

Leccion 1 - Ruby sobre Rieles

De Telefonica I+D - Division 6123

Por: Edwin Plauchu
e-mail: j4nux@gmail.com

Tabla de contenidos

- 1 Lo necesario
 - 1.1 Instalando Rails, Scite y Mysql Server
 - 1.2 Instalacion en windows
- 2 Creando una agenda de contactos en Rails
 - 2.1 Creacion de un Esquema en Mysql
 - 2.2 Creando nuestro primer proyecto en Rails
 - 2.2.1 Creacion de nuestro modelo de datos
 - 2.2.1.1 Creando la Tabla
 - 2.2.1.2 Creando el Modelo
 - 2.2.2 Preparando el Andamiaje
 - 2.2.3 Verificando lo realizado

Lo necesario

Para que nos sea posible empezar a trabajar con Ruby on Rails. Necesitaremos solo 3 cosas:

- Un buen editor para desarrolladores guapos. :)
- La suite de rails
- Y un DBMS soportado por Rails (Oracle, Postgres, Mysql y algunos otros)

Instalando Rails, Scite y Mysql Server

Esto es una de las cosas mas sencillas en Ubuntu linux. Basta con utilizar la herramienta "apt-get":

```

root@pianodaemon-laptop:~# apt-get install rails mysql-server scite
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalaron de forma automática los siguientes paquetes y ya no son necesarios.
  libqt3-mt-psql libqscintilla7
Utilice «apt-get autoremove» para eliminarlos.
Se instalarán los siguientes paquetes extras:
  irb1.8 liberb-ruby libreadline-ruby1.8 libredcloth-ruby1.8
  libsqlite3-ruby1.8 rake rdoc rdoc1.8
Paquetes sugeridos:
  libapache2-mod-ruby libapache-mod-ruby libapache2-mod-fcgid libfcgi-ruby1.8
  graphviz
Paquetes recomendados
  irb
Se instalarán los siguientes paquetes NUEVOS:
  irb1.8 liberb-ruby libreadline-ruby1.8 libredcloth-ruby1.8
  libsqlite3-ruby1.8 rails rake rdoc rdoc1.8
0 actualizados, 9 se instalarán, 0 para eliminar y 0 no actualizados.
2 no instalados del todo o eliminados.
Necesito descargar 3324kB de archivos.
Se utilizarán 20.0MB de espacio de disco adicional después de desempaquetar.
¿Desea continuar [S/n]?

```

Perfecto, ya conseguimos todo lo que necesitabamos. Empecemos.

Instalacion en windows

Busca los instaladores para win32 de mysql, scite y ruby. Tambien los hay..... se trabaja exactamente igual con ellos tanto para windows o linux.

Creando una agenda de contactos en Rails

Creacion de un Esquema en Mysql

Ha estas alturas del partido, tu ya tendras mysql instalado y corriendo en tu sistema. Podras accederlo de la siguiente manera:

```
pianodaemon@pianodaemon-laptop:~$ mysql -u root
Welcome to the MySQL monitor.  Commands END with ; OR \g.
Your MySQL connection id IS 8
Server version: 5.0.45-Debian_1ubuntu3.1-LOG Debian etch distribution

Type 'help;' OR '\h' for HELP. Type '\c' to clear the buffer.

mysql> CREATE schema APP1; GRANT all on APP1.* to pianodaemon@'%' identified BY '123qwe';
```

Hemos creado un schema APP1 que podra ser accesado desde cualquier host por el usuario pianodaemon con password 123qwe :).

Creando nuestro primer proyecto en Rails

```
pianodaemon@pianodaemon-laptop:~$ rails ejemplo_uno
create
create  app/controllers
create  app/helpers
create  app/models
create  app/views/layouts
create  config/environments
create  components
create  db
create  doc
create  lib
create  lib/tasks
create  log
create  public/images
create  public/javascripts
create  public/stylesheets
create  script/performance
create  script/process
create  test/fixtures
create  test/functional
create  test/integration
create  test/mocks/development
create  test/mocks/test
create  test/unit
create  vendor
create  vendor/plugins
create  tmp/sessions
create  tmp/sockets
create  tmp/cache
create  tmp/pids
create  Rakefile
create  README
create  app/controllers/application.rb
create  app/helpers/application_helper.rb
create  test/test_helper.rb
create  config/database.yml
create  config/routes.rb
create  public/.htaccess
create  config/boot.rb
create  config/environment.rb
create  config/environments/production.rb
create  config/environments/development.rb
create  config/environments/test.rb
create  script/about
create  script/breakpointer
create  script/console
create  script/destroy
create  script/generate
create  script/performance/benchmark
create  script/performance/profiler
create  script/process/reaper
create  script/process/spawner
create  script/process/inspector
create  script/runner
create  script/server
create  script/plugin
create  public/dispatch.rb
create  public/dispatch.cgi
create  public/dispatch.fcgi
create  public/404.html
create  public/500.html
create  public/index.html
create  public/favicon.ico
create  public/robots.txt
create  public/images/rails.png
create  public/javascripts/prototype.js
create  public/javascripts/effects.js
create  public/javascripts/dragdrop.js
create  public/javascripts/controls.js
create  public/javascripts/application.js
create  doc/README_FOR_APP
create  log/server.log
create  log/production.log
create  log/development.log
create  log/test.log
```

Editamos el archivo database.yml, para permitir a rails el acceso a esquema de datos en APP1 albergado en nuestro DBMS mysql

```
vi ~/ejemplo_uno/config/database.yml
```

```

# MySQL (default setup). Versions 4.1 and 5.0 are recommended.
#
# Install the MySQL driver:
#   gem install mysql
# On MacOS X:
#   gem install mysql -- --include=/usr/local/lib
# On Windows:
#   gem install mysql
#       Choose the win32 build.
#       Install MySQL and put its /bin directory on your path.
#
# And be sure to use new-style password hashing:
#   http://dev.mysql.com/doc/refman/5.0/en/old-client.html
development:
  adapter: mysql
  database: APP1
  username: pianodaemon
  password: 123qwe
  socket: /var/run/mysqld/mysqld.sock

# Warning: The database defined as 'test' will be erased and
# re-generated from your development database when you run 'rake'.
# Do not set this db to the same as development or production.
test:
  adapter: mysql
  database: ejemplo_uno_test
  username: root
  password:
  socket: /var/run/mysqld/mysqld.sock

production:
  adapter: mysql
  database: APP1
  username: pianodaemon
  password: 123qwe
  socket: /var/run/mysqld/mysqld.sock

```

Listo ya tenemos creado nuestro primer proyecto en rails.

Creacion de nuestro modelo de datos

Los modelos de datos estan muy relacionados a la existencia de una tabla, asi que si creamos un modelo, debera de existir antes su respectiva tabla. Recordar siempre esta reglas

- Modelos de datos en SINGULAR
- Tablas en PLURAL
- Objetos que hacen referencia a estos modelos de dato, estaran en SINGULAR.

Creando la Tabla

Primero creamos la tabla.... Observece que la tabla personas esta en PLURAL y guardara objetos persona que es un SINGULAR:

```

mysql> CREATE TABLE personas (
-> id INT(11) NOT NULL AUTO INCREMENT,
-> primer nombre VARCHAR(50),
-> segundo nombre VARCHAR(50),
-> email VARCHAR(50),
-> PRIMARY KEY (id) );
Query OK, 0 rows affected (0.08 sec)

```

Creando el Modelo

Ahora creamos el modelo:

```

pianodaemon@pianodaemon-laptop:~/ejemplo_uno$ ruby script/generate model persona
exists app/models/
exists test/unit/
exists test/fixtures/
create app/models/persona.rb
create test/unit/persona_test.rb
create test/fixtures/personas.yml
create db/migrate
create db/migrate/001_create_personas.rb

```

Preparando el Andamiaje

Al andamiaje en ingles se le llama **scaffold**, pero que es esto... veamoslo a continuacion:

1. El andamiaje se hace mediante un controller

```
pianodaemon@pianodaemon-laptop:~/ejemplo_uno$ ruby script/generate controller operaciones
exists app/controllers/
exists app/helpers/
create app/views/operaciones
exists test/functional/
create app/controllers/operaciones_controller.rb
create test/functional/operaciones_controller_test.rb
create app/helpers/operaciones_helper.rb
```

Listo hemos creado nuestro controller operaciones... (un controller es como un servlet.. solo que mas efectivo.. y mas bonito!).

Ahora editaremos nuestro controller

```
$ vi ~/ejemplo_uno/app/controllers/operaciones_controller.rb
```

```
class OperacionesController < ApplicationController
  scaffold :persona
end
```

Listo ya tenemos un altas bajas y cambios en ruby on rails

Verificando lo realizado

Arrancamos el servidor web de ruby on rails

```
pianodaemon@pianodaemon-laptop:~/ejemplo_uno$ ruby script/server
=> Booting WEBrick...
=> Rails application started on http://127.0.0.1:3000
=> Ctrl-C to shutdown server; call with --help for options
[2008-02-01 16:10:19] INFO WEBrick 1.3.1
[2008-02-01 16:10:19] INFO ruby 1.8.6 (2007-06-07) [i486-linux]
[2008-02-01 16:10:19] INFO WEBrick::HTTPServer#start: pid=10961 port=3000
```

Sobre un cliente http ingresamos la siguiente URL **http://localhost:3000/operaciones/new**

Obtenido de "[http://172.16.161.129/wiki/index.php/Leccion 1 - Ruby sobre Rieles](http://172.16.161.129/wiki/index.php/Leccion_1_-_Ruby_sobre_Rieles)"

- Esta página fue modificada por última vez el 23:40, 1 feb 2008.