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

$ rails generate model Bird name:string

migration_file only created

$ 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

chicken – [~/Desktop/chicken] – .../db/migrate/20130916185447_create_birds.rb – 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
    - views
  - bin
  - db
    - migrate
      - 20130916185447_create_birds.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 | seeds.rb

```
class CreateBirds < ActiveRecord::Migration
  def change
    create_table :birds do |t|
      t.string :name

      t.timestamps
    end
  end
end
```

methods for defining and destroying tables

folder and file created

rake db:migrate does...

chicken – [~/Desktop/chicken] – .../db/schema.rb – 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
    - views
  - bin
  - config
  - db
    - migrate
      - 20130916185447_create_birds.r
    - development.sqlite3
    - schema.rb
    - seeds.rb
  - lib
    - assets
    - tasks
  - log
  - public
  - test
  - vendor
  - .gitignore
  - config.ru

bird.rb | 20130916185447_create_birds.rb | schema.rb | seeds.rb

```ruby
# 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 that you check this file into your version control system.

ActiveRecord::Schema.define(version: 20130916185447) do

  create_table "birds", force: true do |t|
    t.string   "name"
    t.datetime "created_at"
    t.datetime "updated_at"
  end

end
```

schema.rb created

say, we add feet...

chicken – [~/Desktop/chicken] – .../db/migrate/20130916185447_create_birds.rb – 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
    - views
  - bin
  - config
  - db
    - migrate
      - 20130916185447_create_birds.r
    - development.sqlite3
    - schema.rb
    - seeds.rb
  - lib
    - assets
    - tasks
  - log
  - public
  - test
  - vendor
  - .gitignore
  - config.ru
  - Gemfile

bird.rb | 20130916185447_create_birds.rb | schema.rb | seeds.rb

```
class CreateBirds < ActiveRecord::Migration
  def change
    create_table :birds do |t|
      t.string :name
      t.integer :feet
      t.timestamps
    end
  end
end
```

we add feet field

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

chicken – [~/Desktop/chicken] – .../db/schema.rb – 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
    - views
  - bin
  - config
  - db
    - migrate
      - 20130916185447_create_birds.r
    - development.sqlite3
    - schema.rb
    - seeds.rb
  - lib
    - assets
    - tasks
  - log
  - public
  - test
  - vendor
  - .gitignore
  - config.ru
  - Gemfile

bird.rb | 20130916185447_create_birds.rb | schema.rb | seeds.rb

```ruby
# 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 that you check this file into your version control system.

ActiveRecord::Schema.define(version: 20130916185447) do

  create_table "birds", force: true do |t|
    t.string   "name"
    t.datetime "created_at"
    t.datetime "updated_at"
  end

end
```

!!!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...

chicken – [~/Desktop/chicken] – .../db/schema.rb – 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
    - views
  - bin
  - config
  - db
    - migrate
    - development.sqlite3
    - schema.rb
    - seeds.rb
  - lib
    - assets
    - tasks
  - log
  - public
  - test
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock

bird.rb | 20130916185447_create_birds.rb | schema.rb | seeds.rb

```
# 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 that you check this file into your version control system.

ActiveRecord::Schema.define(version: 0) do

end
```

schema.rb
is emptied

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

Organizer

Build Clean Run Action

PROJECTS & SOURCES

- chicken
  - app
  - config
  - config.ru
  - db
    - development.sqlite3
    - migrate
      - 20111024124255_create_birds.rb
    - schema.rb
    - seeds.rb
  - doc
  - Gemfile
  - Gemfile.lock
  - lib
  - log
  - public
  - Rakefile
  - README
  - script
  - test
  - tmp
  - vendor

schema.rb:24

```
# 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 => 20111024124255) do

  create_table "birds", :force => true do |t|
    t.string   "name"
    t.integer  "feet"
    t.datetime "created_at"
    t.datetime "updated_at"
  end

end
```

aaaaah...
feet are now mentioned

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...

chicken – [~/Desktop/chicken] – .../db/migrate/20130916191319_create_dishes.rb – 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
      - 20130916185447_create_birds.r
      - 20130916191319_create_dishes.
    - development.sqlite3
    - schema.rb
    - seeds.rb
  - lib
    - assets
    - tasks
  - log
  - public
  - test
  - vendor
  - .gitignore

bird.rb | 20130916185447_create_birds.rb | schema.rb | 20130916191319_create_dis

```
class CreateDishes < ActiveRecord::Migration
  def change
    create_table :dishes do |t|
      t.string :name
      t.integer :size

      t.timestamps
    end
  end
end
```

$ rails generate model Dish name:string

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

chicken – [~/Desktop/chicken] – .../db/schema.rb – 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

bird.rb | 20130916185447_create_birds.rb | schema.rb | 20130916191319_create_dish

```ruby
# 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 that you check this file into your version control system.

ActiveRecord::Schema.define(version: 20130916191319) do

  create_table "birds", force: true do |t|
    t.string   "name"
    t.integer  "feet"
    t.datetime "created_at"
    t.datetime "updated_at"
  end

  create_table "dishes", force: true do |t|
    t.string   "name"
    t.integer  "size"
    t.datetime "created_at"
    t.datetime "updated_at"
  end

end
```

both tables are mentioned

then $ rake db:rollback

chicken – [~/Desktop/chicken] – .../db/schema.rb – 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
      - 20130916185447_create_birds.r
      - 20130916191319_create_dishes.
    - development.sqlite3
    - schema.rb
    - seeds.rb
  - lib
    - assets
    - tasks
  - log
  - public
  - test
  - vendor
  - .gitignore

bird.rb | 20130916185447_create_birds.rb | schema.rb | 20130916191319_create_dis

```ruby
# 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 that you check this file into your version control system.

ActiveRecord::Schema.define(version: 20130916185447) do

  create_table "birds", force: true do |t|
    t.string   "name"
    t.integer  "feet"
    t.datetime "created_at"
    t.datetime "updated_at"
  end

end
```

only dishes disappears

then **rake db:rollback** again

chicken – [~/Desktop/chicken] – .../db/schema.rb – 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

bird.rb | 20130916185447_create_birds.rb | schema.rb | 20130916191319_create_dish

```ruby
# 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 that you check this file into your version control system.

ActiveRecord::Schema.define(version: 0) do

end
```

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
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help 36% Fri 17:12

Gemfile - chicken - [~/Dropbox/X_Teaching/Ruby 2013-16/A_Lects&Pracs.2016/RubyWeek11 (Nov 21th).15/RubyLect11.progs/chicken]

Development: chicken

chicken > Gemfile

Project

- chicken ~/Dropbox/X_Teachir
  - app
    - assets
    - controllers
    - helpers
    - mailers
    - models
      - concerns
      - .keep
      - bird.rb
      - dish.rb
    - views
  - bin
  - config
  - db
    - migrate
      - 20130916185447_cre
      - 20130916191319_cre
    - development.sqlite3
    - schema.rb
    - seeds.rb
  - lib
    - assets
    - tasks
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru

seeds.rb | Gemfile | 20130916191319_create_dishes.rb | 20130916185447_create_birds.rb | application.html.erb

```
source 'https://rubygems.org'

# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
gem 'rails', '~> 5.0.0', '>= 5.0.0.1'
# Use sqlite3 as the database for Active Record
gem 'sqlite3'
# Use Puma as the app server
gem 'puma', '~> 3.0'
# Use SCSS for stylesheets
gem 'sass-rails', '~> 5.0'
# Use Uglifier as compressor for JavaScript assets
gem 'uglifier', '>= 1.3.0'
# Use CoffeeScript for .coffee assets and views
gem 'coffee-rails', '~> 4.2'
# See https://github.com/rails/execjs#readme for more supported runtimes
# gem 'therubyracer', platforms: :ruby

# Use jquery as the JavaScript library
gem 'jquery-rails'
# Turbolinks makes navigating your web application faster. Read more: https://github.com/turbolinks/turbolinks
gem 'turbolinks', '~> 5'
# Build JSON APIs with ease. Read more: https://github.com/rails/jbuilder
gem 'jbuilder', '~> 2.5'
# Use Redis adapter to run Action Cable in production
# gem 'redis', '~> 3.0'
# Use ActiveModel has_secure_password
# gem 'bcrypt', '~> 3.1.7'

# Use Capistrano for deployment
# gem 'capistrano-rails', group: :development

group :development, :test do
  # Call 'byebug' anywhere in the code to stop execution and get a debugger console
  gem 'byebug', platform: :mri
end

group :development do
  # Access an IRB console on exception pages or by using <%= console %> anywhere in the code.
```

Gemfile...

Terminal

```
MacBook-Air-4:chicken user$
```

RubyMine Gem Manager
RubyMine has detected that some of the ge
required for 'chicken' are not installed. Insta

6: TODO | Terminal

Gemfile.lock - chicken - [~/Dropbox/X_Teaching/Ruby 2013-16/A_Lects&Pracs.2016/RubyWeek11 (Nov 21th).15/RubyLect11.progs/chicken]

Development: chicken

chicken > Gemfile.lock

Project

- models
  - concerns
  - .keep
  - bird.rb
  - dish.rb
- views
- bin
- config
- db
  - migrate
    - 20130916185447_crea
    - 20130916191319_crea
  - development.sqlite3
  - schema.rb
  - seeds.rb
- lib
  - assets
  - tasks
- log
- public
- test
- tmp
- vendor
- .gitignore
- config.ru
- Gemfile
- Gemfile.lock
- Rakefile
- README.rdoc
- External Libraries

seeds.rb | Gemfile | Gemfile.lock | 20130916191319_create_dishes.rb | 20130916185447_create_birds.rb | application.html.erb

```
GEM
  remote: https://rubygems.org/
  specs:
    actioncable (5.0.0.1)
      actionpack (= 5.0.0.1)
      nio4r (~> 1.2)
      websocket-driver (~> 0.6.1)
    actionmailer (5.0.0.1)
      actionpack (= 5.0.0.1)
      actionview (= 5.0.0.1)
      activejob (= 5.0.0.1)
      mail (~> 2.5, >= 2.5.4)
      rails-dom-testing (~> 2.0)
    actionpack (5.0.0.1)
      actionview (= 5.0.0.1)
      activesupport (= 5.0.0.1)
      rack (~> 2.0)
      rack-test (~> 0.6.3)
      rails-dom-testing (~> 2.0)
      rails-html-sanitizer (~> 1.0, >= 1.0.2)
    actionview (5.0.0.1)
      activesupport (= 5.0.0.1)
      builder (~> 3.1)
      erubis (~> 2.7.0)
      rails-dom-testing (~> 2.0)
      rails-html-sanitizer (~> 1.0, >= 1.0.2)
    activejob (5.0.0.1)
      activesupport (= 5.0.0.1)
      globalid (>= 0.3.6)
    activemodel (5.0.0.1)
      activesupport (= 5.0.0.1)
    activerecord (5.0.0.1)
      activemodel (= 5.0.0.1)
      activesupport (= 5.0.0.1)
      arel (~> 7.0)
    activesupport (5.0.0.1)
      concurrent-ruby (~> 1.0, >= 1.0.2)
      i18n (~> 0.7)
```

bundle info...

Terminal

```
Using web-console 3.3.1
Using rails 5.0.0.1 (was 4.0.0)
Using sass-rails 5.0.6 (was 4.0.5)
Bundle updated!
MacBook-Air-4:chicken user$
```

6: TODO | Terminal

Database connections detector: Database connection properties are detected. // Discover and setup new data sources. (10 minutes ago)

113:17 LF UTF

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**

iTunes: **buy/songo**

Itunes

localhost:3000/buy/songo

Home&Abroad | Finance | Net&Functions | MyCal | Google Scholar | ISI | Health&Exer | Popular | Hacking | Wikis

Buy a Song

Give me your name:

Mark Keabe

Song you want to buy:

Holiday buy

iTunes: **show/actor**

Itunes

localhost:3000/show/actor

Home&Abroad | Finance | Net&Functions | MyCal | Google Scholar | ISI | Health&Exer | Popular | Hacking | Wikis | Other B

Actor Listing

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

Say Goodbye, Mark

iTunes: **goodbye/wave**

iTunes: **show/album**

Itunes

localhost:3000/show/album?utf8=✓&name=Led+Zep+IV&commit=send

Home&Abroad | Finance | Net&Functions | MyCal | Google Scholar | ISI | Health&Exer | Popular | Hacking | Wikis | Other Bookmar

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**

Itunes

localhost:3000/add/song

Home&Abroad | Finance | Net&Functions | MyCal | Google Scholar | ISI | Health&Exer | Popular | Hacking | Wikis | » | Other Bookmar

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**

Itunes

localhost:3000/show/list_song

Home&Abroad | Finance | Net&Functions | MyCal | Google Scholar | ISI | Health&Exer | Popular | Hacking | Wikis | » | Other Bookm

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
```

Mine File Edit View Navigate Code Refactor Run Tools VCS Window Help 33% Fri 17:31

add_controller.rb - itunes - [~/Dropbox/X_Teaching/Ruby 2013-16/A_Lects&Pracs.2016/RubyWeek11 (Nov 21th).15/RubyLect11.progs/itunes]

Development: itunes

itunes > app > controllers > add_controller.rb

Project

- itunes ~/Dropbox/X_Teaching/Ruby
  - app
    - assets
    - controllers
      - add_controller.rb
      - application_controller.rb
      - buy_controller.rb
      - goodbye_controller.rb
      - show_controller.rb
    - helpers
    - mailers
    - models
      - .gitkeep
      - actor.rb
      - song.rb
    - views
      - add
        - song.html.erb
      - buy
      - goodbye
      - layouts
      - show
  - config
  - db
  - doc
  - lib
  - log
  - public
  - script
  - test

show_controller.rb | actor.html.erb | goodbye_controller.rb | wave.html.erb | Gemfile | add_controller.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
```

Terminal

```
MacBook-Air-4:itunes user$ rails server
=> Booting Puma
=> Rails 5.0.0.1 application starting in development on http://localhost:3000
=> Run `rails server -h` for more startup options
config.eager_load is set to nil. Please update your config/environments/*.rb files accordingly:
```

6: TODO | Terminal

Database connections detector: Database connection properties are detected. // Discover and setup new data sources. (15 minutes ago) 8:46 LF UTF

Mine File Edit View Navigate Code Refactor Run Tools VCS Window Help 33% Fri 17:33

show_controller.rb - itunes - [~/Dropbox/X_Teaching/Ruby 2013-16/A_Lects&Pracs.2016/RubyWeek11 (Nov 21th).15/RubyLect11.progs/itunes]

Development: itunes

itunes > app > controllers > show_controller.rb

Project

- itunes ~/Dropbox/X_Teaching/Ruby
  - app
    - assets
    - controllers
      - add_controller.rb
      - application_controller.rb
      - buy_controller.rb
      - goodbye_controller.rb
      - show_controller.rb
    - helpers
    - mailers
    - models
      - .gitkeep
      - actor.rb
      - song.rb
    - views
      - add
        - song.html.erb
      - buy
      - goodbye
      - layouts
      - show
  - config
  - db
  - doc
  - lib
  - log
  - public
  - script
  - test

show_controller.rb | actor.html.erb | goodbye_controller.rb | wave.html.erb | Gemfile | add_controller.rb

```
class ShowController < ApplicationController

  def list_song

  end

  def album



  end

  def actor

  end

end
```

in time, we will fill
in these methods
with things we want to do

Terminal

```
MacBook-Air-4:itunes user$ rails server
=> Booting Puma
=> Rails 5.0.0.1 application starting in development on http://localhost:3000
=> Run `rails server -h` for more startup options
config.eager_load is set to nil. Please update your config/environments/*.rb files accordingly:
```

6: TODO | Terminal

Database connections detector: Database connection properties are detected. // Discover and setup new data sources. (17 minutes ago) 20:38 LF UTF

Load Page

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

andyjp85 | MarksAnalytics Sla | News forum | Action Controller: Exception ca

localhost:3000/show

Apps | Home | Current+Crucial | Home&Abroad | Finance | Net&Functions | Google Scholar | Health&Exer | Popular | Hacking | Wikis | Other Bookmarks

Routing Error

No route matches [GET] "/show"

```
Rails.root: /Users/user/Dropbox/X_Teaching/Ruby 2013-16/A_Lects&Pracs.2016/RubyWeek11 (Nov 21th).15/RubyLect11.progs/itunes
```

Application Trace | Framework Trace | Full Trace

Routes

Routes match in priority from top to bottom

Helper	HTTP Verb	Path	Controller#Action
Path / Url		Path Match	
goodbye_wave_path	GET	/goodbye/wave(.:format)	goodbye#wave
buy_songo_path	GET	/buy/songo(.:format)	buy#songo
add_song_path	GET	/add/song(.:format)	add#song
show_list_song_path	GET	/show/list_song(.:format)	show#list_song
show_album_path	GET	/show/album(.:format)	show#album

given the wrong, path…/show
should be…
/show/list

Load Page

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

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

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 | show_controller.r
add_controller.rb | show/album.html.erb | show/list_song.html.erb | show/

```
<h1>Album Listing</h1>

<%= form_tag('album', :method => 'get') do %>
  <p> Search for an Album named:  </p>
    <%= text_field_tag 'name' %>
    <%= submit_tag 'send' %>
<% end %>

<% if !@album_songs.empty? then %>
<p> The Album "<%=h params[:name] %>" has the following tracks: </p>
<ul>
<% @album_songs.each do |song| %>
    <li> <%=h song.title %> by <%=h song.artist %> </li>
<% end %>
</ul>
<p> The Album is <%=h @album_time %> minutes long. </p>
<% else %>
<p> This Album is not in the database. </p>
<% end %>
```

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

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

```yaml
# 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

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
    - environments
    - initializers
    - locales
    - application.rb
    - boot.rb
    - database.yml
    - environment.rb
    - routes.rb
  - db
  - doc

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/*

20111024135910_create_actors.rb - itunes - [~/Dropbox/X_Teaching/Ruby 2013-16/A_Lects&Pracs.2016/RubyWeek11 (Nov 21th).15/RubyLect11.progs/itunes]

Development: itunes

itunes > db > migrate > 20111024135910_create_actors.rb

Project

- itunes ~/Dropbox/X_Teaching/Ruby
  - app
    - assets
    - controllers
    - helpers
    - mailers
    - models
      - .gitkeep
      - actor.rb
      - song.rb
    - views
  - config
  - db
    - migrate
      - 20111024135910_create_ac
      - 20111024135938_create_sc
    - development.sqlite3
    - schema.rb
    - seeds.rb
    - test.sqlite3
  - doc
  - lib
  - log
  - public
  - script
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile

album.html.erb | database.yml | 20111024135938_create_songs.rb | 20111024135910_create_actors.rb

```
1  class CreateActors < ActiveRecord::Migration
2    def change
3      create_table :actors do |t|
4        t.string :name
5        t.string :likes
6        t
7        t.timestamps
8      end
9    end
10 end
11
```

all this is produced automatically (along with hidden key/id)

Terminal

```
  Song Load (2.9ms)  SELECT "songs".* FROM "songs"
  Rendered show/list_song.html.erb within layouts/application (4.7ms)
Completed 200 OK in 35ms (Views: 30.7ms | ActiveRecord: 2.9ms)
```

6: TODO | Terminal | 1 Event Log

Database connections detector: Database connection properties are detected. // Discover and setup new data sources. (36 minutes ago)

1:1 LF UTF-8

20111024135910_create_actors.rb - itunes - [~/Dropbox/X_Teaching/Ruby 2013-16/A_Lects&Pracs.2016/RubyWeek11 (Nov 21th).15/RubyLect11.progs/itunes]

Development: itunes

itunes > db > migrate > 20111024135910_create_actors.rb

Project

- itunes ~/Dropbox/X_Teaching/Ruby
  - app
    - assets
    - controllers
    - helpers
    - mailers
    - models
      - .gitkeep
      - actor.rb
      - song.rb
    - views
  - config
  - db
    - migrate
      - 20111024135910_create_ac
      - 20111024135938_create_sc
    - development.sqlite3
    - schema.rb
    - seeds.rb
    - test.sqlite3
  - doc
  - lib
  - log
  - public
  - script
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile

album.html.erb | database.yml | 20111024135938_create_songs.rb | 20111024135910_create_actors.rb

```
1  class CreateActors < ActiveRecord::Migration
2    def change
3      create_table :actors do |t|
4        t.string :name
5        t.string :likes
6      t.string :status
7        t.timestamps
8      end
9    end
10 end
11
```

now, lets write in a new field t.string :status

Terminal

```
  Song Load (2.9ms)  SELECT "songs".* FROM "songs"
  Rendered show/list_song.html.erb within layouts/application (4.7ms)
Completed 200 OK in 35ms (Views: 30.7ms | ActiveRecord: 2.9ms)
```

6: TODO | Terminal | 1 Event Log

Database connections detector: Database connection properties are detected. // Discover and setup new data sources. (36 minutes ago)

1:1 LF UTF-8

20111024135938_create_songs.rb - itunes - [~/Dropbox/X_Teaching/Ruby 2013-16/A_Lects&Pracs.2016/RubyWeek11 (Nov 21th).15/RubyLect11.progs/itunes]

Development: itunes

itunes > db > migrate > 20111024135938_create_songs.rb

Project

- itunes ~/Dropbox/X_Teaching/Ruby
  - app
    - assets
    - controllers
    - helpers
    - mailers
    - models
      - .gitkeep
      - actor.rb
      - song.rb
    - views
  - config
  - db
    - migrate
      - 20111024135910_create_ac
      - 20111024135938_create_sc
      - development.sqlite3
      - schema.rb
      - seeds.rb
      - test.sqlite3
  - doc
  - lib
  - log
  - public
  - script
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile

album.html.erb | database.yml | 20111024135938_create_songs.rb | 20111024135910_create_actors.rb

```
1  class CreateSongs < ActiveRecord::Migration
2    def change
3      create_table :songs do |t|
4        t.string :title
5        t.string :in_album
6        t.integer :time
7        t.string :artist
8        t.integer :actor_id
9        t.timestamps
10     end
11   end
12 end
13
```

so add these

nb; here we refer to actor id/key

Terminal

```
  Song Load (2.9ms)  SELECT "songs".* FROM "songs"
  Rendered show/list_song.html.erb within layouts/application (4.7ms)
Completed 200 OK in 35ms (Views: 30.7ms | ActiveRecord: 2.9ms)
```

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/*

actor.rb - itunes - [~/Dropbox/X_Teaching/Ruby 2013-16/A_Lects&Pracs.2016/RubyWeek11 (Nov 21th).15/RubyLect11.progs/itunes]

Development: itunes

itunes > app > models > actor.rb

album.html.erb | database.yml | 20111024135938_create_songs.rb | actor.rb

```
1  class Actor < ActiveRecord::Base
2    has_many :songs
3  end
4
```

relation between actor and songs

- itunes ~/Dropbox/X_Teaching/Ruby
  - app
    - assets
    - controllers
    - helpers
    - mailers
    - models
      - .gitkeep
      - actor.rb
      - song.rb
    - views
  - config
  - db
    - migrate
      - 20111024135910_create_ac
      - 20111024135938_create_sc
    - development.sqlite3
    - schema.rb
    - seeds.rb
    - test.sqlite3
  - doc
  - lib
  - log
  - public
  - script
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile

Terminal

```
  Song Load (2.9ms)  SELECT "songs".* FROM "songs"
  Rendered show/list_song.html.erb within layouts/application (4.7ms)
Completed 200 OK in 35ms (Views: 30.7ms | ActiveRecord: 2.9ms)
```

song.rb - itunes - [~/Dropbox/X_Teaching/Ruby 2013-16/A_Lects&Pracs.2016/RubyWeek11 (Nov 21th).15/RubyLect11.progs/itunes]

Development: itunes

itunes > app > models > song.rb

Project

- itunes ~/Dropbox/X_Teaching/Ruby
  - app
    - assets
    - controllers
    - helpers
    - mailers
    - models
      - .gitkeep
      - actor.rb
      - song.rb
    - views
  - config
  - db
    - migrate
      - 20111024135910_create_ac
      - 20111024135938_create_sc
      - development.sqlite3
      - schema.rb
      - seeds.rb
      - test.sqlite3
  - doc
  - lib
  - log
  - public
  - script
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
  - Gemfile

database.yml | 20111024135938_create_songs.rb | actor.rb | song.rb

```ruby
class Song < ActiveRecord::Base
    belongs_to :actor
end
```

relation between song and actor

Terminal

```
  Song Load (2.9ms)  SELECT "songs".* FROM "songs"
  Rendered show/list_song.html.erb within layouts/application (4.7ms)
Completed 200 OK in 35ms (Views: 30.7ms | ActiveRecord: 2.9ms)
```

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...

schema.rb - itunes - [~/Dropbox/X_Teaching/Ruby 2013-16/A_Lects&Pracs.2016/RubyWeek11 (Nov 21th).15/RubyLect11.progs/itunes]

Development: itunes

itunes > db > schema.rb

Project

- itunes ~/Dropbox/X_Teaching/Ruby
  - app
    - assets
    - controllers
    - helpers
    - mailers
    - models
      - .gitkeep
      - actor.rb
      - song.rb
    - views
  - config
  - db
    - migrate
      - 20111024135910_create_ac
      - 20111024135938_create_sc
    - development.sqlite3
    - schema.rb
    - seeds.rb
    - test.sqlite3
  - doc
  - lib
  - log
  - public
  - script
  - test
  - tmp
  - vendor

20111024135938_create_songs.rb | schema.rb | actor.rb | song.rb

```
# 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 that you check this file into your version control system.

ActiveRecord::Schema.define(version: 20111024135938) do

  create_table "actors", force: :cascade do |t|
    t.string   "name",       limit: 255
    t.string   "likes",      limit: 255
    t.string   "status",     limit: 255
    t.datetime "created_at"
    t.datetime "updated_at"
  end

  create_table "songs", force: :cascade do |t|
    t.string   "title",      limit: 255
    t.string   "in_album",   limit: 255
    t.integer  "time"
    t.string   "artist",     limit: 255
    t.integer  "actor_id"
    t.datetime "created_at"
    t.datetime "updated_at"
  end

end
```

Terminal

```
  * 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

MacBook-Air-4:itunes user$
```

6: TODO | Terminal | Event Log

Database connections detector: Database connection properties are detected. // Discover and setup new data sources. (55 minutes ago) 3:1 LF UTF-8

In Rails 5.0 routes.rb

routes.rb - itunes - [~/Dropbox/X_Teaching/Ruby 2013-16/A_Lects&Pracs.2016/RubyWeek11 (Nov 21th).15/RubyLect11.progs/itunes]

Development: itunes

itunes > config > routes.rb

35938_create_songs.rb | routes.rb | schema.rb | actor.rb | song.rb

```
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
```

Project

- itunes ~/Dropbox/X_Teaching/Ruby
  - app
  - config
    - environments
    - initializers
    - locales
    - application.rb
    - boot.rb
    - database.yml
    - environment.rb
    - routes.rb
  - db
    - migrate
      - 20111024135910_create_ac
      - 20111024135938_create_sc
    - development.sqlite3
    - schema.rb
    - seeds.rb
    - test.sqlite3
  - doc
  - lib
  - log
  - public
  - script
  - test
  - tmp
  - vendor
  - .gitignore

Terminal

```
* 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
```

itunes – [~/Desktop/Teaching/Ruby:2011-13/Lects&Pracs2013/RubyWeek11 (Nov 22th).12/RubyLect11.progs/itunes] – .../config/routes.rb – Je

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
    - environments
    - initializers
    - locales
    - application.rb
    - boot.rb
    - database.yml
    - environment.rb
    - routes.rb
  - doc

show/list_song.html.erb | show/album.html.erb | show/actor.html.erb

application.html.erb | development.sqlite3 | test.sqlite3 | database.yml | sc

```
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
```

delete comment and add action

7: Structure

6: TODO

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

itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../app/controllers/add_controller.rb – JetBrains R

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
        - actor.html.erb
        - album.html.erb
        - list_song.html.erb
  - config
    - environments
    - initializers
    - locales
    - application.rb
    - boot.rb
    - database.yml
    - environment.rb
    - routes.rb
  - db

show/list_song.html.erb | show/actor.html.erb | show_controller.rb | add/song.html
add_controller.rb | show/album.html.erb | database.yml | schema.rb | routes.rb | index.hide

```
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
```

only create a record if
you have been given an input

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

6: TODO

Add View

itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../app/views/add/song.html.erb – JetBrain

Development: itunes

add_controller.rb | show/album.html.erb | database.yml | schema.rb | routes.rb | index.
show/list_song.html.erb | show/actor.html.erb | show_controller.rb | add/song.

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
    - environments
    - initializers
    - locales
    - application.rb
    - boot.rb
    - database.yml
    - environment.rb
    - routes.rb
  - db
  - doc
  - lib
  - log

```
<h1>Add a Song to the DB</h1>
<p> Please enter Details Below: </p>
<%= form_tag('song', :method => 'get') do %>
  <p> Song Name:
    <%= text_field_tag 'title' %>
  <p> Artist:
    <%= text_field_tag 'artist' %>
  <p> Length of Track:
    <%= text_field_tag 'length' %>
  <p> Album its in:
    <%= text_field_tag 'album' %>
  <%= submit_tag 'send' %>
<% end %>

<p> Please
  <%= link_to 'review your inputs', :controller => "show", :action => "list_song"  %></p>
```

sets up the form for entering data

links us to the show/list_song method

6: TODO

add/song Displays...

iGoogle | Jump in foreclosures in Icela | localhost:3000/add/song

localhost:3000/add/song

Home&Abroad | Finance | Net&Functions | MyCal | Google Scholar | ISI | Health&Exer | Popular | Hacking | Wikis | » | Other Bookmark

Add a Song to the DB

Please enter Details Below:

Song Name:

Artist:

Length fof Track:

Album its in: send

Please review your inputs

Buy Controller

itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../app/controllers/buy_controller.rb – JetBrains RubyMine 3.2.

Development: itunes

add_controller.rb | show/album.html.erb | database.yml | schema.rb | routes.rb | index.hideit.html | show/list_song.html.erb | show/actor.html.erb | add/song.html.erb | buy_controller.rb

```
class BuyController < ApplicationController
  def songo
  @actor = Actor.find_by_name(params[:buyer])
    @song1 = Song.find_by_title(params[:song])
  if @song1 && @actor then
     @song2 = Song.create(:artist => @song1.artist, :title => @song1.title,
      :time => @song1.time, :in_album => @song1.in_album, :actor_id => @actor.id)
  end
  end

end
```

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

Project /Users/user/Desktop/X...

- 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
    - environments
    - initializers
    - locales
    - application.rb
    - boot.rb
    - database.yml
    - environment.rb
    - routes.rb
  - db
  - doc
  - lib
  - log

6: TODO

1:1 UTF-8

Buy/songo View

itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../app/views/buy/songo.html.erb – JetBrains RubyM

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
        - songo.html.erb
      - goodbye
      - layouts
      - show
  - config
    - environments
    - initializers
    - locales
    - application.rb
    - boot.rb
    - database.yml
    - environment.rb
    - routes.rb
  - db
  - doc
  - lib

add_controller.rb | show/album.html.erb | database.yml | schema.rb | routes.rb | index.hideit.htm
show/actor.html.erb | add/song.html.erb | buy_controller.rb | buy/songo.html.erb

```
<h1>Buy a Song</h1>

<%= form_tag('songo', :method => 'get') do %>
  <p> Give me your name: </p>
    <%= text_field_tag 'buyer' %>
  <p> Song you want to buy: </p>
    <%= text_field_tag 'song' %>
    <%= submit_tag 'buy' %>
<% end %>

<% if params[:buyer] && params[:song] then %>
    <% if @song1 && @actor then %>
     <p>  "<%=h params[:buyer] %>" you are now the proud owner of "<%=h params[:song] %>" </p>
    <% elsif @actor.nil? then %>
    <p> I am afraid that you are not in my database. </p>
    <% elsif @song1.nil? then %>
    <p> I am fraid I do not have this song: <%=h params[:song] %>. </p>
    <% end %>
<% end %>
```

we only display when we been given an input

we test for what is coming back from the data base

be aware that :find returns arrays and find_by... a record

so, you may need to deal with a nil or a [ ]

6: TODO

19:22 UTF-8

buy/songo Displays...

show Controller

itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../app/controllers/show_controller.rb – JetBrains

Development: itunes

add_controller.rb | show/album.html.erb | database.yml | schema.rb | routes.rb | index.hic
add/song.html.erb | buy_controller.rb | buy/songo.html.erb | show_controlle

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
        - songo.html.erb
      - goodbye
      - layouts
      - show
  - config
    - environments
    - initializers
    - locales
    - application.rb
    - boot.rb
    - database.yml
    - environment.rb
    - routes.rb
  - db
  - doc
  - lib

```
class ShowController < ApplicationController

  def list_song
  @songs = Song.all
  end

  def album
  @name = params[:name]
  if !@name.blank?
  then @album_songs = Song.where(in_album: @name)
        @album_time = (@album_songs.collect {|song| song.time}).sum
  else @album_songs = [] end
  end

  def actor
    @actors = Actor.all
  end

end
```

method for each page

6: TODO

1:71 UTF-8

show/actor View

itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../app/views/show/actor.html.erb – JetBrains

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
        - songo.html.erb
      - goodbye
      - layouts
      - show
        - actor.html.erb
        - album.html.erb
        - list_song.html.erb
  - config
    - environments
    - initializers
    - locales
    - application.rb
    - boot.rb
    - database.yml
    - environment.rb
    - routes.rb

add_controller.rb | database.yml | schema.rb | routes.rb | index.hideit.html | add/song.

buy_controller.rb | buy/songo.html.erb | show_controller.rb | show/actor.htm

```
<h1>Actor Listing</h1>
<ul>
<% @actors.each do |actor| %>
    <li> <%=h actor.name %> the <%=h actor.status %>, likes  <%=h actor.likes %> : <%=h actor.id %> </li>
<% end %>
</ul>


<p> Say
    <%= link_to 'Goodbye, Mark', :controller => 'goodbye', :action => 'wave', :id => "foobar" %></p>
```

6: TODO

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

itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../app/views/show/album.html.erb – JetBr

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
        - songo.html.erb
      - goodbye
      - layouts
      - show
        - actor.html.erb
        - album.html.erb
        - list_song.html.erb
  - config
    - environments
    - initializers
    - locales
    - application.rb
    - boot.rb
    - database.yml
    - environment.rb
    - routes.rb

add_controller.rb | schema.rb | routes.rb | index.hideit.html | add/song.html.erb | buy_
buy/songo.html.erb | show_controller.rb | show/actor.html.erb | show/album

```
<h1>Album Listing</h1>

<%= form_tag('album', :method => 'get') do %>
  <p> Search for an Album named:  </p>
    <%= text_field_tag 'name' %>
    <%= submit_tag 'send' %>
<% end %>

<% if !@album_songs.empty? then %>
<p> The Album "<%=h params[:name] %>" has the following tracks: </p>
<ul>
<% @album_songs.each do |song| %>
    <li> <%=h song.title %> by <%=h song.artist %> </li>
<% end %>
</ul>
<p> The Album is <%=h @album_time %> minutes long. </p>
<% else %>
<p> This Album is not in the database. </p>
<% end %>
```

Displays...

iGoogle | Jump in foreclosures in Icelar | localhost:3000/show/album

localhost:3000/show/album

Home&Abroad | Finance | Net&Functions | MyCal | Google Scholar | ISI | Health&Exer | Popular | Hacking | Wikis | » | Other Bookmar

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

itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../app/views/show/list_song.html.erb – JetBrains Rub

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
        - songo.html.erb
      - goodbye
      - layouts
      - show
        - actor.html.erb
        - album.html.erb
        - list_song.html.erb
  - config
    - environments
    - initializers
    - locales
    - application.rb
    - boot.rb
    - database.yml
    - environment.rb
    - routes.rb

add_controller.rb | routes.rb | index.hideit.html | add/song.html.erb | buy_controller.rb | buy/songo.html.erb | show_controller.rb | show/actor.html.erb | show/album.html.erb | show/list_song.html.erb

```
<h1>Song Listing</h1>
<ul>
<% @songs.each do |song| %>
    <li> <%=h song.title %> by <%=h song.artist %>, in  <%=h song.in_album %> : <%=h song.id %>
      [Owned by Actor ID <%=h song.actor_id %>] </li>
<% end %>
</ul>
```

6: TODO

Displays...

iGoogle | RTE Jump in foreclosures in Icela | localhost:3000/show/song_l

localhost:3000/show/song_list

Home&Abroad | Finance | Net&Functions | MyCal | Google Scholar | ISI | Health&Exer | Popular | Hacking | Wikis » Other Bookm

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...

itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../db/seeds.rb – JetBrains RubyMine 3.2.4

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
  - config
    - environments
    - initializers
    - locales
    - application.rb
    - boot.rb
    - database.yml
    - environment.rb
    - routes.rb
  - db
    - migrate
    - development.sqlite3
    - schema.rb
    - seeds.rb
    - test.sqlite3
  - doc
  - lib
  - log
  - public

add_controller.rb | index.hideit.html | add/song.html.erb | buy_controller.rb | buy/songo.html.erb
show_controller.rb | show/actor.html.erb | show/album.html.erb | show/list_song.html.erb | seeds.rb

```ruby
# This file should contain all the record creation needed to seed the database with its default values.
# The data can then be loaded with the rake db:seed (or created alongside the db with db:setup).
#
# Examples:
#
#   cities = City.create([{ name: 'Chicago' }, { name: 'Copenhagen' }])
#   Mayor.create(name: 'Emanuel', city: cities.first)


Actor.create(:name => 'Mark', :likes => 'reggae', :status  => 'Admin', :id => 1)
Actor.create(:name => 'Bloggie', :likes => 'punk', :status  => 'Buyer', :id => 2)
Song.create(:title => 'Stairway to Heaven', :artist => 'Led Zep', :time  => 4, :in_album => 'Led Zep IV', :actor_id => 1)
Song.create(:title => 'Holiday', :artist => 'Sex Pistols', :time  => 2, :in_album => 'God Save...', :actor_id => 2)
```

6: TODO

1:1 UTF-8

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...

Organizer

Build Clean Run Action

PROJECTS & SOURCES

- chicken
  - app
  - config
  - db
    - development.sqlite3
    - migrate
      - 20101108160428_create_birds.rb
      - 20101108174408_create_dishes.rb
    - schema.rb
    - seeds.rb
  - doc
  - lib
  - log
  - public
  - Rakefile
  - README
  - script
  - test
  - tmp
  - vendor

seeds.rb:13

```
# This file should contain all the record creation needed to seed the database with its d
# The data can then be loaded with the rake db:seed (or created alongside the db with db:
#
# Examples:
#
#   cities = City.create([{ :name => 'Chicago' }, { :name => 'Copenhagen' }])
#   Major.create(:name => 'Daley', :city => cities.first)

Bird.create(:name => 'henny_penny', :feet => 2)
Bird.create(:name => 'cocky_locky', :feet => 1)

Dish.create(:name => 'chicken_tikka')
Dish.create(:name => 'splayed_cockrells')
```

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...

Ryan's Scraps: What's New in | Action Controller: Exception

localhost:3000/add/bird

Home&Abroad | Finance | Net&Functions | MyCal | Google Scholar | ISI | Health&Exer | Popular | Hacking | Wikis » Other Bool

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) ....
```

Easiest way to deal with this is to copy the contents of the Gemfile for an app created with your uptodate libraries and then do:

$ bundle update

then

$ rails server

should work ok

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...

itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../app/views/show/actor.html.erb – JetBr

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
      - buy
      - goodbye
      - layouts
      - show
        - actor.html.erb
        - list_song.html.erb
  - config
  - db
    - migrate
    - development.sqlite3
    - schema.rb
    - seeds.rb
    - test.sqlite3
  - doc
  - lib
  - log
  - public

add_controller.rb | index.hideit.html | add/song.html.erb | buy_controller.rb | buy/sor
show_controller.rb | show/actor.html.erb | show/album.html.erb | show/list_song.html.erb

```
<h1>Actor Listing</h1>
<ul>
<% @actors.each do |actor| %>
    <li> <%=h actor.name %> the <%=h actor.status %>, likes  <%=h actor.likes %> : <%=h actor.id %> </li>
<% end %>
</ul>


<p> Say
    <%= link_to 'Goodbye, Mark', :controller => 'goodbye', :action => 'wave', :id => "foobar" %></p>
```

call wave method
in goodbye controller,
with value “foobar” for :id

add **goodbye controller**

itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../app/controllers/goodbye_controller.rb – JetBrains Rubyl

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
      - buy
      - goodbye
      - layouts
      - show
        - actor.html.erb
        - album.html.erb
        - list_song.html.erb
  - config
  - db
    - migrate
    - development.sqlite3
    - schema.rb
    - seeds.rb
    - test.sqlite3
  - doc
  - lib
  - log
  - public

index.hideit.html | add/song.html.erb | buy_controller.rb | buy/songo.html.erb | show_controller.r
show/actor.html.erb | goodbye_controller.rb | show/album.html.erb | show/list_song.html.erb | seeds

```
class GoodbyeController < ApplicationController
  def wave
    actors = Actor.all
  @actor = actors.last
  @value = params[:id]
  end

end
```

1: Project | 7: Structure | 6: TODO

add **goodbye view**

itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../app/views/goodbye/wave.html.erb – JetBrains RubyMine 3.2.4

Development: itunes

Project /Users/user/Desktop/X_Teaching...

Project

- itunes (~/Desktop/X_Teaching/Ruby:2011-14/Lects
  - app
    - assets
    - controllers
      - add_controller.rb
      - application_controller.rb
      - buy_controller.rb
      - goodbye_controller.rb
      - show_controller.rb
    - helpers
    - mailers
    - models
    - views
      - add
      - buy
      - goodbye
        - wave.html.erb
      - layouts
      - show
  - config
  - db
    - migrate
    - development.sqlite3
    - schema.rb
    - seeds.rb
    - test.sqlite3
  - doc
  - lib
  - log
  - public
  - script
  - test
  - vendor
  - .gitignore

add/song.html.erb | buy_controller.rb | buy/songo.html.erb | show_controller.rb | show/actor.html.erb | goodbye_controller.rb | goodbye/wave.html.erb | show/album.html.erb | show/list_song.html.erb | seeds.rb

```
<h1>Goodbye#actor_wave</h1>

<p>bye bye <%= @actor.to_s %>  <%= @value %> </p>
<p><%= @value %> </p>
```

6: TODO

1:1 UTF-8

show/actor now looks...

when you click you get...

Part E:

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

Look at old bundle...

itunes – [~/Desktop/Teaching/Ruby:2011-12/Lects&Pracs2012/RubyWeek11 (Nov 22th)/RubyLect11.progs/itunes] – .../Gemfile – JetBr...

Development: itunes

schema.rb | show_controller.rb | add_controller.rb | application_controller.rb | application_helper.rb | application.rb | boot.rb | environment.rb | routes.rb | Gemfile

Project /...

- itunes (~/Desktop/Tea
  - app
  - config
  - db
  - doc
  - lib
  - log
  - public
  - script
  - test
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README
- External Libraries

```
source 'http://rubygems.org'

gem 'rails', '3.1.0'

# Bundle edge Rails instead:
# gem 'rails',     :git => 'git://github.com/rails/rails.git'

gem 'sqlite3'


# Gems used only for assets and not required
# in production environments by default.
group :assets do
  gem 'sass-rails', "  ~> 3.1.0"
  gem 'coffee-rails', "~> 3.1.0"
  gem 'uglifier'
end

gem 'jquery-rails'

# Use unicorn as the web server
# gem 'unicorn'

# Deploy with Capistrano
# gem 'capistrano'

# To use debugger
# gem 'ruby-debug19', :require => 'ruby-debug'

group :test do
  # Pretty printed test output
  gem 'turn', :require => false
end
```

You can systematically update....or (lazy guy) just copy the whole contents of the Gemfile from a recent one and paste it in (then do “$bundle update”)

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...

itunes – [~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek11 (Nov 22th).13/RubyLect11.progs/itunes] – .../Gemfile – JetBrains RubyMine 3.2.4

Development: itunes

buy_controller.rb | buy/songo.html.erb | show_controller.rb | show/actor.html.erb | goodbye_controller.rb | goodbye/wave.html.erb | Gemfile | show/album.html.erb | show/list_song.html.erb | seeds.rb

```
source 'http://rubygems.org'

gem 'rails', '4.1.6'

# bundle edge Rails instead:
# gem 'rails',     :git => 'git://github.com/rails/rails.git'

gem 'sqlite3'

# Gems used only for assets and not required
# in production environments by default.
group :assets do
  gem 'sass-rails', "  ~> 4.0.3"
  gem 'coffee-rails', "~> 4.0.1"
  gem 'uglifier'
end

gem 'jquery-rails'

# Use unicorn as the web server
# gem 'unicorn'

# Deploy with Capistrano
# gem 'capistrano'

# To use debugger
# gem 'ruby-debug19', :require => 'ruby-debug'

group :test do
  # Pretty printed test output
  gem 'turn', :require => false
end
```

Project /Users/user/Desktop/X_Teaching...

- itunes (~/Desktop/X_Teaching/Ruby:2011-14/Lects&
  - app
    - assets
    - controllers
    - helpers
    - mailers
    - models
    - views
  - config
  - db
    - migrate
    - development.sqlite3
    - schema.rb
    - seeds.rb
    - test.sqlite3
  - doc
  - lib
  - log
  - public
  - script
  - test
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README
- External Libraries

6: TODO

10:7 UTF-8

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

itunes – [~/Desktop/Teaching/Ruby:2011-12/Lects&Pracs2012/RubyWeek11 (Nov 22th)/RubyLect11.progs/itunes2] – .../Gemfile – JetBrains RubyMin

Development: itunes

Project /Users/mkeane/Desktop/Teaching/...

Project

- itunes2 [itunes] (~/Desktop/Teaching/Ruby:2011-12/Le
  - app
  - config
  - db
  - doc
  - lib
  - log
  - public
  - script
  - test
  - vendor
  - .gitignore
  - config.ru
  - Gemfile
  - Gemfile.lock
  - Rakefile
  - README
- External Libraries

README | Gemfile.lock | config.ru | Rakefile | application_helper.rb | application.rb | boot.rb | environment.rb | routes.rb | Gemfile

```
source 'http://rubygems.org'

gem 'rails', '3.2.8'

# Bundle edge Rails instead:
# gem 'rails',     :git => 'git://github.com/rails/rails.git'

gem 'sqlite3'


# Gems used only for assets and not required
# in production environments by default.
group :assets do
  gem 'sass-rails',   '~> 3.2.3'
  gem 'coffee-rails', '~> 3.2.1'
  gem 'uglifier'
end

gem 'jquery-rails'

# Use unicorn as the web server
# gem 'unicorn'

# Deploy with Capistrano
# gem 'capistrano'

# To use debugger
# gem 'ruby-debug19', :require => 'ruby-debug'
```

1: Project | 7: Structure | 6: TODO

23:82 | UTF-8

```
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
MIS EN BOUTEILLE AU CHÂTEAU
PRODUCE OF FRANCE