

A Rails keretrendszer Gyakorlat

Kovács Gábor

2016. március 8.

1. Ruby telepítése, Gemek

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

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

A Rails adatbázis adapterei és néhány további komponens fordításához szükségünk van C és C++ fordítóra, valamint a fejlesztői függvénykönyvtárakra 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, 2.1-s verzióját használjuk, ami előfeltétele a tavasszal megjelenő Rails 5-ös változatának telepítésének. A C függvénykönyvtárakhoz való illesztéshez szükségünk lesz a Ruby header fájlokra is, ezért telepítjük

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.

azok Linux csomagját is. Ezt a következő paranccsal telepíthetjük rendszer-gazdaként:

```
kovacs@debian:~# sudo bash
[sudo] password for kovacs:
root@debian:/home/kovacs# su -
root@debian:~# ruby
-su: ruby: command not found
root@debian:~# apt-get install ruby ruby-dev ri
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  javascript-common libjs-jquery libruby2.1 libyaml-0-2 ruby2.1 ruby2.1-dev
  ruby2.1-doc rubygems-integration
Suggested packages:
  bundler
The following NEW packages will be installed:
  javascript-common libjs-jquery libruby2.1 libyaml-0-2 ri ruby ruby-dev
  ruby2.1 ruby2.1-dev ruby2.1-doc rubygems-integration
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 8,207 kB of archives.
After this operation, 44.0 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://ftp.hu.debian.org/debian/ stable/main libyaml-0-2 amd64 0.1.6-3
    [50.4 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/ stable/main libjs-jquery all 1.7.2+
    dfsg-3.2 [97.5 kB]
Get:4 http://ftp.hu.debian.org/debian/ stable/main rubygems-integration all
    1.8 [4,514 B]
Get:5 http://ftp.hu.debian.org/debian/ stable/main libruby2.1 amd64 2.1.5-2+
    deb8u2 [3,278 kB]
Get:6 http://ftp.hu.debian.org/debian/ stable/main ruby2.1 amd64 2.1.5-2+
    deb8u2 [275 kB]
Get:7 http://ftp.hu.debian.org/debian/ stable/main ruby2.1-doc all 2.1.5-2+
    deb8u2 [3,368 kB]
Get:8 http://ftp.hu.debian.org/debian/ stable/main ri all 1:2.1.5+deb8u1
    [8,290 B]
Get:9 http://ftp.hu.debian.org/debian/ stable/main ruby all 1:2.1.5+deb8u1
    [9,620 B]
Get:10 http://ftp.hu.debian.org/debian/ stable/main ruby2.1-dev amd64
    2.1.5-2+deb8u2 [1,101 kB]
Get:11 http://ftp.hu.debian.org/debian/ stable/main ruby-dev all 1:2.1.5+
    deb8u1 [8,298 B]
Fetched 8,207 kB in 25s (320 kB/s)
Selecting previously unselected package libyaml-0-2:amd64.
(Reading database ... 148523 files and directories currently installed.)
Preparing to unpack .../libyaml-0-2_0.1.6-3_amd64.deb ...
Unpacking libyaml-0-2:amd64 (0.1.6-3) ...
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_1.7.2+dfsg-3.2_all.deb ...
Unpacking libjs-jquery (1.7.2+dfsg-3.2) ...
Selecting previously unselected package rubygems-integration.
Preparing to unpack .../rubygems-integration_1.8_all.deb ...
Unpacking rubygems-integration (1.8) ...
Selecting previously unselected package libruby2.1:amd64.
Preparing to unpack .../libruby2.1_2.1.5-2+deb8u2_amd64.deb ...
```

```

Unpacking libruby2.1:amd64 (2.1.5-2+deb8u2) ...
Selecting previously unselected package ruby2.1.
Preparing to unpack .../ruby2.1_2.1.5-2+deb8u2_amd64.deb ...
Unpacking ruby2.1 (2.1.5-2+deb8u2) ...
Selecting previously unselected package ruby2.1-doc.
Preparing to unpack .../ruby2.1-doc_2.1.5-2+deb8u2_all.deb ...
Unpacking ruby2.1-doc (2.1.5-2+deb8u2) ...
Selecting previously unselected package ri.
Preparing to unpack .../ri_1%3a2.1.5+deb8u1_all.deb ...
Unpacking ri (1:2.1.5+deb8u1) ...
Selecting previously unselected package ruby.
Preparing to unpack .../ruby_1%3a2.1.5+deb8u1_all.deb ...
Unpacking ruby (1:2.1.5+deb8u1) ...
Selecting previously unselected package ruby2.1-dev:amd64.
Preparing to unpack .../ruby2.1-dev_2.1.5-2+deb8u2_amd64.deb ...
Unpacking ruby2.1-dev:amd64 (2.1.5-2+deb8u2) ...
Selecting previously unselected package ruby-dev.
Preparing to unpack .../ruby-dev_1%3a2.1.5+deb8u1_all.deb ...
Unpacking ruby-dev (1:2.1.5+deb8u1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up libyaml-0-2:amd64 (0.1.6-3) ...
Setting up javascript-common (11) ...
apache2_invoke: Enable configuration javascript-common
Setting up libjs-jquery (1.7.2+dfsg-3.2) ...
Setting up rubygems-integration (1.8) ...
Setting up libruby2.1:amd64 (2.1.5-2+deb8u2) ...
Setting up ruby2.1 (2.1.5-2+deb8u2) ...
Setting up ruby2.1-doc (2.1.5-2+deb8u2) ...
Setting up ri (1:2.1.5+deb8u1) ...
Setting up ruby (1:2.1.5+deb8u1) ...
Setting up ruby2.1-dev:amd64 (2.1.5-2+deb8u2) ...
Setting up ruby-dev (1:2.1.5+deb8u1) ...
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 2.1.5p273 (2014-11-13) [x86_64-linux-gnu]
root@debian:~# gem --version
2.2.2
root@debian:~# gem list

*** LOCAL GEMS ***

bigdecimal (1.2.4)
io-console (0.4.2)
json (1.8.1)
minitest (4.7.5)
psych (2.0.5)
rake (10.1.0)
rdoc (4.1.0)
test-unit (2.1.5.0)

```

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

ban ezeket nem használjuk, teljes telepítést végzünk.

```
root@debian:/home/kovacs#gem install rdoc
Fetching: rdoc-4.2.0.gem (100%)
Depending on your version of ruby, you may need to install ruby rdoc/ri data
:

<= 1.8.6 : unsupported
 = 1.8.7 : gem install rdoc-data; rdoc-data --install
 = 1.9.1 : gem install rdoc-data; rdoc-data --install
>= 1.9.2 : nothing to do! Yay!
Successfully installed rdoc-4.2.0
Parsing documentation for rdoc-4.2.0
Installing ri documentation for rdoc-4.2.0
Done installing documentation for rdoc after 15 seconds
1 gem installed
```

Ezután telepítjük a Rails keretrendszert. A folyamatot a már említett `--no-rdoc --no-ri` kapcsolókkal felgyorsíthatjuk.

```
root@debian:~# gem install rails --no-rdoc --no-ri
Fetching: i18n-0.7.0.gem (100%)
Successfully installed i18n-0.7.0
Fetching: minitest-5.8.4.gem (100%)
Successfully installed minitest-5.8.4
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: activesupport-4.2.6.gem (100%)
Successfully installed activesupport-4.2.6
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: 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.rc3.gem (100%)
Building native extensions. This could take a while...
Successfully installed nokogiri-1.6.8.rc3
Fetching: rails-deprecated_sanitizer-1.0.3.gem (100%)
Successfully installed rails-deprecated_sanitizer-1.0.3
Fetching: rails-dom-testing-1.0.7.gem (100%)
Successfully installed rails-dom-testing-1.0.7
Fetching: crass-1.0.2.gem (100%)
Successfully installed crass-1.0.2
Fetching: loofah-2.1.0.rc2.gem (100%)
Successfully installed loofah-2.1.0.rc2
Fetching: rails-html-sanitizer-1.0.3.gem (100%)
Successfully installed rails-html-sanitizer-1.0.3
Fetching: actionview-4.2.6.gem (100%)
Successfully installed actionview-4.2.6
Fetching: rack-1.6.4.gem (100%)
Successfully installed rack-1.6.4
Fetching: rack-test-0.6.3.gem (100%)
Successfully installed rack-test-0.6.3
Fetching: actionpack-4.2.6.gem (100%)
Successfully installed actionpack-4.2.6
Fetching: activemodel-4.2.6.gem (100%)
```

```

Successfully installed activemodel-4.2.6
Fetching: arel-6.0.3.gem (100%)
Successfully installed arel-6.0.3
Fetching: activerecord-4.2.6.gem (100%)
Successfully installed activerecord-4.2.6
Fetching: globalid-0.3.6.gem (100%)
Successfully installed globalid-0.3.6
Fetching: activejob-4.2.6.gem (100%)
Successfully installed activejob-4.2.6
Fetching: mime-types-data-3.2016.0221.gem (100%)
Successfully installed mime-types-data-3.2016.0221
Fetching: mime-types-3.0.gem (100%)
Successfully installed mime-types-3.0
Fetching: mail-2.6.4.rc2.gem (100%)
Successfully installed mail-2.6.4.rc2
Fetching: actionmailer-4.2.6.gem (100%)
Successfully installed actionmailer-4.2.6
Fetching: thor-0.19.1.gem (100%)
Successfully installed thor-0.19.1
Fetching: railties-4.2.6.gem (100%)
Successfully installed railties-4.2.6
Fetching: bundler-1.12.0.pre.2.gem (100%)
Successfully installed bundler-1.12.0.pre.2
Fetching: concurrent-ruby-1.0.1.gem (100%)
Successfully installed concurrent-ruby-1.0.1
Fetching: sprockets-4.0.0.beta2.gem (100%)
Successfully installed sprockets-4.0.0.beta2
Fetching: sprockets-rails-3.0.4.gem (100%)
Successfully installed sprockets-rails-3.0.4
Fetching: rails-4.2.6.gem (100%)
Successfully installed rails-4.2.6
35 gems installed

```

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

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

```

root@debian:~# gem install passenger
Fetching: passenger-5.0.26.gem (100%)
Building native extensions. This could take a while...
Successfully installed passenger-5.0.26
Parsing documentation for passenger-5.0.26
Installing ri documentation for passenger-5.0.26
Done installing documentation for passenger after 9 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 libmysqlclient-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  sqlite3-doc
The following NEW packages will be installed:
  libmysqlclient-dev libsqlite3-dev
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,495 kB of archives.
After this operation, 7,204 kB of additional disk space will be used.
Get:1 http://ftp.hu.debian.org/debian/ stable/main libmysqlclient-dev amd64
    5.5.46-0+deb8u1 [957 kB]
Get:2 http://ftp.hu.debian.org/debian/ stable/main libsqlite3-dev amd64
    3.8.7.1-1+deb8u1 [538 kB]
Fetched 1,495 kB in 0s (13.0 MB/s)
Selecting previously unselected package libmysqlclient-dev.
(Reading database ... 165232 files and directories currently installed.)
Preparing to unpack .../libmysqlclient-dev_5.5.46-0+deb8u1_amd64.deb ...
Unpacking libmysqlclient-dev (5.5.46-0+deb8u1) ...
Selecting previously unselected package libsqlite3-dev:amd64.
Preparing to unpack .../libsqlite3-dev_3.8.7.1-1+deb8u1_amd64.deb ...
Unpacking libsqlite3-dev:amd64 (3.8.7.1-1+deb8u1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up libmysqlclient-dev (5.5.46-0+deb8u1) ...
Setting up libsqlite3-dev:amd64 (3.8.7.1-1+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: sqlite3-1.3.11.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 1 seconds
1 gem installed
root@debian:~# gem install mysql2
Fetching: mysql2-0.4.3.gem (100%)
Building native extensions. This could take a while...
Successfully installed mysql2-0.4.3
Parsing documentation for mysql2-0.4.3
Installing ri documentation for mysql2-0.4.3
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.

2. Rails alkalmazás létrehozása

A Rails használatához a rails szkriptet használjuk, amely Debian/Ubuntu Linux és 2.1-es ruby esetén a /usr/local/bin/ könyvtárba kerül.

A -d kapcsoló különös jelentőséggel bír számunkra, ezzel adhatjuk meg a használni kívánt adatbáziskezelő típusát, ami lehet `mysql`, `oracle`, `postgresql`, `sqlite3`, `frontbase` vagy `ibm_db`. A gyakorlatok keretében ezek közül a `sqlite3`-ot mint alapértelmezett adatbáziskezelőt és a `mysql`-t fogjuk használni. A -B kapcsoló mellőzi a Rails alkalmazásunk Ruby függőségeinek feloldását, amelyet azonban később pótolnunk kell. Javascript API-nak a `jquery` tökéletesen meg fog felelni számunkra, így azt nem módosítjuk.

```
kovacs@debian:~# rails
Usage:
  rails new APP_PATH [options]

Options:
  -r, [--ruby=PATH]                # Path to the Ruby
    binary of your choice          # Default: /usr/bin
                                   /ruby
  -m, [--template=TEMPLATE]        # Path to some
    application template (can be a filesystem path or URL)
  --skip-gemfile, [--no-skip-gemfile] # Don't create a
    Gemfile
  -B, [--skip-bundle], [--no-skip-bundle] # Don't run bundle
    install
  -G, [--skip-git], [--no-skip-git]    # Skip .gitignore
    file
  --skip-keeps, [--no-skip-keeps]     # Skip source
    control .keep files
  -O, [--skip-active-record], [--no-skip-active-record] # Skip Active
    Record files
  -S, [--skip-sprockets], [--no-skip-sprockets] # Skip Sprockets
    files
  --skip-spring, [--no-skip-spring]   # Don't install
    Spring application preloader
  -d, [--database=DATABASE]          # Preconfigure for
    selected database (options: mysql/oracle/postgresql/sqlite3/frontbase/
    ibm_db/sqlserver/jdbcmysql/jdbcsqlite3/jdbcpostgresql/jdbc)
                                   # Default: sqlite3
  -j, [--javascript=JAVASCRIPT]     # Preconfigure for
    selected JavaScript library     # Default: jquery
  -J, [--skip-javascript], [--no-skip-javascript] # Skip JavaScript
    files
  --dev, [--no-dev]                 # Setup the
    application with Gemfile pointing to your Rails checkout
  --edge, [--no-edge]               # Setup the
    application with Gemfile pointing to Rails repository
  --skip-turbolinks, [--no-skip-turbolinks] # Skip turbolinks
    gem
  -T, [--skip-test-unit], [--no-skip-test-unit] # Skip Test::Unit
    files
  --rc=RC                            # Path to file
    containing extra configuration options for rails command
  --no-rc, [--no-no-rc]             # Skip loading of
```

extra configuration options from .railsrc file

Runtime options:

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

Rails options:

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

Description:

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

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

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

Example:

```
rails new ~/Code/Ruby/weblog
```

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

A Rails keretrendszerben a **rails** parancs az az univerzális eszköz mellyel többek között új komponenseket hozhatunk létre és elindíthatjuk a beépített webszerveret. 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.

```
kovacsg@debian:~# rails new gyakorlat -B
create
create  README.rdoc
create  Rakefile
create  config.ru
create  .gitignore
create  Gemfile
create  app
create  app/assets/javascripts/application.js
create  app/assets/stylesheets/application.css
create  app/controllers/application_controller.rb
create  app/helpers/application_helper.rb
create  app/views/layouts/application.html.erb
create  app/assets/images/.keep
create  app/mailers/.keep
```



```
create app/models/.keep
create app/controllers/concerns/.keep
create app/models/concerns/.keep
create bin
create bin/bundle
create bin/rails
create bin/rake
create bin/setup
create config
create config/routes.rb
create config/application.rb
create config/environment.rb
create config/secrets.yml
create config/environments
create config/environments/development.rb
create config/environments/production.rb
create config/environments/test.rb
create config/initializers
create config/initializers/assets.rb
create config/initializers/backtrace_silencers.rb
create config/initializers/cookies_serializer.rb
create config/initializers/filter_parameter_logging.rb
create config/initializers/inflections.rb
create config/initializers/mime_types.rb
create config/initializers/session_store.rb
create config/initializers/wrap_parameters.rb
create config/locales
create config/locales/en.yml
create config/boot.rb
create config/database.yml
create db
create db/seeds.rb
create lib
create lib/tasks
create lib/tasks/.keep
create lib/assets
create lib/assets/.keep
create log
create log/.keep
create public
create public/404.html
create public/422.html
create public/500.html
create public/favicon.ico
create public/robots.txt
create test/fixtures
create test/fixtures/.keep
create test/controllers
create test/controllers/.keep
create test/mailers
create test/mailers/.keep
create test/models
create test/models/.keep
create test/helpers
create test/helpers/.keep
create test/integration
create test/integration/.keep
create test/test_helper.rb
create tmp/cache
create tmp/cache/assets
create vendor/assets/javascripts
create vendor/assets/javascripts/.keep
```

```
create vendor/assets/stylesheets
create vendor/assets/stylesheets/.keep
```

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 fájlokat, 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 beágyazott 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.

```
kovacs@debian:~# cd gyakorlat/
kovacs@debian:~/gyakorlat# vim Gemfile
kovacs@debian:~/gyakorlat# bundle install
Fetching gem metadata from https://rubygems.org/
Fetching version metadata from https://rubygems.org/
Fetching dependency metadata from https://rubygems.org/
Resolving dependencies .....

Your user account isn't allowed to install to the system Rubygems.
You can cancel this installation and run:

    bundle install --path vendor/bundle

to install the gems into ./vendor/bundle/, or you can enter your password
and install the bundled gems to Rubygems using sudo.

Password:
Installing rake 10.5.0
Using i18n 0.7.0
Installing json 1.8.3 with native extensions
Using minitest 5.8.4
Using thread_safe 0.3.5
Using builder 3.2.2
Using erubis 2.7.0
Installing mini_portile2 2.0.0
Using rack 1.6.4
Installing mime-types 2.99.1
Using arel 6.0.3
Installing debug_inspector 0.0.2 with native extensions
Using bundler 1.12.0.pre.2
Installing byebug 8.2.2 with native extensions
Installing coffee-script-source 1.10.0
Installing execjs 2.6.0
Using thor 0.19.1
Using concurrent-ruby 1.0.1
Installing multi_json 1.11.2
Installing libv8 3.16.14.13
Installing ref 2.0.0
Installing sass 3.4.21
Installing tilt 2.0.2
Installing spring 1.6.4
Using sqlite3 1.3.11
Installing rdoc 4.2.2
Using tzinfo 1.2.2
Installing nokogiri 1.6.7.2 with native extensions
Using rack-test 0.6.3
Installing mail 2.6.3
Installing binding_of_caller 0.7.2 with native extensions
Installing coffee-script 2.4.1
Installing uglifier 2.7.2
Installing sprockets 3.5.2
Installing therubyracer 0.12.2 with native extensions
Installing sdoc 0.4.1
Using activesupport 4.2.6
Installing loofah 2.0.3
Using rails-deprecated_sanitizer 1.0.3
Using globalid 0.3.6
```

```

Using activemodel 4.2.6
Installing jbuilder 2.4.1
Using rails-html-sanitizer 1.0.3
Using rails-dom-testing 1.0.7
Using activejob 4.2.6
Using activerecord 4.2.6
Using actionview 4.2.6
Using actionpack 4.2.6
Using actionmailer 4.2.6
Using railties 4.2.6
Using sprockets-rails 3.0.4
Installing coffee-rails 4.1.1
Installing jquery-rails 4.1.0
Using rails 4.2.6
Installing sass-rails 5.0.4
Installing web-console 2.3.0
Installing turbolinks 2.5.3
Bundle complete! 13 Gemfile dependencies, 57 gems now installed.
Use 'bundle show [gemname]' to see where a bundled gem is installed.
Post-install message from rdoc:
Depending on your version of ruby, you may need to install ruby rdoc/ri data
:

<= 1.8.6 : unsupported
= 1.8.7 : gem install rdoc-data; rdoc-data --install
= 1.9.1 : gem install rdoc-data; rdoc-data --install
>= 1.9.2 : nothing to do! Yay!

```

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)
tzinfo (1.2.2)
uglifier (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")
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"

```

```
In addition to those, 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")
```

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

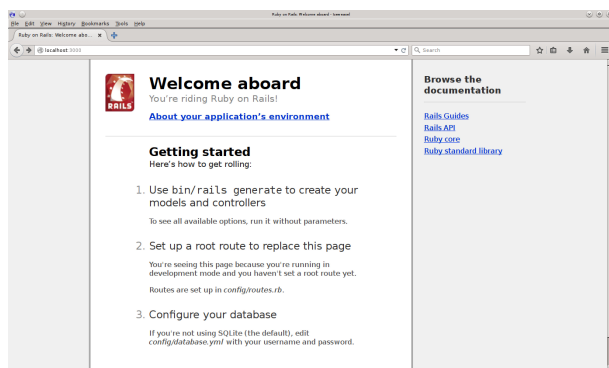
A rails parancs gyakran használt argumentuma a `console` vagy röviden `c`, amivel az első gyakorlatról megismert interaktív Ruby értelmezőt indíthatunk, melyben a Rails alkalmazásunk környezeti beállításai inicializálásra kerültek.

```
kovacs@debian:~/gyakorlat# rails c
Loading development environment (Rails 4.2.6)
irb(main):001:0 >
```

3. Web- és adatbázisszerver kapcsolat beállítása

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

A beépített webszervert a `rails server` vagy röviden `s` opciójával indíthatjuk el a Rails alkalmazásunk tetszőleges könyvtárából. Ha a szerveret háttéralkalmazásként kívánjuk futtatni, akkor a `-d` kapcsolót is hozzá kell fűzünk a parancssorhoz. Ezután a szerver elérhető a `http://localhost:3000` webcímen, lásd 1. ábra.



1. ábra. A beágyazott webszerver elérése

```
kovacs@debian:~/gyakorlat/config# rails s
=> Booting WEBrick
=> Rails 4.2.6 application starting in development on http://localhost:3000
=> Run 'rails server -h' for more startup options
=> Ctrl-C to shutdown server
```

```
[2016-03-08 12:56:21] INFO WEBrick 1.3.1
[2016-03-08 12:56:21] INFO ruby 2.1.5 (2014-11-13) [x86_64-linux-gnu]
[2016-03-08 12:56:21] INFO WEBrick::HTTPServer#start: pid=8458 port=3000
```

A Rails a már említett Passenger plugin segítségével illeszthető 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.

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

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

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

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

```
LoadModule passenger_module /var/lib/gems/2.1.0/gems/passenger-5.0.26/
  buildout/apache2/mod_passenger.so
<IfModule mod_passenger.c>
  PassengerRoot /var/lib/gems/2.1.0/gems/passenger-5.0.26
  PassengerDefaultRuby /usr/bin/ruby2.1
</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 oldalt⁴. Ennek tartalma a telepítő parancs alapján a következő lehet:

```
RackEnv development

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

⁴Alternatív megoldás: `ln -s /etc/apache2/sites-available/gyakorlat.conf /etc/apache2/sites-enabled`

```

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.

```

# 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 használjuk⁵. 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.

```

kovacs@debian:~/gyakorlat/db# rails db
SQLite version 3.7.13
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
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 assets:clean[keep] # Remove old compiled assets
rake assets:clobber # Remove compiled assets
rake assets:environment # Load asset compile environment

```

⁵A Rails 4.2.4-es változatához a következő függőség kell: `gem 'mysql2', '> 0.3.18'` Ne feledkezzünk el a `bundle update` parancs kiadásáról sem.

```

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
    :...
rake db:drop                    # Drops the database from
    DATABASE_URL or config/database.yml for the current RAILS_ENV (use db:dr
    ...
rake db:fixtures:load           # Load fixtures into the current
    environment's database
rake db:migrate                 # Migrate the database (options:
    VERSION=x, VERBOSE=false, SCOPE=blog)
rake db:migrate:status          # Display status of migrations
rake db:rollback                # Rolls the schema back to the
    previous version (specify steps w/ STEP=n)
rake db:schema:cache:clear      # Clear a db/schema_cache.dump file
rake db:schema:cache:dump       # Create a db/schema_cache.dump file
rake db:schema:dump             # Create a db/schema.rb file that is
    portable against any DB supported by AR
rake db:schema:load             # Load a schema.rb file into the
    database
rake db:seed                    # Load the seed data from db/seeds.
    rb
rake db:setup                   # Create the database, load the
    schema, and initialize with the seed data (use db:reset to also dr...
rake db:structure:dump          # Dump the database structure to db/
    structure.sql
rake db:structure:load          # Recreate the databases from the
    structure.sql file
rake db:version                 # Retrieves the current schema
    version number
rake doc:app                    # Generate docs for the app — also
    available doc:rails, doc:guides (options: TEMPLATE=/rdoc-templ...
rake log:clear                  # Truncates all *.log files in log/
    to zero bytes (specify which logs with LOGS=test,development)
rake middleware                  # Prints out your Rack middleware
    stack
rake notes                       # Enumerate all annotations (use
    notes:optimize, :fixme, :todo for focus)
rake notes:custom               # Enumerate a custom annotation,
    specify with ANNOTATION=CUSTOM
rake rails:template             # Applies the template supplied by
    LOCATION=(/path/to/template) or URL
rake rails:update               # Update configs and some other
    initially generated files (or use just update:configs or update:bin)
rake routes                     # Print out all defined routes in
    match order, with names
rake secret                     # Generate a cryptographically
    secure secret key (this is typically used to generate a secret for ...
rake stats                       # Report code statistics (KLOCs, etc
    ) from the application or engine
rake test                       # Runs all tests in test folder
rake test:all                   # Run tests quickly by merging all
    types and not resetting db
rake test:all:db                # Run tests quickly, but also reset
    db
rake test:db                    # Run tests quickly, but also reset
    db

```

```
rake time:zones:all # Displays all time zones, also
                   available: time:zones:us, time:zones:local — filter with OFFSET p...
rake tmp:clear # Clear session, cache, and socket
               files from tmp/ (narrow w/ tmp:sessions:clear, tmp:cache:clear,...)
rake tmp:create # Creates tmp directories for
                sessions, cache, sockets, and pids
```

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

```
kovacs@debian:~/gyakorlat# rake db:create
kovacs@debian:~/gyakorlat# rails db
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 53
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> use gyakorlat_development;
Database changed
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 webservert IP címe után a Controller osztály neve (`:controller`), majd a Controller osztály egy metódusa (`:action`), majd egy adatbázis azonosító (`:id`), és végül formázási útmutató következik, például `.html` vagy `.xml`. Az utolsó három megadása opcionális.

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

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

Hozzunk létre egy új kontrollert a `rails` parancs `generate` opciójával. A második argumentum (`controller`) azt mondja meg, hogy egy új kontrollert

hozunk létre, a harmadik a controller nevét. A negyedik és minden további paraméter a controllerben definiál akciókat. E parancs négy Ruby forrásfájlt és egy könyvárat hoz létre az akcióknak megfelelő weboldalak, view-k számára. A controller nevének megfelelő controller osztályt (`say_controller.rb`), helper osztályt, illetve ezek funkcionális és egységtesztjéhez használható osztályokat.

```
kovacs@debian:~/gyakorlat# 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.

Rails:
  assets
  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:job
  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 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 controllerbe, annak is az akciónak megfelelő metódusába, a `hello`-ba

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

A nézetben pedig csak hivatkozunk a Controller példányváltozóra.

```
<%= @time %>
```

A 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"
```

A nézetünkben (`app/views/say/hello.html.erb`) pedig használjuk a fordítást végző `t`, és a lokalizációt végző `l` függvényeket.

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

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

```
kovacs@debian:~/gyakorlat/app/views/say# rails g model user email:string
name:string password:string birthdate:datetime account_number:string
  invoke  active_record
  create  db/migrate/20160308121613_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`, egy `email` azonosítójú string típusú, egy `birthdate` nevű dátum típusú, és egy `account_number` nevű string típusú attribútum szerepelne.

```
class CreateUsers < ActiveRecord::Migration
  def change
    create_table :users do |t|
      t.string :email
      t.string :name
      t.string :password
      t.datetime :birthdate
      t.string :account_number

      t.timestamps null: false
    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)
== 20160308121613 CreateUsers: migrating
-----
-- create_table(:users)
--> 0.0491s
== 20160308121613 CreateUsers: migrated (0.0496s)
-----
```

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> use gyakorlat_development;
Database changed
mysql> show tables;
+-----+
| Tables_in_gyakorlat_development |
+-----+
| schema_migrations |
| users |
+-----+
2 rows in set (0.00 sec)

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

mysql> Bye

```

Nézzük meg, miként tudunk az adatbázishoz Rubyből hozzáférni. Nyisunk egy konzolt, és hozzunk létre egy felhasználót (1.sor), állítsuk be a tulajdonságait (2-7. sorok), mentjük el az adatbázisba (8. 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 4.2.6)
irb(main):001:0> u = User.new
=> #<User id: nil, email: nil, name: nil, password: nil, birthdate: nil,
  account_number: nil, created_at: nil, updated_at: nil>
irb(main):002:0> u.new_record?
=> true
irb(main):003:0> u.email = 'kovacs@tmit.bme.hu'
=> "kovacs@tmit.bme.hu"
irb(main):004:0> u.name = 'Kovacs_Gabor'
=> "Kovacs_Gabor"
irb(main):005:0> u.password = 'titok'
=> "titok"
irb(main):006:0> u.birthdate = Time.now - 2.years
=> 2014-03-08 13:23:01 +0100
irb(main):007:0> u.account_number = '11111111-11111111-11111111'
=> "11111111-11111111-11111111"
irb(main):008:0> u.save
(0.3ms) BEGIN
SQL (27.6ms) INSERT INTO 'users' ('email', 'name', 'password', 'birthdate', 'account_number', 'created_at', 'updated_at') VALUES ('kovacs@tmit

```

```

    .bme.hu', 'Kovacs_Gabor', 'titok', '2014-03-08_12:23:01', '
    11111111-11111111-11111111', '2016-03-08_12:23:55', '2016-03-08_
    12:23:55')
    (3.4ms) COMMIT
=> true
irb(main):009:0> u.new_record?
=> false
irb(main):010:0> u
=> #<User id: 1, email: "kovacsg@tmit.bme.hu", name: "Kovacs Gabor",
    password: "titok", birthdate: "2014-03-08 12:23:01", account_number:
    "11111111-11111111-11111111", created_at: "2016-03-08 12:23:55",
    updated_at: "2016-03-08 12:23:55">
irb(main):011:0> u = User.find 1
    User Load (6.0ms) SELECT `users`.* FROM `users` WHERE `users`.`id` = 1
    LIMIT 1
=> #<User id: 1, email: "kovacsg@tmit.bme.hu", name: "Kovacs Gabor",
    password: "titok", birthdate: "2014-03-08 12:23:01", account_number:
    "11111111-11111111-11111111", created_at: "2016-03-08 12:23:55",
    updated_at: "2016-03-08 12:23:55">
irb(main):012:0>

```

Ellenőrizzük meg, hogy megjelent-e ez az adatbázisban.

```

mysql> select * from users;
+-----+-----+-----+-----+-----+
| id | email                | name       | password | birthdate                |
|    | account_number       | created_at |          | updated_at               |
+-----+-----+-----+-----+-----+
| 1  | kovacsg@tmit.bme.hu | Kovacs Gabor | titok    | 2014-03-08 12:23:01 |
|    | 11111111-11111111-11111111 | 2016-03-08 12:23:55 | 2016-03-08 12:23:55 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

A kontroller feladata az adatbázisműveletek koordinálása, adjuk át e felhasználó email címét a `hello` akcióhoz tartozó nézetnek, amit módosítsunk úgy, hogy e felhasználó üdvözlje a világot!

```

def hello
  @email = User.find(1).email
  @time = Time.now
end

```

```

<h1><%= t :hello %> by <%= @email %></h1>
<%= l @time, format: :default %>

```

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

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

```

<!DOCTYPE html>
<html>
<head>

```




2. ábra. A hello akció megjelenítve

```
<title>Gyakorlat</title>
<meta name="csrf-param" content="authenticity_token" />
<meta name="csrf-token" content="0
  Y8PVeek024FaSIzIhza8UaQbnGKRKA7wgULI0qmD56JnsQxPRUDGLCGCinhK1gclyjfH40dhRZt
  /SRUS2juw==" />
</head>
<body>

<h1>Hello , world! by kovacsg@tmit.bme.hu</h1>
2016.03.08 13:26:53

</body>
</html>
```

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.