  app/views/ingredients
  create  app/views/ingredients/index.html.erb
  create  app/views/ingredients/edit.html.erb
  create  app/views/ingredients/show.html.erb
  create  app/views/ingredients/new.html.erb
  create  app/views/ingredients/_form.html.erb
  invoke  test_unit
  create  test/controllers/ingredients_controller_test.rb
  invoke  helper
  create  app/helpers/ingredients_helper.rb
  invoke  test_unit
  invoke  jbuilder
  create  app/views/ingredients/index.json.jbuilder
  create  app/views/ingredients/show.json.jbuilder
  create  app/views/ingredients/_ingredient.json.jbuilder
  invoke  assets
  invoke  coffee
  create  app/assets/javascripts/ingredients.coffee
  invoke  scss
  create  app/assets/stylesheets/ingredients.scss
  invoke  scss
  create  app/assets/stylesheets/scaffolds.scss

```

Hajtsuk végre a scaffold létrehozása során keletkezett migrációt.

```

kovacsg@debian:~/gyakorlat/db$ rails db:migrate
== 20170228123223 CreateIngredients: migrating
-----
-- create_table(:ingredients)
--> 0.1732s
== 20170228123223 CreateIngredients: migrated (0.1734s)
-----

```

Majd a böngészőben nyissuk meg az összetevők nézetet (<http://localhost:3000/ingredients>), próbáljuk ki az összetevők létrehozását, törlését listázását. Nézzük meg, hogy létrejött-e a rekord az adatbázisban<sup>67</sup>, nyissuk meg az adatbázis konzolt:

```

kovacsg@debian:~/gyakorlat/app/controllers$ rails db
mysql> select * from ingredients;

```

<sup>6</sup>A második gyakorlaton SQLite adatbáziskezelőt használtunk.

<sup>7</sup>A gyakorlaton jelen lévő hallgatók igen beszédes szót javasoltak, ezért a string attribútum értéke üres.

id	element	amount	unit	created_at	upd
2	liszt	250	gramm	2017-02-28 12:38:24	201 12:38:24

1 row 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.