

A Rails keretrendszer Gyakorlat

Kovács Gábor

2016. szeptember 27.

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 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 ³, 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 ⁴, amely előfeltétele a tavasszal

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.

³A Rails egyik függvénykönyvtára miatt szükségünk lesz még a `zlib1g` csomagra is.

⁴Az `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 rendszer-gazdaként:

```
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 packages were automatically installed and are no longer
required:
  libgnutlsxx28 libp11-kit-dev libssl-doc libtasn1-6-dev libtasn1-doc nettle
  -dev
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  fonts-lato javascript-common libjs-jquery libruby2.3 libssl1.0.2 libyaml
  -0-2 rake ruby-did-you-mean ruby-minitest ruby-net-telnet
  ruby-power-assert ruby-test-unit ruby2.3 ruby2.3-dev ruby2.3-doc rubygems-
  integration
Suggested packages:
  bundler
The following NEW packages will be installed:
  fonts-lato javascript-common libjs-jquery libruby2.3 libssl1.0.2 libyaml
  -0-2 rake ri ruby-dev ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.3 ruby2.3-dev ruby2
  .3-doc rubygems-integration
0 upgraded, 19 newly installed, 0 to remove and 1172 not upgraded.
Need to get 1,516 kB/12.3 MB of archives.
After this operation, 64.2 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 libssl1.0.2 amd64 1.0.2j-1
 [1,291 kB]
Get:2 http://ftp.hu.debian.org/debian/ stable/main javascript-common all 11
 [6,120 B]
Get:3 http://ftp.hu.debian.org/debian/ sid/main libjs-jquery all 1.12.4-1
 [167 kB]
Get:4 http://ftp.hu.debian.org/debian/ sid/main libyaml-0-2 amd64 0.1.7-1
 [47.1 kB]
Get:5 http://ftp.hu.debian.org/debian/ sid/main rubygems-integration all
 1.10 [4,882 B]
Fetched 1,516 kB in 1s (1,074 kB/s)
Preconfiguring packages ...
Selecting previously unselected package fonts-lato.
(Reading database ... 148364 files and directories currently installed.)
Preparing to unpack .../fonts-lato_2.0-1_all.deb ...
Unpacking fonts-lato (2.0-1) ...
Selecting previously unselected package libssl1.0.2:amd64.
Preparing to unpack .../libssl1.0.2_1.0.2j-1_amd64.deb ...
Unpacking libssl1.0.2:amd64 (1.0.2j-1) ...
Selecting previously unselected package javascript-common.
Preparing to unpack .../javascript-common_11_all.deb ...
Unpacking javascript-common (11) ...
Selecting previously unselected package libjs-jquery.
Preparing to unpack .../libjs-jquery_1.12.4-1_all.deb ...
Unpacking libjs-jquery (1.12.4-1) ...
Selecting previously unselected package libyaml-0-2:amd64.
```

```

Preparing to unpack .../libyaml-0-2_0.1.7-1_amd64.deb ...
Unpacking libyaml-0-2:amd64 (0.1.7-1) ...
Selecting previously unselected package rubygems-integration.
Preparing to unpack .../rubygems-integration_1.10_all.deb ...
Unpacking rubygems-integration (1.10) ...
Selecting previously unselected package ruby-did-you-mean.
Preparing to unpack .../ruby-did-you-mean_1.0.0-2_all.deb ...
Unpacking ruby-did-you-mean (1.0.0-2) ...
Selecting previously unselected package ruby-minitest.
Preparing to unpack .../ruby-minitest_5.9.0-1_all.deb ...
Unpacking ruby-minitest (5.9.0-1) ...
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack .../ruby-net-telnet_0.1.1-2_all.deb ...
Unpacking ruby-net-telnet (0.1.1-2) ...
Selecting previously unselected package ruby-power-assert.
Preparing to unpack .../ruby-power-assert_0.3.0-1_all.deb ...
Unpacking ruby-power-assert (0.3.0-1) ...
Selecting previously unselected package ruby-test-unit.
Preparing to unpack .../ruby-test-unit_3.1.7-2_all.deb ...
Unpacking ruby-test-unit (3.1.7-2) ...
Selecting previously unselected package libruby2.3:amd64.
Preparing to unpack .../libruby2.3_2.3.1-5_amd64.deb ...
Unpacking libruby2.3:amd64 (2.3.1-5) ...
Selecting previously unselected package ruby2.3.
Preparing to unpack .../ruby2.3_2.3.1-5_amd64.deb ...
Unpacking ruby2.3 (2.3.1-5) ...
Selecting previously unselected package ruby.
Preparing to unpack .../ruby_1%3a2.3.0+4_amd64.deb ...
Unpacking ruby (1:2.3.0+4) ...
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 ruby2.3-doc.
Preparing to unpack .../ruby2.3-doc_2.3.1-5_all.deb ...
Unpacking ruby2.3-doc (2.3.1-5) ...
Selecting previously unselected package ri.
Preparing to unpack .../ri_1%3a2.3.0+4_all.deb ...
Unpacking ri (1:2.3.0+4) ...
Selecting previously unselected package ruby2.3-dev:amd64.
Preparing to unpack .../ruby2.3-dev_2.3.1-5_amd64.deb ...
Unpacking ruby2.3-dev:amd64 (2.3.1-5) ...
Selecting previously unselected package ruby-dev:amd64.
Preparing to unpack .../ruby-dev_1%3a2.3.0+4_amd64.deb ...
Unpacking ruby-dev:amd64 (1:2.3.0+4) ...
Processing triggers for fontconfig (2.11.0-6.3) ...
Processing triggers for libc-bin (2.19-18+deb8u2) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up fonts-lato (2.0-1) ...
Setting up libssl1.0.2:amd64 (1.0.2j-1) ...
Setting up javascript-common (11) ...
apache2_invoke: Enable configuration javascript-common
apache2_reload: Your configuration is broken. Not reloading Apache 2
apache2_reload: apache2: Syntax error on line 140 of /etc/apache2/apache2.
conf: Syntax error on line 1 of /etc/apache2/mods-enabled/rails.load:
Cannot load /var/lib/gems/2.1.0/gems/passenger-5.0.26/buildout/apache2/
mod_passenger.so into server: /var/lib/gems/2.1.0/gems/passenger-5.0.26/
buildout/apache2/mod_passenger.so: cannot open shared object file: No
such file or directory
Setting up libjs-jquery (1.12.4-1) ...
Setting up libyaml-0-2:amd64 (0.1.7-1) ...
Setting up rubygems-integration (1.10) ...
Setting up ruby-did-you-mean (1.0.0-2) ...

```

```

Setting up ruby-minitest (5.9.0-1) ...
Setting up ruby-net-telnet (0.1.1-2) ...
Setting up ruby-power-assert (0.3.0-1) ...
Setting up ruby-test-unit (3.1.7-2) ...
Setting up ruby2.3-doc (2.3.1-5) ...
Setting up ruby2.3 (2.3.1-5) ...
Setting up ri (1:2.3.0+4) ...
Setting up ruby (1:2.3.0+4) ...
Setting up rake (10.5.0-2) ...
Setting up libruby2.3:amd64 (2.3.1-5) ...
Setting up ruby2.3-dev:amd64 (2.3.1-5) ...
Setting up ruby-dev:amd64 (1:2.3.0+4) ...
Processing triggers for libc-bin (2.19-18+deb8u2) ...

```

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 2.3.1p112 (2016-04-26) [x86_64-linux-gnu]
root@debian:~# gem --version
2.5.1
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 (2.0.17)
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 előszerttel 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-0.7.0.gem (100%)
Successfully installed i18n-0.7.0
Fetching: thread_safe-0.3.5.gem (100%)
Successfully installed thread_safe-0.3.5
Fetching: tzinfo-1.2.2.gem (100%)
Successfully installed tzinfo-1.2.2
Fetching: concurrent-ruby-1.0.2.gem (100%)
Successfully installed concurrent-ruby-1.0.2
Fetching: activesupport-5.0.0.1.gem (100%)
Successfully installed activesupport-5.0.0.1
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: pkg-config-1.1.7.gem (100%)
Successfully installed pkg-config-1.1.7
Fetching: nokogiri-1.6.8.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-1.6.8/ext/nokogiri
      /usr/bin/ruby2.3 -r ./siteconf20160913-32548-jb8mlp.rb extconf.rb
      Using pkg-config version 1.1.7
      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-1.6.8/mkmf.log

      extconf failed, exit code 1

      Gem files will remain installed in /var/lib/gems/2.3.0/gems/nokogiri-1.6.8
      for inspection.
      Results logged to /var/lib/gems/2.3.0/extensions/x86_64-linux/2.3.0/nokogiri
      -1.6.8/gem_make.out

```

Az XML API telepítése során történt a hiba. A „Building native extensions.” 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 packages were automatically installed and are no longer
required:
  libgnutlsxx28 libp11-kit-dev libssl-doc libtasn1-6-dev libtasn1-doc nettle
  -dev
Use 'apt-get autoremove' to remove them.
The following NEW packages will be installed:
  zlib1g-dev
0 upgraded, 1 newly installed, 0 to remove and 265 not upgraded.
Need to get 0 B/206 kB of archives.
After this operation, 462 kB of additional disk space will be used.
Selecting previously unselected package zlib1g-dev:amd64.
(Reading database ... 164998 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-1.6.8
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-2.0.1.gem (100%)
Successfully installed rails-dom-testing-2.0.1
Fetching: builder-3.2.2.gem (100%)
Successfully installed builder-3.2.2
Fetching: erubis-2.7.0.gem (100%)
Successfully installed erubis-2.7.0
Fetching: actionview-5.0.0.1.gem (100%)
Successfully installed actionview-5.0.0.1
Fetching: actionpack-5.0.0.1.gem (100%)
Successfully installed actionpack-5.0.0.1
Fetching: activemodel-5.0.0.1.gem (100%)
Successfully installed activemodel-5.0.0.1
Fetching: arel-7.1.2.gem (100%)
Successfully installed arel-7.1.2
Fetching: activerecord-5.0.0.1.gem (100%)
Successfully installed activerecord-5.0.0.1
Fetching: globalid-0.3.7.gem (100%)
Successfully installed globalid-0.3.7
Fetching: activejob-5.0.0.1.gem (100%)
Successfully installed activejob-5.0.0.1
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-5.0.0.1.gem (100%)
Successfully installed actionmailer-5.0.0.1
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-0.6.4.gem (100%)
Building native extensions. This could take a while...
Successfully installed websocket-driver-0.6.4
Fetching: actioncable-5.0.0.1.gem (100%)
Successfully installed actioncable-5.0.0.1
Fetching: thor-0.19.1.gem (100%)
Successfully installed thor-0.19.1
Fetching: method_source-0.8.2.gem (100%)
Successfully installed method_source-0.8.2
Fetching: railties-5.0.0.1.gem (100%)
Successfully installed railties-5.0.0.1
Fetching: bundler-1.13.1.gem (100%)
Successfully installed bundler-1.13.1
Fetching: sprockets-3.7.0.gem (100%)
Successfully installed sprockets-3.7.0
Fetching: sprockets-rails-3.2.0.gem (100%)
Successfully installed sprockets-rails-3.2.0
Fetching: rails-5.0.0.1.gem (100%)
Successfully installed rails-5.0.0.1
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
Building native extensions. This could take a while...
Successfully installed passenger-5.0.30
Parsing documentation for passenger-5.0.30
Installing ri documentation for passenger-5.0.30
Done installing documentation for passenger after 82 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
The following packages were automatically installed and are no longer
required:
  libgnutlsxx28 libp11-kit-dev libssl-doc libtasn1-6-dev libtasn1-doc nettle
  -dev
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  libsqlite3-0 sqlite3
Suggested packages:
  sqlite3-doc
The following NEW packages will be installed:
  libsqlite3-dev
The following packages will be upgraded:
  libsqlite3-0 sqlite3
2 upgraded, 1 newly installed, 0 to remove and 263 not upgraded.
Need to get 1,077 kB of archives.
After this operation, 1,542 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://ftp.hu.debian.org/debian/ stable/main sqlite3 amd64 3.8.7.1-1+
  deb8u2 [101 kB]
Get:2 http://ftp.hu.debian.org/debian/ stable/main libsqlite3-0 amd64
  3.8.7.1-1+deb8u2 [438 kB]
Get:3 http://ftp.hu.debian.org/debian/ stable/main libsqlite3-dev amd64
  3.8.7.1-1+deb8u2 [538 kB]
Fetched 1,077 kB in 0s (2,702 kB/s)
Reading changelogs... Done
(Reading database ... 165028 files and directories currently installed.)
Preparing to unpack .../sqlite3_3.8.7.1-1+deb8u2_amd64.deb ...
Unpacking sqlite3 (3.8.7.1-1+deb8u2) over (3.8.7.1-1+deb8u1) ...
Preparing to unpack .../libsqlite3-0_3.8.7.1-1+deb8u2_amd64.deb ...
Unpacking libsqlite3-0:amd64 (3.8.7.1-1+deb8u2) over (3.8.7.1-1+deb8u1) ...
Selecting previously unselected package libsqlite3-dev:amd64.
Preparing to unpack .../libsqlite3-dev_3.8.7.1-1+deb8u2_amd64.deb ...
Unpacking libsqlite3-dev:amd64 (3.8.7.1-1+deb8u2) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up libsqlite3-0:amd64 (3.8.7.1-1+deb8u2) ...
Setting up sqlite3 (3.8.7.1-1+deb8u2) ...
Setting up libsqlite3-dev:amd64 (3.8.7.1-1+deb8u2) ...
Processing triggers for libc-bin (2.19-18+deb8u2) ...
root@debian:~# apt-get install libmysqlclient-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer
required:
  libgnutlsxx28 libp11-kit-dev libssl-doc libtasn1-6-dev libtasn1-doc nettle
  -dev
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  libmysqlclient18 mysql-common
The following NEW packages will be installed:
  libmysqlclient-dev
The following packages will be upgraded:
  libmysqlclient18 mysql-common
2 upgraded, 1 newly installed, 0 to remove and 261 not upgraded.
Need to get 1,711 kB of archives.
After this operation, 5,656 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://ftp.hu.debian.org/debian/ stable/main mysql-common all
  5.5.50-0+deb8u1 [81.8 kB]
Get:2 http://ftp.hu.debian.org/debian/ stable/main libmysqlclient18 amd64
  5.5.50-0+deb8u1 [675 kB]

```



```

Get:3 http://ftp.hu.debian.org/debian/ stable/main libmysqlclient-dev amd64
      5.5.50-0+deb8u1 [955 kB]
Fetched 1,711 kB in 1s (1,550 kB/s)
Reading changelogs... Done
(Reading database ... 165039 files and directories currently installed.)
Preparing to unpack .../mysql-common_5.5.50-0+deb8u1_all.deb ...
Unpacking mysql-common (5.5.50-0+deb8u1) over (5.5.46-0+deb8u1) ...
Preparing to unpack .../libmysqlclient18_5.5.50-0+deb8u1_amd64.deb ...
Unpacking libmysqlclient18:amd64 (5.5.50-0+deb8u1) over (5.5.46-0+deb8u1)
...
Selecting previously unselected package libmysqlclient-dev.
Preparing to unpack .../libmysqlclient-dev_5.5.50-0+deb8u1_amd64.deb ...
Unpacking libmysqlclient-dev (5.5.50-0+deb8u1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up mysql-common (5.5.50-0+deb8u1) ...
Setting up libmysqlclient18:amd64 (5.5.50-0+deb8u1) ...
Setting up libmysqlclient-dev (5.5.50-0+deb8u1) ...
Processing triggers for libc-bin (2.19-18+deb8u2) ...

```

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: sqlite3-1.3.11.gem (100%)
Building native extensions. This could take a while...
Successfully installed sqlite3-1.3.11
Parsing documentation for sqlite3-1.3.11
Installing ri documentation for sqlite3-1.3.11
Done installing documentation for sqlite3 after 0 seconds
1 gem installed
root@debian:~# gem install mysql2
Fetching: mysql2-0.4.4.gem (100%)
Building native extensions. This could take a while...
Successfully installed mysql2-0.4.4
Parsing documentation for mysql2-0.4.4
Installing ri documentation for mysql2-0.4.4
Done installing documentation for mysql2 after 0 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: libv8-3.16.14.15-x86_64-linux.gem (100%)
Successfully installed libv8-3.16.14.15-x86_64-linux
Fetching: therubyracer-0.12.2.gem (100%)
Building native extensions. This could take a while...
Successfully installed therubyracer-0.12.2
Parsing documentation for ref-2.0.0

```

```
Installing ri documentation for ref-2.0.0
Parsing documentation for libv8-3.16.14.15-x86_64-linux
Installing ri documentation for libv8-3.16.14.15-x86_64-linux
Parsing documentation for therubyracer-0.12.2
Installing ri documentation for therubyracer-0.12.2
Done installing documentation for ref, libv8, therubyracer after 1 seconds
```

A telepítés után a következő Ruby függvénykönyvtáraink érhetőek el:

```
root@debian:~# gem list

*** LOCAL GEMS ***

actioncable (5.0.0.1)
actionmailer (5.0.0.1)
actionpack (5.0.0.1)
actionview (5.0.0.1)
activejob (5.0.0.1)
activemodel (5.0.0.1)
activerecord (5.0.0.1)
activesupport (5.0.0.1)
arel (7.1.2)
bigdecimal (1.2.8)
builder (3.2.2)
bundler (1.13.1)
concurrent-ruby (1.0.2)
did_you_mean (1.0.0)
erubis (2.7.0)
execjs (2.7.0)
globalid (0.3.7)
i18n (0.7.0)
io-console (0.4.5)
json (1.8.3)
libv8 (3.16.14.15 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 (0.4.4)
net-telnet (0.1.1)
nio4r (1.2.1)
nokogiri (1.6.8)
passenger (5.0.30)
pkg-config (1.1.7)
power_assert (0.2.7)
psych (2.0.17)
rack (2.0.1)
rack-test (0.6.3)
rails (5.0.0.1)
rails-dom-testing (2.0.1)
rails-html-sanitizer (1.0.3)
railties (5.0.0.1)
rake (10.5.0)
rdoc (4.2.1)
ref (2.0.0)
sprockets (3.7.0)
sprockets-rails (3.2.0)
sqlite3 (1.3.11)
test-unit (3.1.7)
```

```
therubyracer (0.12.2)
thor (0.19.1)
thread_safe (0.3.5)
tinfo (1.2.2)
websocket-driver (0.6.4)
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 rendszergazdaként, 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.

```
kovacs@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
                                   # Default: jquery
  [--skip-gemfile], [--no-skip-gemfile] # Don't create a
                                   Gemfile
  -B, [--skip-bundle], [--no-skip-bundle] # Don't run bundle
                                   install
  -G, [--skip-git], [--no-skip-git]    # Skip .gitignore
                                   file
  [--skip-keeps], [--no-skip-keeps]    # Skip source
                                   control .keep files
  -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 szerIALIZED formában magát az adatbázis tartalmazza. A `doc` a dokumentációk gyűjtőhelye, a `lib` és `vendor` könyvtárak mások által készített

Ruby, illetve Rails függvénykönyvtárakat tartalmazhatnak. A `public` könyvtár a beépített webservert területe, az összes ott bekövetkezett esemény a `log` könyvtárban található az aktuális Rails környezetnek megfelelő állományban kerülnek naplózásra, a webservert `tmp` könyvtárban helyezheti el az átmeneti fájljait, mint például `session` azonosítókat, sütiket.

A `Gemfile` és `Rakefile` a Rails alkalmazásunk által használt Ruby függvénykönyvtárakat specifikálja, amelyek vagy a telepített Ruby és Rails környezeten vagy a már említett `lib` és `vendor` könyvtárakban érhetők 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 kommentjel 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$ bundle install
Using rake 11.3.0
Using concurrent-ruby 1.0.2
Using i18n 0.7.0
Using minitest 5.9.1
Using thread_safe 0.3.5
Using builder 3.2.2
Using erubis 2.7.0
Using mini_portile2 2.1.0
Using pkg-config 1.1.7
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 7.1.2
Using byebug 9.0.5
Using coffee-script-source 1.10.0
Using execjs 2.7.0
Using method_source 0.8.2
Using thor 0.19.1
Using debug_inspector 0.0.2
Using ffi 1.9.14
Using multi_json 1.12.1
Using libv8 3.16.14.15
Using rb-fsevent 0.9.7
Using puma 3.6.0
Using bundler 1.13.1
Using ref 2.0.0
Using sass 3.4.22
Using tilt 2.0.5
Using spring 1.7.2
Using sqlite3 1.3.11
Using turbolinks-source 5.0.0
Using tzinfo 1.2.2
Using nokogiri 1.6.8
Using rack-test 0.6.3
Using sprockets 3.7.0
Using websocket-driver 0.6.4
Using mime-types 3.1
Using coffee-script 2.4.1
Using uglifier 3.0.2
Using rb-inotify 0.9.7
Using therubyracer 0.12.2
Using turbolinks 5.0.1
Using activesupport 5.0.0.1
Using loofah 2.0.3
Using mail 2.6.4
Using listen 3.0.8
Using rails-dom-testing 2.0.1
Using globalid 0.3.7
Using activemodel 5.0.0.1
Using jbuilder 2.6.0
Using rails-html-sanitizer 1.0.3
Using spring-watcher-listen 2.0.0
Using activejob 5.0.0.1
Using activerecord 5.0.0.1
Using actionview 5.0.0.1
Using actionpack 5.0.0.1
Using actioncable 5.0.0.1
Using actionmailer 5.0.0.1
Using railties 5.0.0.1
Using sprockets-rails 3.2.0
Using coffee-rails 4.2.1
Using jquery-rails 4.2.1
Using web-console 3.3.1
Using rails 5.0.0.1
Using sass-rails 5.0.6
Bundle complete! 16 Gemfile dependencies, 66 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:

```
kovacs@debian:~/gyakorlat# gem list
```



```

*** LOCAL GEMS ***

actionmailer (4.2.6)
actionpack (4.2.6)
actionview (4.2.6)
activejob (4.2.6)
activemodel (4.2.6)
activerecord (4.2.6)
activesupport (4.2.6)
arel (6.0.3)
bigdecimal (1.2.4)
binding_of_caller (0.7.2)
builder (3.2.2)
bundler (1.12.0.pre.2)
byebug (8.2.2)
coffee-rails (4.1.1)
coffee-script (2.4.1)
coffee-script-source (1.10.0)
concurrent-ruby (1.0.1)
crass (1.0.2)
debug_inspector (0.0.2)
erubis (2.7.0)
execjs (2.6.0)
globalid (0.3.6)
i18n (0.7.0)
io-console (0.4.2)
jbuilder (2.4.1)
jquery-rails (4.1.0)
json (1.8.3, 1.8.1)
libv8 (3.16.14.13 x86_64-linux)
loofah (2.1.0.rc2, 2.0.3)
mail (2.6.4.rc2, 2.6.3)
mime-types (3.0, 2.99.1)
mime-types-data (3.2016.0221)
mini_portile2 (2.1.0, 2.0.0)
minitest (5.8.4, 4.7.5)
multi_json (1.11.2)
mysql2 (0.4.3)
nokogiri (1.6.8.rc3, 1.6.7.2)
passenger (5.0.26)
pkg-config (1.1.7)
psych (2.0.5)
rack (1.6.4)
rack-test (0.6.3)
rails (4.2.6)
rails-deprecated_sanitizer (1.0.3)
rails-dom-testing (1.0.7)
rails-html-sanitizer (1.0.3)
railties (4.2.6)
rake (10.5.0, 10.1.0)
rdoc (4.2.2, 4.1.0)
ref (2.0.0)
sass (3.4.21)
sass-rails (5.0.4)
sdoc (0.4.1)
spring (1.6.4)
sprockets (4.0.0.beta2, 3.5.2)
sprockets-rails (3.0.4)
sqlite3 (1.3.11)
test-unit (2.1.5.0)
therubyracer (0.12.2)
thor (0.19.1)

```

```
thread_safe (0.3.5)
tilt (2.0.2)
turbolinks (2.5.3)
tinfo (1.2.2)
uglifyer (2.7.2)
web-console (2.3.0)
```

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 webservert 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 database	Set the environment value for the database
db:fixtures:load environment's database	Loads fixtures into the current environment's database
db:migrate x, VERBOSE=false, SCOPE=blog)	Migrate the database (options: VERSION=x, VERBOSE=false, SCOPE=blog)
db:migrate:status	Display status of migrations
db:rollback version (specify steps w/ STEP=n)	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 database	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 structure.sql	Dumps the database structure to db/structure.sql
db:structure:load structure.sql file	Recreates the databases from the structure.sql file
db:version number	Retrieves the current schema version number
dev:cache	Toggle development mode caching on/off
initializers the order they are invoked by Rails	Print out all defined initializers in the order they are invoked by Rails
log:clear log/ to zero bytes (specify which logs with LOGS=test,development)	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 optimize, :fixme, :todo for focus)	Enumerate all annotations (use notes: optimize, :fixme, :todo for focus)
notes:custom with ANNOTATION=CUSTOM	Enumerate a custom annotation, specify with ANNOTATION=CUSTOM
restart	Restart app by touching tmp/restart.txt
routes order, with names	Print out all defined routes in match order, with names
secret secret key (this is typically used to generate a secret for cookie sessions)	Generate a cryptographically secure secret key (this is typically used to generate a secret for cookie sessions)
stats from the application or engine	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] country code ('rails time:zones[US]'), or list by UTC offset ('rails time:zones[-8]')	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 (narrow w/ tmp:cache:clear, tmp:sockets: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.

```
kovacs@debian:~/gyakorlat/config# rails s
kovacs@debian:~/gyakorlat$ rails s
=> Booting Puma
=> Rails 5.0.0.1 application starting in development on http://localhost
:3000
=> Run 'rails server -h' for more startup options
Puma starting in single mode...
* Version 3.6.0 (ruby 2.3.1-p112), codename: Sleepy Sunday Serenity
* 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 2016-09-13 20:48:05 +0200
Processing by Rails::WelcomeController#index as HTML
  Parameters: {"internal"=>true}
  Rendering /var/lib/gems/2.3.0/gems/railties-5.0.0.1/lib/rails/templates/
rails/welcome/index.html.erb
  Rendered /var/lib/gems/2.3.0/gems/railties-5.0.0.1/lib/rails/templates/
rails/welcome/index.html.erb (6.7ms)
Completed 200 OK in 35ms (Views: 18.9ms | ActiveRecord: 0.0ms)
```

A Rails a már említett Passenger plugin segítségével illeszthető Apache2 webszerverhez. Ehhez az alábbi kódrészletben mutatott parancsot kell kiadnunk. A parancs lefordítja, majd telepíti az Apache webszerver 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 , v5.0.30.

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

```

-----

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-5.0.30/
        buildout/apache2/mod_passenger.so
<IfModule mod_passenger.c>
    PassengerRoot /var/lib/gems/2.3.0/gems/passenger-5.0.30
    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 gé-

pen kezelte 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-5.0.30/
  buildout/apache2/mod_passenger.so
<IfModule mod_passenger.c>
  PassengerRoot /var/lib/gems/2.3.0/gems/passenger-5.0.30
  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 engedélyezük a gyakorlat oldalt⁵. Ennek tartalma a telepítő parancs alapján a következő 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.

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

```
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 mindegyik definiált környezethez, vagyis a fejlesztési (`development`), tesztelési (`test`) és éles (`production`). Az egyes környezetek adapter opciója határozza meg az adatbáziskezelő típusát és a használható paraméterkészletet. Az SQLite az adatait fájlokban tárolja, amelyek a Rails alkalmazás `db` könyvtárában találhatóak, és a környezet nevével azonosítottak e fájl szerint.

```
# SQLite version 3.x
# gem install sqlite3
#
# Ensure the SQLite 3 gem is defined in your Gemfile
# gem 'sqlite3'
#
default: &default
  adapter: sqlite3
  pool: 5
  timeout: 5000

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

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

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

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

Távoli gépen található MySQL esetén szükséges a Rails felhasználó hozzáférési jogosultságainak beállítása, amelyet a MySQL `mysql` adatbázisának `user` táblájában kell megtennünk. A Rails szerverének IP címére engedélyez-

nünk 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.

```
kovacs@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 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 olyan műveletekre használunk, mint az adatbázis menedzsmentje vagy a tesztvégrehajtás vezérlése. A műveletek listája alább látható.

```
kovacs@debian:~/gyakorlat# 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_ENV is development, it defaults to creating the development
               and test databases
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 development, it defaults to dropping the development and
             test databases
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

```

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

```

kovacs@debian:~/gyakorlat$ rake db:create
Created database 'gyakorlat_development'
Created database 'gyakorlat_test'
kovacs@debian:~/gyakorlat$ rails db
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 46
Server version: 5.5.46-0+deb8u1 (Debian)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its

```

```

affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

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 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.

```

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 controllert a `rails` parancs `generate` opciójával. A második argumentum (`controller`) azt mondja meg, hogy egy új controllert hozunk létre, a harmadik a controller nevét. A negyedik és minden további paraméter a controllerben definiál akciókat. E parancs négy Ruby forrásfájlt és egy 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.

```

kovacsg@debian:~/gyakorlat$ rails g
Usage: rails generate GENERATOR [args] [options]

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

Please choose a generator below.

```

```

Rails:
  assets
  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 g controller say hello
  create  app/controllers/say_controller.rb
  route   get 'say/hello'
  invoke  erb
  create  app/views/say
  create  app/views/say/hello.html.erb
  invoke  test_unit
  create  test/controllers/say_controller_test.rb
  invoke  helper
  create  app/helpers/say_helper.rb
  invoke  test_unit
  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 Rails-ben. Az `app/views/say` könyvtárban létrehozunk egy `hello.html.erb` nevű view-t, amely a `say` controller `hello` akciójához kötődik. Az eredményt a `http://localhost:3000/say/hello` linken ellenőrizhetjük.

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

Ez dinamikussá tehetjük az aktuális idő kiírásával.

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

Mivel a nézetbe nem illik logikát rakni, csak a megjelenítendő értéket, ezért áttesszük az idő lekérdezését a 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 YML fájlokkal érhetjük el. Definiáljuk a `hello` fordítását és egy időformátumot.

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

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

A nézetünkben (`app/views/say/hello.html.erb`) pedig használjuk a fordítást végző `t` azonosítójú, és a lokalizációt végző `l` 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 oszlály neve, amely a konvenció alapján egy egyes számban megadó és a szavakat `_` szimbólummal összefűző string. Ennek többes számú változatával jön létre az az adatbázisban egy tábla. A parancs kiadása négy fájlt hoz létre: egy adatbázis migrációs Ruby szkriptet, egy `ActiveRecord::Base` leszármazottat a modell osztályok közé, egy egységteszteket tartalmazó osztályt és egy tesztadatokat tartalmazó YAML fájlt.

```
kovacs@debian:~/gyakorlat/config/locales$ rails g model user username:
string password:string email:string
  invoke  active_record
  create  db/migrate/20160913192820_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 :username
      t.string :password
      t.string :email

      t.timestamps
    end
  end
end
```

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

```
kovacs@debian:~/gyakorlat/db/migrate$ rake db:migrate
(in /home/kovacs/gyakorlat)
== 20160913192820 CreateUsers: migrating ==
-- create_table(:users)
--> 0.1454s
== 20160913192820 CreateUsers: migrated (0.1458s) ==
```

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 |
| username   | 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 Rubyből 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 5.0.0.1)
irb(main):001:0> User.new
=> #<User id: nil, username: nil, password: nil, email: nil, created_at: nil,
  updated_at: nil>
irb(main):002:0> u = User.new
=> #<User id: nil, username: nil, password: nil, email: nil, created_at: nil,
  updated_at: nil>
irb(main):003:0> u.username = 'Valaki'
=> "Valaki"
irb(main):004:0> u.password = 'titok'
=> "titok"
irb(main):005:0> u.email = 'valaki@mail.bme.hu'
=> "valaki@mail.bme.hu"
irb(main):006:0> u.email
=> "valaki@mail.bme.hu"
irb(main):007:0> u.save
(0.4ms) BEGIN
SQL (7.4ms) INSERT INTO 'users' ('username', 'password', 'email', '
  created_at', 'updated_at') VALUES ('Valaki', 'titok', 'valaki@mail.bme
.hu', '2016-09-13_19:35:23', '2016-09-13_19:35:23')
(103.5ms) COMMIT
=> true
irb(main):008:0> u.id
=> 1
irb(main):009:0>
```

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

```
kovacs@debian:~/gyakorlat/db/migrate$ rails db
mysql> select * from users;
+----+-----+-----+-----+-----+
| id | username | password | email | created_at |
| updated_at |
+----+-----+-----+-----+-----+
| 1 | Valaki | titok | valaki@mail.bme.hu | 2016-09-13 19:35:23 |
| 2016-09-13 19:35:23 |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> Bye
```

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.