Ruby Explorations XI

Mark Keane...CSI...UCD

Rails Basics

More Ruby on Rails:

A: understanding Rake's migration & rollback

B: itunes in Rails

C: using **rails console** to look at db

D: **link_to** has a third possible arg...

E: When time passes and my bundle is old...

Controller

Model

$ rake db:migrate

Views

$ rails server

Part A:

Understanding Rake migration & rollback

Recall...

Database tables are produced by a combination of migrations and editing of tables (and changing dbs directly)...in all of this **schema.rb** is the definitive source for the db structure

it is very easy to fail to understand how this all works and to encounter strange errors

so, lets look at it carefully....

http://guides.rubyonrails.org/command_line.html

Steps

$ rails new chicken

$ cd chicken

migration_file only created

$ rails generate model Bird name:string

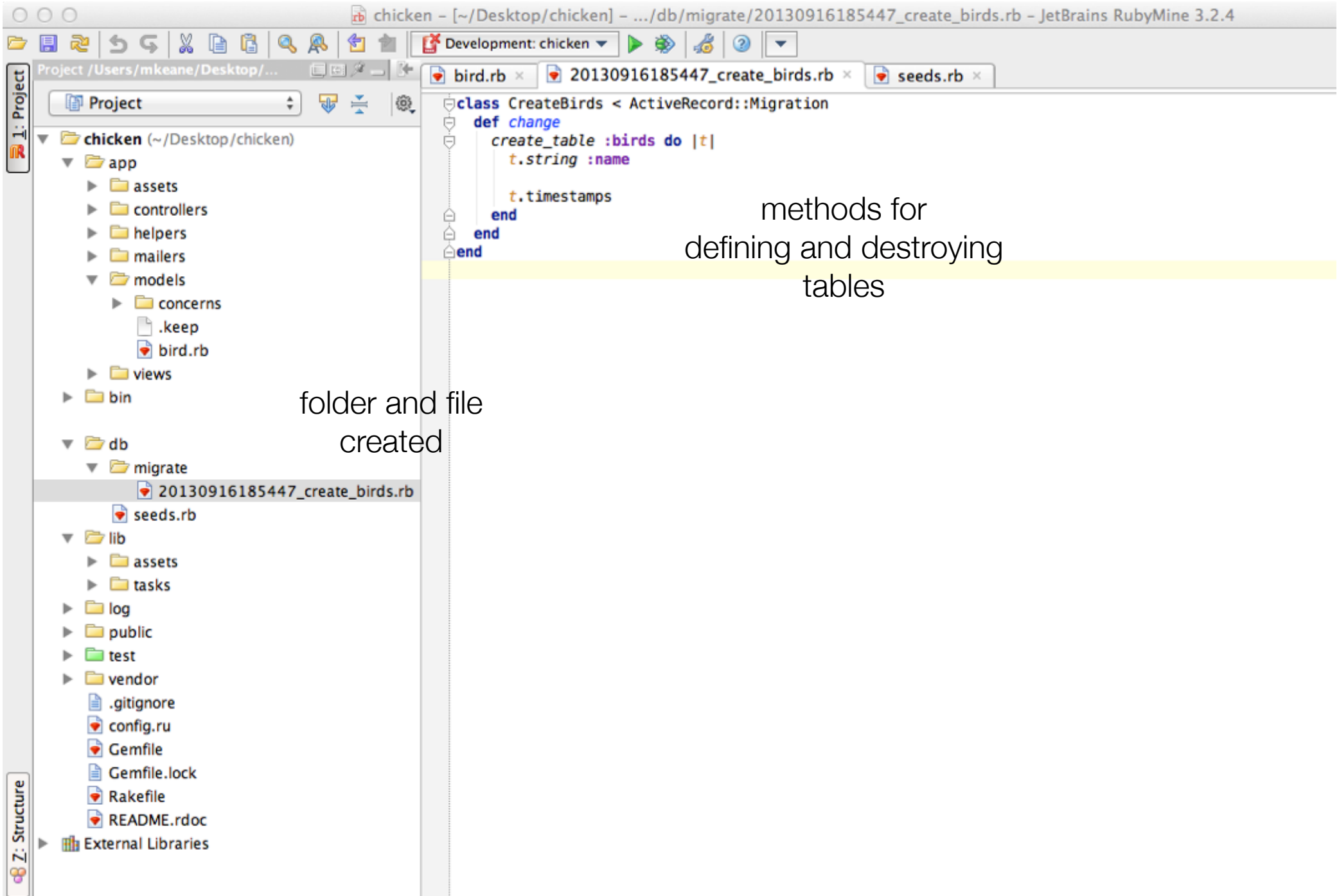
$ rake db:migrate

schema.rb file created

....now imagine that we go into the migration file and alter it...and re-run **rake db:migrate**...

what will **schema.rb** look like now ?

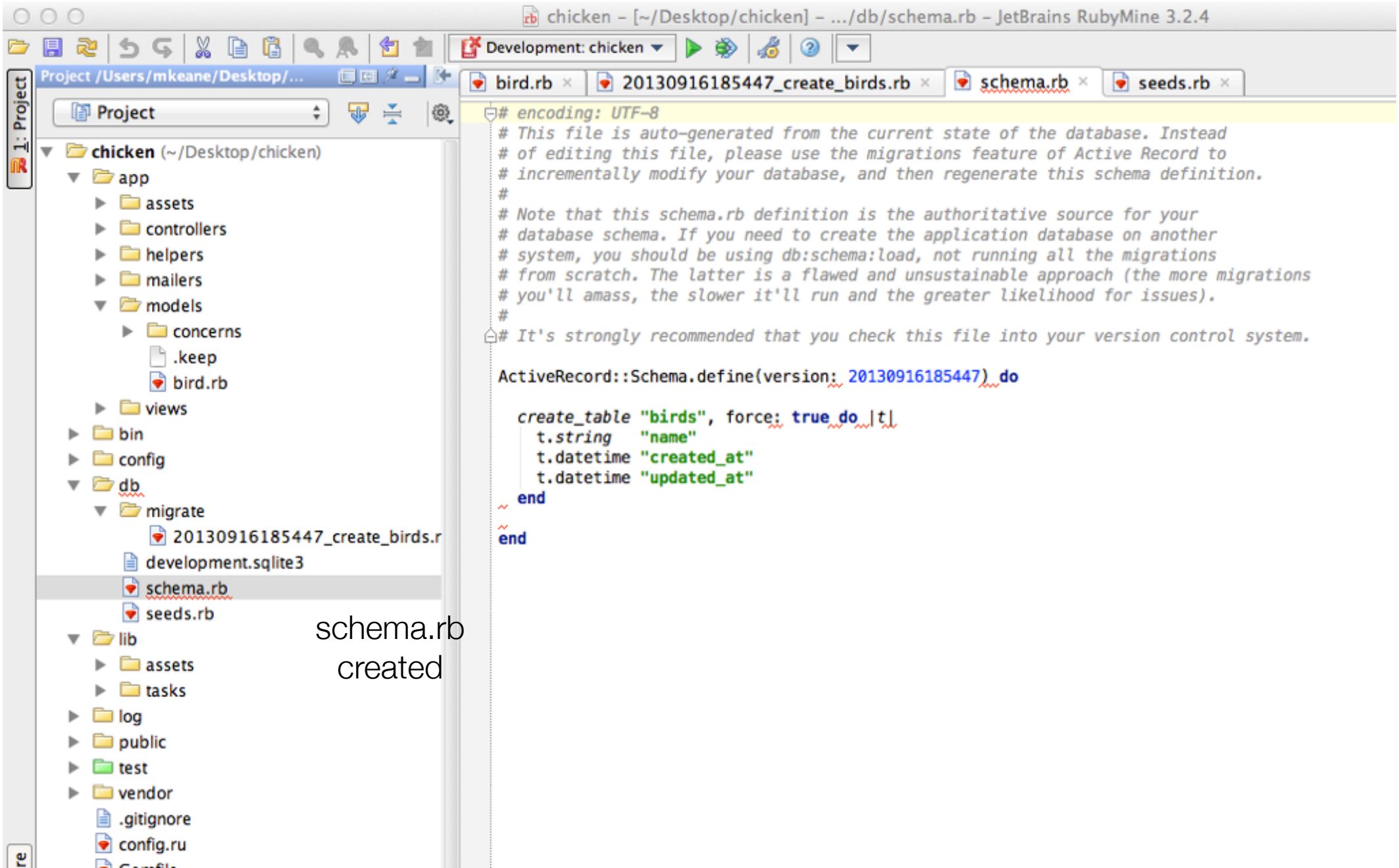
$ rails generate model Bird name:string



folder and file created

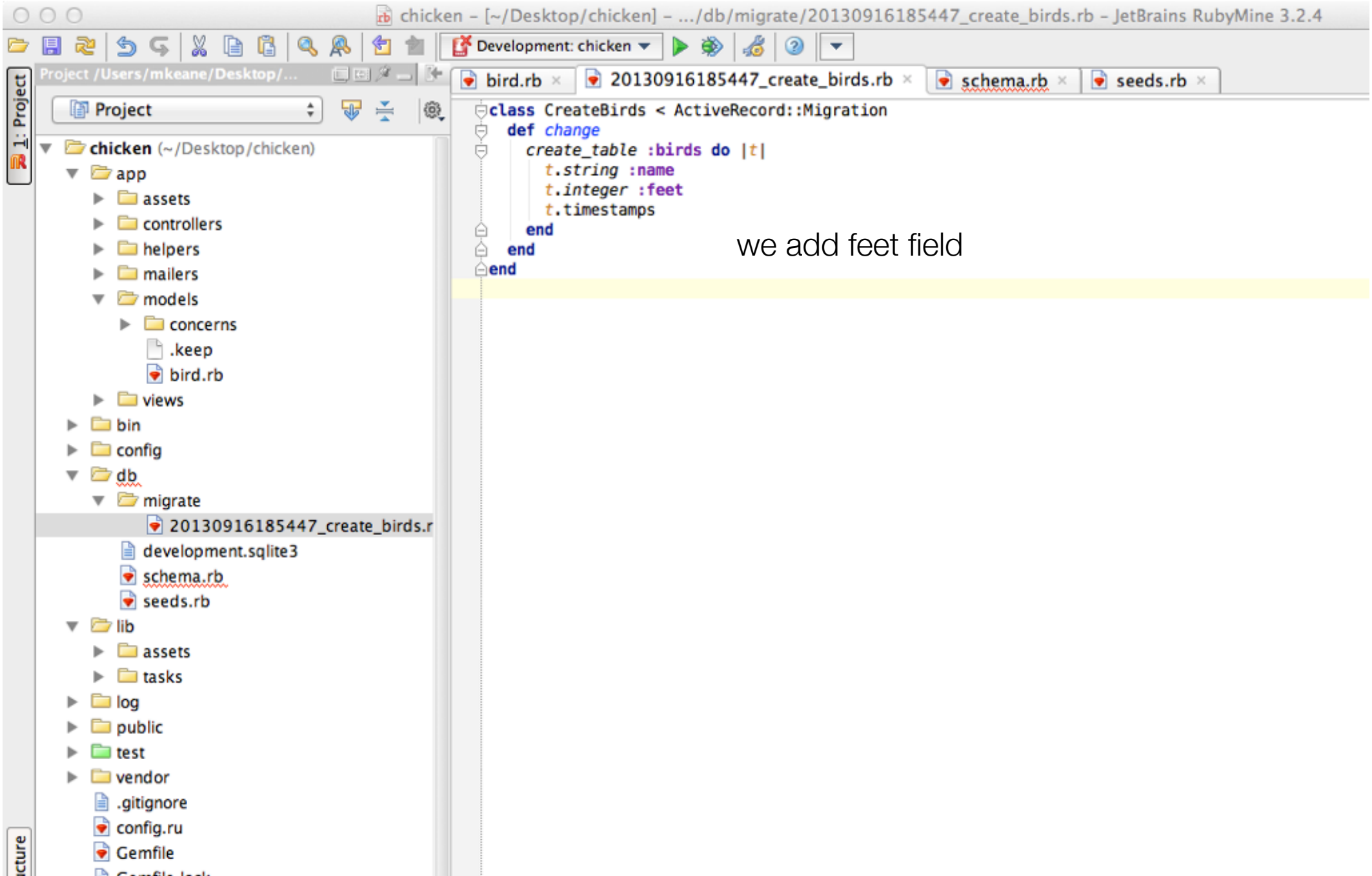
methods for defining and destroying tables

rake db:migrate does...

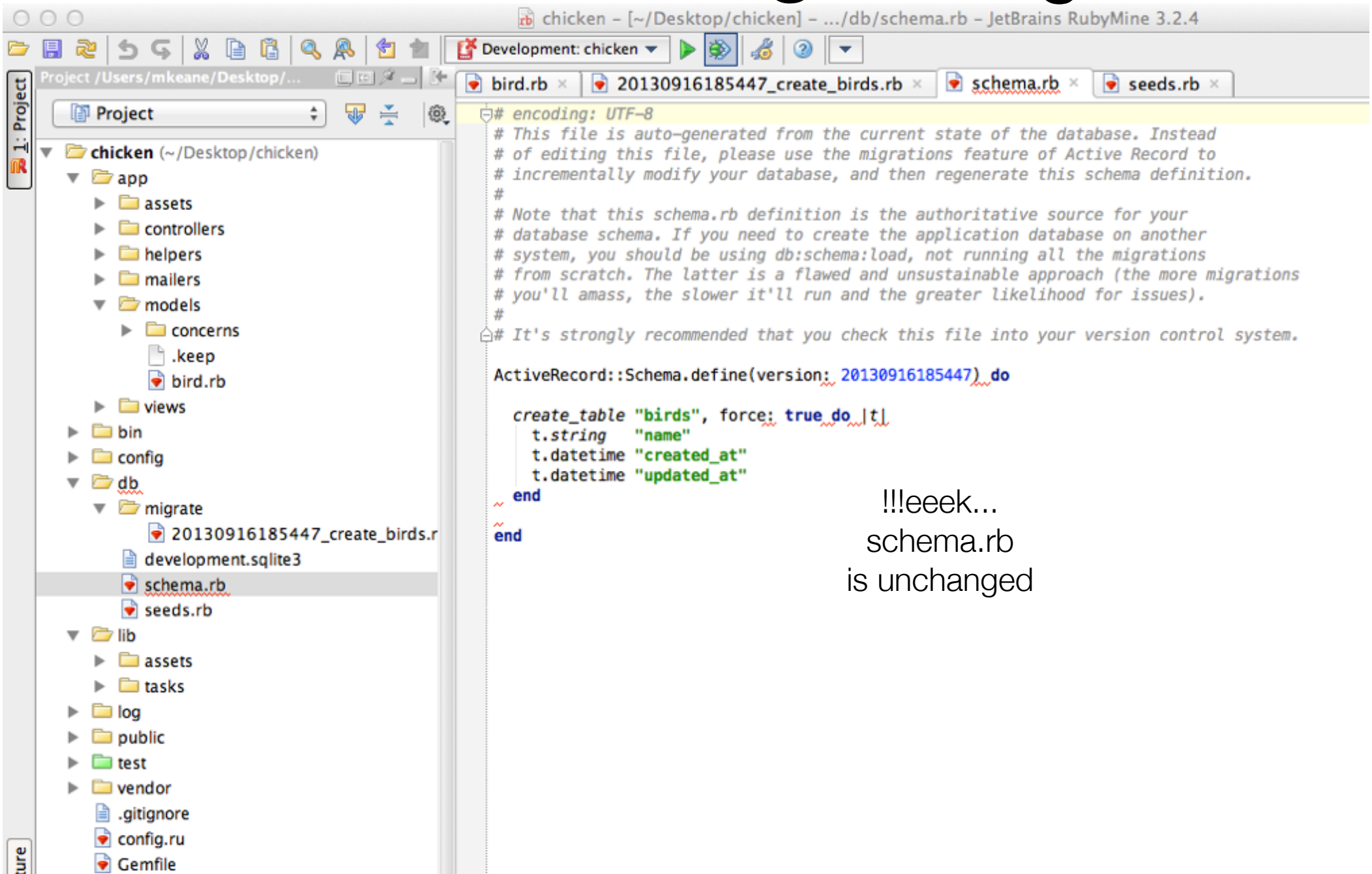


schema.rb created

say, we add feet...



and do **rake db:migrate** again...



!!!eeek...
schema.rb
is unchanged

rake db:migrate

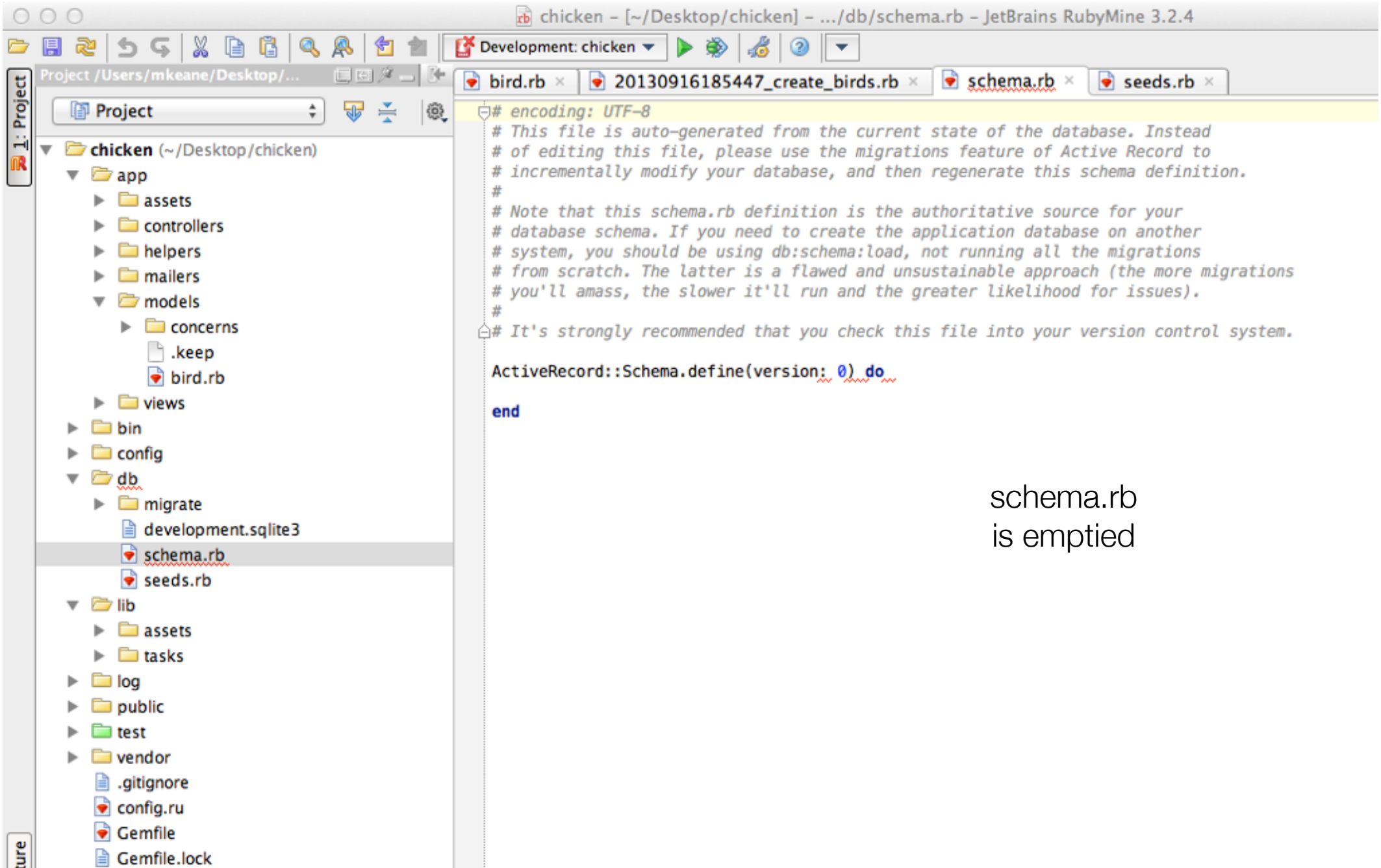
does not re-run the edited file...it just looks to see if there are any explicitly undone migrations

as we did the Bird migration already...it does not pick up the edits...for safety reasons

so, we need to explicitly rollback the db and rebuild the tables to get **schema.rb** updated

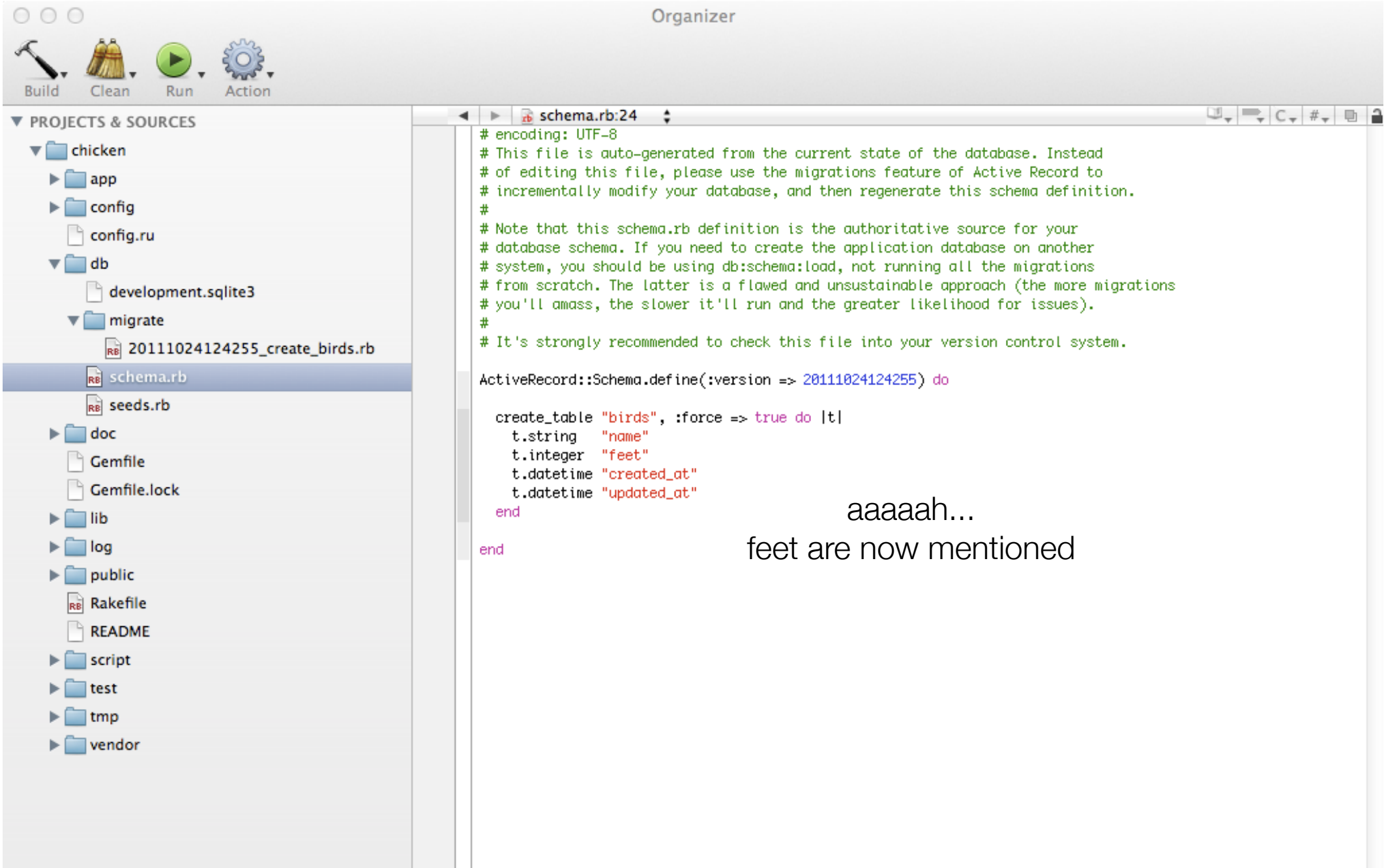
so, we never attempt to change **schema.rb** directly by editing it !

rake db:rollback does this...



schema.rb
is emptied

then redo **rake db:migrate** ...



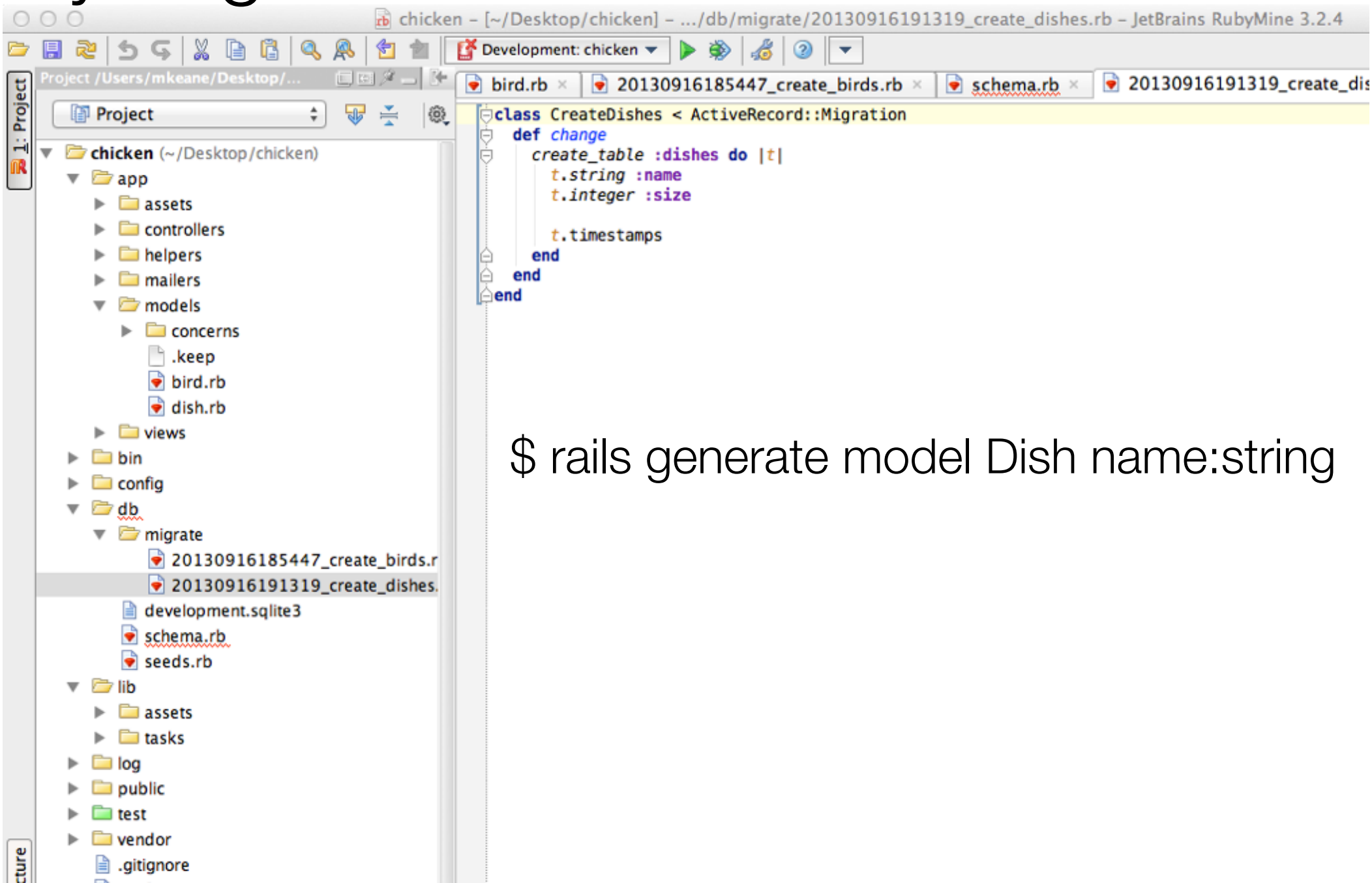
rake db:migrate...again

and that's not the end of it...if you have done several migrations then rollback goes back through them one step at a time

so, in more complex apps with multiple models you need to make sure you have rolled back right and your **schema.rb** file is what you want

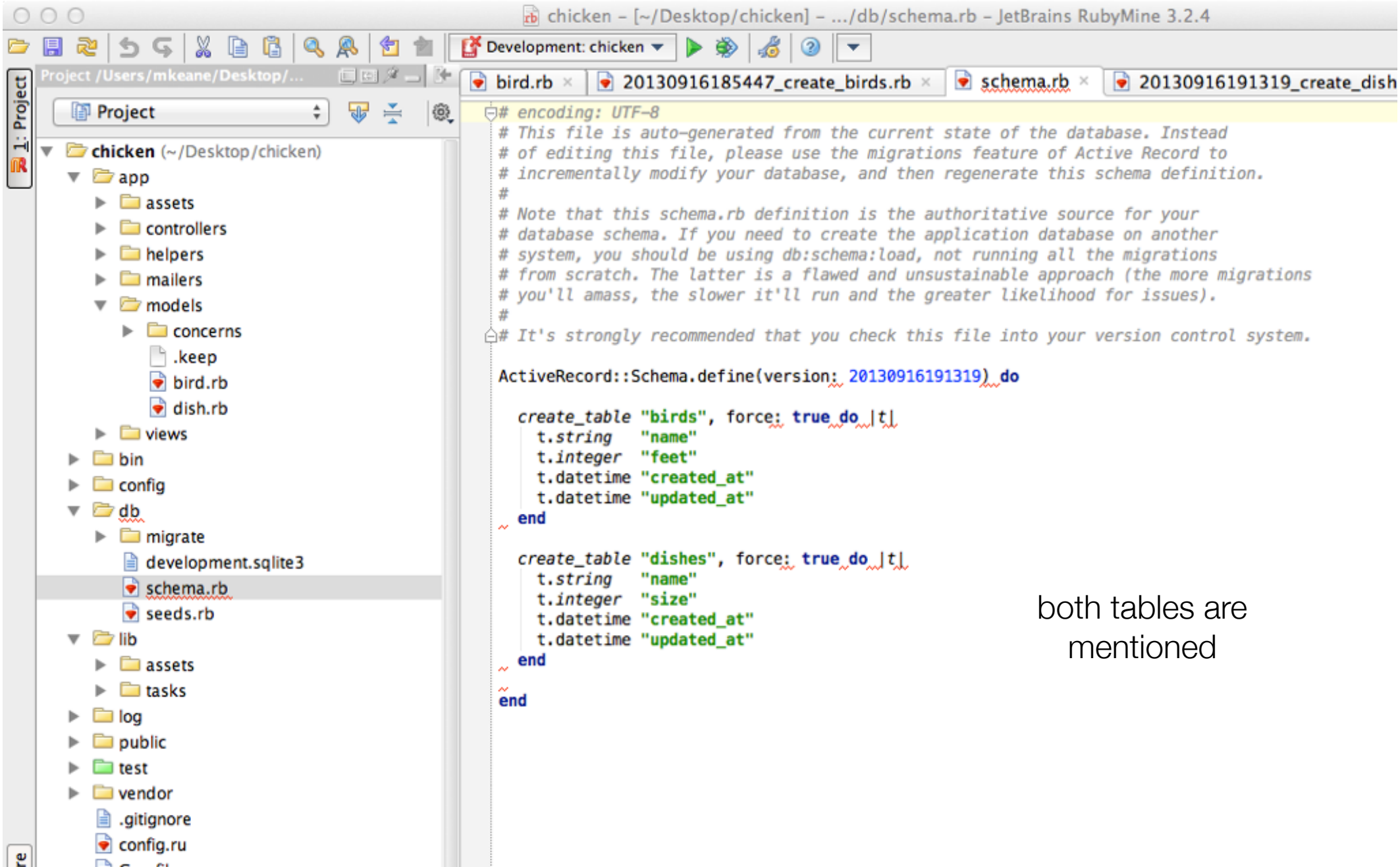
this is only the tip of the iceberg...online discussions suggest that **schema.rb** may be in part build from the current state of the db

if you generate a new model...



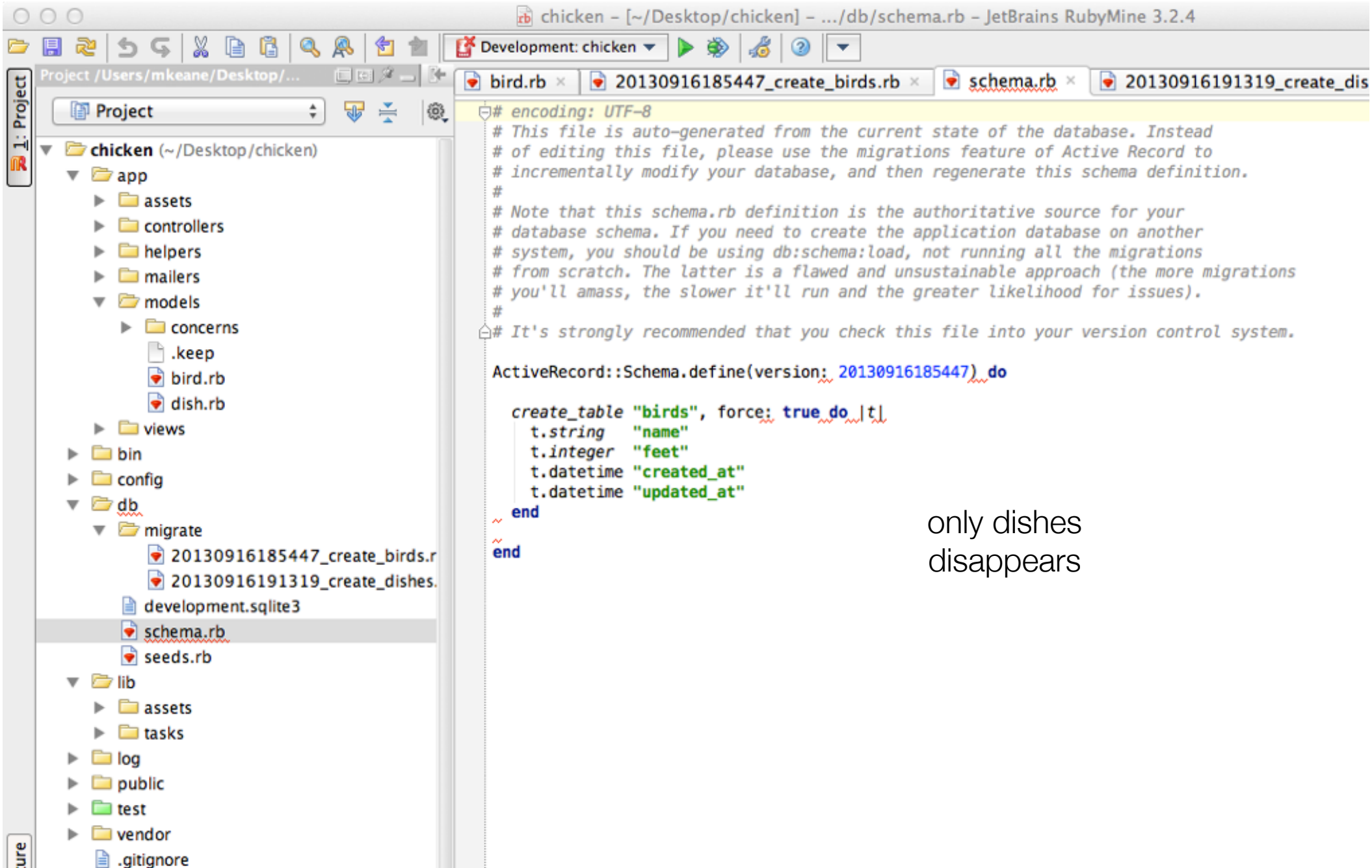
$ rails generate model Dish name:string

then do **rake db:migrate** ...



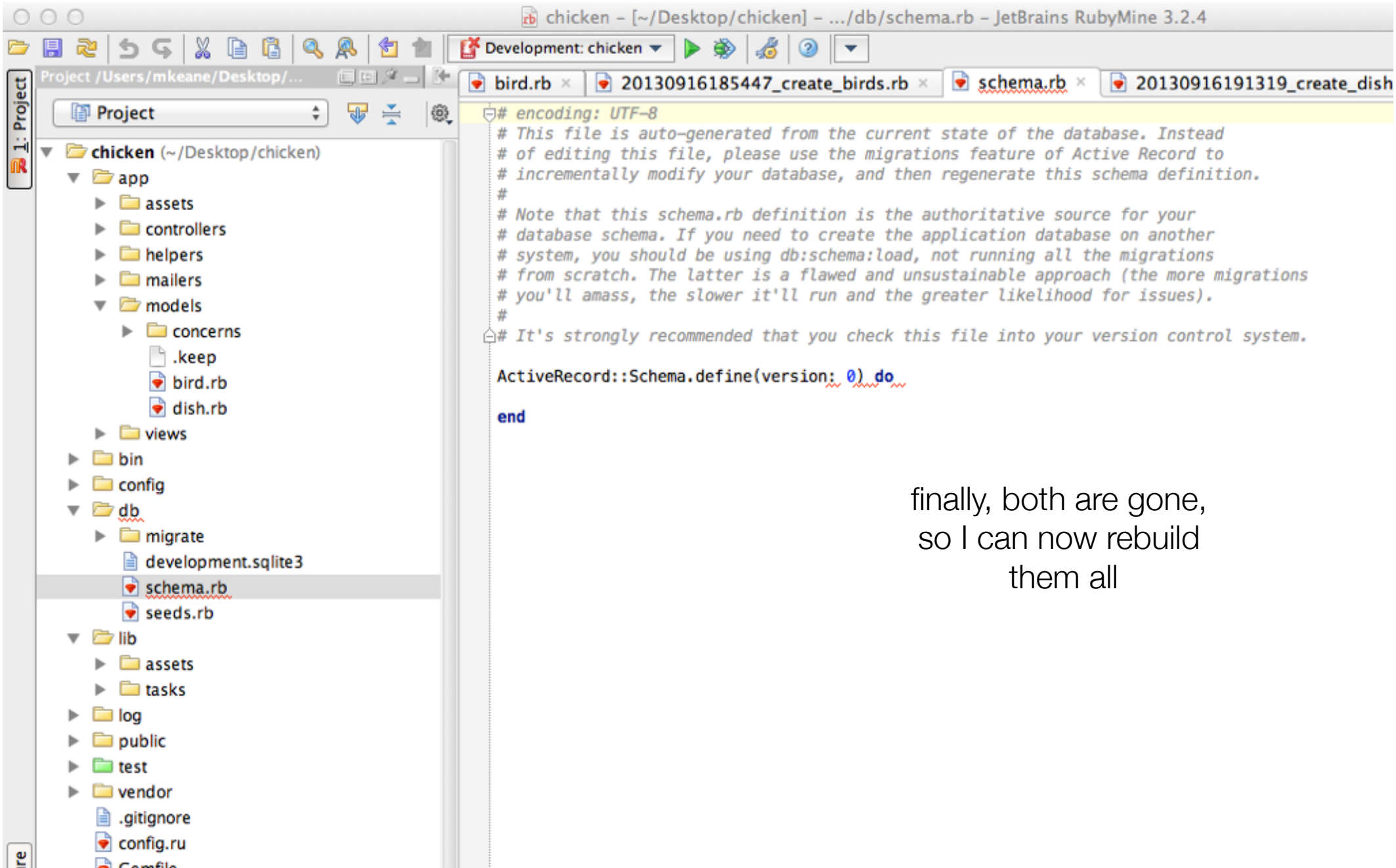
both tables are mentioned

then $ rake db:rollback



only dishes
disappears

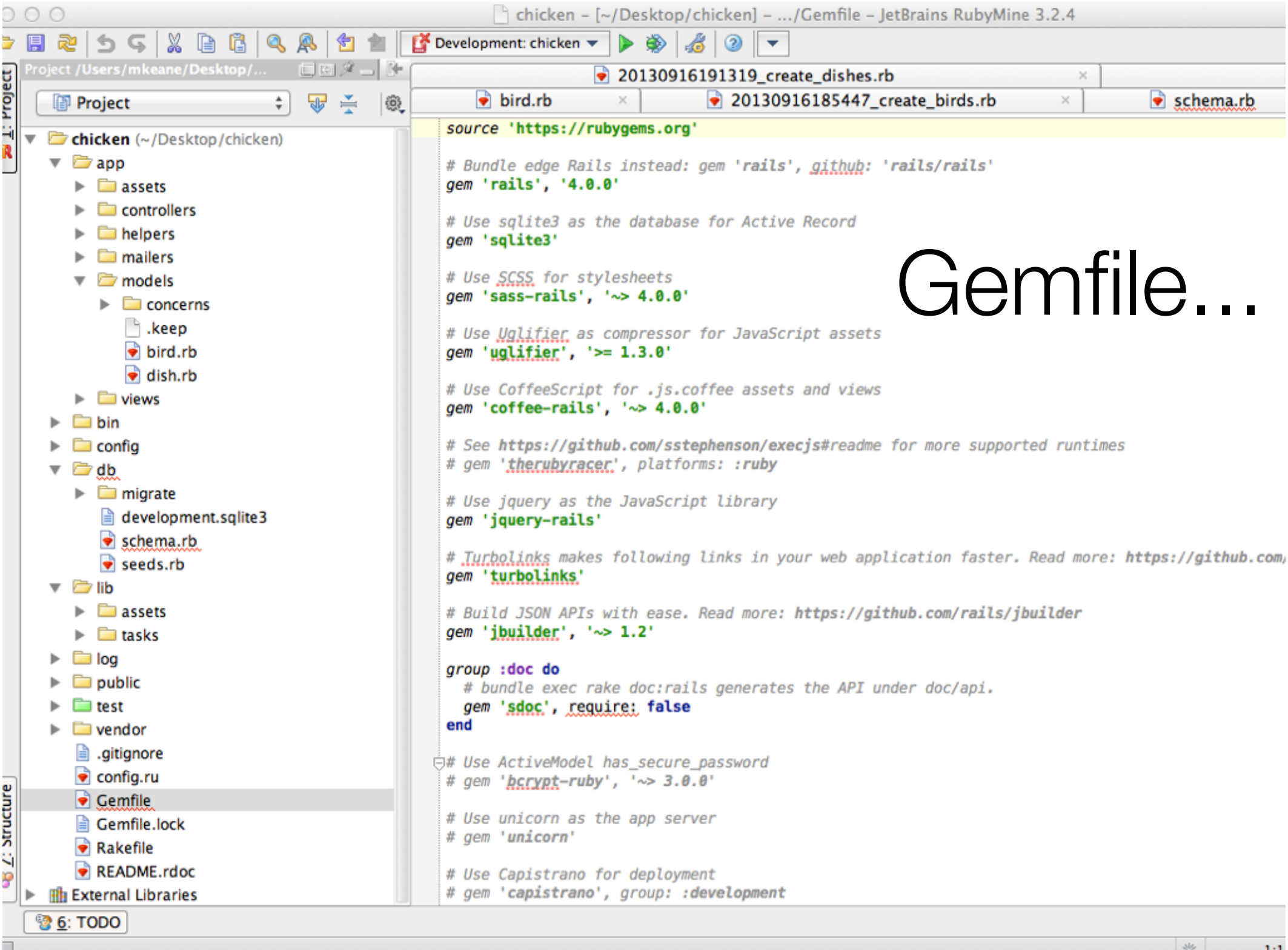
then **rake db:rollback** again

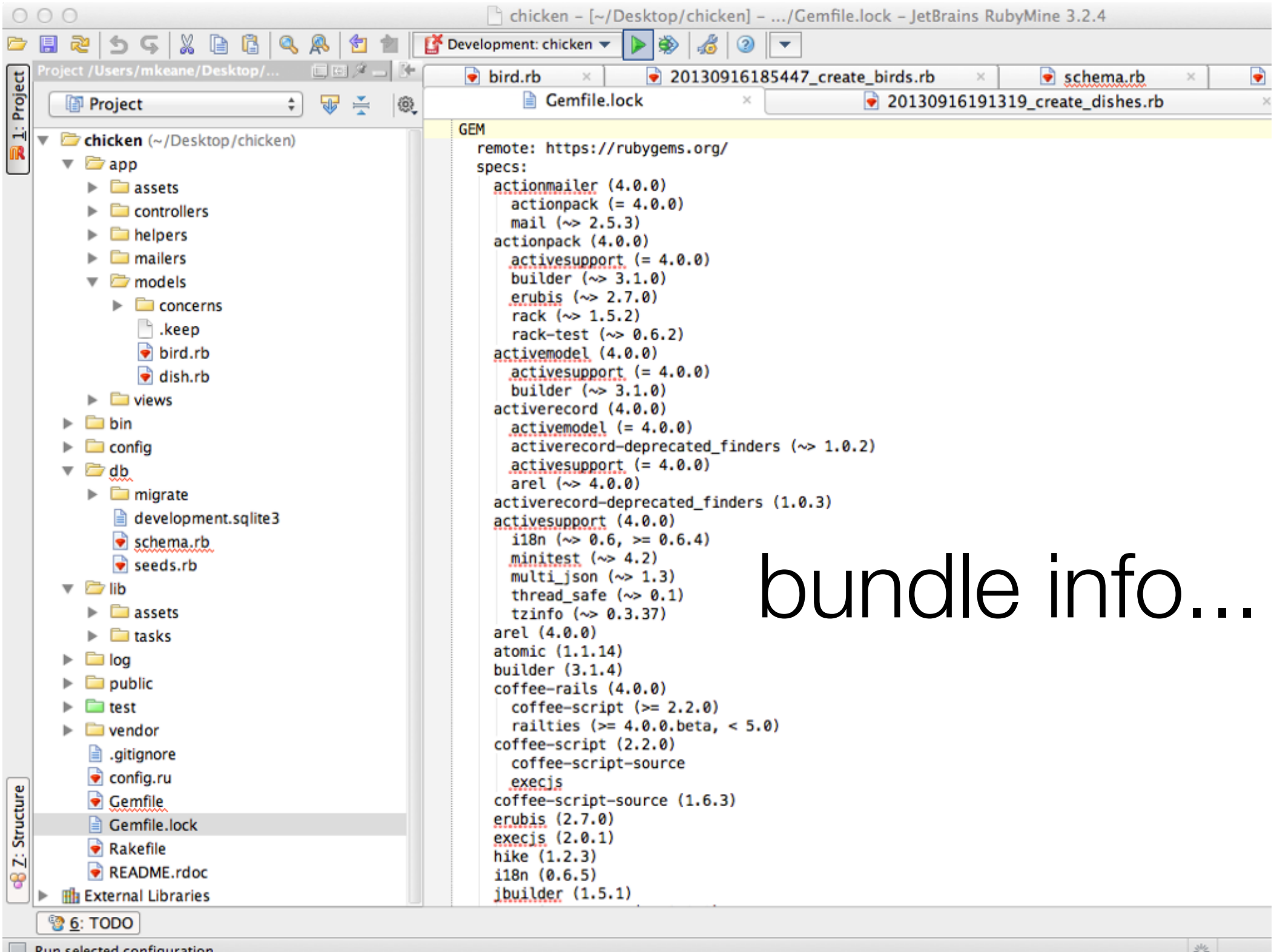


finally, both are gone,
so I can now rebuild
them all

rake has a lot options...

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


chicken – [~/Desktop/chicken] – .../Gemfile – JetBrains RubyMine 3.2.4
Development: chicken
Project /Users/mkeane/Desktop/...
Project
chicken (~/Desktop/chicken)
app
assets
controllers
helpers
mailers
models
concerns
.keep
bird.rb
dish.rb
views
bin
config
db
migrate
development.sqlite3
schema.rb
seeds.rb
lib
assets
tasks
log
public
test
vendor
.gitignore
config.ru
Gemfile
Gemfile.lock
Rakefile
README.rdoc
External Libraries
20130916191319_create_dishes.rb
bird.rb
20130916185447_create_birds.rb
schema.rb
source 'https://rubygems.org'
Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
gem 'rails', '4.0.0'
Use sqlite3 as the database for Active Record
gem 'sqlite3'
Use SCSS for stylesheets
gem 'sass-rails', '~> 4.0.0'
Use Uglifier as compressor for JavaScript assets
gem 'uglifier', '>= 1.3.0'
Use CoffeeScript for .js.coffee assets and views
gem 'coffee-rails', '~> 4.0.0'
See https://github.com/sstephenson/execjs#readme for more supported runtimes
gem 'therubyracer', platforms: :ruby
Use jquery as the JavaScript library
gem 'jquery-rails'
Turbolinks makes following links in your web application faster. Read more: https://github.com/
gem 'turbolinks'
Build JSON APIs with ease. Read more: https://github.com/rails/jbuilder
gem 'jbuilder', '~> 1.2'
group :doc do
bundle exec rake doc:rails generates the API under doc/api.
gem 'sdoc', require: false
end
Use ActiveModel has_secure_password
gem 'bcrypt-ruby', '~> 3.0.0'
Use unicorn as the app server
gem 'unicorn'
Use Capistrano for deployment
gem 'capistrano', group: :development
6: TODO
Gemfile...


chicken – [~/Desktop/chicken] – .../Gemfile.lock – JetBrains RubyMine 3.2.4
Development: chicken
Project /Users/mkeane/Desktop/...
Project
chicken (~/Desktop/chicken)
app
assets
controllers
helpers
mailers
models
concerns
.keep
bird.rb
dish.rb
views
bin
config
db
migrate
development.sqlite3
schema.rb
seeds.rb
lib
assets
tasks
log
public
test
vendor
.gitignore
config.ru
Gemfile
Gemfile.lock
Rakefile
README.rdoc
External Libraries
bird.rb
20130916185447_create_birds.rb
schema.rb
Gemfile.lock
20130916191319_create_dishes.rb
GEM
remote: https://rubygems.org/
specs:
actionmailer (4.0.0)
actionpack (= 4.0.0)
mail (~> 2.5.3)
actionpack (4.0.0)
activesupport (= 4.0.0)
builder (~> 3.1.0)
erubis (~> 2.7.0)
rack (~> 1.5.2)
rack-test (~> 0.6.2)
activemodel (4.0.0)
activesupport (= 4.0.0)
builder (~> 3.1.0)
activerecord (4.0.0)
activemodel (= 4.0.0)
activerecord-deprecated_finders (~> 1.0.2)
activesupport (= 4.0.0)
arel (~> 4.0.0)
activerecord-deprecated_finders (1.0.3)
activesupport (4.0.0)
i18n (~> 0.6, >= 0.6.4)
minitest (~> 4.2)
multi_json (~> 1.3)
thread_safe (~> 0.1)
tzinfo (~> 0.3.37)
arel (4.0.0)
atomic (1.1.14)
builder (3.1.4)
coffee-rails (4.0.0)
coffee-script (>= 2.2.0)
railties (>= 4.0.0.beta, < 5.0)
coffee-script (2.2.0)
coffee-script-source
execjs
coffee-script-source (1.6.3)
erubis (2.7.0)
execjs (2.0.1)
hike (1.2.3)
i18n (0.6.5)
jbuilder (1.5.1)
1: Project
7: Structure
6: TODO
bundle info...

Some Useful Links...

http://rails-nutshell.labs.oreilly.com/ch03.html

http://guides.rubyonrails.org/active_record_querying.html

http://stackoverflow.com/questions/1007187/exactly-what-does-rake-dbmigrate-do

http://railstutorial.org/book#sec:the_first_application

http://api.rubyonrails.org/classes/ActiveRecord/Migration.html

Part B

itunes in Rails...the long, dark tea-time of the soul

Part B(i):

iTunes Rails Design...

General Design

Models: DB tables for songs and actors (more plausible to create albums on the fly)

Views: there will be the usual views for each action/screen; could also have one for different users (like Buyer, Administrator)

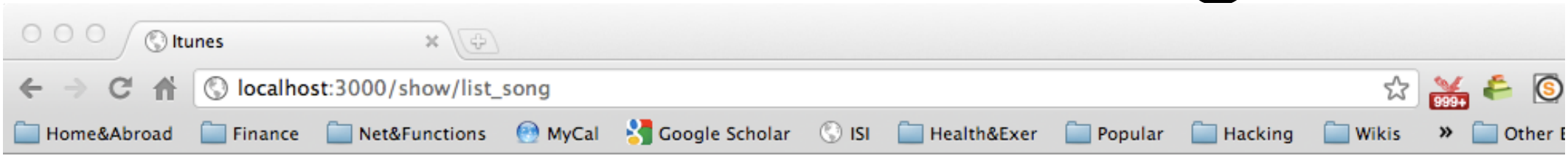
Controllers: methods to show songs/albums/ actors, buy, add songs...

Typically, every screen has a method-view pair and a model too...

Part B(ii):

Some Screens...its not pretty...but hey it works

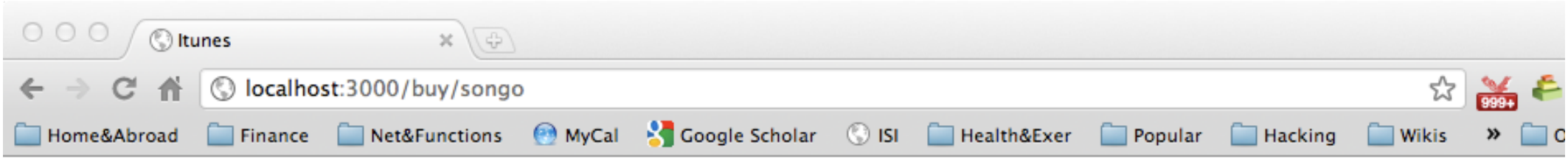
iTunes: **show/list_song**



Song Listing

- Stairway to Heaven by Led Zep, in Led Zep IV : 1 [Owned by Actor ID 1]
- Holiday by Sex Pistols, in God Save... : 2 [Owned by Actor ID 2]

iTunes: **buy/songo**



Buy a Song

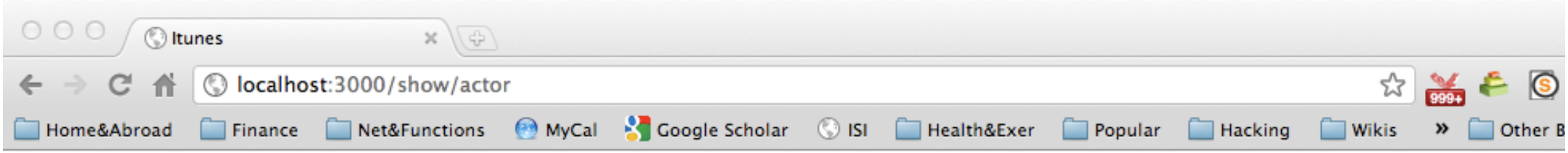
Give me your name:

Mark Keabe

Song you want to buy:

Holiday buy

iTunes: **show/actor**

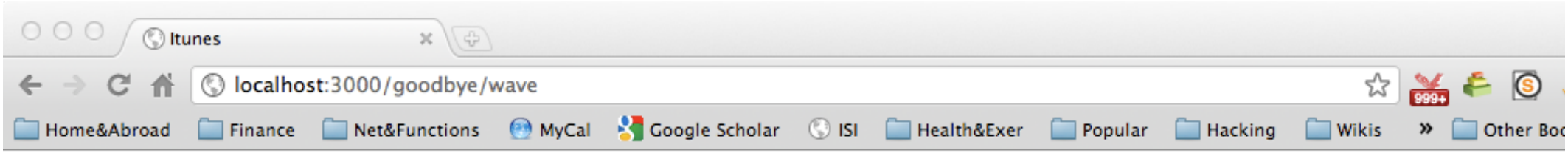


Actor Listing

- Mark the Admin, likes reggae : 1
- Bloggie the Buyer, likes punk : 2

Say Goodbye, Mark

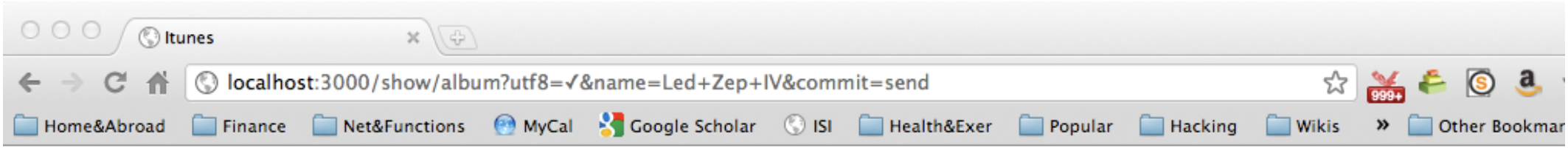
iTunes: **goodbye/wave**



Goodbye#actor_wave

bye bye mark

iTunes: **show/album**



Album Listing

Search for an Album named:

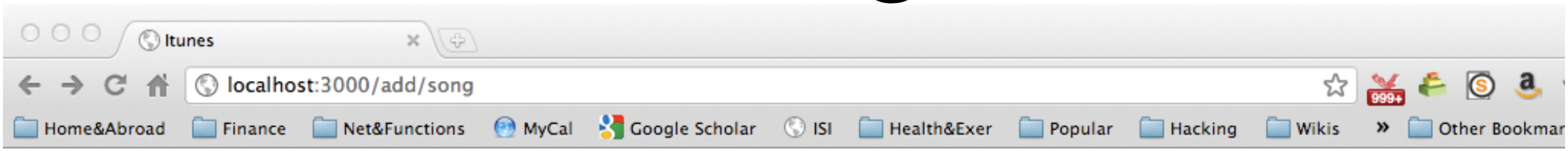
send

The Album "Led Zep IV" has the following tracks:

- Stairway to Heaven by Led Zep

The Album is 4 minutes long.

iTunes: **add/song**



Add a Song to the DB

Please enter Details Below:

Song Name: Gloria

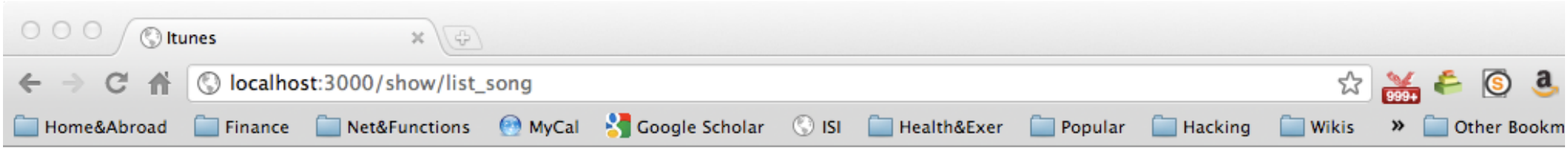
Artist: Patti Smith

Length of Track: 3

Album its in: Horses send

Please review your inputs

iTunes: **show/list_song**



Song Listing

- Stairway to Heaven by Led Zep, in Led Zep IV : 1 [Owned by Actor ID 1]
- Holiday by Sex Pistols, in God Save... : 2 [Owned by Actor ID 2]
- Gloria by Patti Smith, in Horses : 3 [Owned by Actor ID]

Part B(iii):

Walk through of iTunes Rails

Setup: Controllers & Models

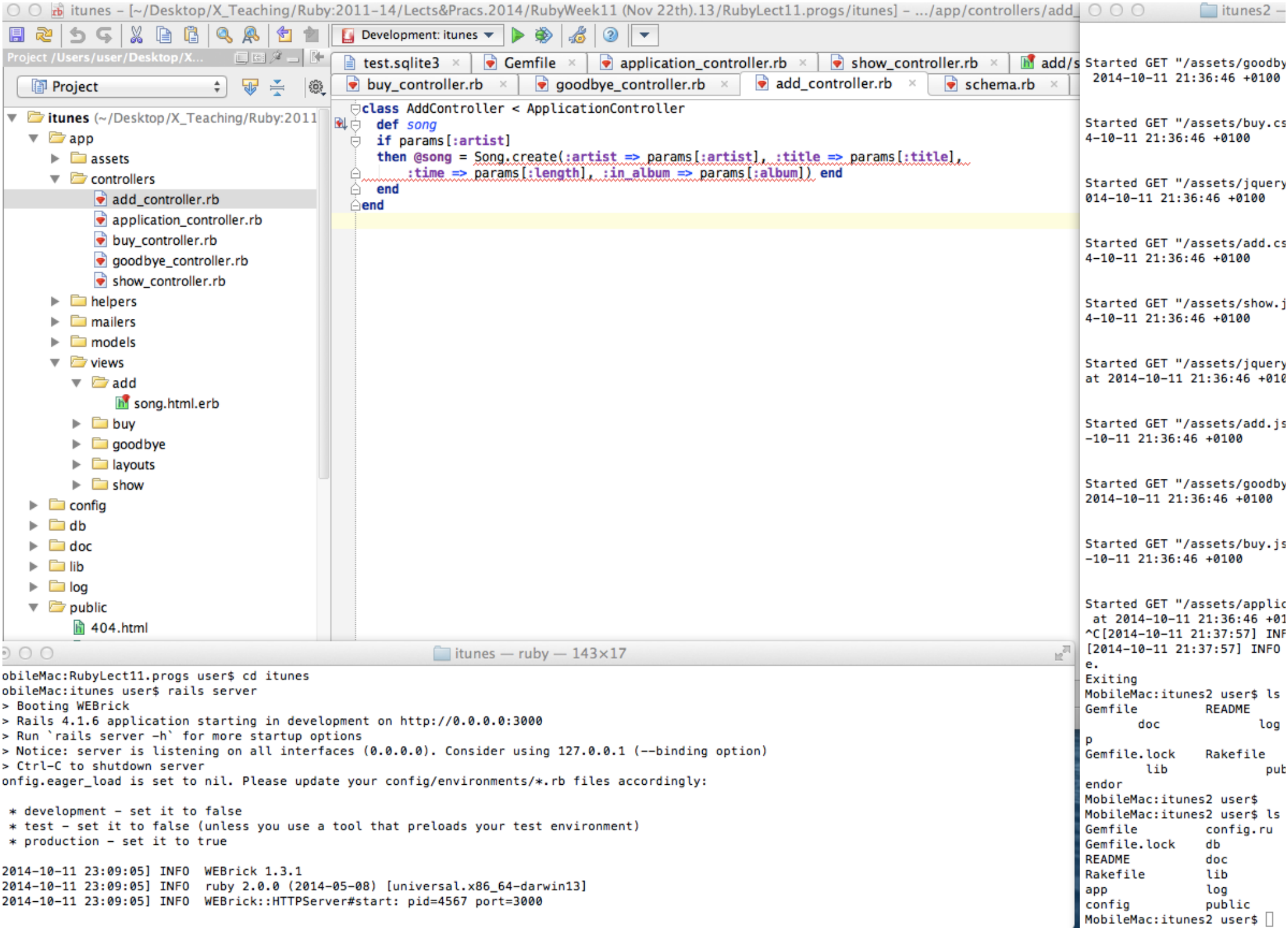
```
$ rails new itunes
$ cd itunes
$ rails generate controller show list_song album actor
$ rails generate controller buy songo
$ rails generate controller add song
$ rails generate model Actor name:string likes:string
$ rails generate model Song
$ rails server
```

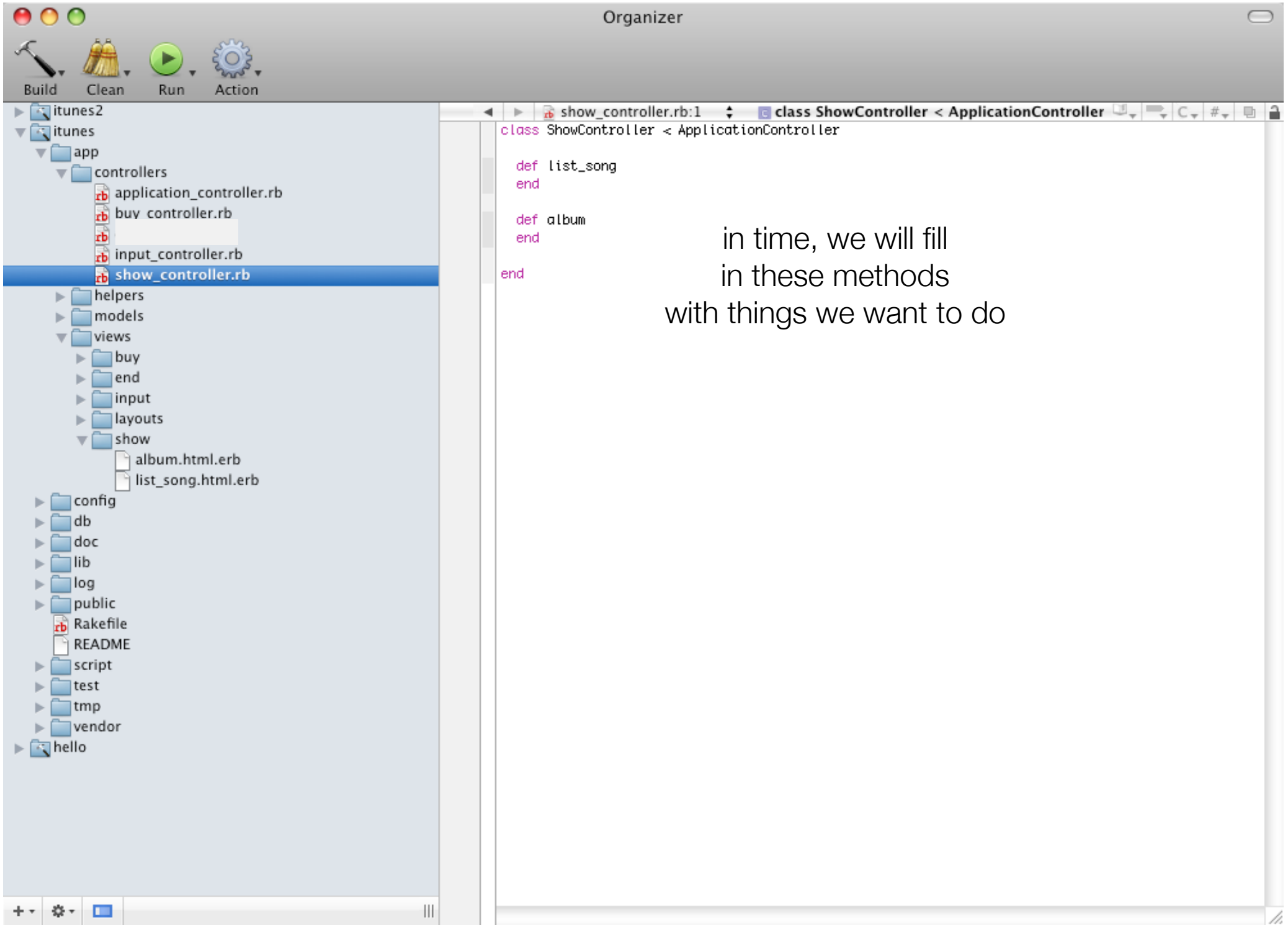
...if you make a mistake you can do...

```
$ rails destroy controller buy
```

Setup: Model...

```
$ rails generate model Song
      invoke  active_record
      create    db/migrate/20101115115939_create_songs.rb
      create    app/models/song.rb
      invoke    test_unit
      create      test/unit/song_test.rb
      create      test/fixtures/songs_yml
```

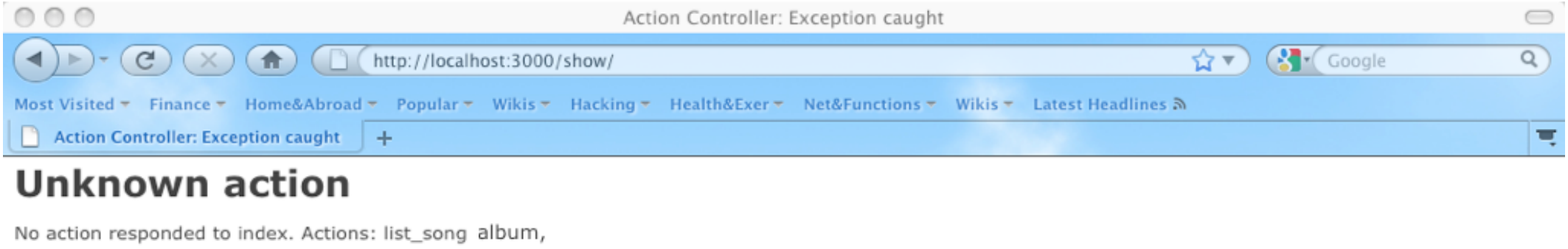

itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../app/controllers/add_
Development: itunes
Project /Users/user/Desktop/X...
Project
itunes (~/Desktop/X_Teaching/Ruby:2011
app
assets
controllers
add_controller.rb
application_controller.rb
buy_controller.rb
goodbye_controller.rb
show_controller.rb
helpers
mailers
models
views
add
song.html.erb
buy
goodbye
layouts
show
config
db
doc
lib
log
public
404.html
test.sqlite3
Gemfile
application_controller.rb
show_controller.rb
buy_controller.rb
goodbye_controller.rb
add_controller.rb
schema.rb
class AddController < ApplicationController
def song
if params[:artist]
then @song = Song.create(:artist => params[:artist], :title => params[:title],
:time => params[:length], :in_album => params[:album]) end
end
end
itunes — ruby — 143×17
obileMac:RubyLect11.progs user$ cd itunes
obileMac:itunes user$ rails server
> Booting WEBrick
> Rails 4.1.6 application starting in development on http://0.0.0.0:3000
> Run `rails server -h` for more startup options
> Notice: server is listening on all interfaces (0.0.0.0). Consider using 127.0.0.1 (--binding option)
> Ctrl-C to shutdown server
onfig.eager_load is set to nil. Please update your config/environments/*.rb files accordingly:
* development - set it to false
* test - set it to false (unless you use a tool that preloads your test environment)
* production - set it to true
2014-10-11 23:09:05] INFO WEBrick 1.3.1
2014-10-11 23:09:05] INFO ruby 2.0.0 (2014-05-08) [universal.x86_64-darwin13]
2014-10-11 23:09:05] INFO WEBrick::HTTPServer#start: pid=4567 port=3000
itunes2 –
Started GET "/assets/goodby
2014-10-11 21:36:46 +0100
Started GET "/assets/buy.cs
4-10-11 21:36:46 +0100
Started GET "/assets/jquery
014-10-11 21:36:46 +0100
Started GET "/assets/add.cs
4-10-11 21:36:46 +0100
Started GET "/assets/show.j
4-10-11 21:36:46 +0100
Started GET "/assets/jquery
at 2014-10-11 21:36:46 +010
Started GET "/assets/add.js
-10-11 21:36:46 +0100
Started GET "/assets/goodby
2014-10-11 21:36:46 +0100
Started GET "/assets/buy.js
-10-11 21:36:46 +0100
Started GET "/assets/applic
at 2014-10-11 21:36:46 +01
^C[2014-10-11 21:37:57] INF
[2014-10-11 21:37:57] INFO
e.
Exiting
MobileMac:itunes2 user$ ls
Gemfile README
doc log
p
Gemfile.lock Rakefile
lib pub
endor
MobileMac:itunes2 user$
MobileMac:itunes2 user$ ls
Gemfile config.ru
Gemfile.lock db
README doc
Rakefile lib
app log
config public
MobileMac:itunes2 user$


Organizer
Build
Clean
Run
Action
itunes2
itunes
app
controllers
application_controller.rb
buy_controller.rb
input_controller.rb
show_controller.rb
helpers
models
views
buy
end
input
layouts
show
album.html.erb
list_song.html.erb
config
db
doc
lib
log
public
Rakefile
README
script
test
tmp
vendor
hello
show_controller.rb:1
class ShowController < ApplicationController
class ShowController < ApplicationController
def list_song
end
def album
end
end
in time, we will fill
in these methods
with things we want to do

Load Page

nb: will not load if
browser is in offline mode;
even tho' local files

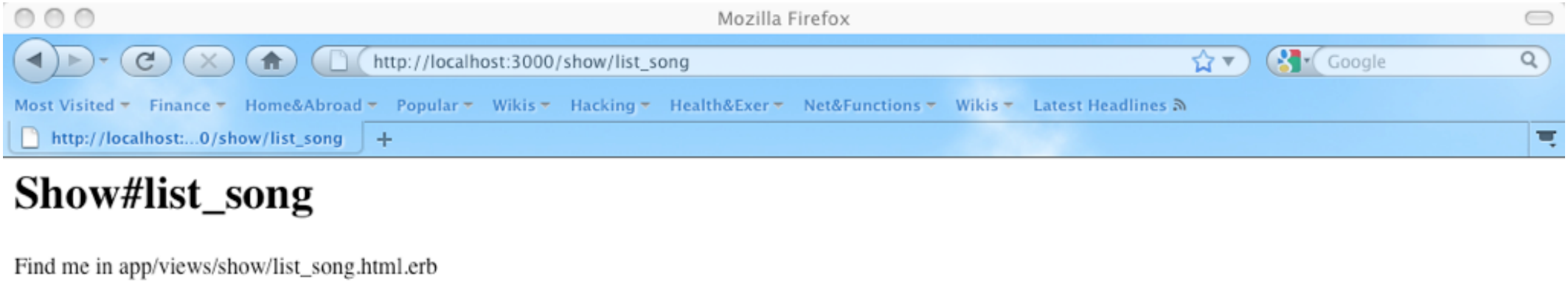
load page in browser:
http://localhost:3000/show

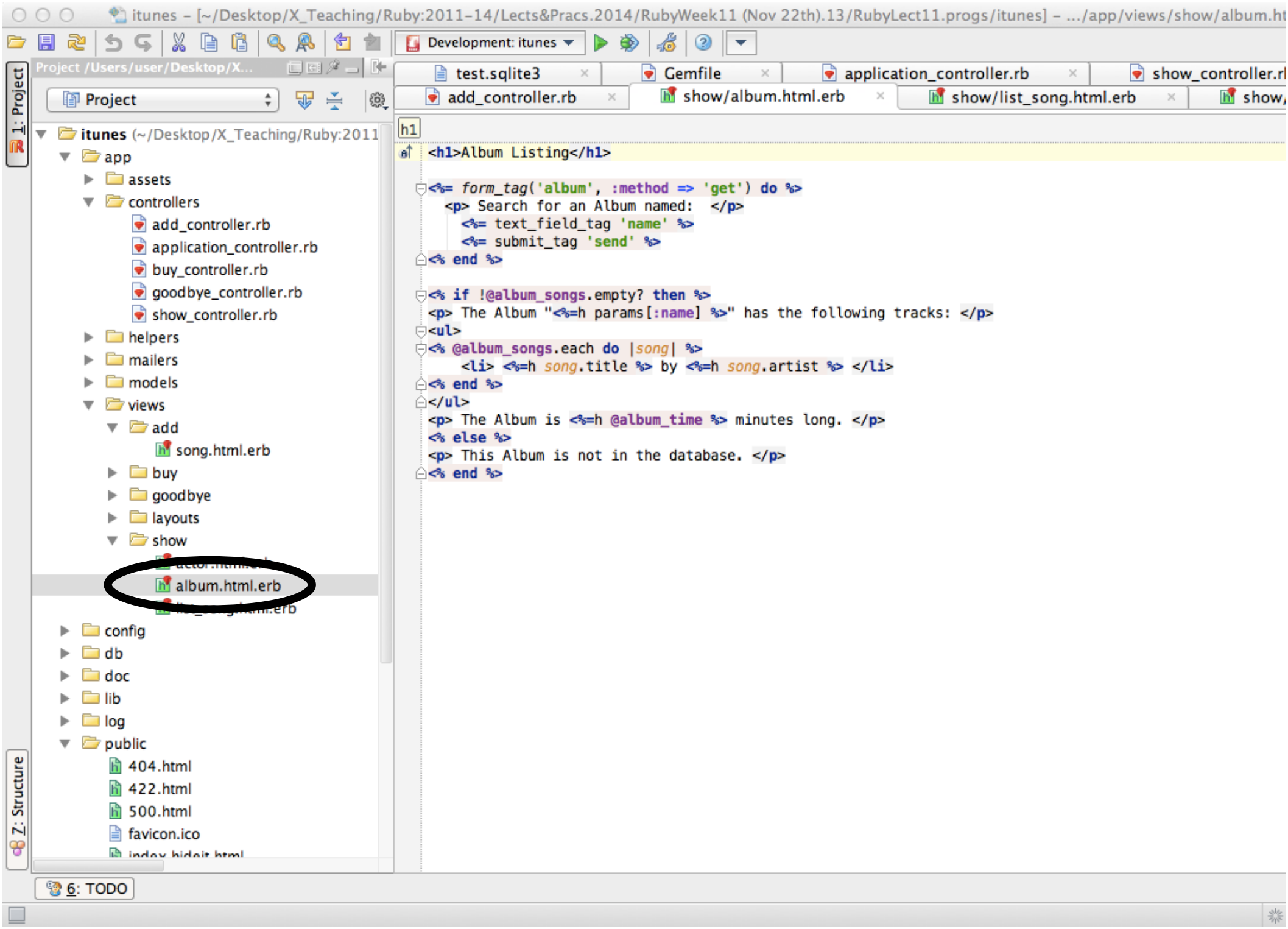


have only given controller
but not method name

Load Page

load page in browser:
http://localhost:3000/show/list_song




itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../app/views/show/album.ht
Development: itunes
Project /Users/user/Desktop/X...
Project
itunes (~/Desktop/X_Teaching/Ruby:2011
app
assets
controllers
add_controller.rb
application_controller.rb
buy_controller.rb
goodbye_controller.rb
show_controller.rb
helpers
mailers
models
views
add
song.html.erb
buy
goodbye
layouts
show
album.html.erb
config
db
doc
lib
log
public
404.html
422.html
500.html
favicon.ico
test.sqlite3
Gemfile
application_controller.rb
add_controller.rb
show/album.html.erb
show/list_song.html.erb
h1
<h1>Album Listing</h1>
<%= form_tag('album', :method => 'get') do %>
 Search for an Album named:
<%= text_field_tag 'name' %>
<%= submit_tag 'send' %>
<% end %>
<% if !@album_songs.empty? then %>
 The Album "<%=h params[:name] %>" has the following tracks:

<% @album_songs.each do |song| %>
 <%=h song.title %> by <%=h song.artist %>
<% end %>

 The Album is <%=h @album_time %> minutes long.
<% else %>
 This Album is not in the database.
<% end %>
1: Project
7: Structure
6: TODO

Part B(iv):

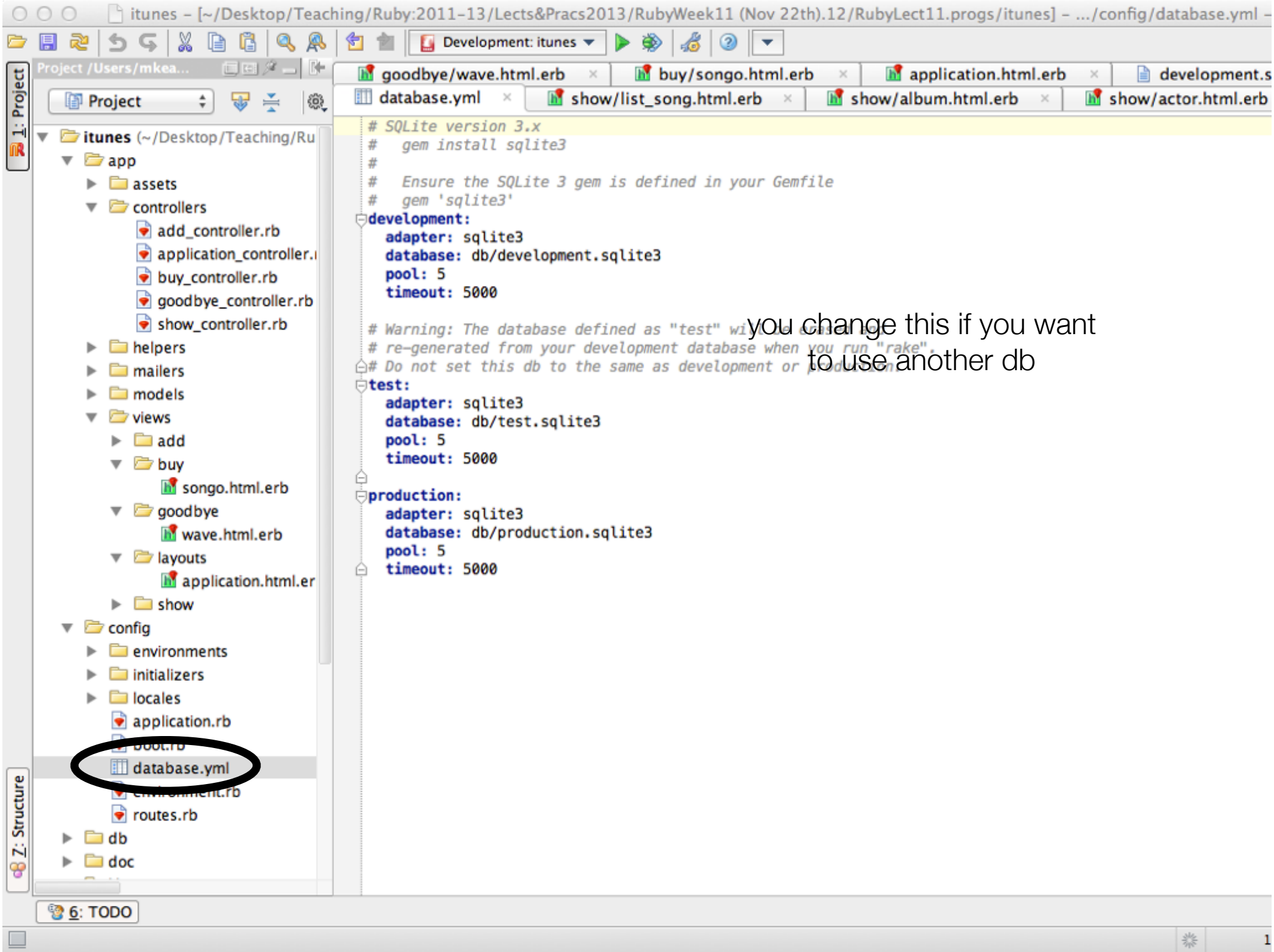
Setting up tables, models and starting the db

What's happening, mammy?

Within Rails' bootage ActiveRecord connects to a database based on what is in **itunes/config/database.yml**

When you have defined your migrations/models (relations), you use **rake db:migrate** to create your **schema.rb** file (but wait til end)

rake db:seed can also load in a set of seed records from **seeds.rb**


itunes – [~/Desktop/Teaching/Ruby:2011-13/Lects&Pracs2013/RubyWeek11 (Nov 22th).12/RubyLect11.progs/itunes] – .../config/database.yml –
Development: itunes
Project /Users/mkea...
Project
goodbye/wave.html.erb
buy/songo.html.erb
application.html.erb
development.s
database.yml
show/list_song.html.erb
show/album.html.erb
show/actor.html.erb
itunes (~/Desktop/Teaching/Ru
app
assets
controllers
add_controller.rb
application_controller.
buy_controller.rb
goodbye_controller.rb
show_controller.rb
helpers
mailers
models
views
add
buy
songo.html.erb
goodbye
wave.html.erb
layouts
application.html.er
show
config
environments
initializers
locales
application.rb
database.yml
routes.rb
db
doc
SQLite version 3.x
gem install sqlite3
#
Ensure the SQLite 3 gem is defined in your Gemfile
gem 'sqlite3'
development:
adapter: sqlite3
database: db/development.sqlite3
pool: 5
timeout: 5000
Warning: The database defined as "test" will be erased and
re-generated from your development database when you run "rake".
Do not set this db to the same as development or production.
test:
adapter: sqlite3
database: db/test.sqlite3
pool: 5
timeout: 5000
production:
adapter: sqlite3
database: db/production.sqlite3
pool: 5
timeout: 5000
you change this if you want
to use another db
1: Project
7: Structure
6: TODO

Setup tables etc...

We need to set up a bunch of migrations to define our tables for the db

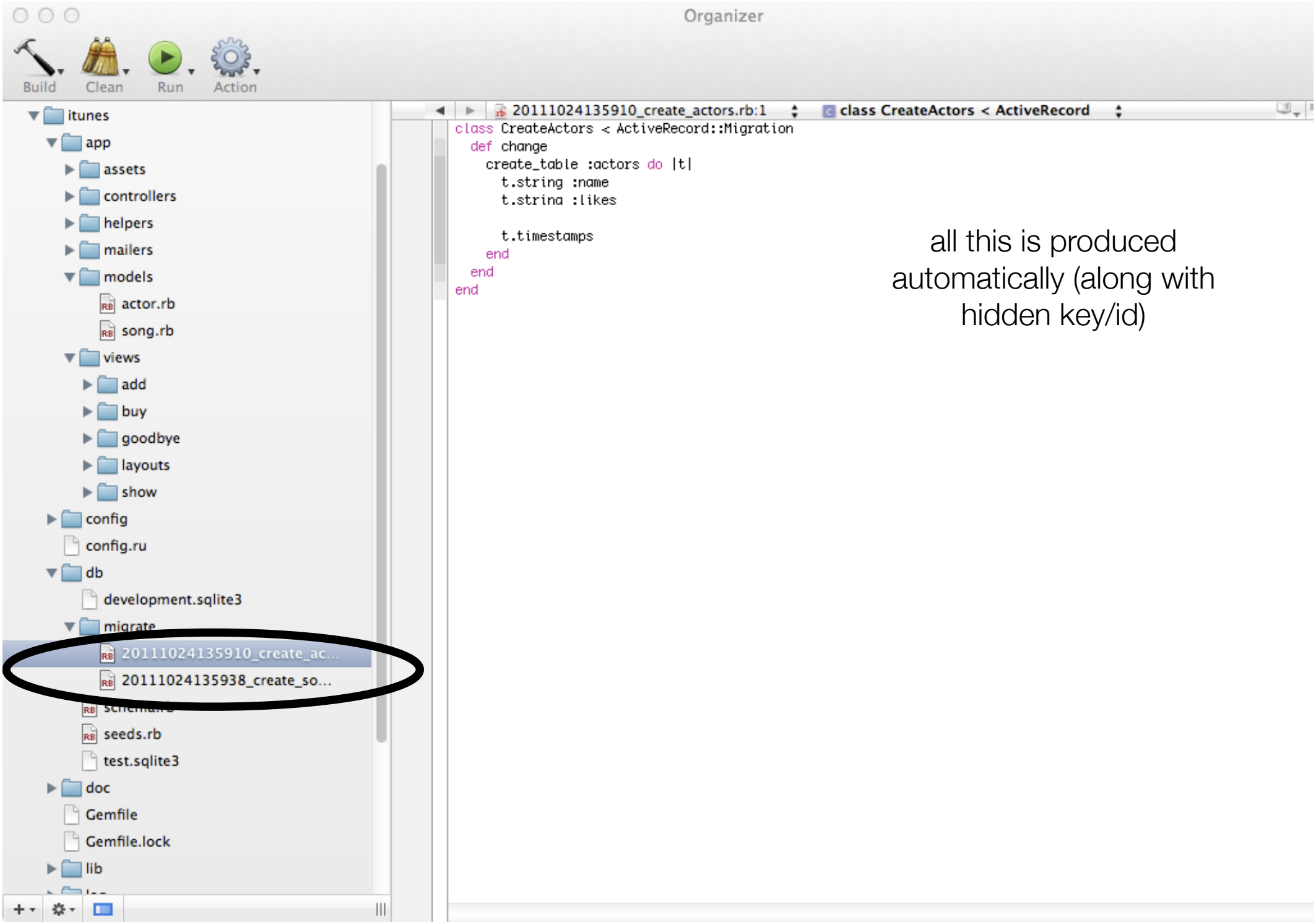
Our initial commands have created a number of files for for actors and song:

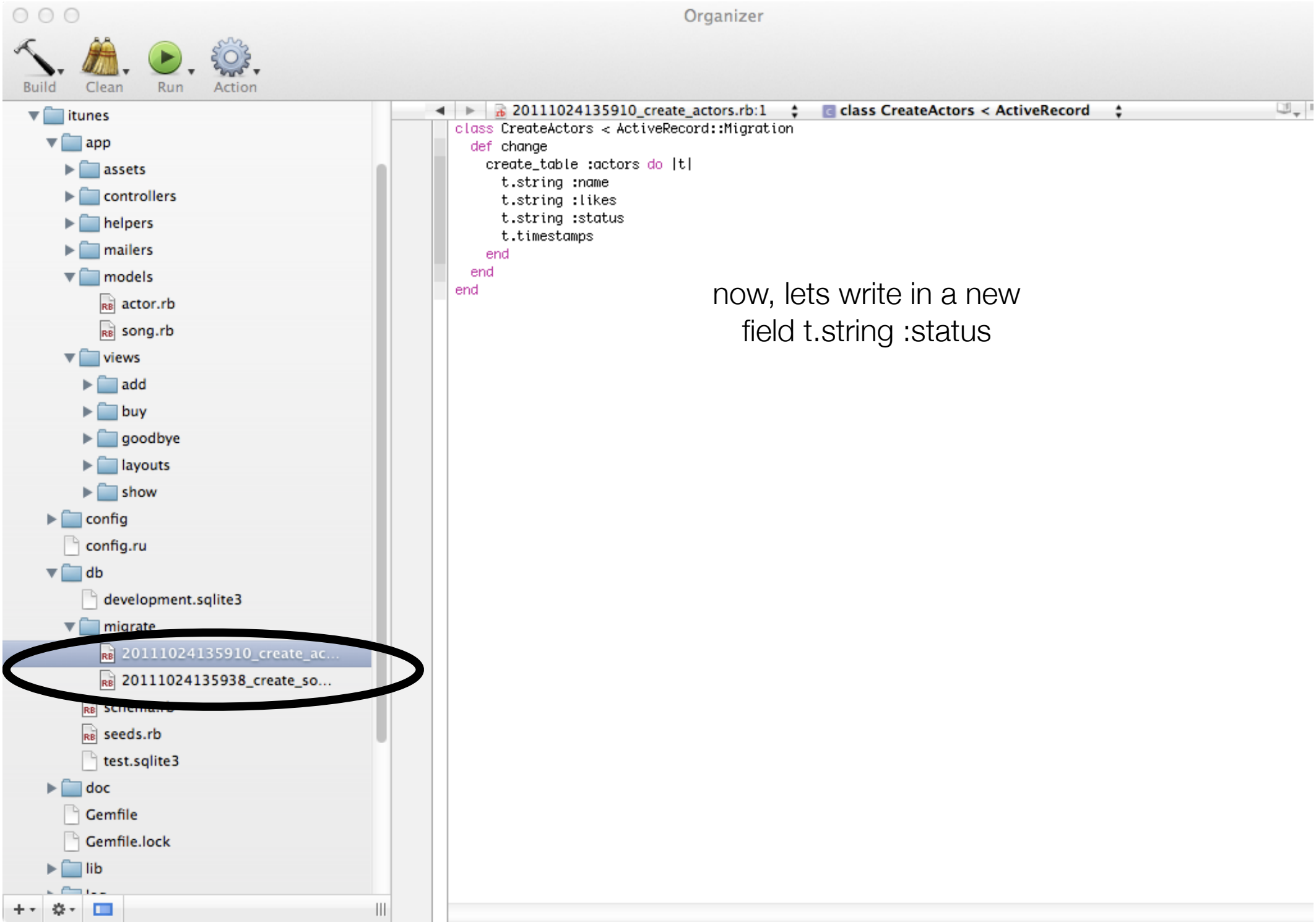
$ rails generate model Actor name:string likes:string

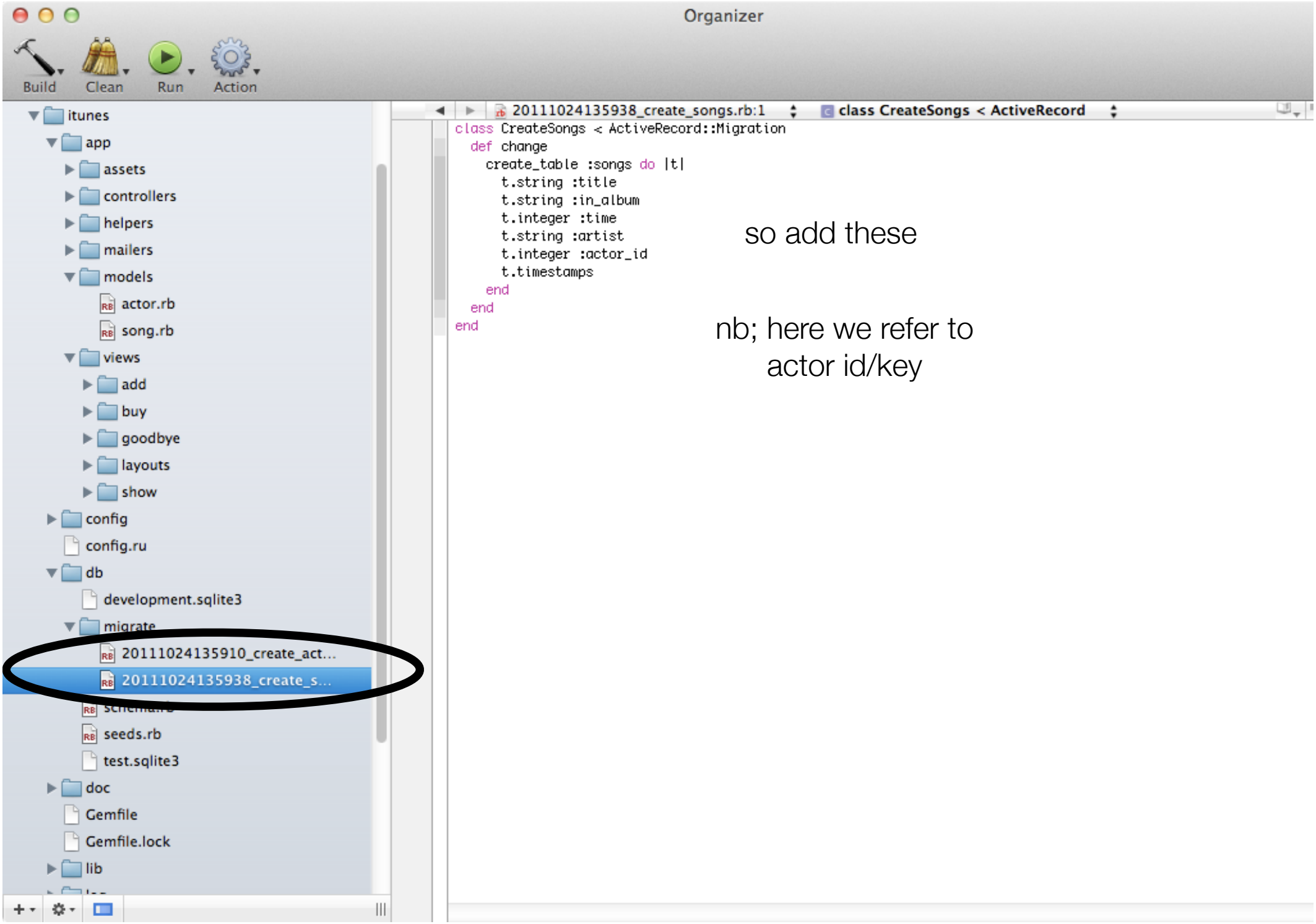
Look in *itunes/db/migrate*

We need to define the relations between those tables

Look in *itunes/app/models/*


Organizer
Build
Clean
Run
Action
itunes
app
assets
controllers
helpers
mailers
models
actor.rb
song.rb
views
add
buy
goodbye
layouts
show
config
config.ru
db
development.sqlite3
migrate
20111024135910_create_ac...
20111024135938_create_so...
seeds.rb
test.sqlite3
doc
Gemfile
Gemfile.lock
lib
20111024135910_create_actors.rb:1
class CreateActors < ActiveRecord
class CreateActors < ActiveRecord::Migration
def change
create_table :actors do |t|
t.string :name
t.string :likes
t.timestamps
end
end
end
all this is produced automatically (along with hidden key/id)


Organizer
Build
Clean
Run
Action
itunes
app
assets
controllers
helpers
mailers
models
actor.rb
song.rb
views
add
buy
goodbye
layouts
show
config
config.ru
db
development.sqlite3
migrate
20111024135910_create_ac...
20111024135938_create_so...
seeds.rb
test.sqlite3
doc
Gemfile
Gemfile.lock
lib
20111024135910_create_actors.rb:1
class CreateActors < ActiveRecord
class CreateActors < ActiveRecord::Migration
def change
create_table :actors do |t|
t.string :name
t.string :likes
t.string :status
t.timestamps
end
end
end
now, lets write in a new
field t.string :status


Organizer
Build
Clean
Run
Action
itunes
app
assets
controllers
helpers
mailers
models
actor.rb
song.rb
views
add
buy
goodbye
layouts
show
config
config.ru
db
development.sqlite3
migrate
20111024135910_create_act...
20111024135938_create_s...
seeds.rb
test.sqlite3
doc
Gemfile
Gemfile.lock
lib
20111024135938_create_songs.rb:1
class CreateSongs < ActiveRecord
class CreateSongs < ActiveRecord::Migration
def change
create_table :songs do |t|
t.string :title
t.string :in_album
t.integer :time
t.string :artist
t.integer :actor_id
t.timestamps
end
end
end
so add these
nb; here we refer to
actor id/key

Types in ActiveRecord

Abstract Type	SQLite Type	Ruby Class
integer	integer	`Fixnum`
decimal	decimal	`BigDecimal`
float	float	`Float`
string	varchar(255)	`String`
text	text	`String`
binary	blob	`String`
date	date	`Date`
datetime	datetime	`Time`
timestamp	datetime	`Time`
time	time	`Time`
boolean	boolean	`TrueClass/FalseClass`

Setup tables etc...

We need to set up a bunch of migrations to define our tables for the db

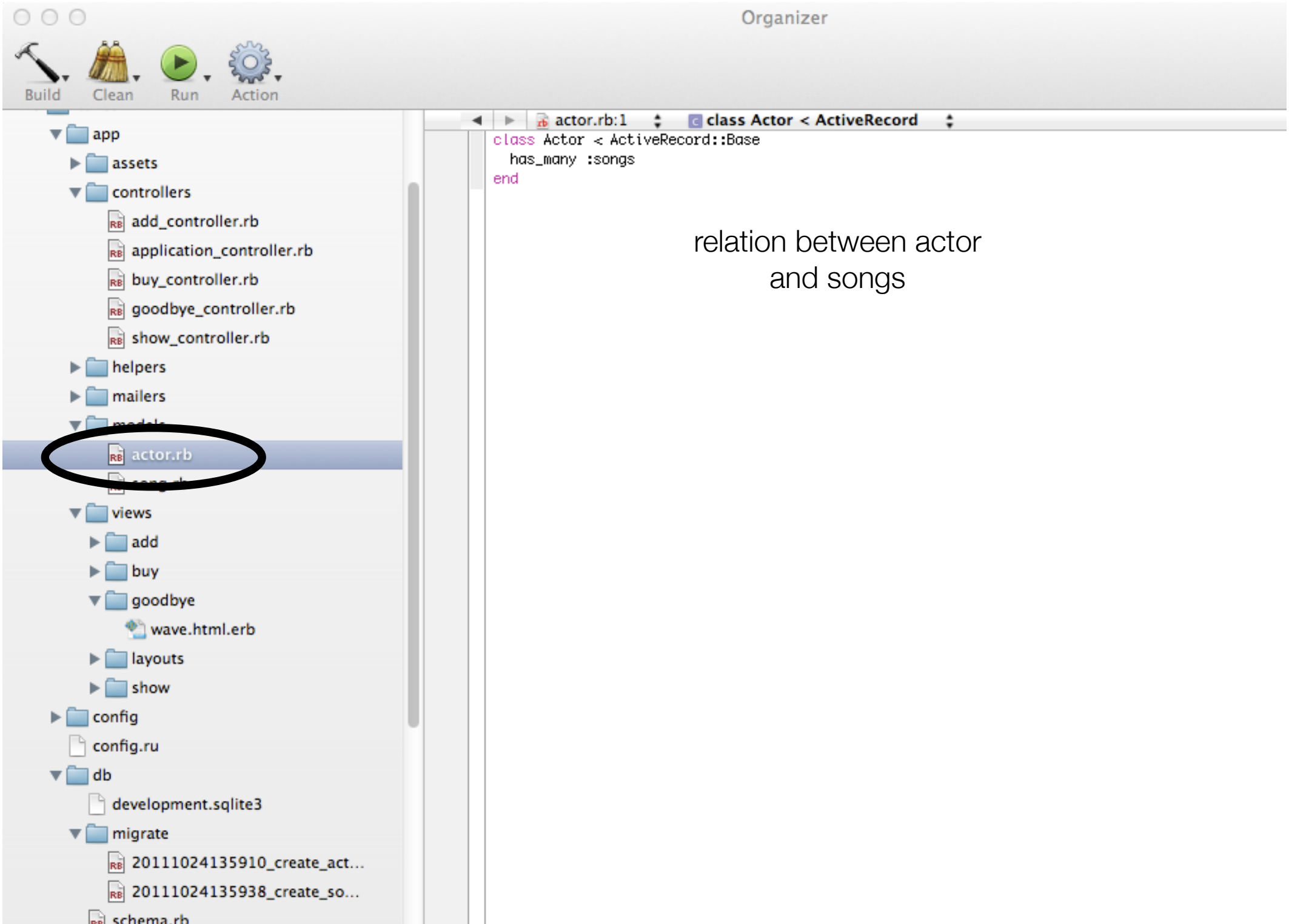
Our initial commands have created a number of files for for actors and song:

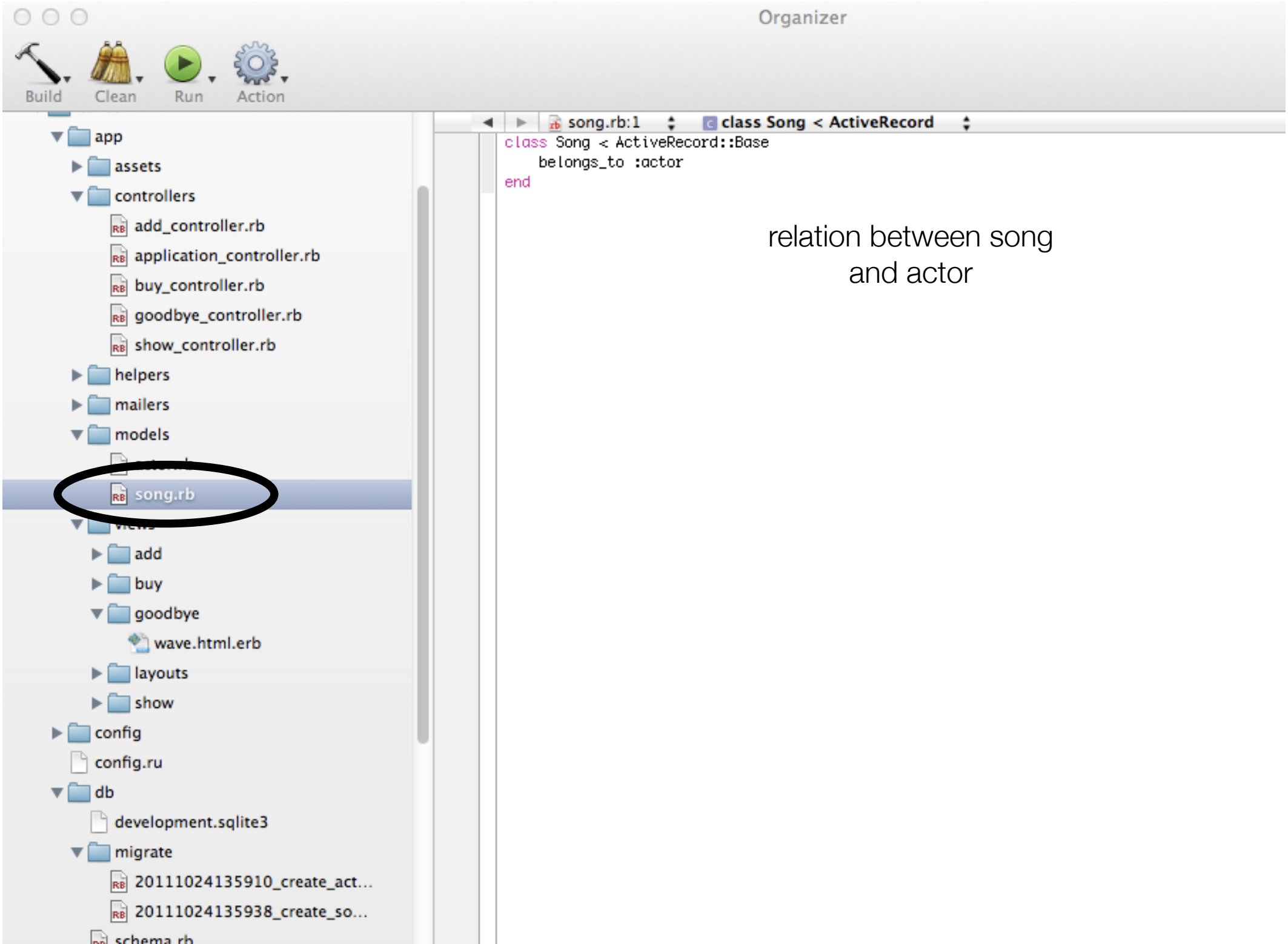
$ rails generate model Actor name:string likes:string

Look in *itunes/db/migrate*

We need to define the relations between those tables

Look in *itunes/app/models/*


Organizer
Build
Clean
Run
Action
app
assets
controllers
add_controller.rb
application_controller.rb
buy_controller.rb
goodbye_controller.rb
show_controller.rb
helpers
mailers
actor.rb
views
add
buy
goodbye
wave.html.erb
layouts
show
config
config.ru
db
development.sqlite3
migrate
20111024135910_create_act...
20111024135938_create_so...
schema.rb
actor.rb:1
class Actor < ActiveRecord
class Actor < ActiveRecord::Base
has_many :songs
end
relation between actor
and songs


Organizer
Build
Clean
Run
Action
app
assets
controllers
add_controller.rb
application_controller.rb
buy_controller.rb
goodbye_controller.rb
show_controller.rb
helpers
mailers
models
song.rb
add
buy
goodbye
wave.html.erb
layouts
show
config
config.ru
db
development.sqlite3
migrate
20111024135910_create_act...
20111024135938_create_so...
schema.rb
song.rb:1
class Song < ActiveRecord
class Song < ActiveRecord::Base
belongs_to :actor
end
relation between song
and actor

Types in ActiveRecord

Do not do **rake db:migrate**
until you have all your models
and migrations are defined

rake db:migrate creates a **schema.rb** file that records the current version of the migrations

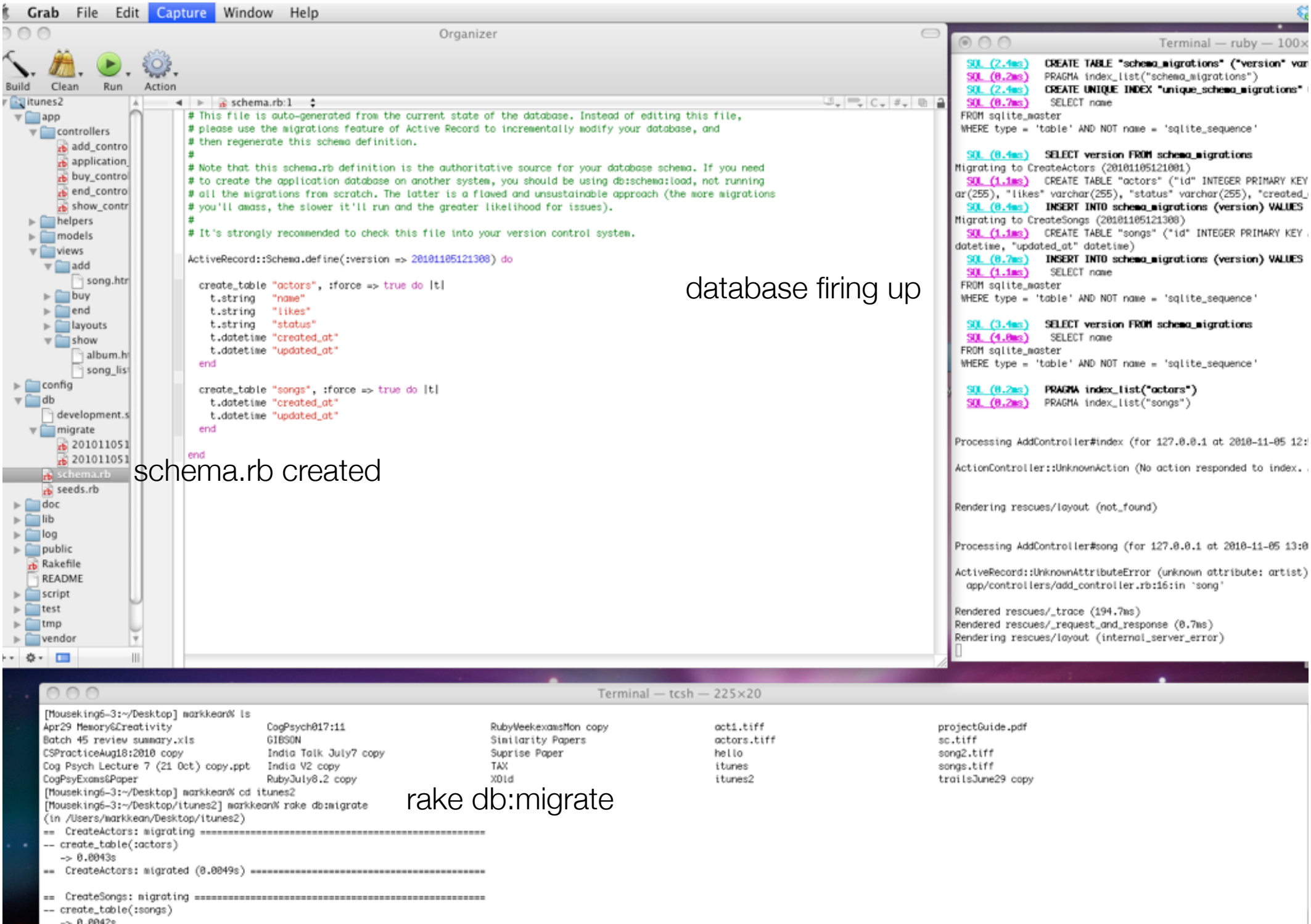
...never change **schema.rb** by hand

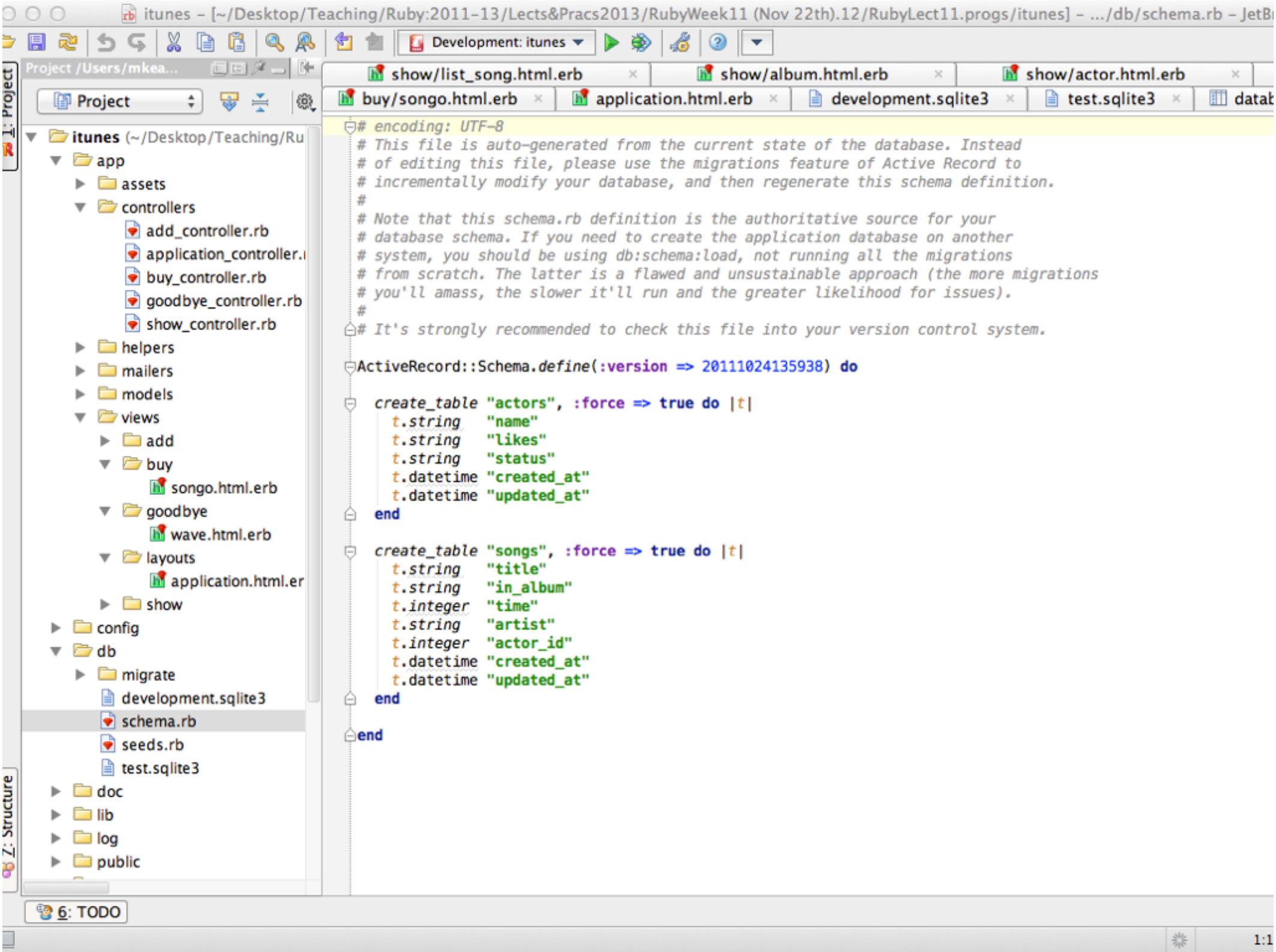
...never create supporting files for controller by hand

Models recipe...

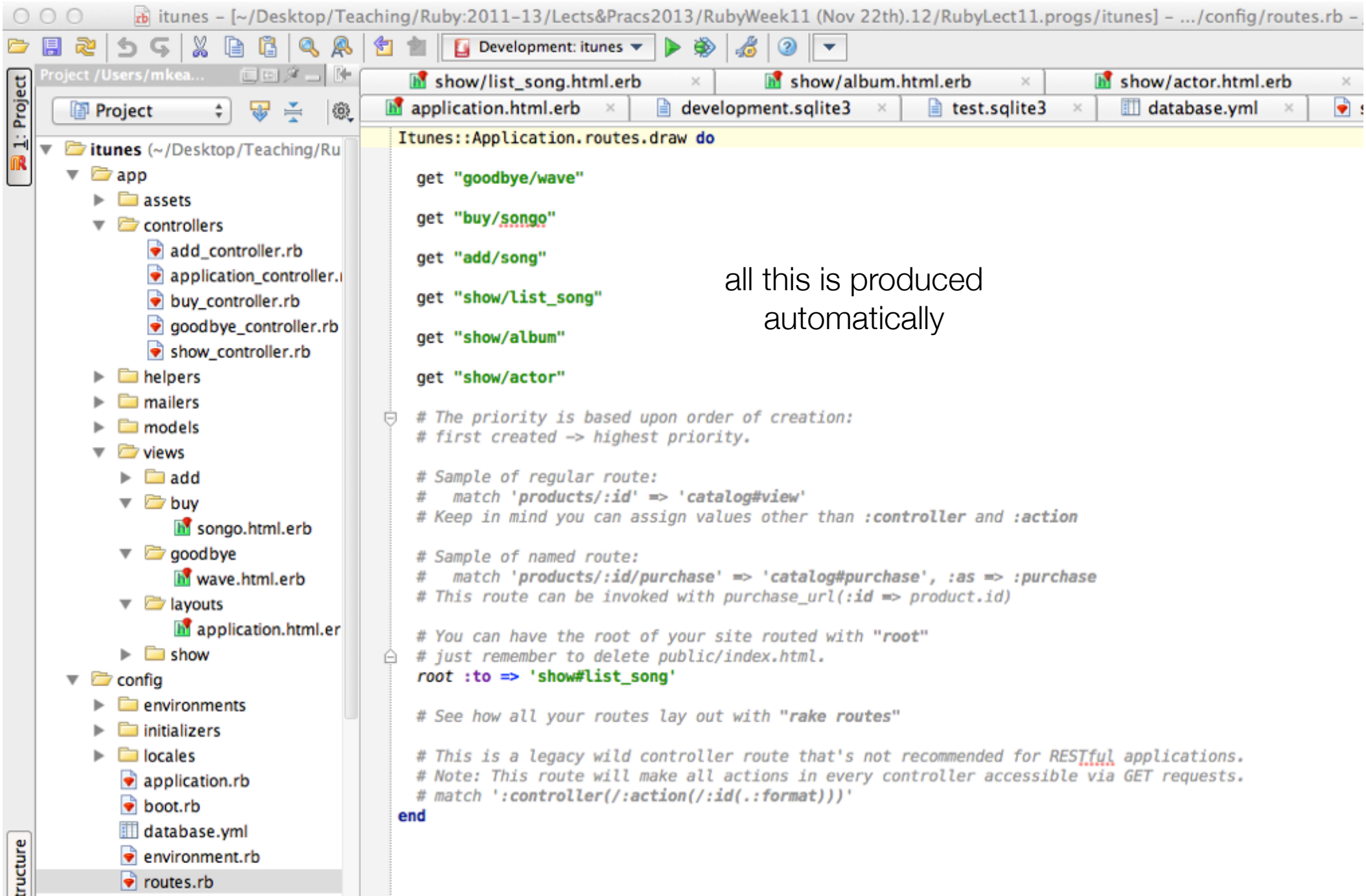
1) set up models
2) check and/or edit migration files
3) include db relations (if required) in model files (eg actor.rb)
4) edit your seed files (will return to this later)
5) do **$ rake db:migrate**
6) do **$ rake db:seed**

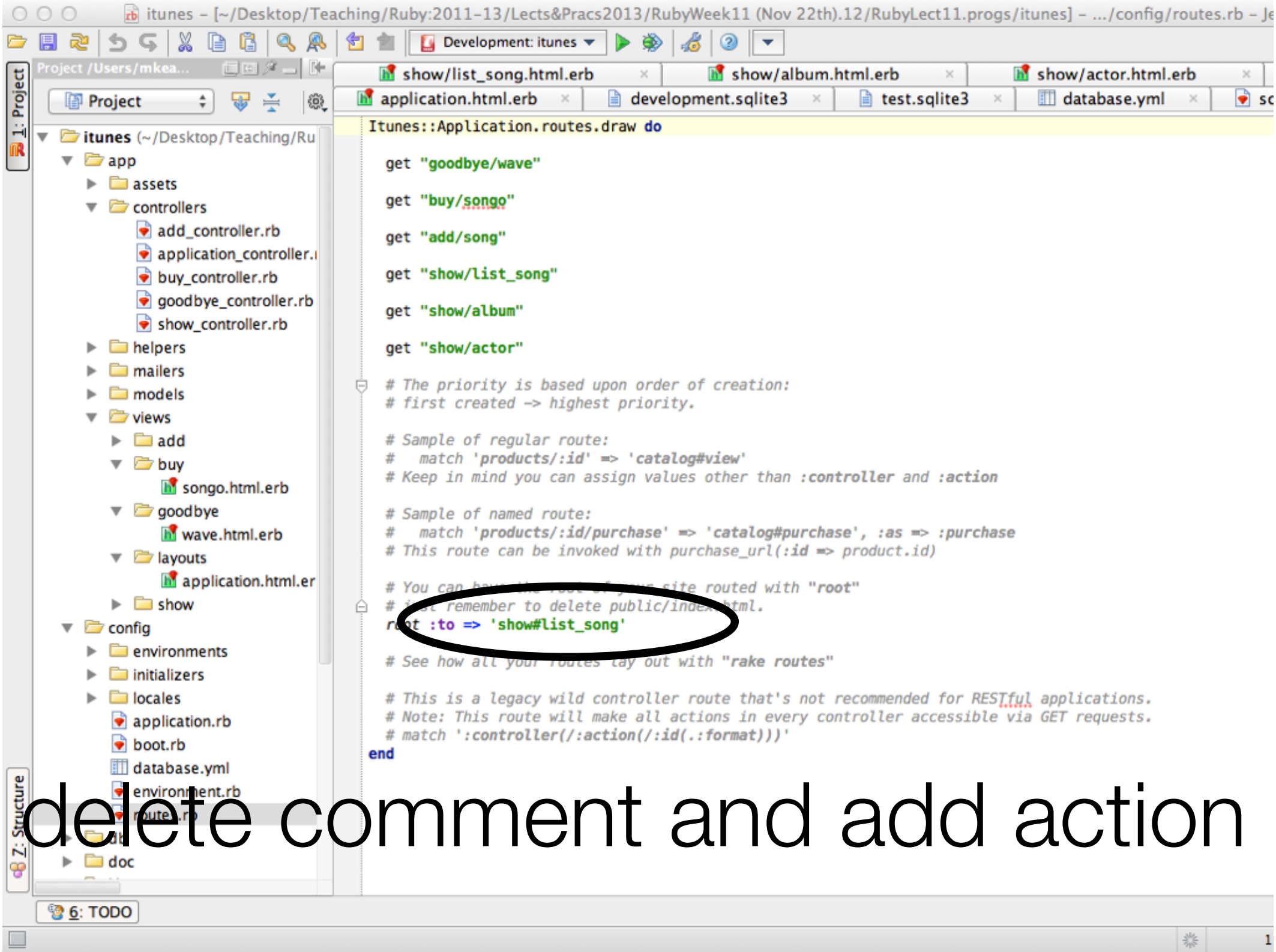
...then you can move to your browser and start running...


Grab File Edit Capture Window Help
Organizer
Build Clean Run Action
itunes2
app
controllers
add_contro
application_
buy_contro
end_contro
show_contr
helpers
models
views
add
song.htr
buy
end
layouts
show
album.h
song_lis
config
db
development.s
migrate
201011051
201011051
schema.rb
seeds.rb
doc
lib
log
public
Rakefile
README
script
test
tmp
vendor
schema.rb:1
This file is auto-generated from the current state of the database. Instead of editing this file,
please use the migrations feature of Active Record to incrementally modify your database, and
then regenerate this schema definition.
#
Note that this schema.rb definition is the authoritative source for your database schema. If you need
to create the application database on another system, you should be using db:schema:load, not running
all the migrations from scratch. The latter is a flawed and unsustainable approach (the more migrations
you'll amass, the slower it'll run and the greater likelihood for issues).
#
It's strongly recommended to check this file into your version control system.
ActiveRecord::Schema.define(:version => 20101105121308) do
create_table "actors", :force => true do |t|
t.string "name"
t.string "likes"
t.string "status"
t.datetime "created_at"
t.datetime "updated_at"
end
create_table "songs", :force => true do |t|
t.datetime "created_at"
t.datetime "updated_at"
end
end
schema.rb created
database firing up
Terminal — ruby — 100×
SQL (2.4ms) CREATE TABLE "schema_migrations" ("version" var
SQL (0.2ms) PRAGMA index_list("schema_migrations")
SQL (2.4ms) CREATE UNIQUE INDEX "unique_schema_migrations"
SQL (0.7ms) SELECT name
FROM sqlite_master
WHERE type = 'table' AND NOT name = 'sqlite_sequence'
SQL (0.4ms) SELECT version FROM schema_migrations
Migrating to CreateActors (20101105121001)
SQL (1.1ms) CREATE TABLE "actors" ("id" INTEGER PRIMARY KEY
ar(255), "likes" varchar(255), "status" varchar(255), "created_
SQL (0.4ms) INSERT INTO schema_migrations (version) VALUES
Migrating to CreateSongs (20101105121308)
SQL (1.1ms) CREATE TABLE "songs" ("id" INTEGER PRIMARY KEY
datetime, "updated_at" datetime)
SQL (0.7ms) INSERT INTO schema_migrations (version) VALUES
SQL (1.1ms) SELECT name
FROM sqlite_master
WHERE type = 'table' AND NOT name = 'sqlite_sequence'
SQL (3.4ms) SELECT version FROM schema_migrations
SQL (4.0ms) SELECT name
FROM sqlite_master
WHERE type = 'table' AND NOT name = 'sqlite_sequence'
SQL (0.2ms) PRAGMA index_list("actors")
SQL (0.2ms) PRAGMA index_list("songs")
Processing AddController#index (for 127.0.0.1 at 2010-11-05 12:
ActionController::UnknownAction (No action responded to index.
Rendering rescues/layout (not_found)
Processing AddController#song (for 127.0.0.1 at 2010-11-05 13:0
ActiveRecord::UnknownAttributeError (unknown attribute: artist)
app/controllers/add_controller.rb:16:in 'song'
Rendered rescues/_trace (194.7ms)
Rendered rescues/_request_and_response (0.7ms)
Rendering rescues/layout (internal_server_error)
Terminal — tcsh — 225×20
[Mouseking6-3:~/Desktop] markkean% ls
Apr29 Memory&Creativity
Batch 45 review summary.xls
CSPracticeAug18:2010 copy
Cog Psych Lecture 7 (21 Oct) copy.ppt
CogPsyExams&Paper
CogPsych017:11
GIBSON
India Talk July7 copy
India V2 copy
RubyJuly8.2 copy
RubyWeekexamsMon copy
Similarity Papers
Suprise Paper
TAX
XOld
act1.tiff
actors.tiff
hello
itunes
itunes2
projectGuide.pdf
sc.tiff
song2.tiff
songs.tiff
trailsJune29 copy
[Mouseking6-3:~/Desktop] markkean% cd itunes2
[Mouseking6-3:~/Desktop/itunes2] markkean% rake db:migrate
(in /Users/markkean/Desktop/itunes2)
== CreateActors: migrating ==
-- create_table(:actors)
-> 0.0043s
== CreateActors: migrated (0.0049s) ==
== CreateSongs: migrating ==
-- create_table(:songs)
rake db:migrate

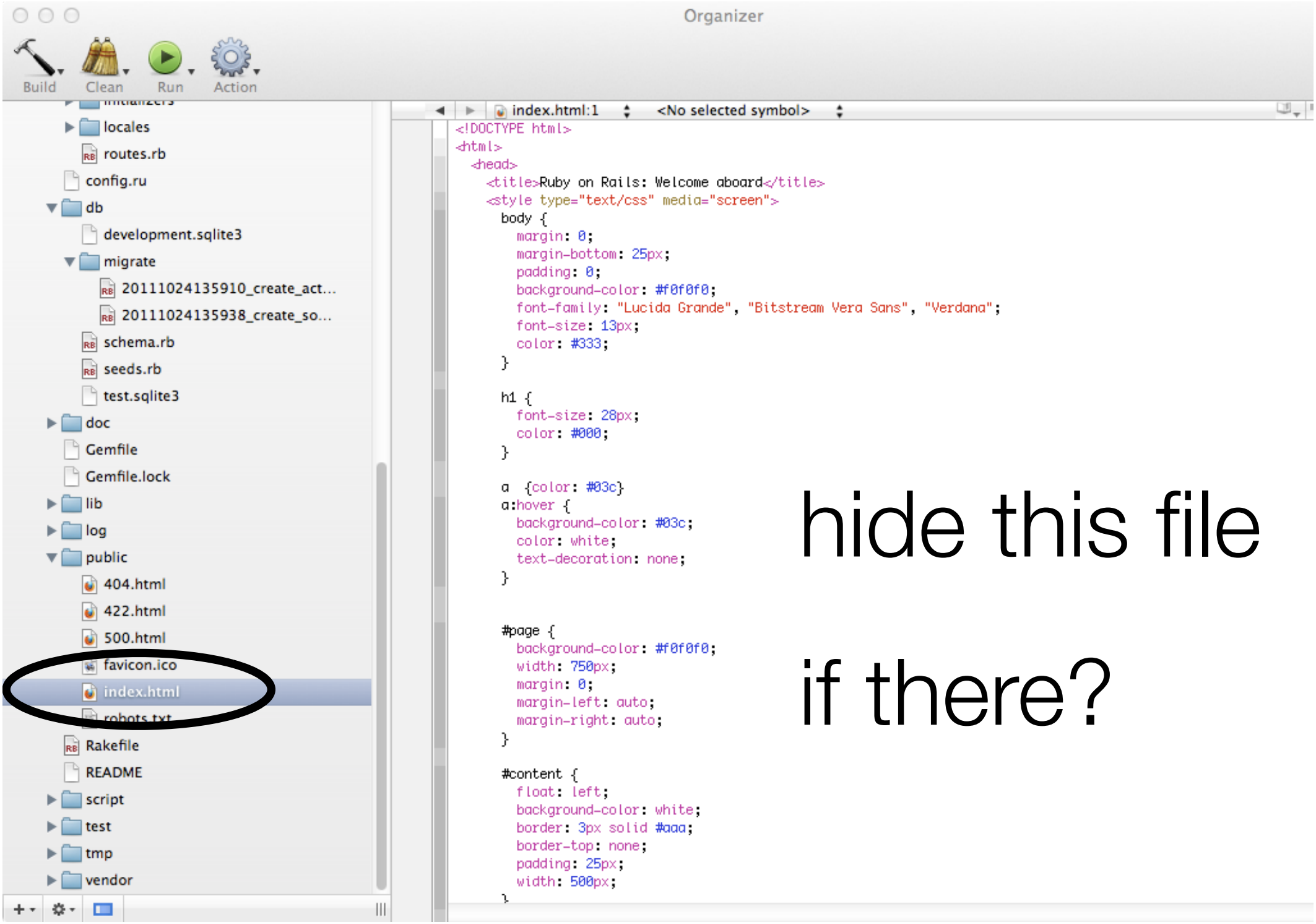

itunes – [~/Desktop/Teaching/Ruby:2011-13/Lects&Pracs2013/RubyWeek11 (Nov 22th).12/RubyLect11.progs/itunes] – .../db/schema.rb – JetB
Development: itunes
Project /Users/mkea...
Project
itunes (~/Desktop/Teaching/Ru
app
assets
controllers
add_controller.rb
application_controller.
buy_controller.rb
goodbye_controller.rb
show_controller.rb
helpers
mailers
models
views
add
buy
songo.html.erb
goodbye
wave.html.erb
layouts
application.html.er
show
config
db
migrate
development.sqlite3
schema.rb
seeds.rb
test.sqlite3
doc
lib
log
public
show/list_song.html.erb
show/album.html.erb
show/actor.html.erb
buy/songo.html.erb
application.html.erb
development.sqlite3
test.sqlite3
encoding: UTF-8
This file is auto-generated from the current state of the database. Instead
of editing this file, please use the migrations feature of Active Record to
incrementally modify your database, and then regenerate this schema definition.
#
Note that this schema.rb definition is the authoritative source for your
database schema. If you need to create the application database on another
system, you should be using db:schema:load, not running all the migrations
from scratch. The latter is a flawed and unsustainable approach (the more migrations
you'll amass, the slower it'll run and the greater likelihood for issues).
#
It's strongly recommended to check this file into your version control system.
ActiveRecord::Schema.define(:version => 20111024135938) do
create_table "actors", :force => true do |t|
t.string "name"
t.string "likes"
t.string "status"
t.datetime "created_at"
t.datetime "updated_at"
end
create_table "songs", :force => true do |t|
t.string "title"
t.string "in_album"
t.integer "time"
t.string "artist"
t.integer "actor_id"
t.datetime "created_at"
t.datetime "updated_at"
end
end
6: TODO
1:1

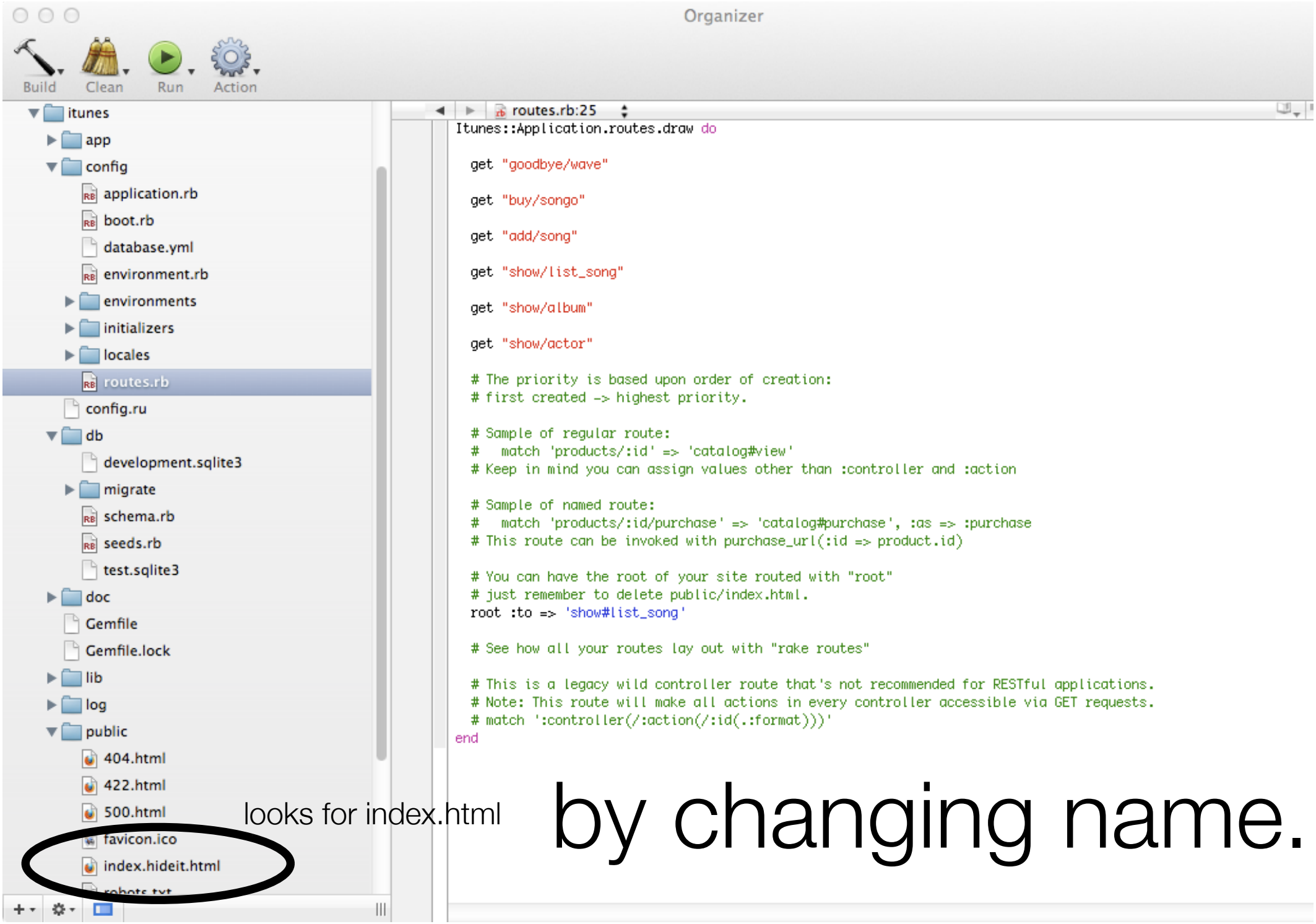
In Rails 4.0 routes.rb

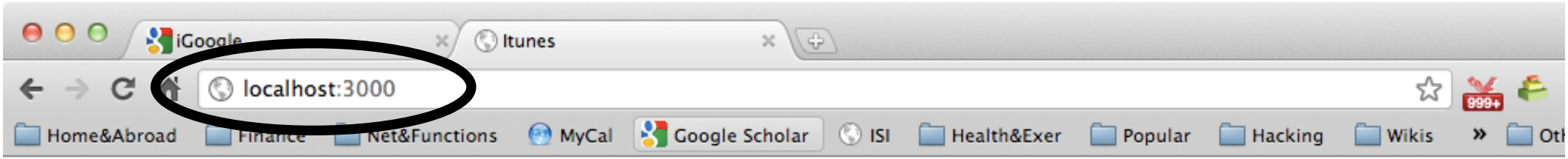




delete comment and add action


Organizer
Build
Clean
Run
Action
locales
routes.rb
config.ru
db
development.sqlite3
migrate
20111024135910_create_act...
20111024135938_create_so...
schema.rb
seeds.rb
test.sqlite3
doc
Gemfile
Gemfile.lock
lib
log
public
404.html
422.html
500.html
favicon.ico
index.html
robots.txt
Rakefile
README
script
test
tmp
vendor
index.html:1
<No selected symbol>
<!DOCTYPE html>
<html>
<head>
<title>Ruby on Rails: Welcome aboard</title>
<style type="text/css" media="screen">
body {
margin: 0;
margin-bottom: 25px;
padding: 0;
background-color: #f0f0f0;
font-family: "Lucida Grande", "Bitstream Vera Sans", "Verdana";
font-size: 13px;
color: #333;
}
h1 {
font-size: 28px;
color: #000;
}
a {color: #03c}
a:hover {
background-color: #03c;
color: white;
text-decoration: none;
}
#page {
background-color: #f0f0f0;
width: 750px;
margin: 0;
margin-left: auto;
margin-right: auto;
}
#content {
float: left;
background-color: white;
border: 3px solid #aaa;
border-top: none;
padding: 25px;
width: 500px;
}
hide this file
if there?


Organizer
Build
Clean
Run
Action
itunes
app
config
application.rb
boot.rb
database.yml
environment.rb
environments
initializers
locales
routes.rb
config.ru
db
development.sqlite3
migrate
schema.rb
seeds.rb
test.sqlite3
doc
Gemfile
Gemfile.lock
lib
log
public
404.html
422.html
500.html
favicon.ico
index.hideit.html
routes.rb:25
Itunes::Application.routes.draw do
get "goodbye/wave"
get "buy/songo"
get "add/song"
get "show/list_song"
get "show/album"
get "show/actor"
The priority is based upon order of creation:
first created -> highest priority.
Sample of regular route:
match 'products/:id' => 'catalog#view'
Keep in mind you can assign values other than :controller and :action
Sample of named route:
match 'products/:id/purchase' => 'catalog#purchase', :as => :purchase
This route can be invoked with purchase_url(:id => product.id)
You can have the root of your site routed with "root"
just remember to delete public/index.html.
root :to => 'show#list_song'
See how all your routes lay out with "rake routes"
This is a legacy wild controller route that's not recommended for RESTful applications.
Note: This route will make all actions in every controller accessible via GET requests.
match ':controller(/:action(/:id(.:format)))'
end
looks for index.html
by changing name.



Song Listing

- Stairway to Heaven by Led Zep, in Led Zep IV : 1 [Owned by Actor ID 1]
- Holiday by Sex Pistols, in God Save... : 2 [Owned by Actor ID 2]
- Gloria by Patti Smith, in Horses : 3 [Owned by Actor ID]

Part B(v):

Editing the Controllers & Views

Controllers & Views

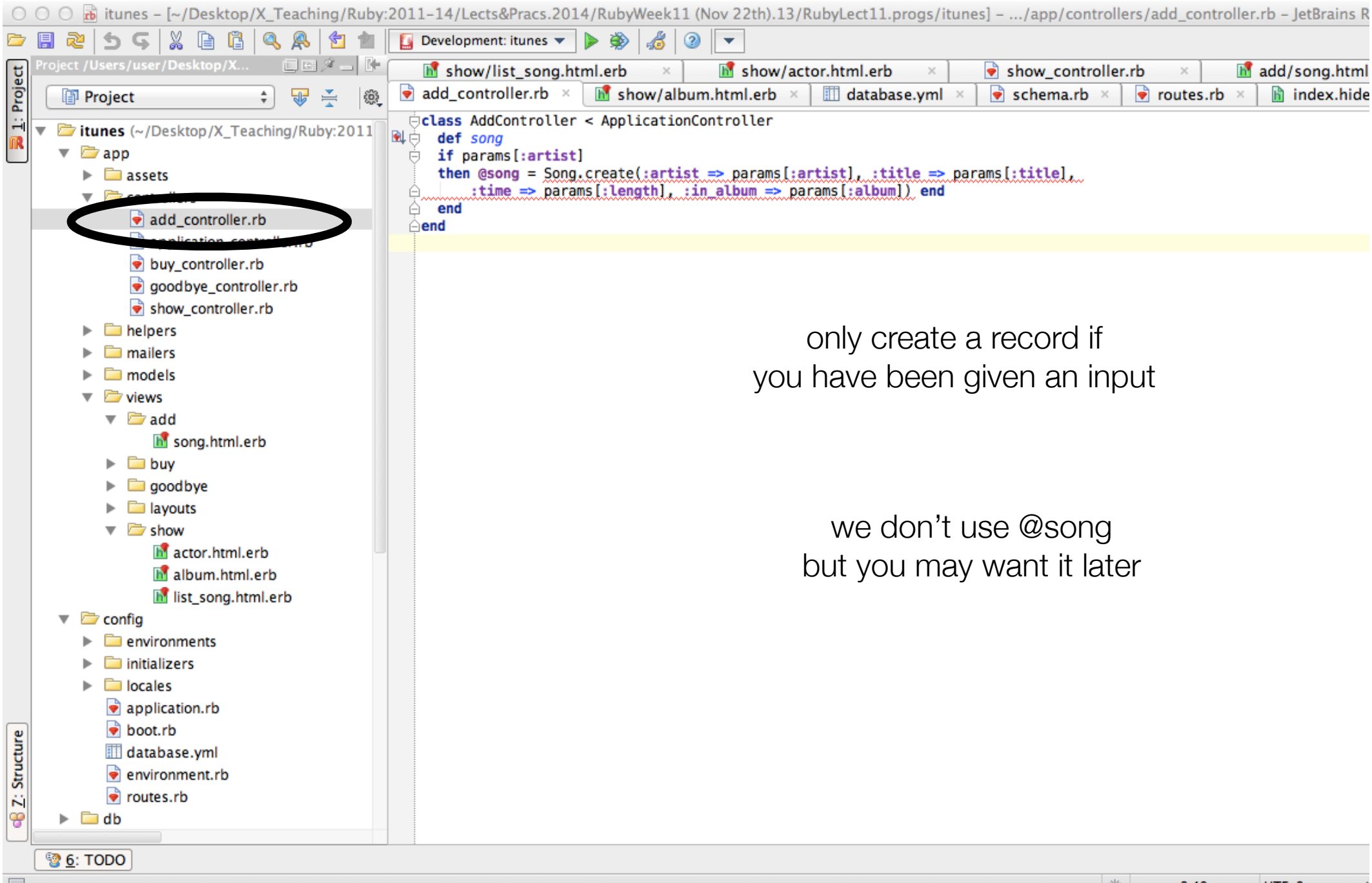
most of our controllers are set up with empty methods and without view templates

so, we need to write the methods and the ERb to display what we want to display

there should be no surprises here as we do not do anything beyond the functionality we saw in earlier rails apps

I am ignoring layout issues, for simplicity

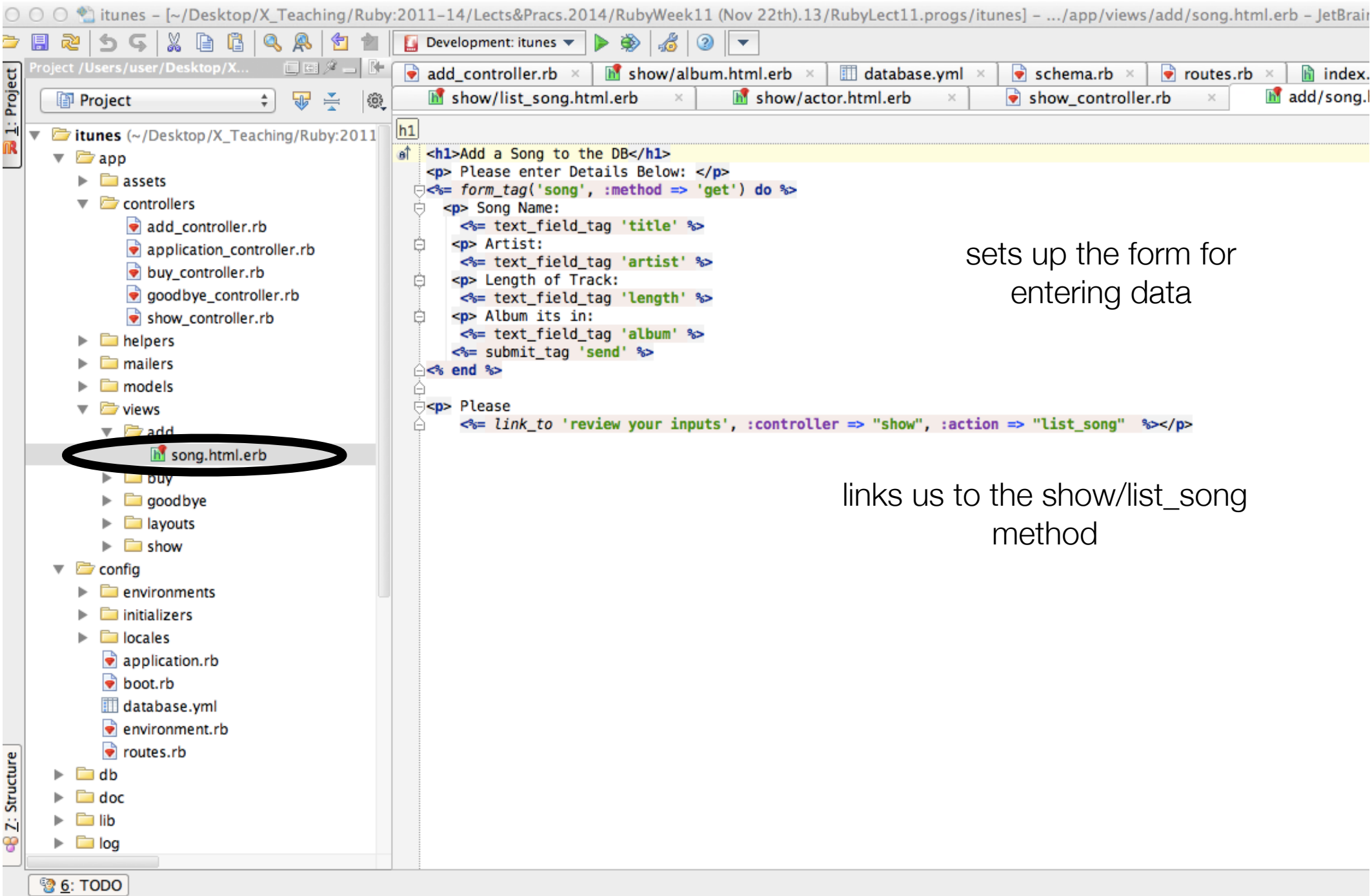
Add Controller



only create a record if
you have been given an input

we don't use @song
but you may want it later

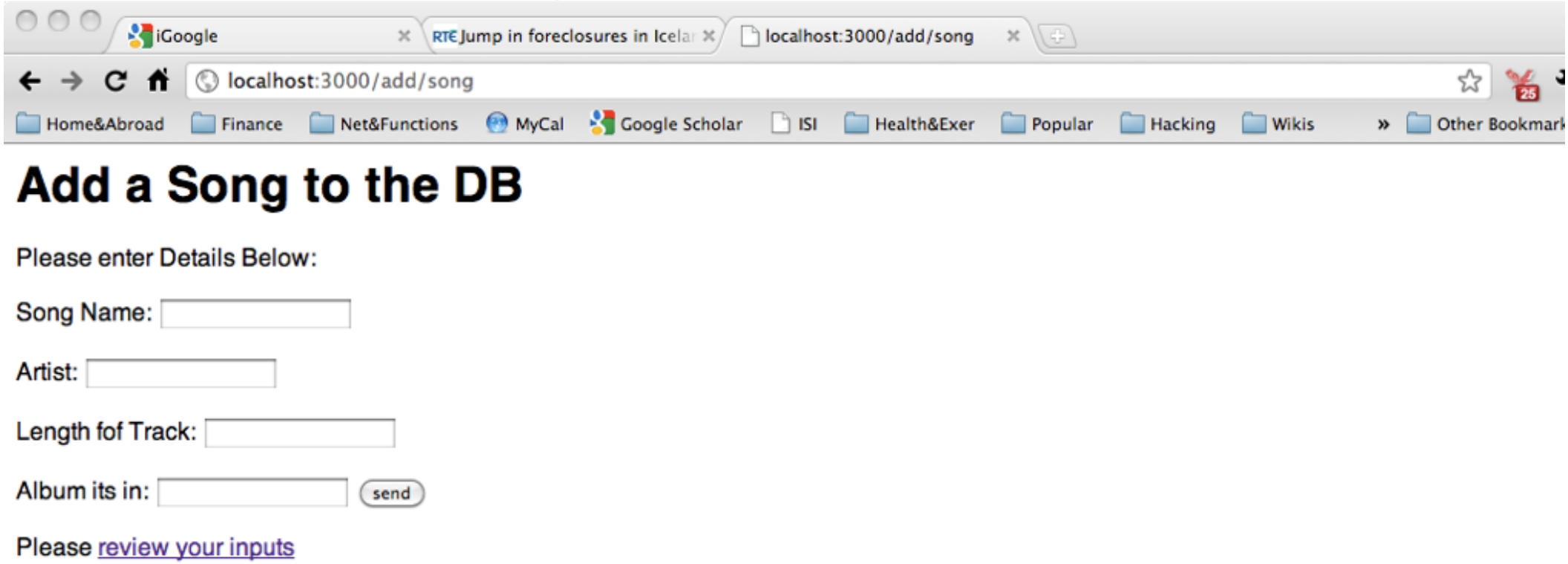
Add View



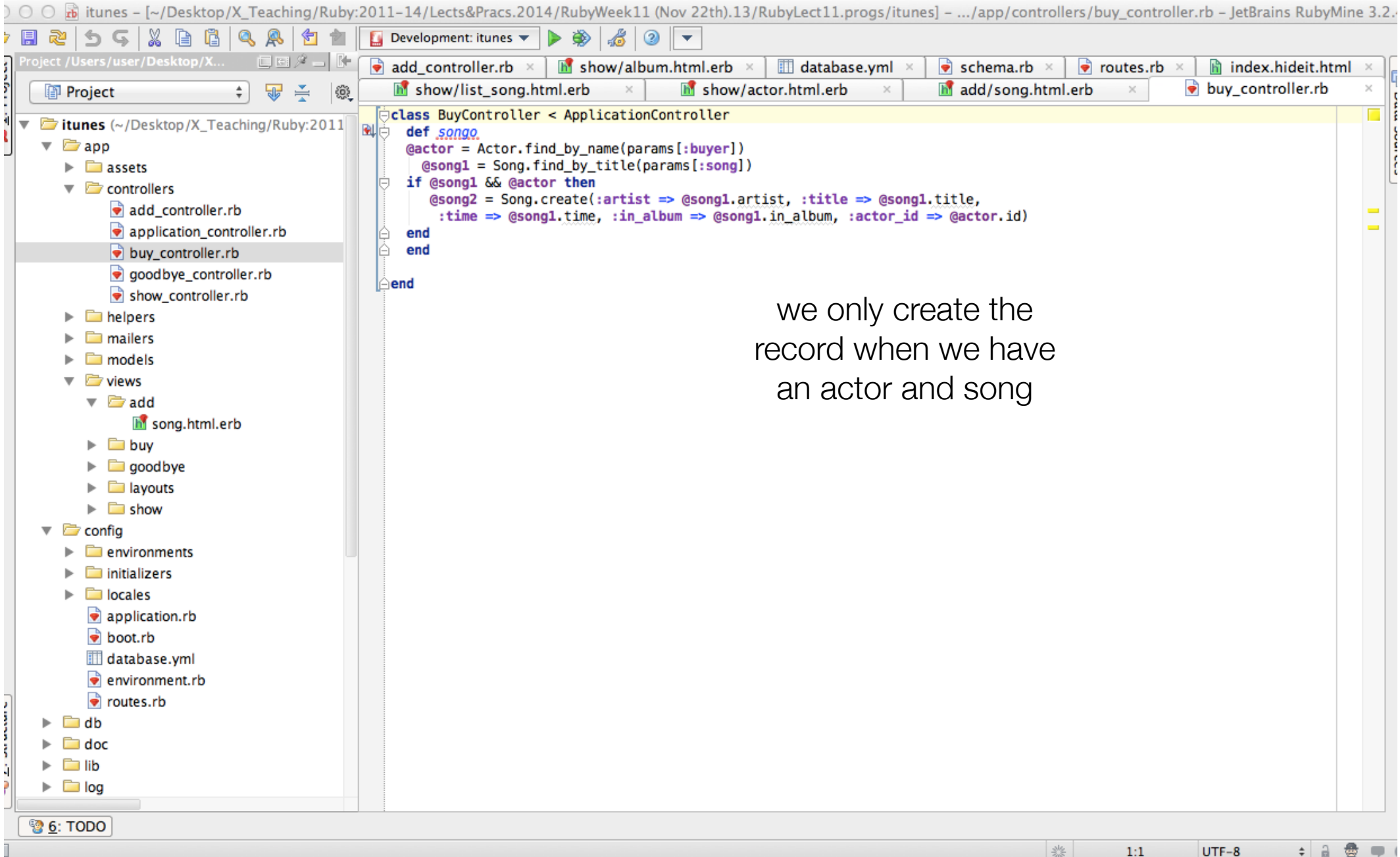
sets up the form for entering data

links us to the show/list_song method

add/song Displays...

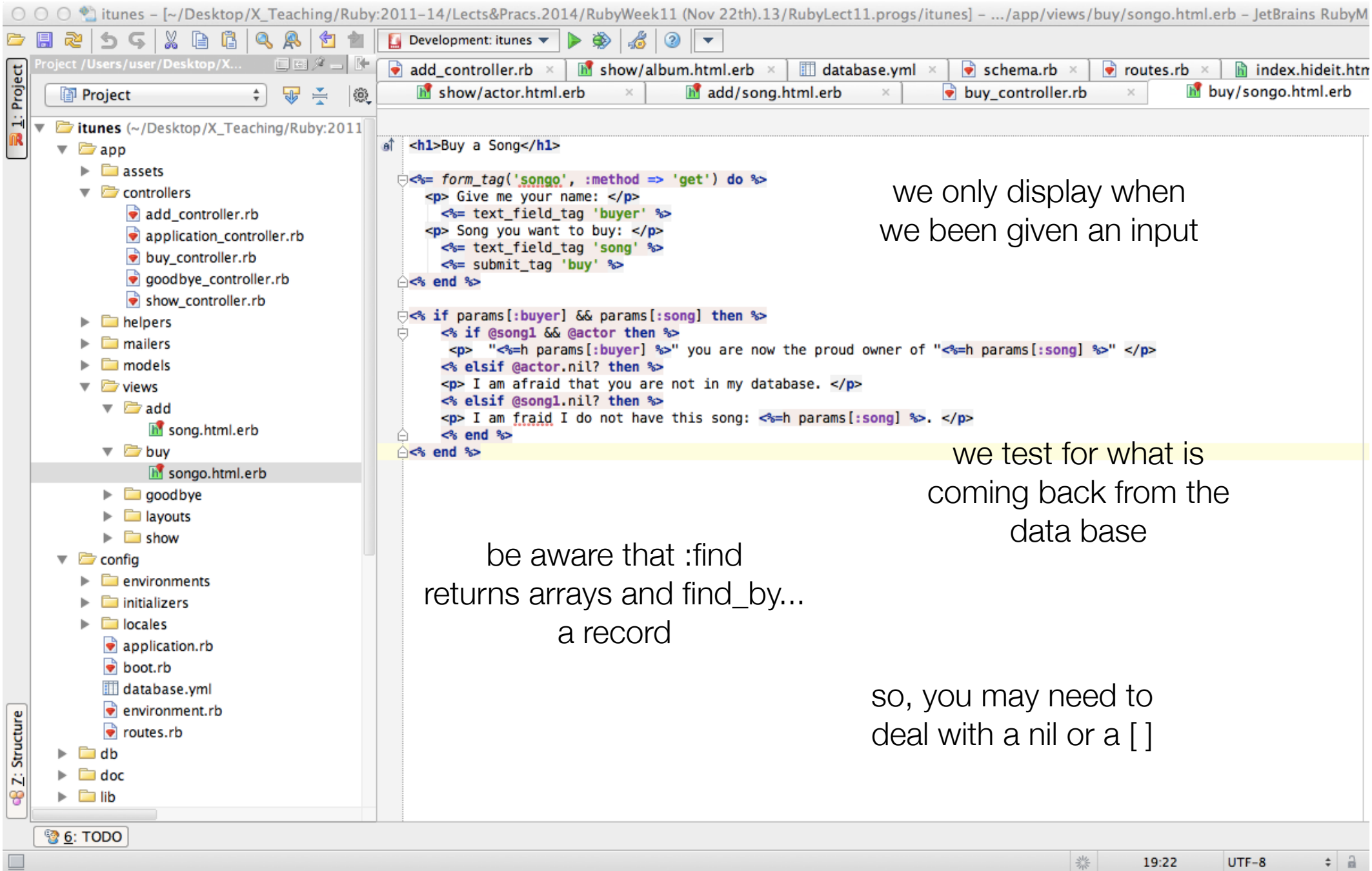


Buy Controller

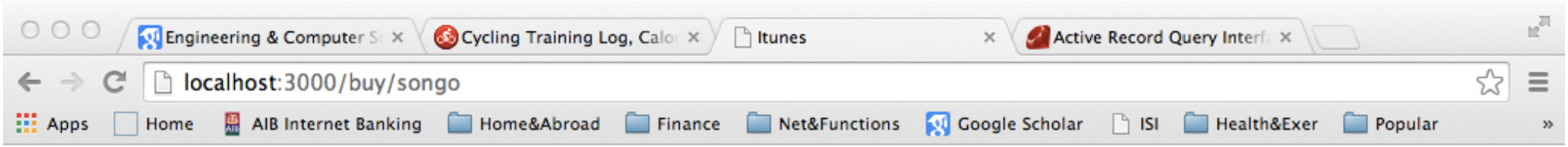


we only create the record when we have an actor and song

Buy/songo View



buy/songo Displays…



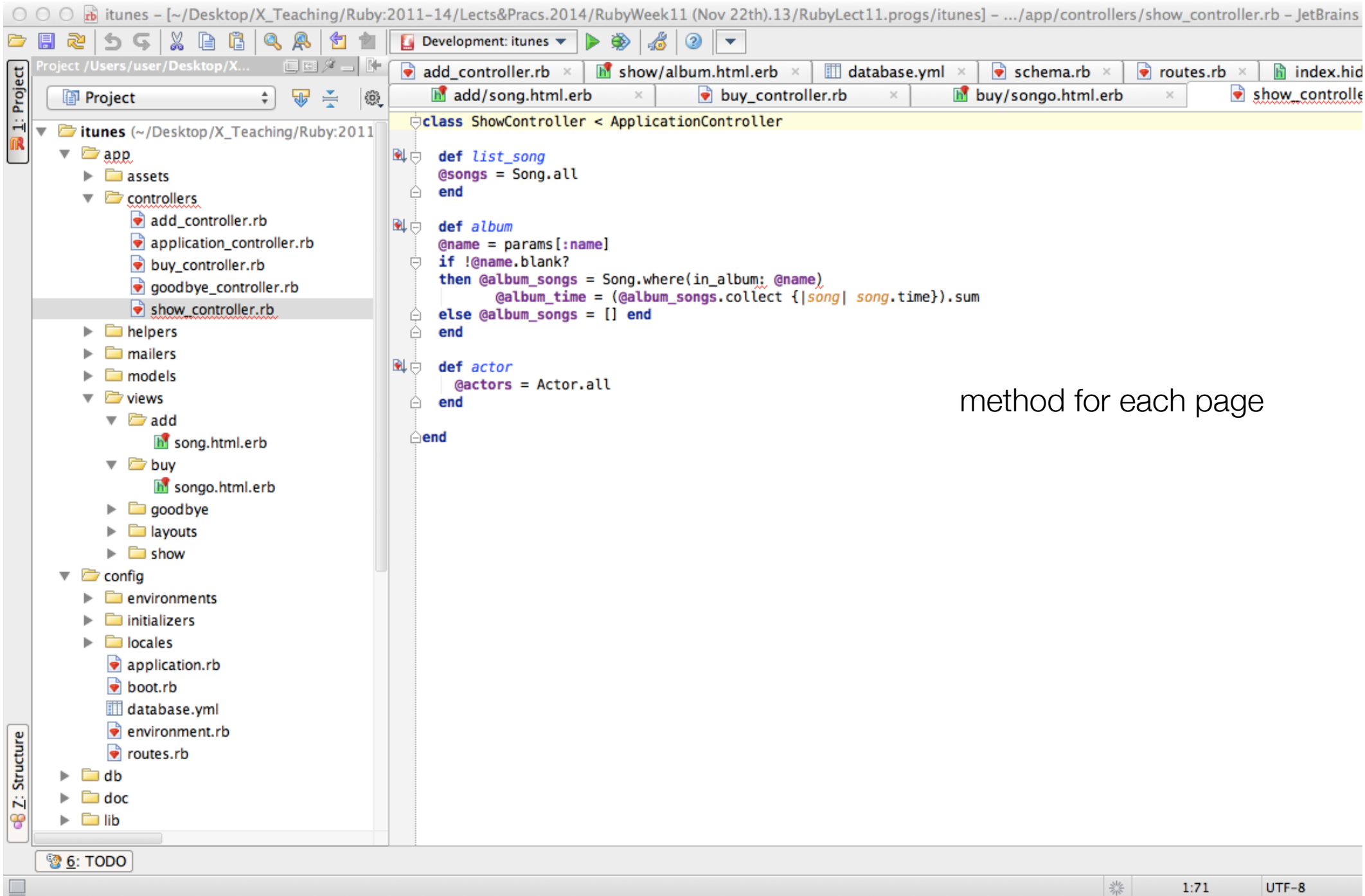
Buy a Song

Give me your name:

Song you want to buy:

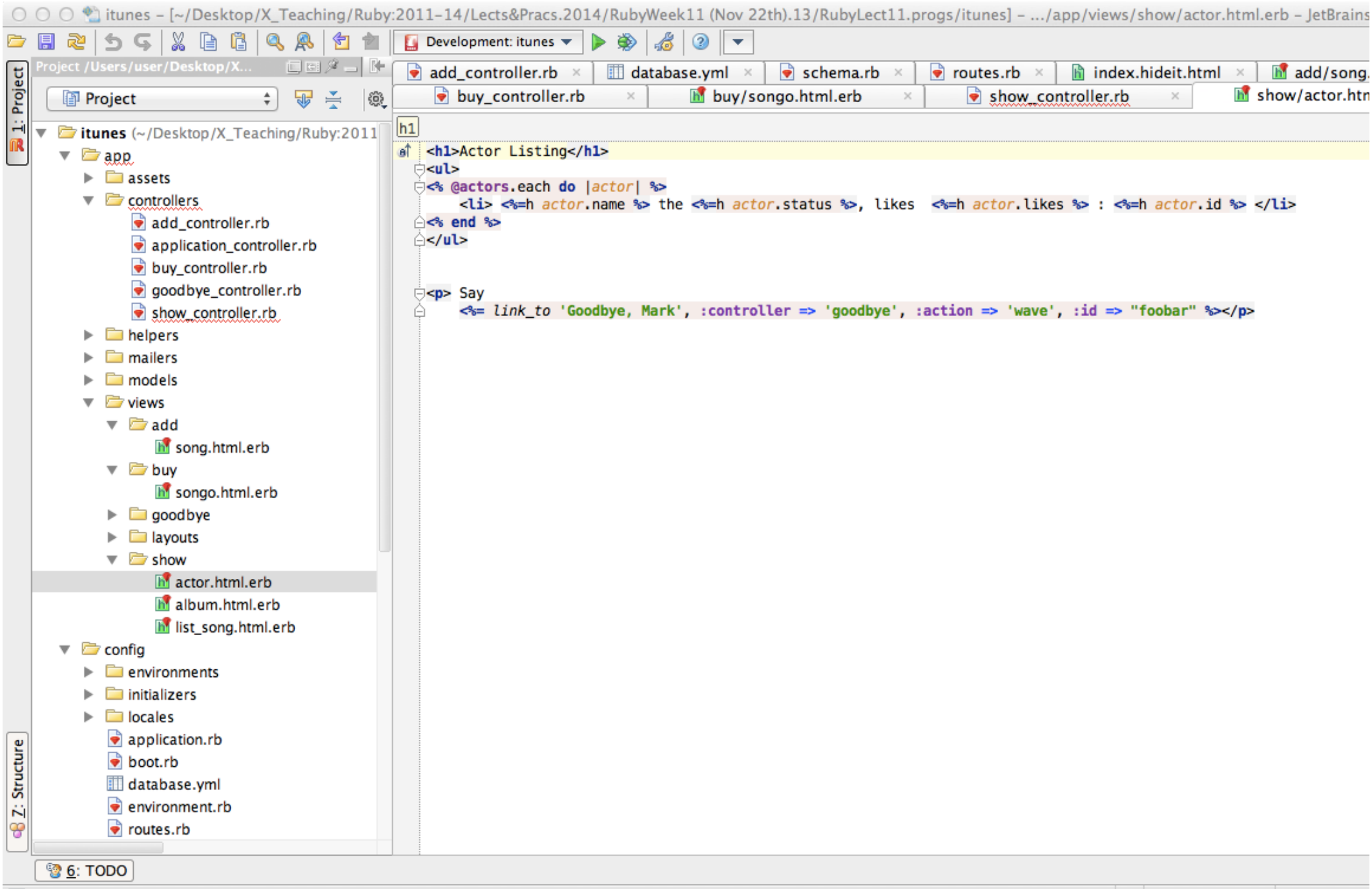
buy

show Controller

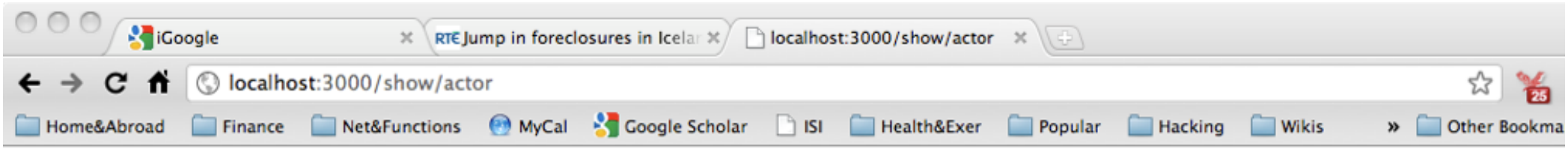


method for each page

show/actor View



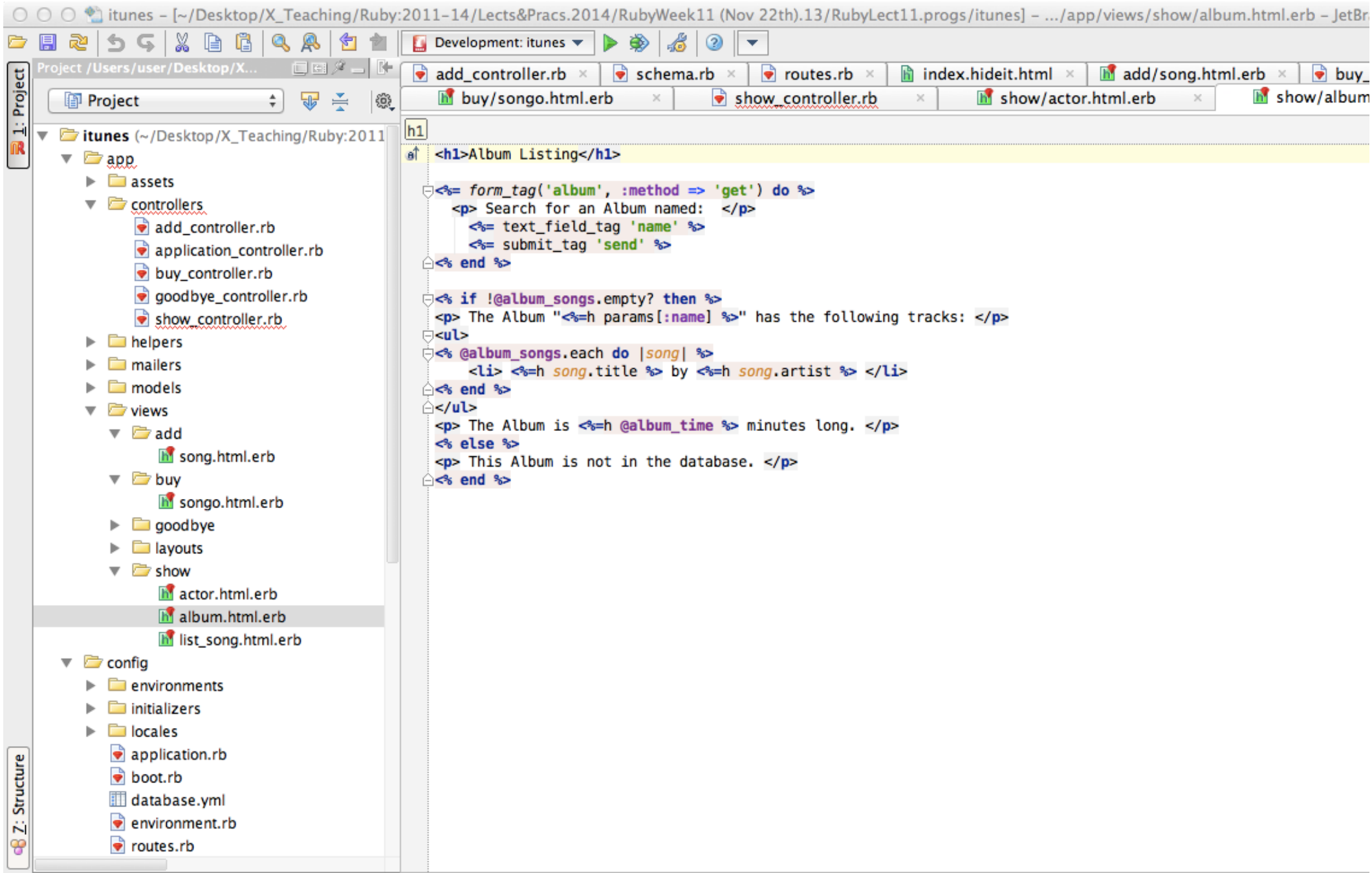
Displays...



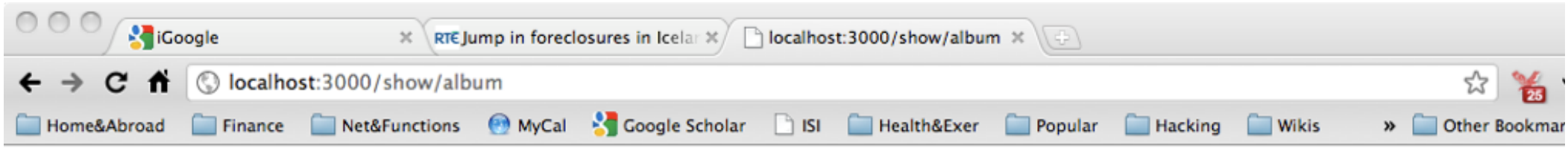
Actor Listing

- Mark the Admin, likes reggae : 1
- Bloggie the Buyer, likes punk : 2
- Mark the Admin, likes reggae : 3
- Bloggie the Buyer, likes punk : 4
- Mark the Admin, likes reggae : 5
- Bloggie the Buyer, likes punk : 6

show/album View



Displays...



Album Listing

Search for an Album named:

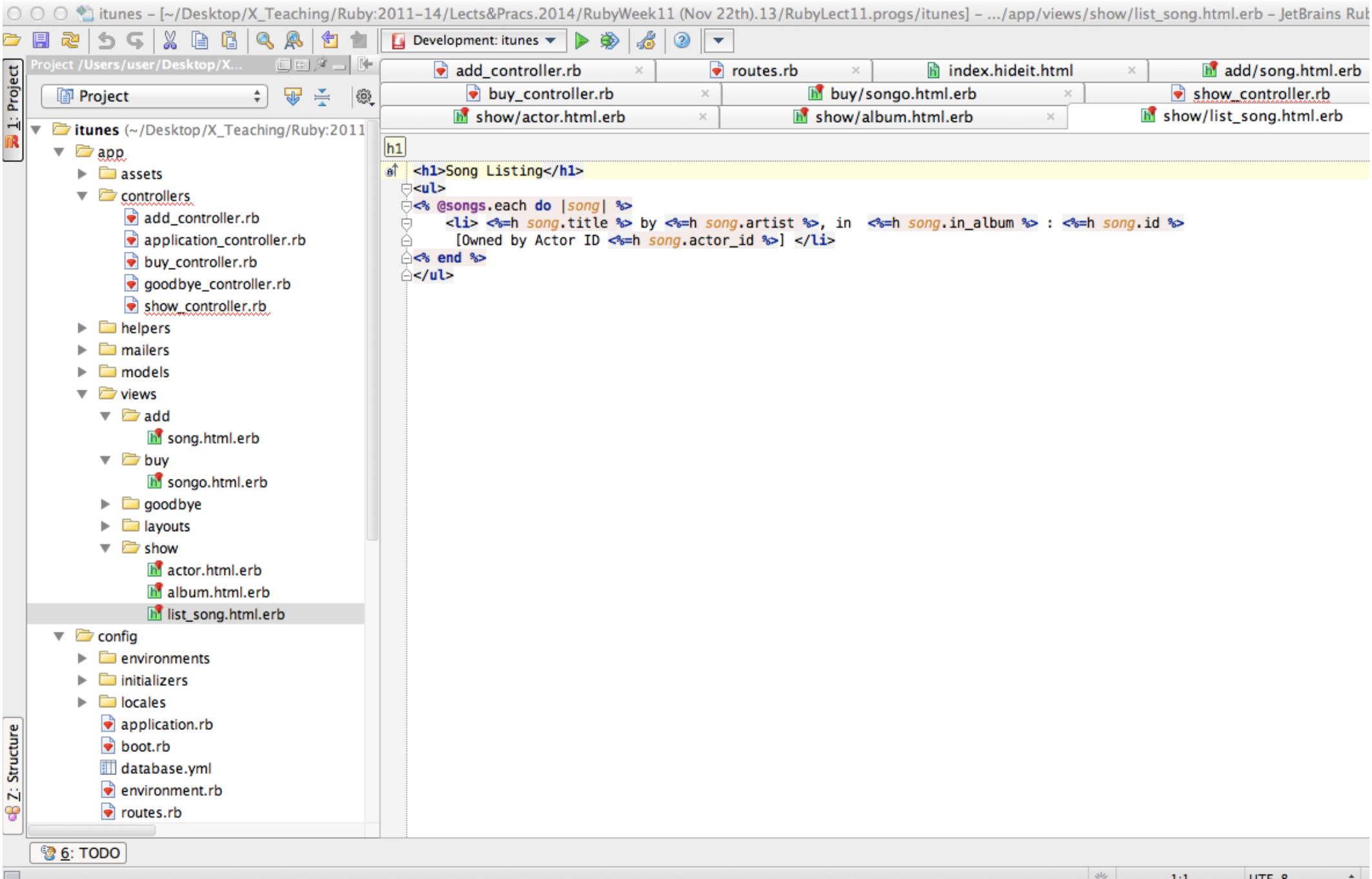
send

The Album "Led Zep IV" has the following tracks:

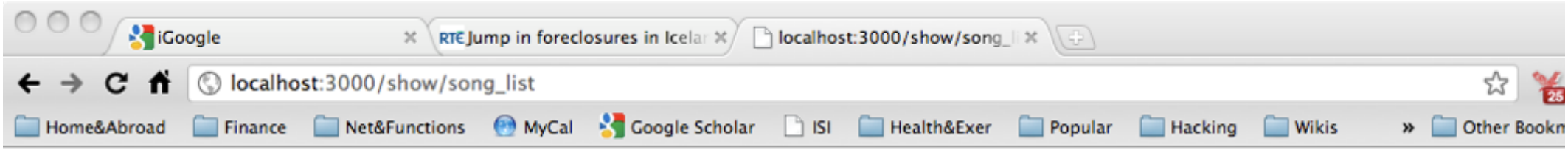
- Stairway to Heaven by Led Zep

The Album is 4 minutes long.

show/song_list View



Displays...



Song Listing

- Stairway to Heaven by Led Zep, in Lex Zep IV : 1 [Owned by Actor ID 1]
- Holiday by Seex Pistols, in God Save... : 2 [Owned by Actor ID 2]
- Forest by The Cure, in Seventheen Seconds : 3 [Owned by Actor ID]
- The end by The cure, in Three Imaginary Boys : 4 [Owned by Actor ID]
- Holiday by Seex Pistols, in God Save... : 5 [Owned by Actor ID 2183125080]
- Holiday by Seex Pistols, in God Save... : 6 [Owned by Actor ID 2182658940]
- Holiday by Seex Pistols, in God Save... : 7 [Owned by Actor ID 2182305980]
- Holiday by Seex Pistols, in God Save... : 8 [Owned by Actor ID 2182928100]
- Holiday by Seex Pistols, in God Save... : 9 [Owned by Actor ID 2182511280]
- Holiday by Seex Pistols, in God Save... : 10 [Owned by Actor ID 2182153620]
- Holiday by Seex Pistols, in God Save... : 11 [Owned by Actor ID 2181724300]
- Holiday by Seex Pistols, in God Save... : 12 [Owned by Actor ID 2183213180]
- Stairway to Heaven by Led Zep, in Led Zep IV : 13 [Owned by Actor ID 1]
- Holiday by Sex Pistols, in God Save... : 14 [Owned by Actor ID 2]

Oops...I forgot a controller method

if you forgot to input a method for a controller; say buy song

then it is best to re-generate it...(or use destroy then...)

$ rails generate controller buy song
will prompt you for Y to overwrite your file and check to see whether you want to overwrite view too

Don’t be tempted to do it yourself...by creating files..

Oops...I forgot to put in..

```
$ rails generate controller buy  song
      exists  app/controllers/
      exists  app/helpers/
      exists  app/views/buy
      exists  test/functional/
      exists  test/unit/helpers/
overwrite app/controllers/buy_controller.rb? (enter "h" for help) [Ynaqdh] Y
       force  app/controllers/buy_controller.rb
   identical  test/functional/buy_controller_test.rb
   identical  app/helpers/buy_helper.rb
   identical  test/unit/helpers/buy_helper_test.rb
overwrite app/views/show/song.html.erb? (enter "h" for help) [Ynaqdh] n
        create  app/views/show/song.html.erb
```

Oops...I forgot to put in..db

you can rollback and re-migrate everything with edits

or just create a new migrate, to fix the problem

better practice, because it is more explicit for dbs

Commands in migrations

`create_table(name, options)` : creates a table

`drop_table(name)`: Drops the table called `name`.

`rename_table(old_name, new_name)`: Renames the table called `old_name` to `new_name`.

`add_column(table_name, column_name, type, options)`: Adds a new column

`rename_column(table_name, column_name, new_column_name)`: Renames a column

`change_column(table_name, column_name, type, options)`: Changes the column

`remove_column(table_name, column_name)`: Removes the column named

`add_index(table_name, column_names, options)`: Adds a new index with the name of the column. Other options include `:name` and `:unique` (e.g. `{ :name => "users_name_index", :unique => true }`).

`remove_index(table_name, index_name)`: Removes the index specified by `index_name`.

Migrations to fix things...

Not all migrations change the schema. Some just fix the data:

```
class RemoveEmptyTags < ActiveRecord::Migration
  def self.up
    Tag.find(:all).each { |tag| tag.destroy if
          tag.pages.empty? }
  end

  def self.down
    # not much we can do to restore deleted data
    raise ActiveRecord::IrreversibleMigration,
  "Can't recover the deleted tags"
  end
end
```

Others remove columns when they migrate up instead of down:

```
class RemoveUnnecessaryItemAttributes < ActiveRecord::Migration
  def self.up
    remove_column :items, :incomplete_items_count
    remove_column :items, :completed_items_count
  end

  def self.down
    add_column :items, :incomplete_items_count
    add_column :items, :completed_items_count
  end
end
```

Part B(vi):

Seeding the DB

Seed.rb

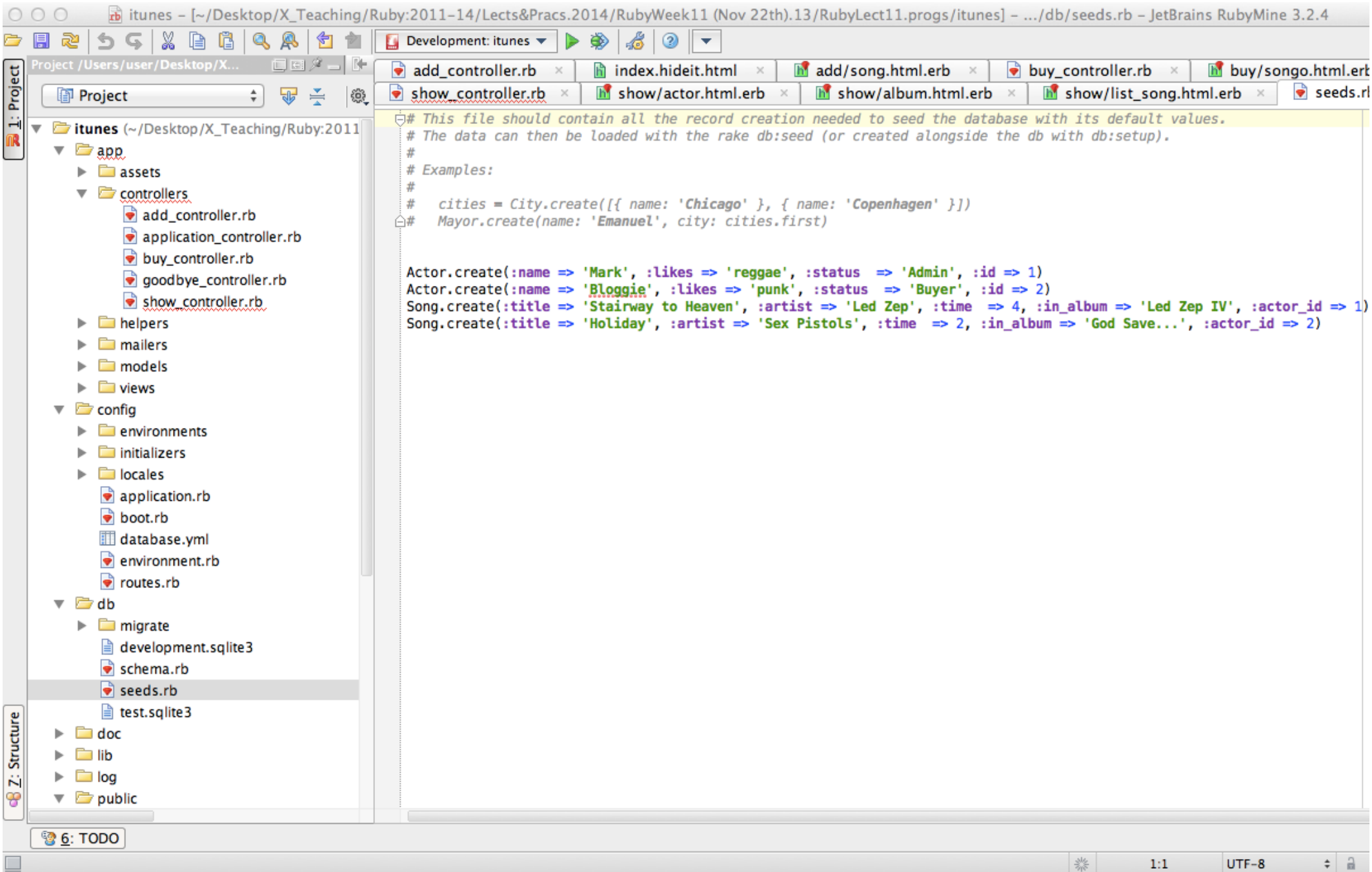
this file which sits in the db directory is a clean and quick way of seeding a db with initial records

it is clean ‘cos you just write object descriptions, like you normally do in Ruby

there is an explicit **rake** command to load

it looks like this...

Write **seeds.rb** like this...



rake db:seed

allows you to seed the database with a few initial instances

you just write them as object instance creations

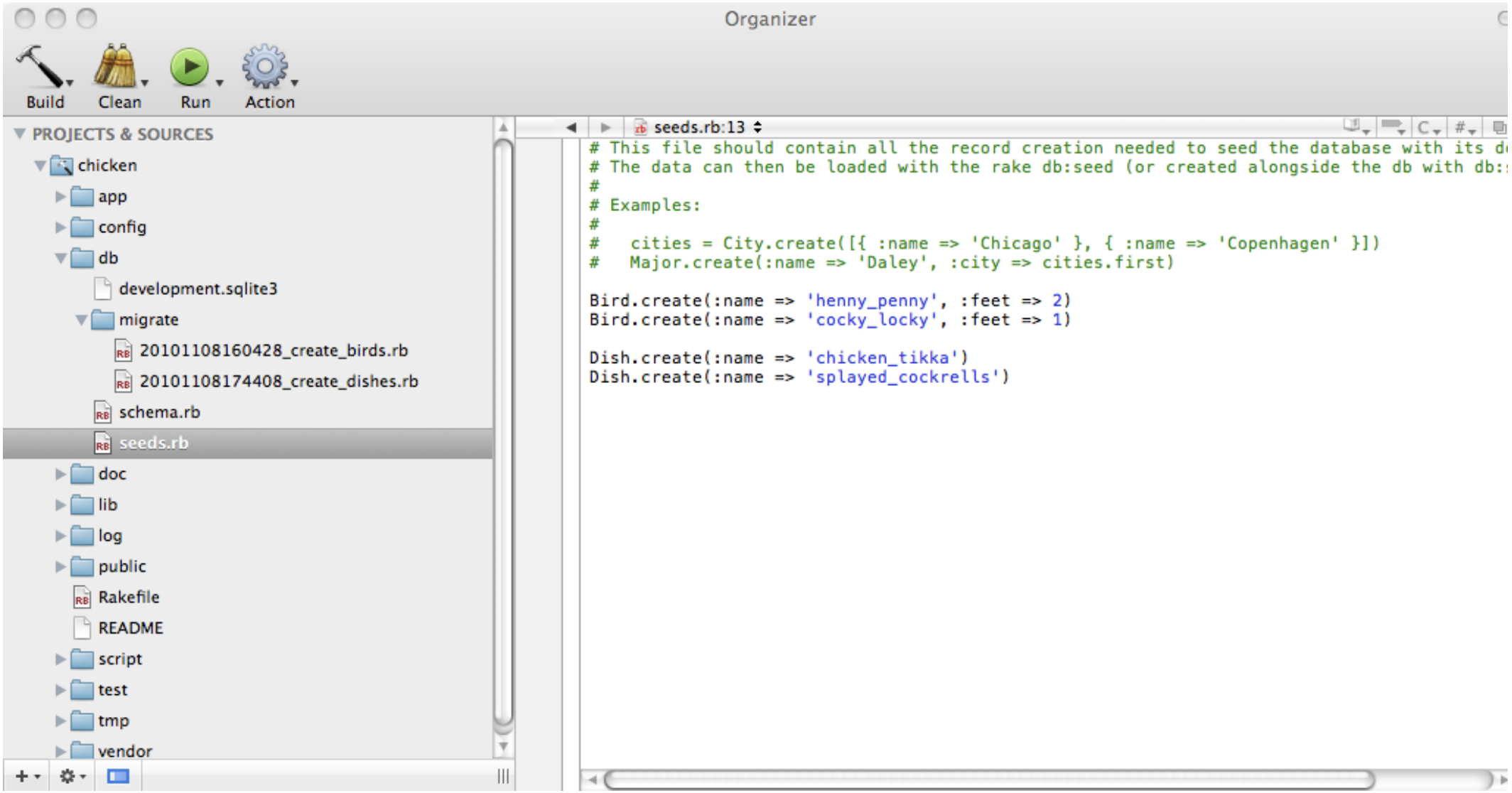
then you run **rake db:seed** to load them in

Models recipe... **REM:**

1) set up models
2) check and/or edit migration files
3) include db relations (if required) in model files (eg actor.rb)
4) edit your seed files (will return to this later)
5) do **$ rake db:migrate**
6) do **$ rake db:seed**

...then you can move to your browser and start running...

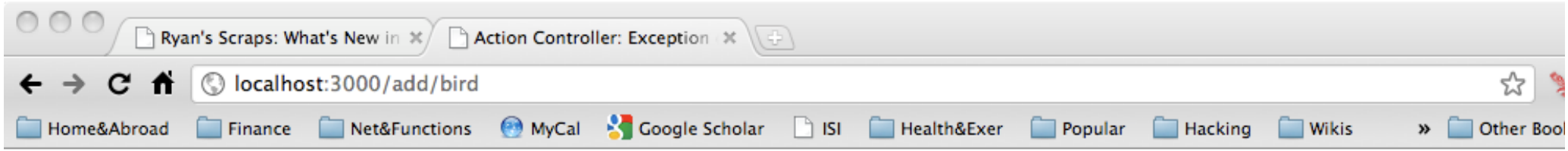
another **seeds.rb** eg...



then load them in...

```
$ rake db:seed
(in /Users/mkeane/Desktop/chicken)
rake aborted!
Could not find table 'birds'
$ rake db:migrate
(in /Users/mkeane/Desktop/chicken)
==  CreateBirds: migrating ==-- create_table(:birds)
   -> 0.0018s
==  CreateBirds: migrated (0.0023s)
==  CreateDishes: migrating
-- create_table(:dishes)
   -> 0.0013s
==  CreateDishes: migrated (0.0015s)
$ rake db:seed
(in /Users/mkeane/Desktop/chicken)
```

When your **schema.rb** is wrong...



ActiveRecord::UnknownAttributeError in AddController#bird

```
unknown attribute: name
```

RAILS_ROOT: /Users/mkeane/Desktop/chicken

Application Trace | Framework Trace | Full Trace

```
/Library/Ruby/Gems/1.8/gems/activerecord-2.3.8/lib/active_record/base.rb:2906:in `assign_attributes'
/Library/Ruby/Gems/1.8/gems/activerecord-2.3.8/lib/active_record/base.rb:2902:in `each'
/Library/Ruby/Gems/1.8/gems/activerecord-2.3.8/lib/active_record/base.rb:2902:in `assign_attributes'
/Library/Ruby/Gems/1.8/gems/activerecord-2.3.8/lib/active_record/base.rb:2775:in `attributes='
/Library/Ruby/Gems/1.8/gems/activerecord-2.3.8/lib/active_record/base.rb:2473:in `initialize'
/Library/Ruby/Gems/1.8/gems/activerecord-2.3.8/lib/active_record/base.rb:724:in `new'
/Library/Ruby/Gems/1.8/gems/activerecord-2.3.8/lib/active_record/base.rb:724:in `create'
/Users/mkeane/Desktop/chicken/app/controllers/add_controller.rb:3:in `bird'
/Library/Ruby/Gems/1.8/gems/actionpack-2.3.8/lib/action_controller/base.rb:1331:in `send'
/Library/Ruby/Gems/1.8/gems/actionpack-2.3.8/lib/action_controller/base.rb:1331:in `perform_action_without_filters'
/Library/Ruby/Gems/1.8/gems/actionpack-2.3.8/lib/action_controller/filters.rb:617:in `call_filters'
/Library/Ruby/Gems/1.8/gems/actionpack-2.3.8/lib/action_controller/filters.rb:610:in `perform_action_without_benchmark'
/Library/Ruby/Gems/1.8/gems/actionpack-2.3.8/lib/action_controller/benchmarking.rb:68:in `perform_action_without_rescue'
/Library/Ruby/Gems/1.8/gems/activesupport-2.3.8/lib/active_support/core_ext/benchmark.rb:17:in `ms'
/Library/Ruby/Gems/1.8/gems/activesupport-2.3.8/lib/active_support/core_ext/benchmark.rb:17:in `ms'
/Library/Ruby/Gems/1.8/gems/actionpack-2.3.8/lib/action_controller/benchmarking.rb:68:in `perform_action_without_rescue'
/Library/Ruby/Gems/1.8/gems/actionpack-2.3.8/lib/action_controller/rescue.rb:160:in `perform_action_without_flash'
/Library/Ruby/Gems/1.8/gems/actionpack-2.3.8/lib/action_controller/flash.rb:151:in `perform_action'
/Library/Ruby/Gems/1.8/gems/actionpack-2.3.8/lib/action_controller/base.rb:532:in `send'
/Library/Ruby/Gems/1.8/gems/actionpack-2.3.8/lib/action_controller/base.rb:532:in `process_without_filters'
/Library/Ruby/Gems/1.8/gems/actionpack-2.3.8/lib/action_controller/filters.rb:606:in `process'
/Library/Ruby/Gems/1.8/gems/actionpack-2.3.8/lib/action_controller/base.rb:391:in `process'
```

or this in command line...

$ rake db:seed

(in /Users/mkeane/Desktop/chicken)

rake aborted!

unknown attribute: feet

(See full trace by running task with --trace)

Bundle...

$ bundle exec rake db:migrate

explicitly use the version of rake in the gemfile

(or update your gem file version)

Running old apps...

$ cd itunes2

$ rails server

Could not find rake-10.0.2 in any of the sources

Run `bundle install` to install missing gems.

$ bundle install

Fetching gem metadata from http://rubygems.org/.....

Fetching gem metadata from http://rubygems.org/..

Enter password to install the bundled RubyGems to system:

Installing rake (10.0.2)

Installing i18n (0.6.1) ....

Part C:

Using **rails console** to talk to the db

Console interact with db

you can get a interactive console to query the database with ruby commands, uses irb

in a command line of the app directory do :

$ **rails console**

Console interact with db

```
MobileMac:itunes2 user$ rails console
config.eager_load is set to nil. Please update your config/environments/*.rb files accordingly:

  * development - set it to false
  * test - set it to false (unless you use a tool that preloads your test environment)
  * production - set it to true

Loading development environment (Rails 4.1.6)
irb(main):001:0> Song.all
  Song Load (1.2ms)  SELECT "songs".* FROM "songs"
=> #<ActiveRecord::Relation [#<Song id: 1, title: "Stairway to Heaven", in_album: "Led Zep IV", time: 4, artist: "Led Zep",
 actor_id: 1, created_at: "2011-10-25 15:27:37", updated_at: "2011-10-25 15:27:37">, #<Song id: 2, title: "Holiday", in_alb
um: "God Save...", time: 2, artist: "Sex Pistols", actor_id: 2, created_at: "2011-10-25 15:27:37", updated_at: "2011-10-25
15:27:37">, #<Song id: 3, title: "Gloria", in_album: "Horses", time: 3, artist: "Patti Smith", actor_id: nil, created_at: "
2011-10-25 15:42:57", updated_at: "2011-10-25 15:42:57">, #<Song id: 4, title: "Gloria", in_album: "Horses", time: 3, artis
t: "Patti Smith", actor_id: 1, created_at: "2011-10-26 10:47:40", updated_at: "2011-10-26 10:47:40">, #<Song id: 5, title:
"Gloria", in_album: "Horses", time: 3, artist: "Patti Smith", actor_id: 1, created_at: "2011-10-26 10:47:48", updated_at: "
2011-10-26 10:47:48">, #<Song id: 6, title: "Bibble", in_album: "Babble", time: 4, artist: "Bobble", actor_id: nil, created
_at: "2012-11-19 15:10:45", updated_at: "2012-11-19 15:10:45">]>
irb(main):002:0>
```

Part D:

link_to has a third possible arg...

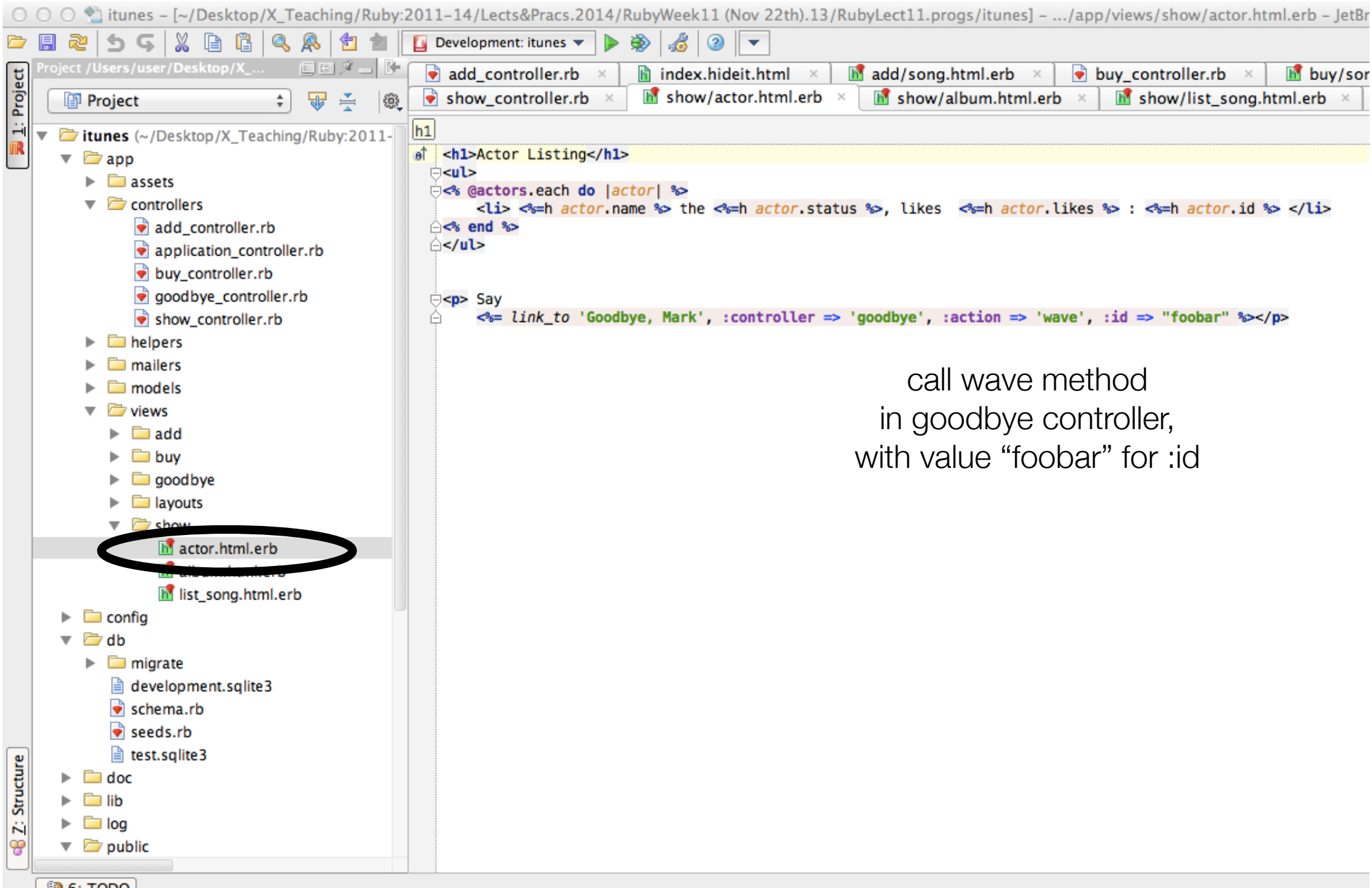
What we will do...

we will have a **link_to** in show/actor page, that will call goodbye/wave AND pass a value to that controller

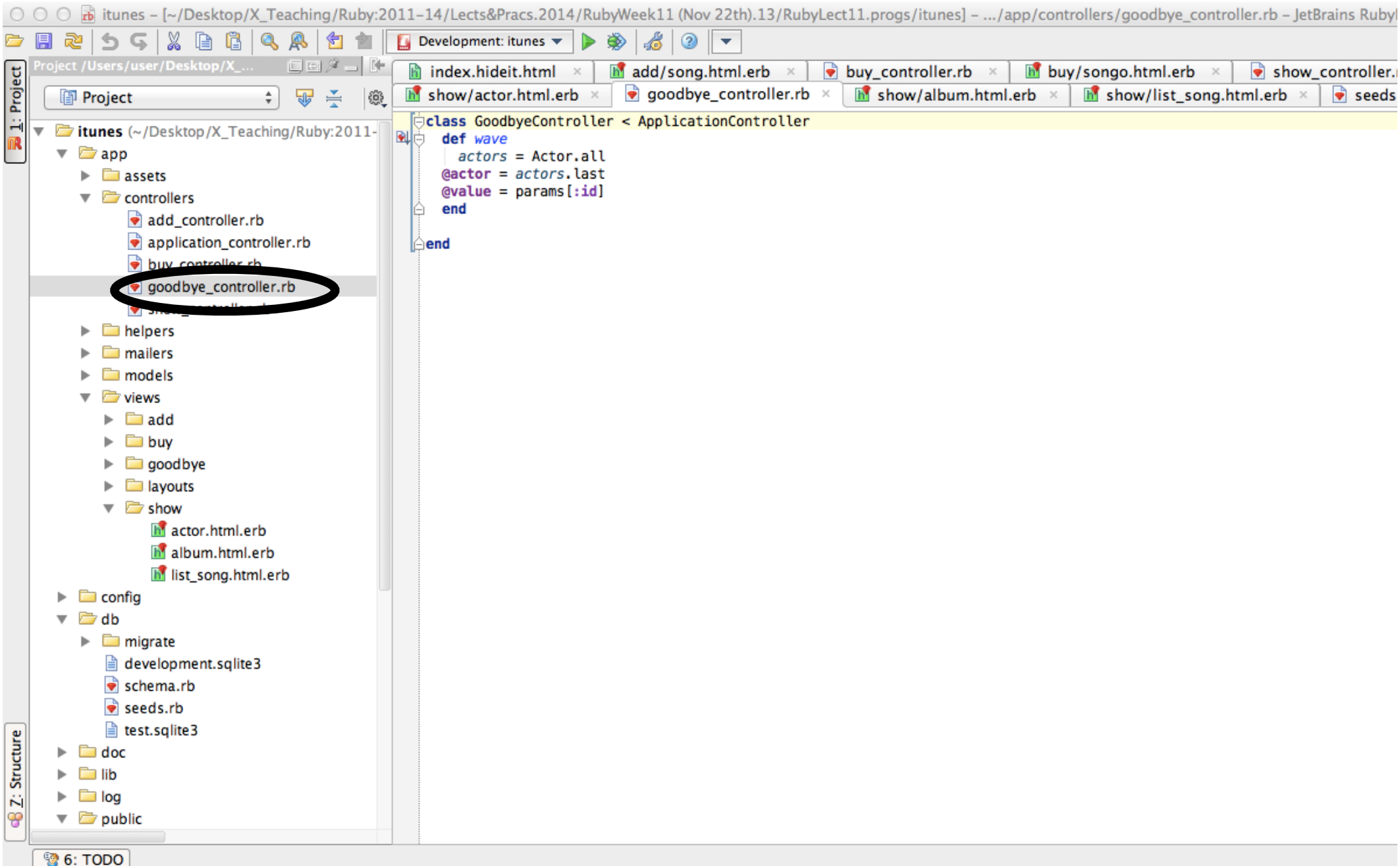
we will then use that value in the page-view that controller renders; to just see it

could do the same thing to pass parameters to a method

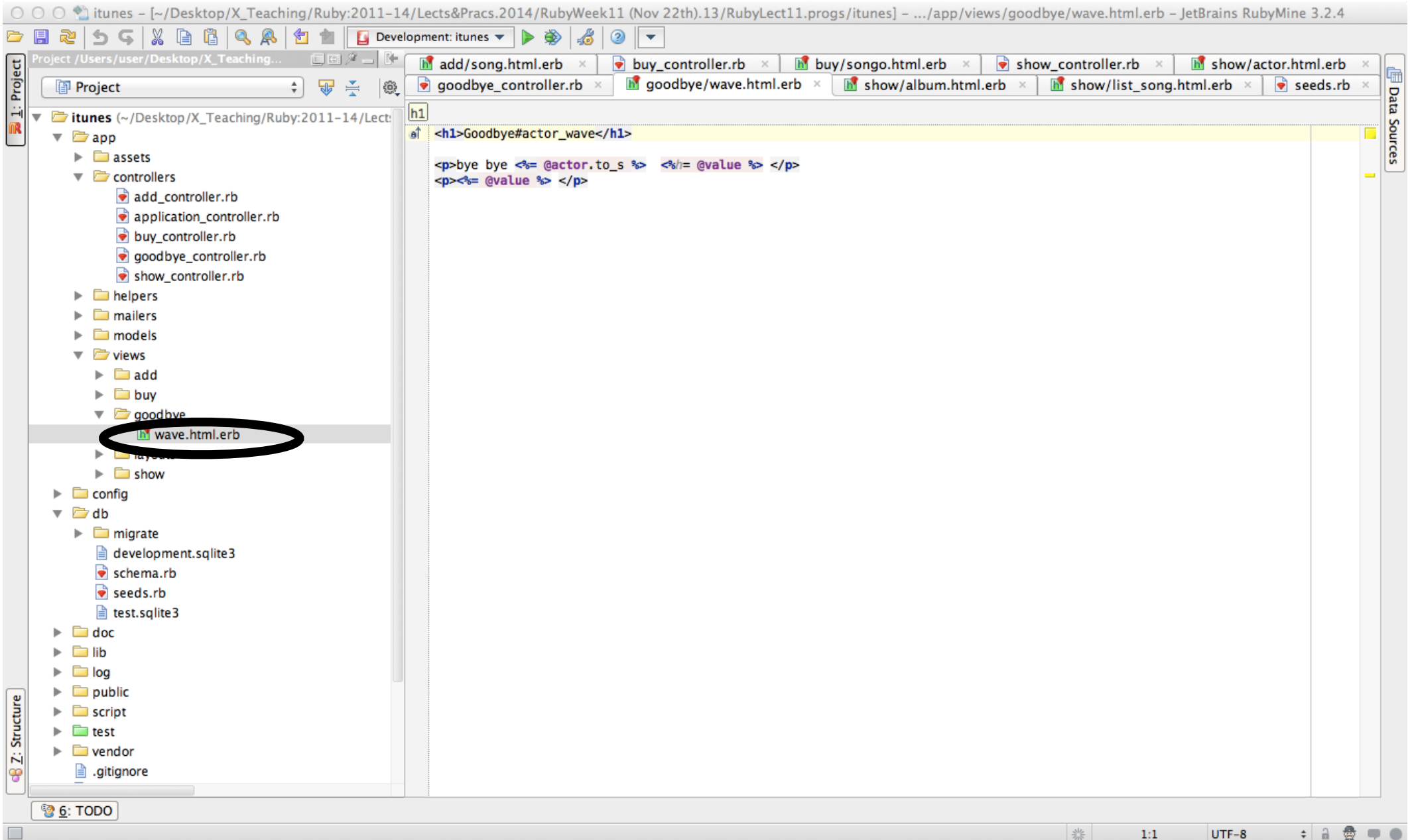
add link_to to actor...



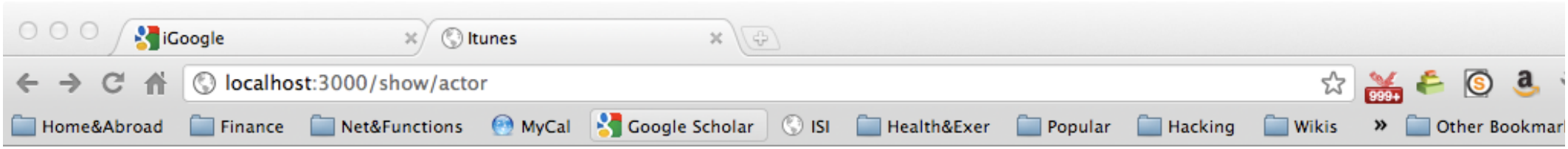
add **goodbye controller**



add **goodbye view**



show/actor now looks...

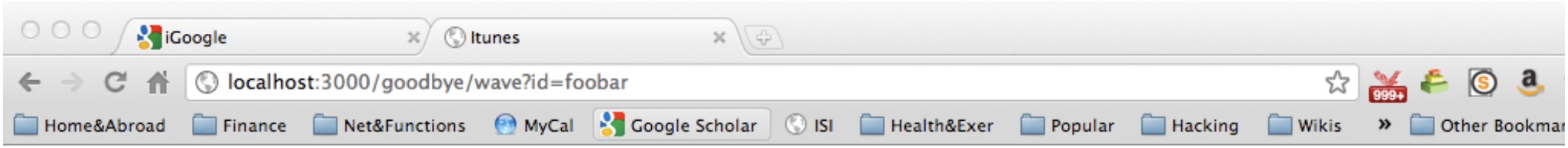


Actor Listing

- Mark the Admin, likes reggae : 1
- Bloggie the Buyer, likes punk : 2

Say Goodbye, Mark

when you click you get...



Goodbye#actor_wave

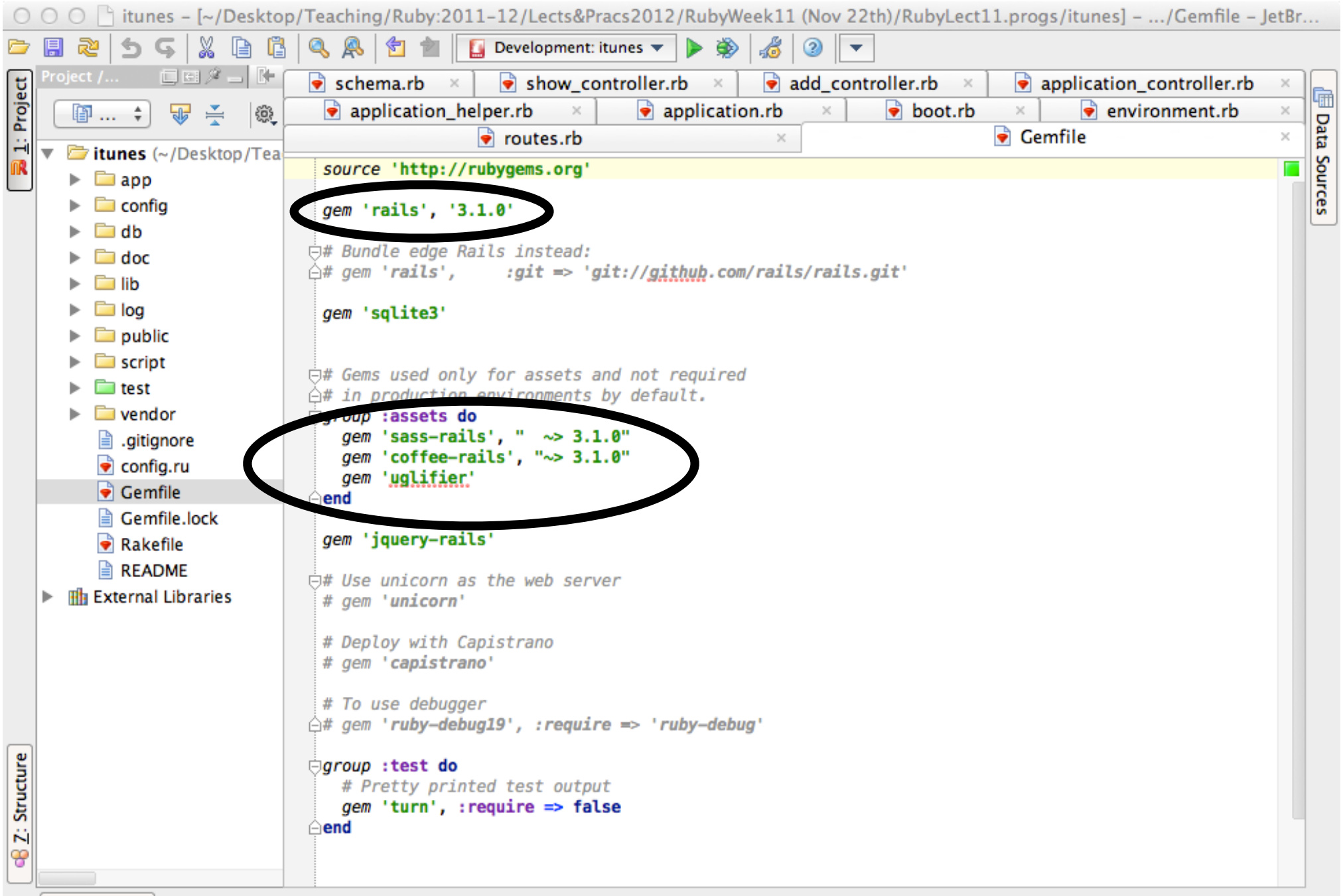
bye bye #<Actor:0x007ffa7cf49998>

foobar

Part E:

Time has passed...how do I run my site with a new rails and gems?

Look at old bundle...

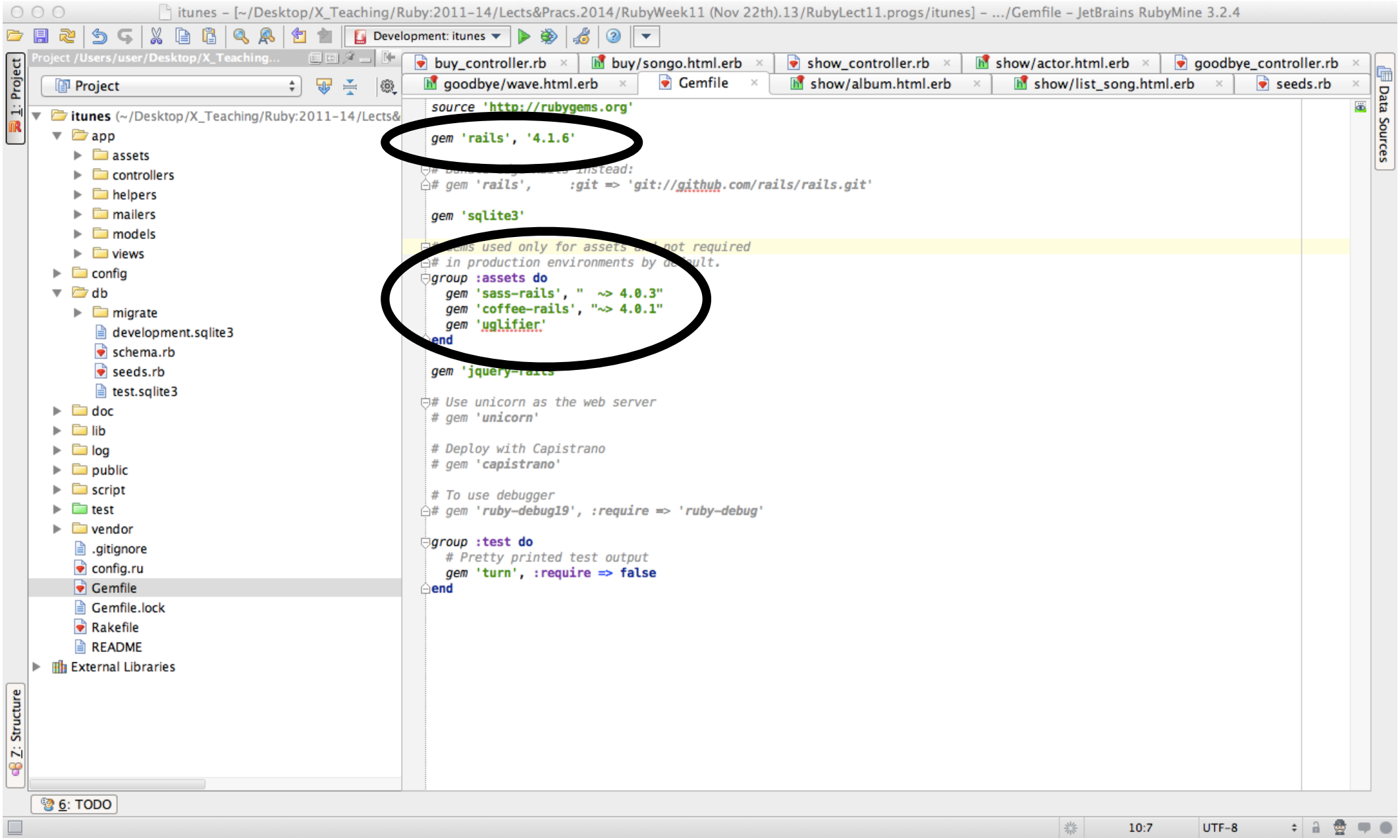


Do **bundle update**

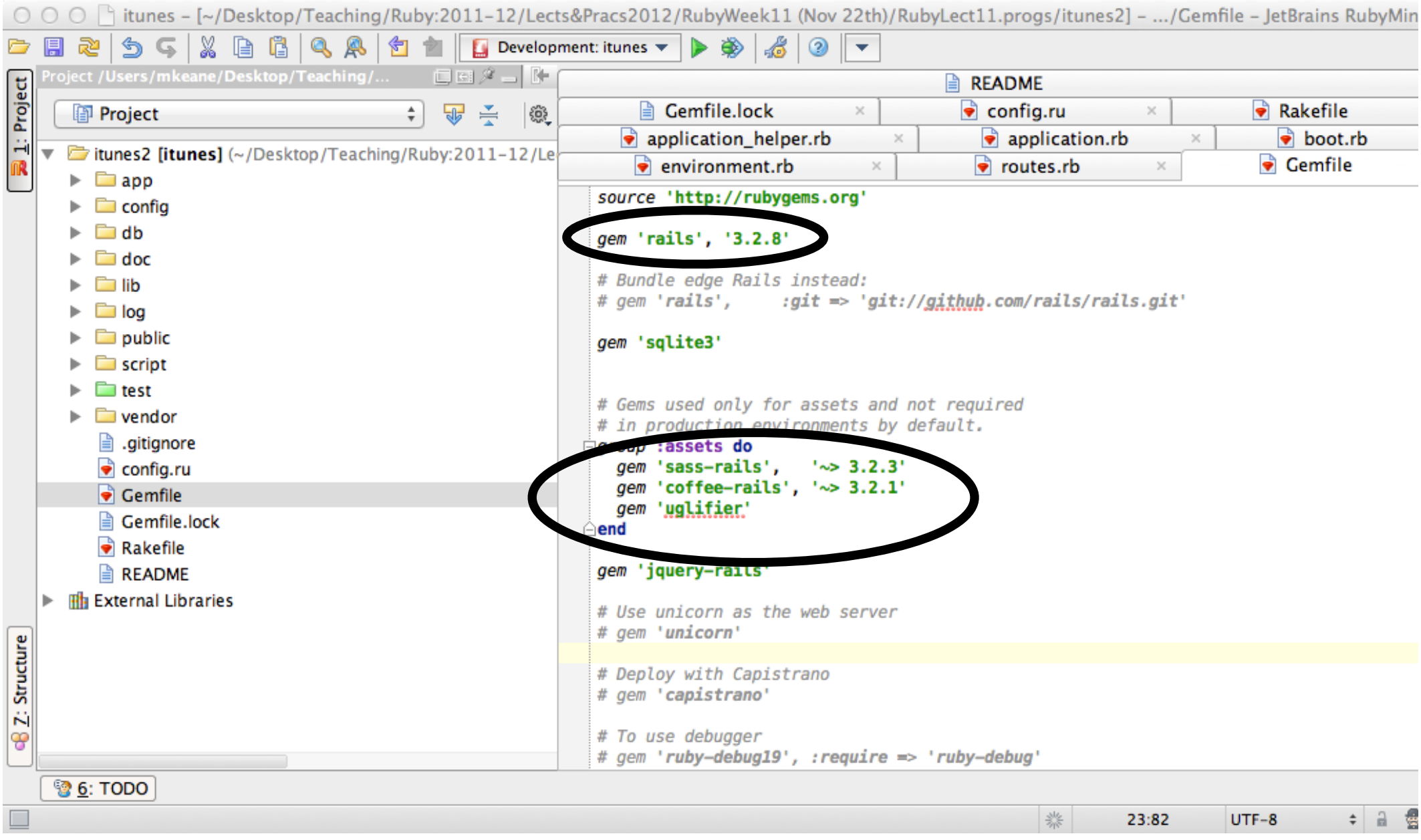
```
Marks-MacBook-Pro-2:itunes2 mkeane$ bundle update
Fetching gem metadata from http://rubygems.org/.........
Fetching gem metadata from http://rubygems.org/..
Bundler could not find compatible versions for gem "railties":
  In Gemfile:
    coffee-rails (~> 3.1.0) ruby depends on
      railties (~> 3.1.0.rc1) ruby

    rails (= 3.2.8) ruby depends on
      railties (3.2.8)
```

Ukkk ! Need to update gems

Change old **versions** to new...



Change old ***versions*** to new...



```
Marks-MacBook-Pro-2:itunes2 mkeane$ bundle update
Fetching gem metadata from http://rubygems.org/.........
Fetching gem metadata from http://rubygems.org/..
Enter your password to install the bundled RubyGems to your system:
Using rake (10.0.2)
Using i18n (0.6.1)
Using multi_json (1.3.7)
Using activesupport (3.2.8)
Using builder (3.0.4)
Using activemodel (3.2.8)
Using erubis (2.7.0)
Using journey (1.0.4)
Using rack (1.4.1)
Using rack-cache (1.2)
Using rack-test (0.6.2)
Using hike (1.2.1)
Using tilt (1.3.3)
Using sprockets (2.1.3)
Using actionpack (3.2.8)
Using mime-types (1.19)
Using polyglot (0.3.3)
Using treetop (1.4.12)
Using mail (2.4.4)
Using actionmailer (3.2.8)
Using arel (3.0.2)
Using tzinfo (0.3.35)
Using activerecord (3.2.8)
Using activeresource (3.2.8)
Using ansi (1.4.3)
Using bundler (1.2.1)
Using coffee-script-source (1.4.0)
Using execjs (1.4.0)
Using coffee-script (2.2.
Using rack-ssl (1.3.2)
Using json (1.7.5)
Using rdoc (3.12)
Using thor (0.16.0)
Using railties (3.2.8)
Using coffee-rails (3.2.2
Using jquery-rails (2.1.3
Using rails (3.2.8)
Using sass (3.2.3)
Using sass-rails (3.2.5)
Using sqlite3 (1.3.6)
Using turn (0.9.6)
Using uglifier (1.3.0)
Your bundle is updated! Use `bundle show [gemname]` to see where a bundled gem is inst
alled.
Marks-MacBook-Pro-2:itunes2 mkeane$
```

Now, our rails site is running with new gems...but note you need to still test it, to see is all ok (see itunes2)

Drinks ?...


CHATEAU COUTET
PREMIER CRU CLASSÉ
2005
SAUTERNES-BARSAC
APPELLATION BARSAC CONTRÔLÉE
PRODUCE OF FRANCE