

Ruby Explorations X

Mark Keane...CSI...UCD



Rails Basics

Ruby on Rails I:

- A: the design theory behind Rails framework
- B: walk-thru' of simple Rails system
- C: other Rails functionality...neat stuff
- D: now for models in Hello
- E: some residual things to know...
- F: fixing the db problem in Hello

My New Puppy

- ✿ this is my second puppy
- ✿ her name is ...



Part A:

practical and theoretical background

What is Rails...

- software framework, a set of libraries to support rapid web-site development
- when you type ***rails new sitename***, a folder is opened up with bags and bags of sub-folders and files
- these sub-folders contain file templates for coding and ready-made methods, a web server (WEBrick) and a database (SQLite3); related by ***conventions***
- the thing is a monster... ALSO...nb with Rails 5.0.0

If an error about missing gems in sitename dir do:
\$bundle update

welcome_controller.rb - newsite - [~/Dropbox/X_Teaching/Ruby 2013-16/A_Lects&Pracs.2016/RubyWeek10 (Nov 17).15/xRubyLect10.progs/newsite]

Development: newsite

newsite app controllers welcome_controller.rb

Project Project Files

1: Project 2: Structure 3: Favorites

newsite ~/Dropbox/X_Teaching/Ruby 2013-16/app/assets/channels/controllers/concerns/controllers/views/welcome/helpers/jobs/mailers/models/layouts/welcome/bin/config/db/lib/log/public/test/tmp/vendor/.gitignore/config.ru/Gemfile/Gemfile.lock

application_controller.rb x welcome_controller.rb x hello.html.erb x

```
1 class WelcomeController < ApplicationController
2   def hello
3     @message = "Hi, there do da !"
4   end
5 end
```

Database

Completed 200 OK in 33ms (Views: 28.5ms | ActiveRecord: 0.0ms)

Started GET "/welcome/hello" for ::1 at 2016-09-29 09:27:37 +0100
Processing by WelcomeController#hello as HTML
 Rendering welcome/hello.html.erb within layouts/application
 Rendered welcome/hello.html.erb within layouts/application (0.3ms)
Completed 200 OK in 31ms (Views: 26.4ms | ActiveRecord: 0.0ms)

Terminal Messages Run TODO Event Log

Platform and Plugin Updates: RubyMine is ready to update. (16 minutes ago) 6:17 LF+ UTF-8+

What is Rails...

- originally, web-pages were static html but this was very inflexible and hard to maintain
- then...they became more dynamic with databases behind them...then people started putting code in so that the page was created on the fly (e.g., personalised)
- this improved flexibility and maintenance but introduced complexity and security issues
- **Rails** is a response to these issues...

What is Rails...MVC

- complexity is handled by dividing task into model (db setup), view (screens) and controller (code to do things)
- in simple terms, what was a single html file with code, gets broken up into 3-4 files, which combine to create a file that is sent to browser to be rendered
- its secure 'cos website users can only see the views and user have no sight of code for views or models
- the model/controller bit just separates db issues from code that modifies objects (sort of)

MVC framework

- ***model-view-controller*** is a software architecture/architectural pattern (used in Rails)
- ***model*** is domain-specific rep of the data used by the application; manages data and informs of changes
- ***view*** renders model into suitable form for interactions (ie, at interface); may be multiple views of same model
- ***controller*** receives input and initiates response, making calls to model data (sort-of guts of app)

Rails...runs off...

- Rails is backed by libraries, (some) we have seen:
 - ActionMailer: for mailing
 - ActiveRecord: for interface to db
 - Rake: for setting up, updating and clearing out db
 - Erb: for writing embedded ruby commands in docs
- we will adopt a hide-the-db approach

Controller



Model



Views



MVC flow of control

1. **user** interacts with interface (mouse-click)
2. **controller** receives input event and converts it to suitable user action, understandable for model
3. **controller** notifies **model** of user action, poss. changing model's state (e.g., update shopping cart)
4. once **controller** does actions, based on **model** data, flow of control may pass back to **view** to render output (or move on to next page/action)
5. **user** interface may await further user input, cycle restarts

Controller



Model



Views



Practical Issues

Or Xcode...if you
know/prefer it

- we will work with four windows: Rubymine, 2 command-line windows and a browser
- Rubymine helps us navigate folders and edits files
- Terminal will run the server
- If you need it, you can Run a second Terminal window to run external scripts (e.g. rake)
- Browser will look at web-site pages running
- get used to moving between all four windows

The screenshot shows the RubyMine IDE interface. The top navigation bar displays the file 'welcome_controller.rb - newsite' and the development environment 'Development: newsite'. The left sidebar shows the project structure under 'newsite' with various files and folders like 'app/controllers/welcome_controller.rb'. The main editor window contains the following code:

```
1 class WelcomeController < ApplicationController
2   def hello
3     @message = "Hi, there do da !"
4   end
5 end
```

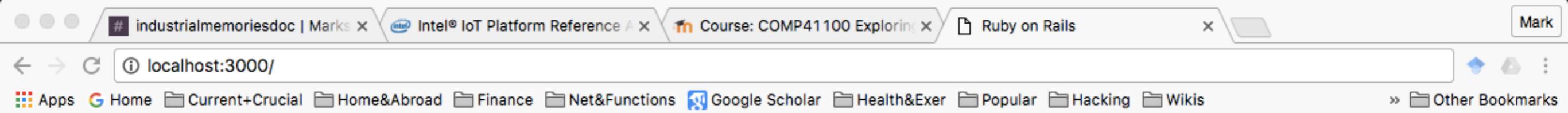
The bottom terminal window shows the command 'rails server' being run in the 'newsite' directory. The output shows several failed connection attempts before the server starts successfully.

Annotations:

- A large watermark 'Window-1' is overlaid across the center of the image.
- A large watermark 'Outputs' is positioned in the lower right area.
- A black arrow points from the text 'DDO' in the terminal's status bar to the 'DDO' tab in the bottom navigation bar.

```
+ ^CExiting
✖ MacBook-Air-4:newsite user$ 
✖ MacBook-Air-4:newsite user$ rails server
[1]: DDO
```

Platform and Plugin Updates: RubyMine is ready to update. (today 09:15)



Browser



Yay! You're on Rails!



Rails version: 5.0.0.1

Ruby version: 2.3.1 (x86_64-darwin15)

Part B(i):
simplest possible 7-step recipe to use Rails...

Controller



Model



Views



Controller



Model

```
$ rake db:migrate
```



Views

```
$ rails server
```

Before Cookbook Run

1. Please re-adjust your trousers...

2. Make sure your gems are in place...run any or all of dese

\$ gem install rails (make sure **rake** is there)

\$ gem update --system **sudo if needed**

\$ install a_gem (if **a_gem** is missing)

\$ gem install bundle

\$ bundle install

\$ bundle update **a_particular_gem** (to get latest v)

Cookbook Run

1. \$ rails new newsite
2. \$ cd newsite
3. \$ rails generate controller welcome hello
4. look at /newsite/app/controllers/welcome_controller.rb
add **@message = “Hi there do da !”**
to *welcome* method
5. look at newsite/app/views/welcome/hello.html.erb
add **<p> <%= @message %></p>** to file
6. \$ rails server
7. load in browser: <http://localhost:3000/welcome/hello>

first time may pause at “run bundle”
if impatient use this... **-- skip-bundle**
Mac issues on privileges...

Cookbook Run: Terminal

```
markkean% rails new newsite
      create ...a whole bunch of crap....
markkean% cd newsite
markkean% ls
README      config          lib       script        vendor
Rakefile    db             log       test
markkean% rails generate controller welcome hello
      create app/controllers/test_controller.rb
      route  get "welcome/hello"
      invoke erb
      create app/views/welcome
      create app/views/welcome/hello.html.erb
      invoke test_unit
      create test/functional/welcome_controller_test.rb
.....
markkean% rails server
=> Booting WEBrick
=> Rails 3.1.0 application starting in development on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
[2011-10-11 18:56:20] INFO  WEBrick 1.3.1
[2011-10-11 18:56:20] INFO  ruby 1.9.2 (2011-07-09) [x86_64-darwin11]
[2011-10-11 18:56:20] INFO  WEBrick::HTTPServer#start: pid=1581
port=3000
Started GET "/test/hello" for 127.0.0.1 at 2011-10-11 19:09:48 +0100
Processing by WelcomeController#hello as HTML
Rendered test/hello.html.erb within layouts/application (25.4ms)
Compiled application.css (28ms) (pid 1581)
```

Cookbook Run

1. \$ rails new newsite
2. \$ cd newsite
3. \$ rails generate controller welcome hello
4. look at /newsite/app/controllers/welcome_controller.rb
add **@message = “Hi there do da !”**
to welcome method
5. look at newsite/app/views/welcome/hello.html.erb
add **<p> <%= @message %></p>** to file
6. \$ rails server
7. load in browser: <http://localhost:3000/welcome/hello>

Cookbook Run

1. \$ rails new newsite
2. \$ cd newsite
3. \$ rails generate controller welcome hello
4. look at /newsite/app/controllers/
welcome_controller.rb
add **@message = “Hi there do da !”**
to *welcome* method
5. look at newsite/app/views/
welcome/hello.html.erb
add **<p> <%= @message %></p>** to file
6. \$ rails server
7. load page in browser: [http://localhost:
3000/welcome/hello](http://localhost:3000/welcome/hello)

\$ rake db:create
*is done first time
you use db*

\$ rake db:migrate
*is also done when
you have defined
model*

Cookbook Run: Rubymine I

The screenshot shows the RubyMine IDE interface. The left sidebar displays the project structure for a 'newsite' application, which includes directories for assets, channels, controllers (containing application_controller.rb and welcome_controller.rb), helpers, jobs, mailers, models, views (with layouts and a welcome folder containing hello.html.erb), bin, config, db, lib, log, public, test, tmp, vendor, and several configuration files (.gitignore, config.ru, Gemfile, Gemfile.lock). A blue callout bubble points to the 'welcome_controller.rb' file in the controllers directory, with the text 'free empty controller' inside.

The main editor window shows the 'welcome_controller.rb' file content:

```
1 class WelcomeController < ApplicationController
2   def hello
3     @message = "Hi, there do da !"
4   end
5 end
```

The terminal window at the bottom shows the command 'rails server' being run in the 'newsite' directory, with the output '^CExiting' and several blank lines indicating the server has stopped.

Bottom status bar: Platform and Plugin Updates: RubyMine is ready to update. (today 09:15) | 5.4 | LF | UTF-8 | Event Log

Cookbook Run:Xcode II

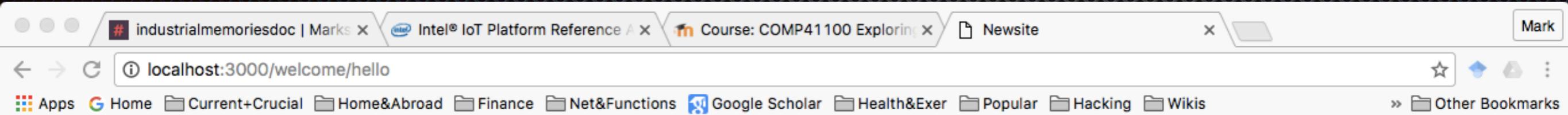
The screenshot shows the RubyMine IDE interface. The left sidebar displays the project structure under '1: Structure'. The 'newsite' project contains an 'app' directory with subfolders like 'assets', 'channels', 'controllers', 'helpers', 'jobs', 'mailers', 'models', 'views', 'layouts', and 'welcome'. Inside 'welcome', there is a file named 'hello.html.erb'. The code editor on the right shows the contents of 'hello.html.erb':

```
1 <h1>Welcome#hello</h1>
2 <p>  <%= @message %> <p>
3 <p>Find me in app/views/welcome/hello.html.erb</p>
```

A blue callout bubble points to the third line of code with the text 'defines the view'.

The top status bar shows the path: 'hello.html.erb - newsite - [~/Dropbox/X_Teaching/Ruby 2013-16/A_Lects&Pracs.2016/RubyWeek10 (Nov 17).15/xRubyLect10.progs/newsite]'. The bottom status bar includes tabs for Terminal, Messages, Run, TODO, and Event Log, along with system status icons.

Cookbook Run: Browser



Welcome#hello

Hi, there do da !

Find me in app/views/welcome/hello.html.erb

Controller



localhost/welcome/hello

where is the ?



Model

render
hello.html.erb
with @message



Views

Caveats

- generally, each controller-method maps to a view file
- we have not defined a layout (general frame of page)
- we have not defined models
- we have not looked at stylesheets
- we have not looked at routing
- but...we will

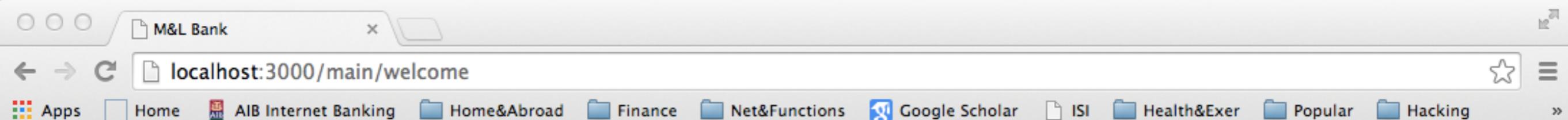
Part B2(i):

Walk-through of (less) simple Rails: The screens

A Banking Site

- takes you through mortgage application
- uses forms to take inputs
- re-presents that information
- does some minimal computation behind scenes
- uses a db to store the interaction; that persists over different sessions with user

The Site: Screen I (start)



Welcome to Monster & Leinster's Home Page



[click logo to enter]

Monster & Leinster Inc. © 2011

The Site: Screen IIa

Ruby on Rails Guides: Layout × Google Image Result for http × localhost:3000/other/name ×

localhost:3000/other/name

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

999+

The Monster & Leinster Bank



We are very pleased to welcome you our home mortgage page.

Please enter your first and last name below:

First Name:

Last Name:

Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are not responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

The Site: Screen IIb

Ruby on Rails Guides: Layout × Google Image Result for http × localhost:3000/other/name? × +

localhost:3000/other/name?utf8=✓&fst_name=Mark&lst_name=Keane&commit=send

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

The Monster & Leinster Bank



We are very pleased to welcome you our home mortgage page.

Please enter your first and last name below:

First Name:

Last Name:

Welcome Mark, let's go to the [next page](#).

Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are not responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

The Site: Screen III

Ruby on Rails Guides: Layout × Google Image Result for http × localhost:3000/other/index ×

localhost:3000/other/index

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

The Monster & Leinster Bank



Your Mortgage Details

So, Mark, let us get some more of your details:

First Name: Mark

Last Name: Keane

Address:

Salary:

Loan Amount:

Loan Reason:

Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are not responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

The Site: Screen IV

Ruby on Rails Guides: Layout × Google Image Result for http × localhost:3000/other/index? × +

localhost:3000/other/index?utf8=✓&st_name=15+Flitwick+Close&salary=5000&loan=100000&reason=I%27m+broke&c... 999+

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

The Monster & Leinster Bank



Your Mortgage Details

So, we have your details as:

First Name: Mark

Last Name: Keane

Address: 15 Flitwick Close

Salary: 5000

Loan: 100000

Reason: I'm broke

Your ID: 27

Do you want to [Ask For Quote](#) or [Go Back](#).

Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are not responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

The Site: Screen V

Ruby on Rails Guides: Layout × Google Image Result for http × You goddam pauper, you asl ×

localhost:3000/other/quote

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

The Monster & Leinster Bank



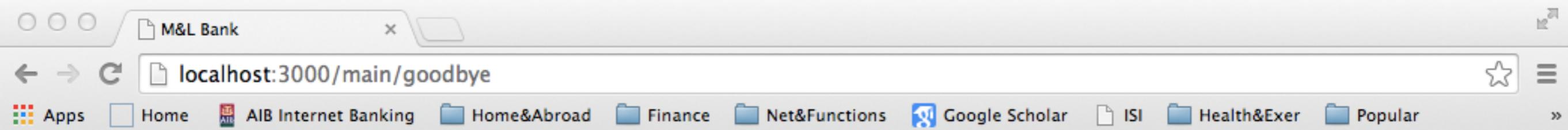
Your Quote

You goddam pauper, you asked us for a loan that is: more than three times your salary. Please go back and re-enter your correct salary.

Would you like to [Finish](#) or [Change Details](#)

Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are not responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

The Site: Screen VI (end)



Goodbye to Monster & Leinster's End Page



[click logo to re-enter site]

- ,,,1
- mark,keane,,2
- mark,keane,2000,31223
- mark,keane,,4
- mark,keane,12,20000005
- ,,,6
- Mark,Keane,5000,1000007
- ,,,8
- ,,,9
- Mark,Keane,5000,10000010

Part B2(ii):

Walk-through of simple Rails: The Setup

Steps

1. \$ rails new banko
2. \$ cd banko
3. \$ rails generate controller main welcome goodbye
4. \$ rails generate controller other name index show
change quote
5. \$ rails generate model Entry
6. lets...edit a whole bunch of files...
7. \$ rails server
8. load page in browser:
<http://localhost:3000/main/welcome/>

\$ rake db:create
*is done first time
you use db*

\$ rake db:migrate
*is also done when
you have defined
model*

Steps

1. \$ rails new bank
set up dir
2. \$ cd bank
go to dir
3. \$ rails generate controller main welcome goodbye
create controller
main with method **welcome**
4. \$ rails generate controller other name index show
change quote
create controller
other with method **name**
5. \$ rails generate model Entry
create model
called **entry**
6. lets...edit a whole bunch of files...
7. \$ rails server
start server; you must be in the **bank** directory
8. load page in browser:
<http://localhost:3000/main/welcome/>
call first page

Controller



Model

```
$ rake db:migrate
```

Views

```
$ rails server
```

What you get...

The screenshot shows the JetBrains RubyMine 3.2.4 IDE interface. The title bar indicates the project is named 'banko' and the current file is 'main_controller.rb'. The left sidebar displays the project structure under 'Project' mode, showing the 'app' directory which contains 'assets', 'controllers', 'views', and other files like 'application_controller.rb', 'other_controller.rb', etc. The right side shows the code editor with the following content:

```
class MainController < ApplicationController
  layout 'start_to_end'

  def welcome
    @message = "Welcome"
  end

  def goodbye
    @message = "Goodbye"
    @entries = Entry.all
  end
end
```

The 'goodbye' method is currently selected. The status bar at the bottom shows '6: TODO'.

What you get....

The screenshot shows a Ruby on Rails application structure in JetBrains RubyMine 3.2.4. The project tree on the left shows the following structure:

- banko
- app
 - assets
 - images
 - javascripts
 - stylesheets
 - controllers
 - concerns
 - application_controller.rb
 - main_controller.rb
 - other_controller.rb
 - helpers
 - mailers
 - models
 - views
 - layouts
 - application.html.erb
 - start_to_end.erb
 - main
 - goodbye.html.erb
 - welcome.html.erb
 - other
- bin
- config
- db
- lib
- log
- public
- test
- vendor
- .gitignore
- config.ru
- Gemfile
- Gemfile.lock

for **welcome** method
in **main**

What you get...

The screenshot shows the JetBrains RubyMine 3.2.4 IDE interface. The title bar indicates the current project is 'banko' and the file being edited is 'main/welcome.html.erb'. The left side features a 'Project' tool window showing the directory structure of the 'banko' application. Two specific files are highlighted with red circles and arrows: 'main_controller.rb' in the 'controllers' folder and 'welcome.html.erb' in the 'views/main' folder. A blue callout bubble on the right side points to 'welcome.html.erb' with the text 'the main/welcome view'. The code editor on the right displays the HTML and ERB code for the 'welcome.html.erb' view.

```
<h1><p align="center"> <%= h @message %> to Monster & Leinster's</h1>
<h1><p align="center"> Home Page</h1>
<p align="center"><%= link_to image_tag("monster.jpg"),
:controller => "other", :action => "name" %> </p>
<h5> <p align="center"> [click logo to enter] </p></h5>
```

What you get...

The diagram illustrates the relationship between the `main_controller.rb` file and the `start_to_end.html.erb` layout file. A red arrow points from the `layout 'start_to_end'` line in the code editor to the `start_to_end.html.erb` file in the project structure. A blue callout bubble labeled "main layout" is positioned below the `start_to_end.html.erb` file.

Project /Users/user/Desktop/X_Teachin... main_controller.rb application.html.erb application_controller.rb application.css

1: Project

banko (~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek10 (Nov 15th).13/RubyLect10.progs/banko) – .../app/controllers/main_controller.rb – JetBrains RubyMine 3.2.4

File Edit View Tools Window Help

Development: banko

Project

banko (~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek10 (Nov 15th).13/RubyLect10.progs/banko)

app

assets

images

javascripts

stylesheets

controllers

concerns

application_controller.rb

main_controller.rb

other_controller.rb

helpers

mailers

models

views

layouts

application.html.erb

start_to_end.html.erb

main

goodbye.html.erb

welcome.html.erb

other

bin

config

db

lib

log

public

test

vendor

.gitignore

config.ru

Gemfile

Gemfile.lock

6: TODO

11:6 UTF-8

main_controller.rb

```
class MainController < ApplicationController
  layout 'start_to_end'

  def welcome
    @message = "Welcome"
  end

  def goodbye
    @message = "Goodbye"
    @entries = Entry.all
  end
end
```

application.html.erb

application_controller.rb

application.css

main layout

What you get...

The screenshot shows the JetBrains RubyMine 3.2.4 IDE interface. The title bar indicates the project is named 'banko' and the current file is 'application_controller.rb'. The left sidebar shows the project structure under 'Project'. The main area displays the code for 'application_controller.rb'.

```
<!DOCTYPE html>
<html>
<head>
  <title> M&L Bank </title>
</head>
<body>

<%= yield %>
<p align="center"> <font size="1"> Monster & Leinster Inc. &copy 2011 </font> </p>

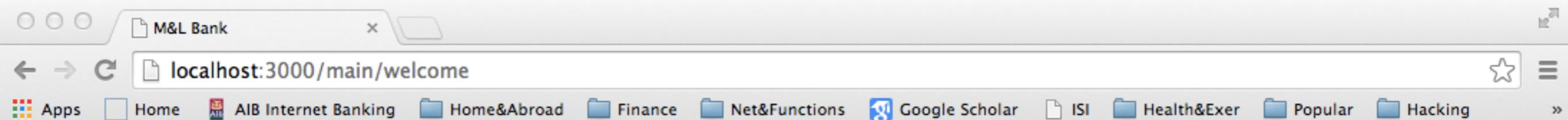
</body>
</html>
```

Two specific files are highlighted with red circles and arrows pointing to them in the code editor:

- 'main_controller.rb' in the controllers directory
- 'start_to_end.erb' in the layouts/main directory

A large blue callout bubble on the right side of the screen contains the text 'start_to_end layout'.

The Site: Screen I



Welcome to Monster & Leinster's Home Page



[click logo to enter]

Monster & Leinster Inc. © 2011

REM

in simple terms, what was a single html file with code, gets broken up into 3-4 files, which combine to create a file that is sent to browser to be rendered

The image shows a developer's workspace with three windows:

- start_to_end.erb**: A layout template containing the text "start_to_end layout". It includes a red arrow pointing to the `<%= yield %>` placeholder.
- main/welcome.html.erb**: A view file containing the following code:

```
<h1><p align="center"> <%= h @message %> to Monster & Leinster's</h1>
<h1><p align="center"> Home Page</h1>
<p align="center"><%= link_to image_tag("monster.jpg"),
:controller => "other", :action => "name" %> </p>
<h5> <p align="center"> [click logo to enter]</p></h5>
```

A red arrow points from this window to the `@message` variable in the controller code.
- main_controller.rb**: A controller file with the following code:

```
class MainController < ApplicationController
layout 'start_to_end'

def welcome
@message = "Welcome"
end

def goodbye
@message = "Goodbye"
@entries = Entry.all
end
```

A red arrow points from this window to the `layout` directive in the layout file.

Below the windows, a large red arrow points from the "welcome content" in the view to the "welcome method" in the controller.

...if you understand this you know how controllers work with views

The Site: Screen I

M&L Bank

localhost:3000/main/welcome

layout bits

Welcome to Monster & Leinster's
Home Page

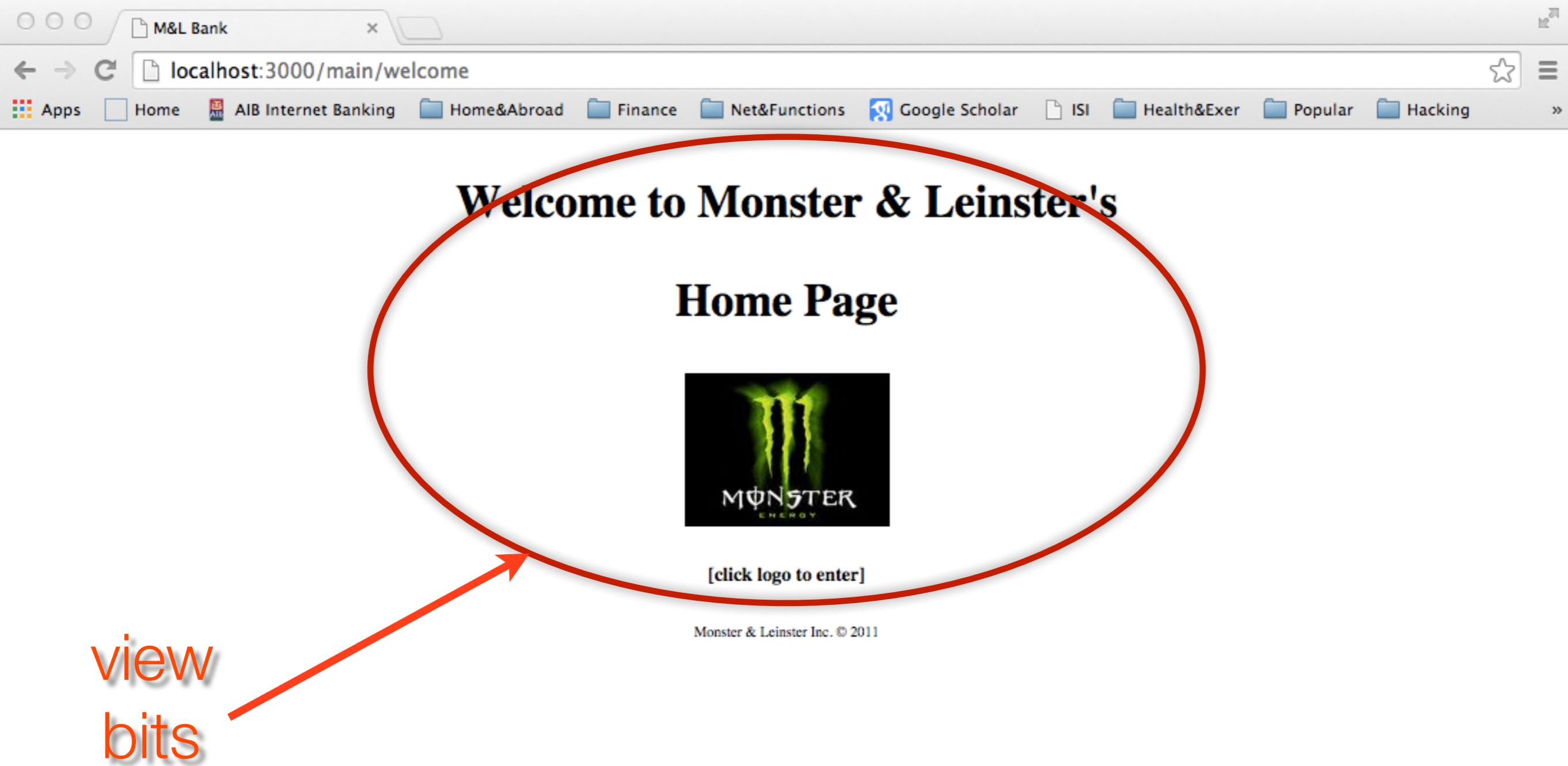
MONSTER ENERGY

[click logo to enter]

Monster & Leinster Inc. © 2011

A screenshot of a web browser window displaying a homepage. The title bar shows 'M&L Bank'. The address bar shows 'localhost:3000/main/welcome'. The page content includes a large heading 'Welcome to Monster & Leinster's Home Page' with a 'MONSTER ENERGY' logo below it. At the bottom, there is a link '[click logo to enter]' and the copyright notice 'Monster & Leinster Inc. © 2011'. Red annotations are present: a red circle highlights the title bar, a red arrow points from the text 'layout bits' to the title bar, another red arrow points from 'layout bits' to the footer area, and a red circle highlights the footer area.

The Site: Screen Begin



The Site: Screen Begin

M&L Bank localhost:3000/main/welcome

Apps Home AIB Internet Banking Home&Abroad Finance Net&Functions Google Scholar ISI Health&Exer Popular Hacking

Welcome to Monster & Leinster's
Home Page

 [click logo to enter]

Monster & Leinster Inc. © 2011

@message bit

A red arrow points from the text '@message bit' to the word 'Welcome' in the main heading.

Part B2(iii):

Walk-through: Screen II-IV ~ Application Layout

Application Controller



Model



render
application.html.erb
with application
spreadsheet



Views

localhost/any/action

Layouts: Application

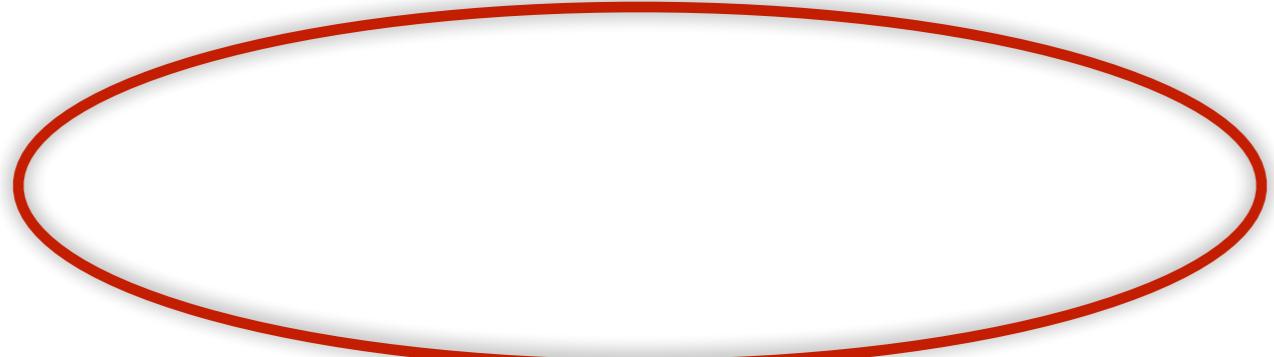
Mozilla Firefox

http://localhost:3000/other/name

Most Visited ▾ Finance ▾ Home&Abroad ▾ Hacking ▾ Net&Functions ▾ News ▾ Wikis ▾ Latest Headlines ▾ NYT > Home Page

We're sorry, but something we... http://localhost:3000/other/name

The Monster & Leinster Bank



Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

Done

Layouts: Application

Mozilla Firefox

http://localhost:3000/other/name

Most Visited ▾ Finance ▾ Home&Abroad ▾ Hacking ▾ Net&Functions ▾ News ▾ Wikis ▾ Latest Headlines ▾ NYT > Home Page ▾

We're sorry, but something we... http://localhost:3000/other/name

The Monster & Leinster Bank

layout bits

specific content will go here...

don't worry where the image is for now

Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

Done

Layouts: Frame

Mozilla Firefox

http://localhost:3000/other/name

Most Visited ▾ Finance ▾ Home&Abroad ▾ Hacking ▾ Net&Functions ▾ News ▾ Wikis ▾ Latest Headlines ▾ NYT > Home Page

We're sorry, but something we... http://localhost:3000/other/name

The Monster & Leinster Bank

...will come from different views

Monster & Leinster Inc accept no responsibility if you fall into arrears on your loan and are responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

banko/app/views/layout/application.html.erb

Done

Set Up Layout

1. for layout we edit:

app/controllers/ application_controller.rb

app/views/layout/application.html.erb

app/assets/stylesheets/application.css

2. for specific content, we edit:

app/controllers/other#name

app/views/other/name.html.erb

...

Set Up Layout

1. for frame we edit:

app/controllers/ application_controller.rb

which we can leave

app/views/layout/application.html.erb

where we add the layout

app/assets/stylesheets/application.css

where we define style

2. for specific content, we edit:

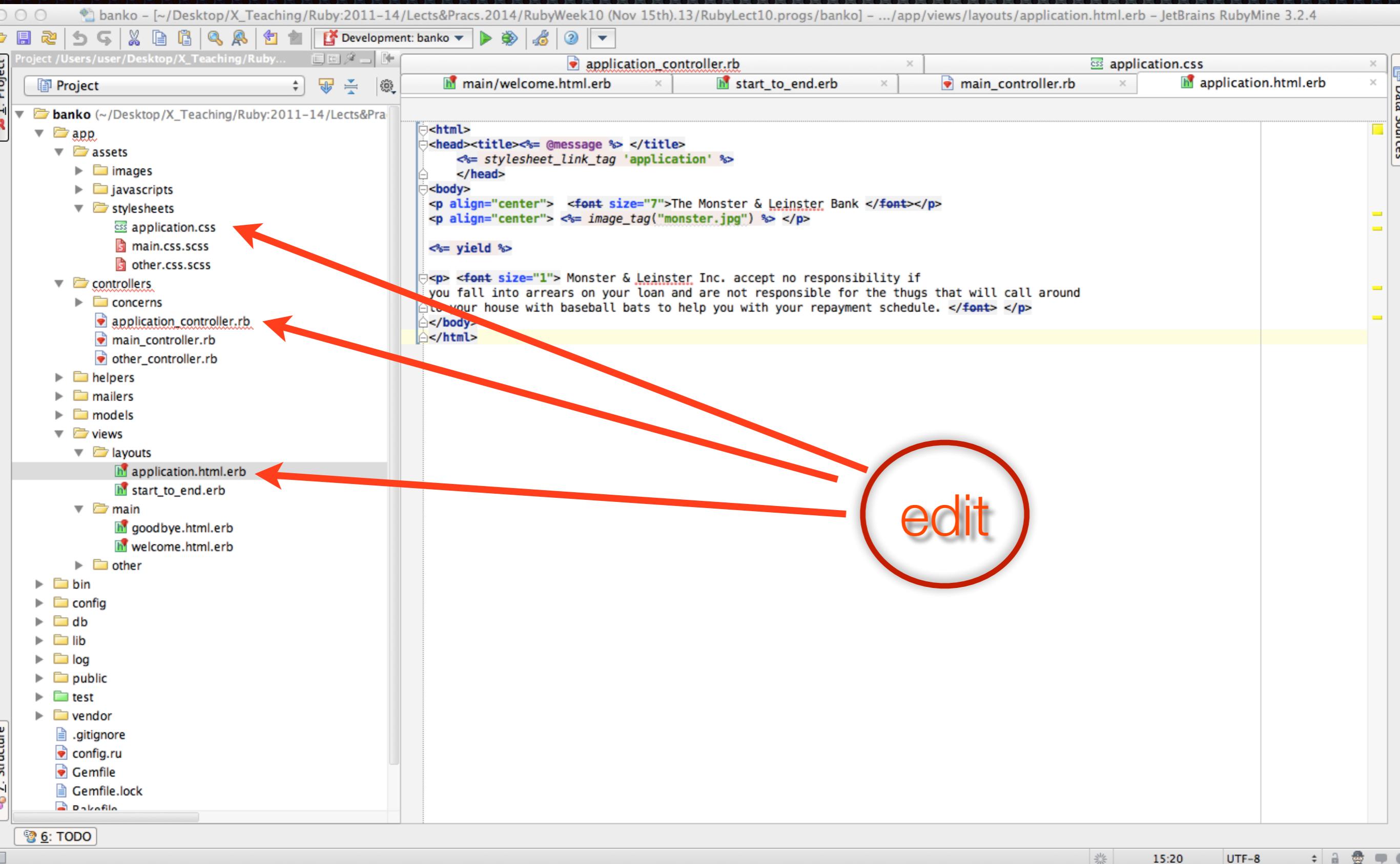
app/controllers/other#name

app/views/other/name.html.erb

just to show something

....

Steps to Set Up Layout



The Monster & Leinster Bank



Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

```
application_controller.rb:2  ↴ class ApplicationController < ActionController
class ApplicationController < ActionController::Base
  #protect_from_forgery
end
```

controller
is empty

The Monster & Leinster Bank



application.html.erb:17

```
<html>
<head><title><%= @message %> </title>
  <%= stylesheet_link_tag 'application' %>
</head>
<body>
<p align="center"> <font size="7">The Monster & Leinster Bank </font></p>
<p align="center"> <%= image_tag("monster.jpg") %> </p>

<%= yield %>

<p> <font size="1"> Monster & Leinster Inc. accept no responsibility if
you fall into arrears on your loan and are not responsible for the thugs that will call around
to your house with baseball bats to help you with your repayment schedule. </font> </p>
</body>
</html>
```

layout
has this

The Monster & Leinster Bank



```
application.css:16 f h1

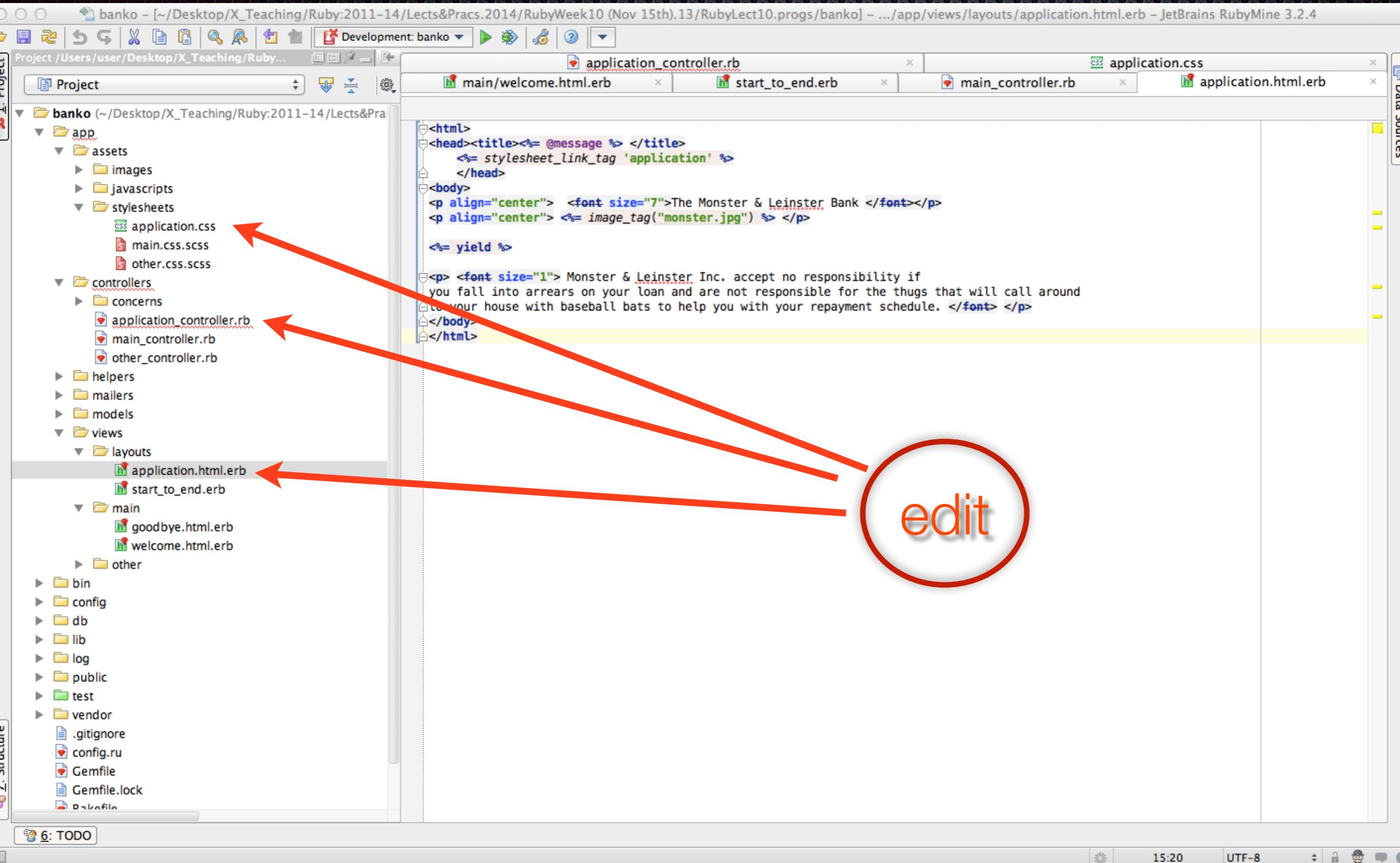
/*
 * This is a manifest file that'll automatically include all the stylesheets available in this
 * and any sub-directories. You're free to add application-wide styles to this file and they'll
 * be included in the compiled file, but it's generally better to create a new file per style scope
 *= require_self
 *= require_tree .
 */

body {font-family:sans-serif;
}

h1 {font-family:times;
   font-size: 24pt;
   font-weight: bold;
   color:#F00 ;
}
```

stylesheet
has this

Steps to Set Up Layout



Specific Content

Ruby on Rails Guides: Layout × Google Image Result for http × localhost:3000/other/name ×

localhost:3000/other/name

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

The Monster & Leinster Bank



We are very pleased to welcome you our home mortgage page.

Please enter your first and last name below:

First Name:

Last Name: send

from name view file

Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are not responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

Steps to Set Up Layout

The screenshot shows the JetBrains RubyMine IDE interface. The title bar indicates the project is named 'banko' and the current file is 'other/name.html.erb'. The left sidebar displays the project structure under 'banko', including 'app' (assets, controllers, helpers, mailers, models, views), 'bin', 'config', and 'db'. The main editor window shows the following ERB code:

```
<p> We are very pleased to welcome you our home mortgage page. </p>
<p> Please enter your first and last name below: </p>

<%= form_tag("name", :method => 'get') do %>
  <p> First Name:
    <%= text_field_tag 'fst_name' %>
  <p> Last Name:
    <%= text_field_tag 'lst_name' %>
    <%= submit_tag 'send' %>
<% end %>

<% if @fname then %>
  <p> Welcome <= h @entry.first_name %>, let's go to the
  <%= link_to 'next page', :controller => "other", :action => "index" %>.</p>
<% end %>
```

A large red watermark-style text overlay on the right side of the code area reads: "this goes into the yield part of the layout".

Part B3:
Looking at what is behind Screen II

The Site: Screen IIa

Ruby on Rails Guides: Layout × Google Image Result for http × localhost:3000/other/name ×

localhost:3000/other/name

Home&Abroad Finance Net&Functions MyCal Google Scholar ISI Health&Exer Popular Hacking Wikis Other Bookmarks 999+

The Monster & Leinster Bank



We are very pleased to welcome you our home mortgage page.

Please enter your first and last name below:

First Name:

Last Name:

Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are not responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

Other Controller...

```
def name
  @fname = params[:fst_name]
  @lname = params[:lst_name]
  @entry = Entry.create({:first_name => @fname, :last_name => @lname})
end

def index
  @person = Entry.last
  @fname = @person.first_name
  @lname = @person.last_name
  @person.update_attributes({:address => params[:st_name],
    :salary => params[:salary], :loan => params[:loan],
    :loan_reason => params[:reason]})
```

The code snippet shows the implementation of several controller actions:

- name**: Creates a new entry with first and last names from the parameters.
- index**: Fetches the last entry from the database and updates its attributes based on the provided parameters.
- show**: Finds the last entry in the database.
- change**: Fetches the last entry and creates a new entry with the same first name and last name, but with salary and loan values from the parameters.
- quote**: Fetches the last entry and calculates a message based on its salary and loan amount. If the loan is more than three times the salary, it's considered a pauper.

The 'name.html.erb' view associated with the 'name' action is also highlighted and circled in red.

Other Controller...

The screenshot shows the JetBrains RubyMine IDE interface. The top bar displays the project name "banko" and the current file path: ".../app/views/other/name.html.erb". The main window has several tabs open: database.yml, main_controller.rb, schema.rb, seeds.rb, main/goodbye.html.erb, start_to_end.erb, other_controller.rb, and routes.rb. The left sidebar shows the project structure under "Project". The "app/controllers" folder contains "other_controller.rb", which is highlighted with a red oval. The "app/views/other" folder contains "name.html.erb", also highlighted with a red oval. A red arrow points from the "name.html.erb" file in the views list to the "name.html.erb" file in the code editor. The code editor displays the following content:

```
<p> We are very pleased to welcome you our home mortgage page. </p>
<p> Please enter your first and last name below: </p>

<%= form_tag("name", :method => 'get') do %>
  <p> First Name:
    <%= text_field_tag 'fst_name' %>
  <p> Last Name:
    <%= text_field_tag 'lst_name' %>
    <%= submit_tag 'send' %>
<% end %>

<% if @fname then %>
  <p> Welcome <%= h @entry.first_name %>, let's go to the
  <%= link_to 'next page', :controller => "other", :action => "index" %>.</p>
<% end %>
```

REM

in simple terms, what was a single html file with code, gets broken up into 3 files, which combine to create a file that is sent to browser to be rendered

```
application.html.erb:17
```

```
<html>
<head><title><%= @message %> </title>
  <%= stylesheet_link_tag 'application' %>
</head>
<body>
<p align="center"> <font size="7">The Monster & Leinster Bank </font></p>
<p align="center"> <%= image_tag("monster.jpg") %> </p>

<%= yield %>

<p> <font size="1"> Monster & Leinster Inc. accept no responsibility if
you fall into arrears on your loan and are not responsible for the thugs that will call around
to your house with baseball bats to help you with your repayment schedule. </font> </p>
</body>
</html>
```

application layout

```
name.html.erb:8
```

```
<p> We are very pleased to welcome you our home mortgage page. </p>
<p> Please enter your first and last name below: </p>

<%= form_tag("name", :method => 'get') do %>
  <p> First Name:<br>
    <%= text_field_tag 'fst_name' %>
  <p> Last Name:<br>
    <%= text_field_tag 'lst_name' %>
  <%= submit_tag 'send' %>
<% end %>

<% if @fname then %>
  <p> Welcome <%= h @entry.first_name %>, let's go to the
  <%= link_to 'next page', :controller => "other", :action => "index" %>.</p>
<% end %>
```

name content

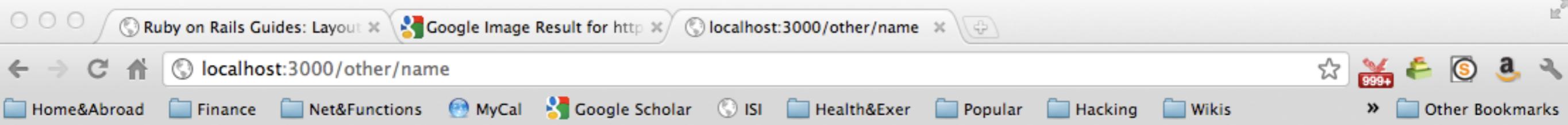
```
other_controller.rb:1
```

```
class OtherController < ApplicationController

  def name
    @fname = params[:fst_name]
    @lname = params[:lst_name]
    @entry = Entry.create({:first_name => @fname, :l
```

name method

The Site: Screen Ia



The Monster & Leinster Bank



We are very pleased to welcome you our home mortgage page.

Please enter your first and last name below:

First Name:

Last Name:

Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are not responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

The Site: Screen IIb

Ruby on Rails Guides: Layout × Google Image Result for http × localhost:3000/other/name? ×

localhost:3000/other/name?utf8=✓&fst_name=Mark&lst_name=Keane&commit=send

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

The Monster & Leinster Bank



We are very pleased to welcome you our home mortgage page.

Please enter your first and last name below:

First Name:

Last Name:

Welcome Mark, let's go to the [next page](#).

Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are not responsib

```
name.html.erb:8
<p> We are very pleased to welcome you our home mortgage page. </p>
<p> Please enter your first and last name below: </p>

<%= form_tag("name", :method => 'get') do %>
  <p> First Name:
    <%= text_field_tag 'fst_name' %>
  <p> Last Name:
    <%= text_field_tag 'lst_name' %>
    <%= submit_tag 'send' %>
<% end %>

<% if @fname then %>
  <p> Welcome <%= h @entry.first_name %>, let's go to the
    <%= link_to 'next page', :controller => "other", :action => "index" %>.</p>
<% end %>
```

Pause...Paws...Pose...

- Ok, so, you need to digest all of this
- we have covered how you set up the initial files
- how you edit those files, how controllers, layouts and views combine to produce the final page
- ...the one remaining big thing is to talk about models
- but...before we do that we'll back up into layouts...



REM:

MVC framework

- ***model-view-controller*** is a software architecture/architectural pattern used in Rails
- ***model*** is domain-specified rep of the data used by the application; manages data and informs of changes
- ***view*** renders model into suitable form for interactions (at interface); may have multiple views of same model
- ***controller*** receives input and initiates responds making calls to model data (operational guts of the app)

Controller



Model

```
$ rake db:migrate
```

Views

```
$ rails server
```

Part C:

A note Rails on functionality...in the foregoing...

Neat Stuff Ia

- Controllers are just Ruby:
 - layout is a new primitive
 - nb inheritance from libraries

```
other_controller.rb:13  class OtherController < ApplicationController
class OtherController < ApplicationController

def name
  @fname = params[:fst_name]
  @lname = params[:lst_name]
  @entry = Entry.create({:first_name => @fname, :last_name => @lname})
end

end
```

Neat Stuff I^b

```
other_controller.rb:13  class OtherController < ApplicationController
class OtherController < ApplicationController

def name
  @entries = Entry.find(:all)
  @entries.each {|entry| entry.destroy}
  @fname = params[:fst_name]
  @lname = params[:lst_name]
  @entry = Entry.create({:first_name => @fname, :last_name => @lname})
  @entries = Entry.find(:all)
end
end
```

db access

db change

form variables

db item create

db access

oval refers to access to tags in the form used; need to look in views

Embedded Ruby

We are very pleased to welcome you our home mortgage page.

Please enter your first and last name below:

First Name:

Last Name:

```
name.html.erb:8
<p> We are very pleased to welcome you our home mortgage page. </p>
<p> Please enter your first and last name below: </p>

<% form_tag("name", :method => 'get') do %>
  <p> First Name:
    <%= text_field_tag 'fst_name' %>
  <p> Last Name:
    <%= text_field_tag 'lst_name' %>
    <%= submit_tag 'send' %>
<% end %>

<% if @fname then %>
  <p> Welcome <%= h @entry.first_name %>, let's go to the
  <%= link_to 'next page', :controller => "other", :action => "index" %>.</p>
<% end %>
```

ERb stuff

sets up form

ERb is inside <% or <%= and %>

ERb arbitrarily embedded within html

Part D:

But, Mark, what about de models...



Recall Banko....

- we just discussed controllers, views and layouts
- but, we said nothing about models
- yet, we did issue a command originally...
- and, performed some edits...

Controller



CHANEL

Model

\$ rake db:migrate



Views

\$ rails server

Steps

1. \$ rails new **banko**
set up dir
go to dir
2. \$ cd bank
3. \$ rails generate controller main welcome
4. \$ rails generate controller other name
5. \$ rails generate model Entry first_name:string ...
create model
called **entry**
with field:type
6. lets...edit a whole bunch of files...
7. \$ rake db:migrate
\$ rake db:rollback

REM:

\$ rake db:migrate:reset
clears out all records

set up tables

rollback to early migration of db

Countries: Define tables REM:

```
ActiveRecord::Schema.define do
  create_table :countries do |table|
    table.column :name, :string
    table.column :continent, :string
    table.column :size, :integer
  end

  create_table :regions do |table|
    table.column :country_id, :integer
    table.column :region_size, :integer
    table.column :name, :string
  end
end
...
```

neatdb.rb

Rails uses ActiveRecord
via Rake

- two tables
- region references the country it is part of



Countries: Create associations **REM:**

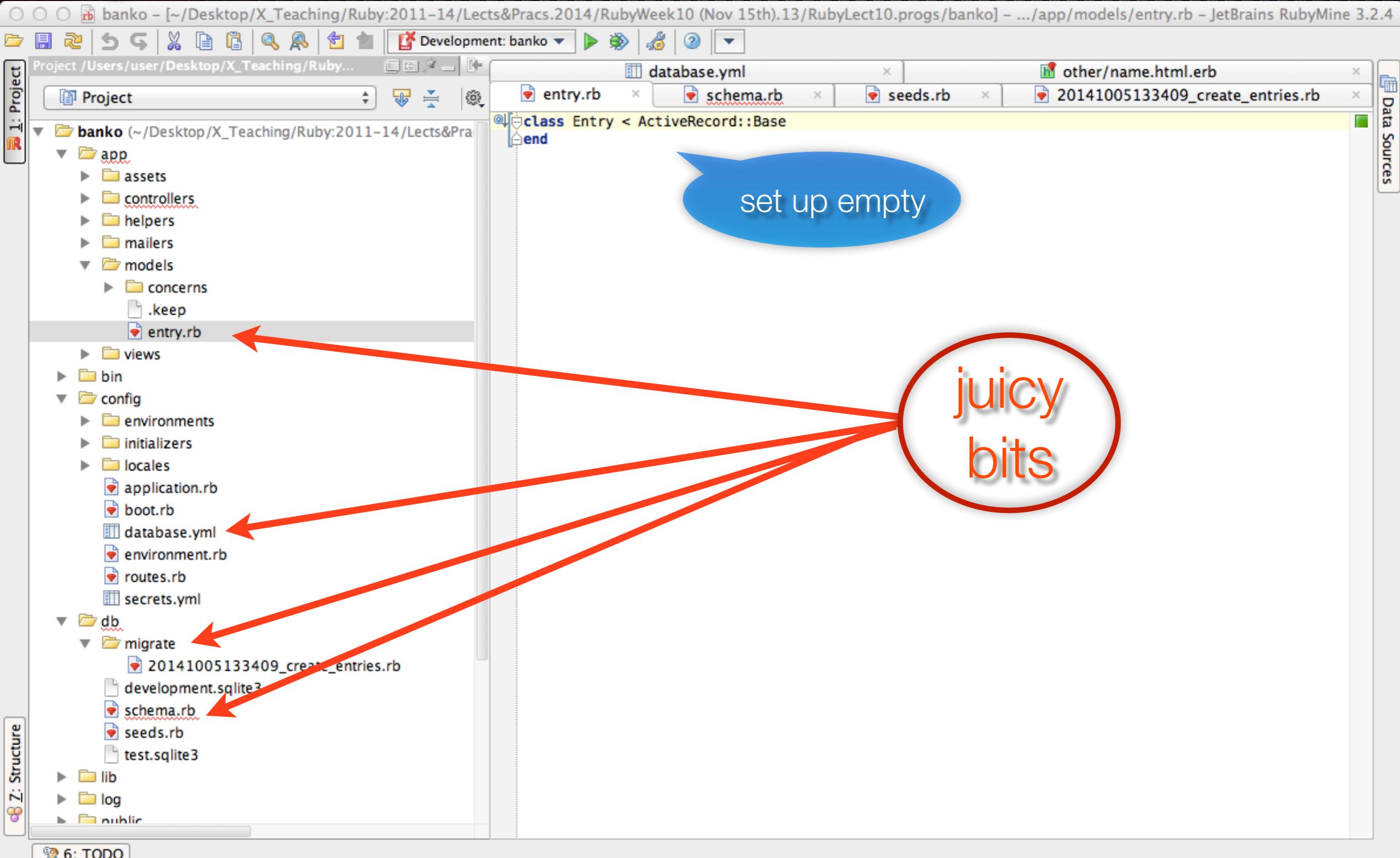
```
class Country < ActiveRecord::Base  
  has_many :regions  
end  
  
class Region < ActiveRecord::Base  
  belongs_to :country  
end  
...
```

neatdb.rb

- **Country** and **Region** are subclasses of ActiveRecord
- describes relationships between models (tables)
- here, **has_many** and **belongs_to** define a 1:n



generate model Entry...does



generate model Entry...does

The screenshot shows the JetBrains RubyMine 3.2.4 IDE interface. The title bar indicates the project is named "banko" and the current file is "app/models/entry.rb". The left sidebar displays the project structure under "banko", with the "entry.rb" file highlighted by a red oval. The main editor window shows the following code:

```
class Entry < ActiveRecord::Base
end
```

A large blue speech bubble points from the text "model Entry" in the explanatory text below to the word "Entry" in the code. The explanatory text reads:

model **Entry**
where you put the
has_many and
belongs_to stuff

The bottom status bar shows "6: TODO".

generate model Entry...does

The screenshot shows a Java-based IDE (like IntelliJ IDEA) with a Rails project named 'banko'. The project structure on the left includes 'app', 'config', 'db', and 'lib' directories. The 'db/migrate' folder contains a file named '20141005133409_create_entries.rb', which is circled in red at the bottom left. The main editor window displays the content of this migration file:

```
class CreateEntries < ActiveRecord::Migration
  def change
    create_table :entries do |t|
      t.string :first_name
      t.string :last_name
      t.string :address
      t.integer :loan
      t.string :loan_reason
      t.integer :salary
      t.timestamps
    end
  end
end
```

Below the code, there are tabs for 'database.yml', 'entry.rb', 'schema.rb', 'seeds.rb', and 'other/name.html.erb'. The 'entry.rb' tab is currently selected.

**\$ rails generate model Entry first_name:string
last_name:string address:string loan:integer...**

OR

you can write it directly into the migrate file

generate model Entry...does

The screenshot shows a Java-based IDE (like IntelliJ IDEA) with a Rails project named 'banko'. The project structure on the left includes 'app', 'config', 'db', and 'lib' directories. The 'db/migrate' folder contains a file named '20141005133409_create_entries.rb'. A red circle highlights this file. The main editor window shows the content of 'entry.rb':

```
class CreateEntries < ActiveRecord::Migration
  def change
    create_table :entries do |t|
      t.string :first_name
      t.string :last_name
      t.string :address
      t.integer :loan
      t.string :loan_reason
      t.integer :salary
      t.timestamps
    end
  end
end
```

A blue speech bubble points to the code with the text: "class to create entries, Entry.up sets up tables and Entry.down gets rid of them". Another blue speech bubble points to the 'db/migrate' folder with the text: "add a bunch of fields, specifying their type; nb db gives :id for primary key". Red text at the bottom provides alternative commands:

\$ rails generate model Entry first_name:string
last_name:string address:string loan:integer...
OR
you can write it directly into the migrate file

rb banko - [/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek10 (Nov 15th).13/RubyLect10.progs/banko] - .../db/schema.rb - JetBrains RubyMine 3.2.

Development: banko

entry.rb schema.rb seeds.rb 20141005133409_create_entries.rb database.yml

Project

banko (~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek10 (Nov 15th).13/RubyLect10.progs/banko)

app

- assets
- controllers
- helpers
- mailers
- models
 - concerns
 - .keep
 - entry.rb
- views

bin

config

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

db

migrate

- 20141005133409_create_entries.rb
- development.sqlite3
- schema.rb
- seeds.rb
- test.sqlite3

lib

log

public

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: 20141005133409) do
 create_table "entries", force: true do |t|
 t.string "first_name"
 t.string "last_name"
 t.string "address"
 t.integer "loan"
 t.string "loan_reason"
 t.integer "salary"
 t.datetime "created_at"
 t.datetime "updated_at"
 end
end

schema produced automatically by rake when you do \$ rake db:migrate

\$ rake db:rollback steps back 'true earlier migrations'

banko - [/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek10 (Nov 15th).13/RubyLect10.progs/banko] - .../config/database.yml - JetBrains RubyMine 3.2

Project /Users/user/Desktop/X_Teaching/Ruby... Development: banko entry.rb schema.rb seeds.rb 20141005133409_create_entries.rb database.yml

Project

banko (~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pra...

app assets controllers helpers mailers models concerns .keep entry.rb views bin config environments initializers locales application.rb boot.rb database.yml environment.rb routes.rb secrets.yml db migrate 20141005133409_create_entries.rb development.sqlite3 schema.rb seeds.rb test.sqlite3 lib log public

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

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

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

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

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

config file for database

banko - [/Users/user/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek10 (Nov 15th).13/RubyLect10.progs/banko] - .../db/seeds.rb - JetBrains RubyMine 3.2.4

Project /Users/user/Desktop/X_Teaching/Ruby... Development: banko entry.rb schema.rb seeds.rb 20141005133409_create_entries.rb database.yml

Project

banko (~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pra

app

- assets
- controllers
- helpers
- mailers
- models
 - concerns
 - .keep
 - entry.rb
- views

bin

config

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

db

migrate

- 20141005133409_create_entries.rb
- development.sqlite3
- schema.rb
- seeds.rb
- test.sqlite3

lib

log

public

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)

can introduce seed records here

Steps

1. \$ rails new bank
set up dir
 2. \$ cd banko
go to dir
 3. \$ rails generate controller main welcome
 4. \$ rails generate controller other name
 5. \$ rails generate model Entry first_name:string ...
create model called **entry** with **field:type**
 6. lets...edit a whole bunch of files...
 7. \$ rake db:migrate
set up tables
- \$ rake db:rollback
rollback to early stage of db

REM:

\$ rake db:migrate:reset
clears out all records

The Site: Screen VI

REM:

M&L Bank localhost:3000/main/goodbye

Apps Home AIB Internet Banking Home&Abroad Finance Net&Functions Google Scholar ISI Health&Exer Popular

what's this !

Goodbye to Monster & Leinster's End Page

MONSTER ENERGY

[click logo to re-enter site]

• ,,,1
• mark,keane,,2
• mark,keane,2000,31223
• mark,keane,,4
• mark,keane,12,20000005
• ,,,6
• Mark,Keane,5000,1000007
• ,,,8
• ,,,9
• Mark,Keane,5000,10000010



Monster & Leinster Inc. © 2011

banko - [/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek10 (Nov 15th).13/RubyLect10.progs/banko] - .../app/views/main/goodbye.html.erb - Jet

Development: banko

Project

banko (~/Desktop/X_Teaching/Ruby:
app
assets
controllers
concerns
application_controller.rb
main_controller.rb
other_controller.rb
helpers
mailers
models
views
layouts
main
goodbye.html.erb
welcome.html.erb
other
bin
config
db
lib
log
public
test
vendor
.gitignore
config.ru
Gemfile
Gemfile.lock
Rakefile
README.rdoc
External Libraries

6: TODO

```
20141005133409_create_entries.rb
```

entry.rb schema.rb seeds.rb main/goodbye.html.erb database.yml

```
h1
  <h1><p align="center"> <%= h @message %> to Monster & Leinster's</h1>

  <h1><p align="center"> End Page</h1>

  <p align="center"><%= link_to image_tag("monster.jpg"),
    :controller => "main", :action => "welcome" %> </p>

  <h5> <p align="center"> [click logo to re-enter site] </p></h5>

  <ul>
    <% @entries.each do |entry| %>
      <li> <%=h entry.first_name %>, <%=h entry.last_name %> , <%=h entry.salary %>, <%=h entry.loan %> <%=h entry.id %>
    <% end %>
  </ul>
```

Problem Banko...

- this, finally, clears up the mystery of what all the crap on the goodbye page is about...
- however, it also shows that something odd is going on in our db
- why are there two entries, some of which have no values? What's the other crap about ?
- ...a question not to be asked...

Part E:

Some Residuals in Banko... controllers, routing & images

Basic Banko Functionality...

- note, that each screen has an associated, controller-method, view and layout
- controllers/main_controller.rb has two methods **welcome** and **goodbye**
- controllers/other_controller.rb has four methods **index**, **name**, **show**, **change**, and **quote**
- there are all fairly simple...its the co-ordination that is complicated

The Main Controller

A screenshot of a Java IDE (JetBrains) showing the code for the `main_controller.rb` file. The code defines a `MainController` class that inherits from `ApplicationController` and uses a layout named `'start_to_end'`. It contains two methods: `welcome` and `goodbye`. Both methods set a message and retrieve all entries from the database. The `welcome` method destroys each entry, while the `goodbye` method only retrieves them. The code ends with an `end` keyword.

```
class MainController < ApplicationController
  layout 'start_to_end'

  def welcome
    @message = "Welcome to DB version"
    @entries = Entry.all
    @entries.each { |entry| entry.destroy}
  end

  def goodbye
    @message = "Goodbye to the Db version"
    @entries = Entry.all
  end
end
```

The `main_controller.rb` file is highlighted in the code editor. The `other_controller.rb` file is also visible in the editor tab bar. A red oval highlights the `main_controller.rb` file in the project tree on the left, which lists the `app/controllers` directory containing `application_controller.rb`, `main_controller.rb`, and `other_controller.rb`.

The Other Controller

Organizer

Build Clean Run Action

PROJECTS & SOURCES

- hellodb
 - app
 - assets
 - controllers
 - application_controller.rb
 - main_controller.rb
 - other_controller.rb
 - helpers
 - mailers
 - models
 - views
 - config
 - config.ru
 - db
 - doc
 - Gemfile
 - Gemfile.lock
 - lib
 - log
 - public
 - Rakefile
 - README
 - script
 - test
 - tmp
 - vendor

other_controller.rb:42 class OtherController < ApplicationController

```
class OtherController < ApplicationController

  def name
    @fname = params[:fst_name]
    @lname = params[:lst_name]
    @entry = Entry.create({:first_name => @fname, :last_name => @lname})
  end

  def index
    @person = Entry.find(:last)
    @fname = @person.first_name
    @lname = @person.last_name
    @person.update_attributes({:address => params[:st_name],
      :salary => params[:salary], :loan => params[:loan],
      :loan_reason => params[:reason]})
    if !@person.address.nil? then render "show" end
  end

  def show
    @person = Entry.find(:last)
  end

  def change
    @person = Entry.find(:last)
    @fname = @person.first_name
    @lname = @person.last_name
    @entry = Entry.create({:first_name => @fname, :last_name => @lname,
      :salary => params[:salary], :loan => params[:loan]})

  end

  def quote
    @person = Entry.find(:last)
    if !@person.salary.nil? then
      if (@person.salary * 3) < @person.loan
        then @message = "You goddam pauper, you asked us for a loan that is:
          more than three times your salary. \n Please go back
          and re-enter your correct salary."
      else @message = "Sure thing baby..."
      end
    else @message = "Seem to have an empty record??"
    end
  end
end
```

Banko Routing...

- **Rails** style is sort-of weird in the way it creates objects and calls methods
- essentially, methods run by linking to pages with uri-s; when a page is entered the method runs
- action tags in forms also call methods (e.g., **submit_tag**; they are usually helper methods)
- obviously, methods call other methods too (more conventionally)

Banko Routing...

- this means that routing becomes very important, as it is a sort-of shorthand for calling methods
- will return to it in detail, but just note it here
- btw, its is often a source of error in the browser, as you fail to address the page with the right uri
- what is the “right uri” will be set by stuff in banko/config/routes.rb



will
cause
bugs

routes.rb on setup....

these were created automatically

```
Rails.application.routes.draw do
  get 'other/name'

  get 'other/index'

  get 'other/show'

  get 'other/change'

  get 'other/quote'

  get 'main/welcome'

  get 'main/goodbye'

  # The priority is based upon order of creation: first created -> highest priority.
  # See how all your routes lay out with "rake routes".

  # You can have the root of your site routed with "root"
  # root 'welcome#index'

  # Example of regular route:
  #   get 'products/:id' => 'catalog#view'

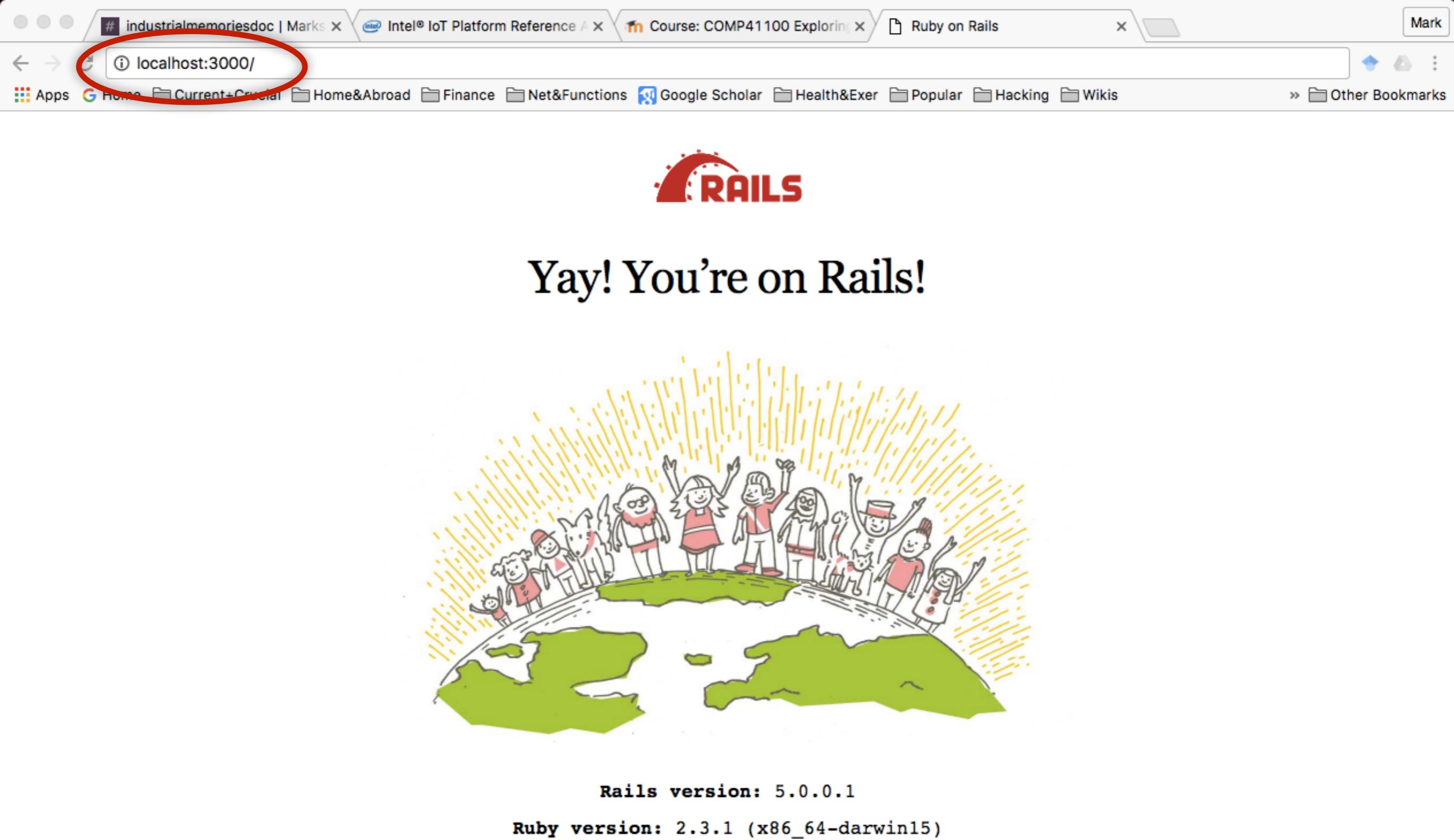
  # Example of named route that can be invoked with purchase_url(id: product.id)
  #   get 'products/:id/purchase' => 'catalog#purchase', as: :purchase
end
```

The screenshot shows the JetBrains RubyMine IDE interface. The left sidebar displays the project structure for a 'banko' application, with the 'routes.rb' file circled in red at the bottom of the 'config' folder. The main editor window shows the content of the 'routes.rb' file, which defines several GET routes for 'other' and 'main' controllers. A large blue speech bubble points to the first few routes, with the text 'these were created automatically'. The status bar at the bottom right shows the time as 28:1 and the encoding as UTF-8.

Banko Routing...

- one of the things you can do is change the routing statements in this routes.rb file
- often people want to have the browser default connect to the first page in your site

not like this...



routes.rb...what we do

Project /Users/user/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek10 (Nov 15th).13/RubyLect10.progs/bankodb - .../config/routes.rb - JetBrains RubyMine 3.2.4

1: Project

Project banko [banko] (~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek10 (Nov 15th).13/RubyLect10.progs/bankodb)

- app
- assets
- controllers
 - concerns
 - application_controller.rb
 - main_controller.rb
 - other_controller.rb
 - helpers
 - mailers
 - models
 - views
- bin
- config
 - environments
 - initializers
 - locales
 - application.rb
 - boot.rb
 - database.yml
 - environment.rb
 - routes.rb
 - secret.yml
- db
- lib
- log
- public
- test
- vendor
 - .gitignore
 - config.ru
 - Gemfile
 - Gemfile.lock
 - Rakefile
 - README.rdoc
- External Libraries

Development: banko

entry.rb schema.rb seeds.rb main/goodbye.html.erb
other/name.html.erb other_controller.rb main_controller.rb
routes.rb 20141005133409_create_entries.rb database.yml

```
Rails.application.routes.draw do
  get 'other/name'
  get 'other/index'
  get 'other/show'
  get 'other/change'
  get 'other/quote'
  get 'main/welcome'
  get 'main/goodbye'

  # The priority is based upon order of creation: first created → highest priority.
  # See how all your routes lay out with "rake routes".
  #
  # You can have the root of your site routed with "root"
  root 'main#welcome'

  # Example of regular route:
  #   get 'products/:id' => 'catalog#view'

  # Example of named route that can be invoked with purchase_url(id: product.id)
  #   get 'products/:id/purchase' => 'catalog#purchase', as: :purchase

  # Example resource route (maps HTTP verbs to controller actions automatically):
  #   resources :products

  # Example resource route with options:
  #   resources :products do
  #     member do
  #       get 'short'
  #       post 'toggle'
  #     end
  #     #
  #     collection do
  #       get 'sold'
  #     end
  #   end

  # Example resource route with sub-resources:
  #   resources :products do
  #     resources :comments, :sales
  #   end

  # And finally, you can name your routes
  #   get 'products/:id/comments/:comment_id' => 'comments#destroy', as: :comment_destroy
end
```

uncomment this line and refer it to the controller#method of the first page

so, we now get this...

Safari

ActionView::Helpers::FormTag

TeachingPipe

M&L Bank

localhost:3000

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

999+

Welcome to DB version to Monster & Leinster's Home Page

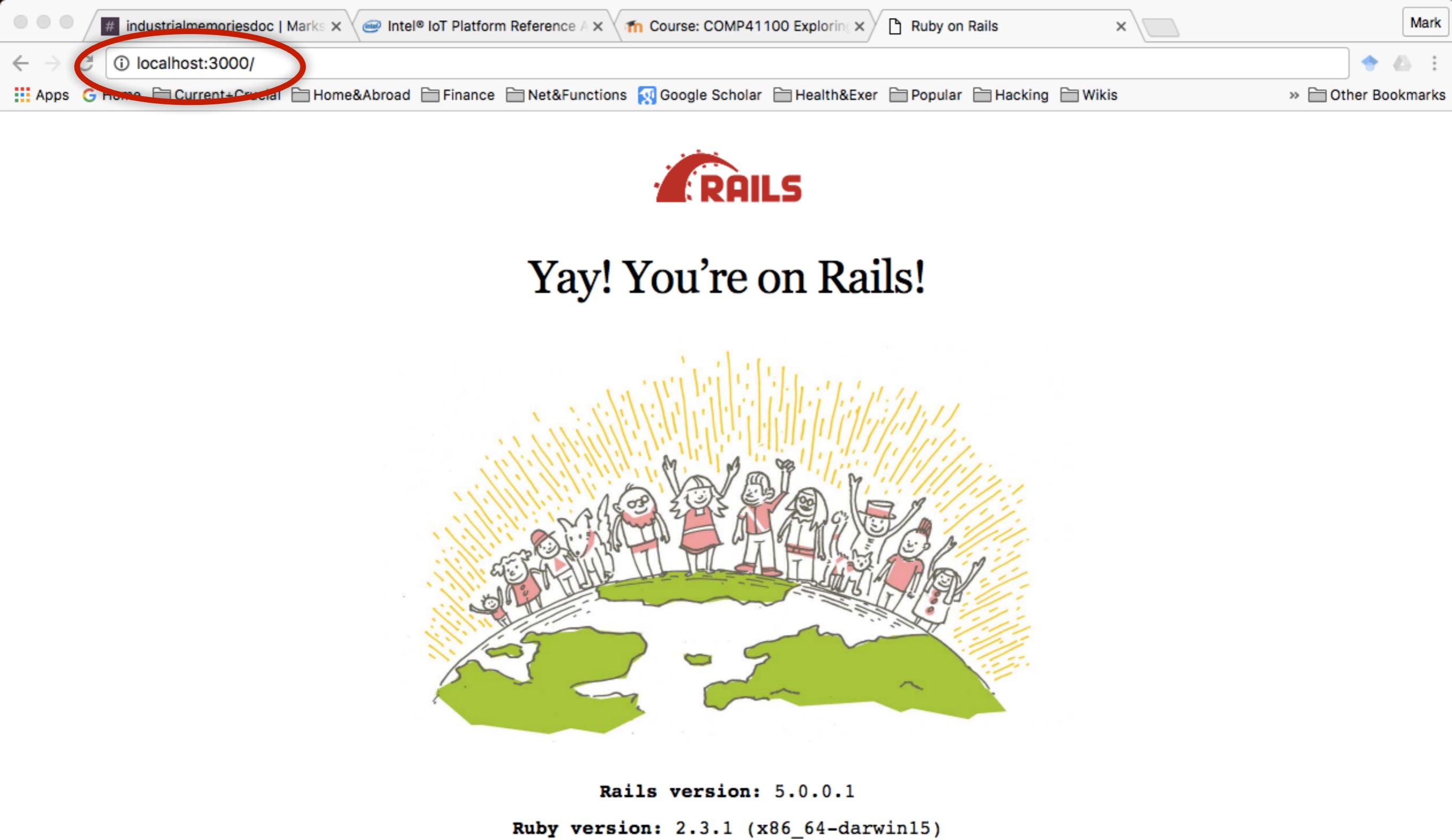
MONSTER ENERGY

[click logo to enter]

Monster & Leinster Inc. © 2011

The screenshot shows a web browser window with multiple tabs open. The active tab displays a modified homepage for 'Monster & Leinster'. The URL 'localhost:3000' is circled in red at the top left of the browser. The main content of the page is a welcome message: 'Welcome to DB version to Monster & Leinster's Home Page'. Below this, there is a large 'MONSTER ENERGY' logo with glowing green vertical bars. A link '[click logo to enter]' is positioned below the logo. At the bottom of the page, the copyright notice 'Monster & Leinster Inc. © 2011' is visible. The browser interface includes standard navigation buttons (back, forward, search) and a menu bar with various links like 'Home&Abroad', 'Finance', 'Net&Functions', etc.

Instead of this...



BY THE WAY..

user\$ rake routes

Prefix	Verb	URI Pattern	Controller#Action
other_name	GET	/other/name(.:format)	other#name
other_index	GET	/other/index(.:format)	other#index
other_show	GET	/other/show(.:format)	other#show
other_change	GET	/other/change(.:format)	other#change
other_quote	GET	/other/quote(.:format)	other#quote
main_welcome	GET	/main/welcome(.:format)	main#welcome
main_goodbye	GET	/main/goodbye(.:format)	main#goodbye
root	GET	/	main#welcome

user\$

Where are Images ?



- **Rails** stores images in banko/app/assets/images
- they are then referenced in various view and layout files



Where are images...

banko - [/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek10 (Nov 15th).13/RubyLect10.progs/bankodb] - .../app/assets/images/monster.jpg - JetBrains...

Project /Users/user/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek10 (Nov 15th).13/RubyLect10.progs/bankodb

Development: banko

face.jpg routes.rb monster.jpg environment.rb

143x107 JPEG (24-bit color) 2.54Kb

Project

1: Project

bankodb [banko] (~/Desktop/X_Teaching/Ruby:2011-14/Lects&Pracs.2014/RubyWeek10 (Nov 15th).13/RubyLect10.progs/bankodb)

- app
 - assets
 - images
 - .keep
 - face.jpg
 - monster.jpg
 - javascripts
 - stylesheets
 - controllers
 - concerns
 - application_controller.rb
 - main_controller.rb
 - other_controller.rb
 - helpers
 - mailers
 - models
 - views
 - bin
 - config
 - db
 - lib
 - log
 - public
 - test
 - vendor
 - .gitignore
 - config.ru
 - Gemfile
 - Gemfile.lock
 - Rakefile
 - README.rdoc

External Libraries

Data Sources

The screenshot shows a JetBrains IDE interface with a project named "banko". The "Project" tool window on the left displays the file structure of the "app" directory, specifically the "assets/images" folder. Inside this folder, two files are listed: "face.jpg" and "monster.jpg". The "monster.jpg" file is circled in red. To the right of the project tree, there is a preview area showing the "monster.jpg" image, which is the iconic green "MONSTER ENERGY" logo. The status bar at the bottom right indicates the image's dimensions as 143x107 pixels, its color depth as 24-bit color, and its size as 2.54Kb.

Layouts: Frame

Mozilla Firefox

http://localhost:3000/other/name

Most Visited ▾ Finance ▾ Home&Abroad ▾ Hacking ▾ Net&Functions ▾ News ▾ Wikis ▾ Latest Headlines ▾ NYT > Home Page ▾

We're sorry, but something we... http://localhost:3000/other/name

The Monster & Leinster Bank



Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

Done

Where are images...ref'ed

```
application.html.erb:17
<html>
<head><title><%= @message %> </title>
  <%= stylesheet_link_tag 'application' %>
</head>
<body>
<p align="center"> <font size="7">The Monster & Leinster Bank </font></p>
<p align="center"> <%= image_tag("monster.jpg") %> </p>
<%= yield %>

<p> <font size="1"> Monster & Leinster Inc. accept no responsibility if
you fall into arrears on your loan and are not responsible for the thugs that will call around
to your house with baseball bats to help you with your repayment schedule. </font> </p>
</body>
</html>
```

Part F:

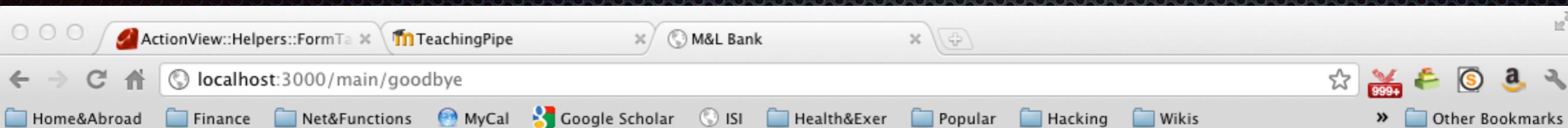
Fixing Banko's Problems...

Problem Banko...

REM:

- if you look at the goodbye page, which prints out the state of the db at the end; it progressively seems to spit out loads of crap
- why do we have these odd entries...

The Problem



Goodbye to Monster & Leinster's

End Page



[click logo to re-enter site]

- , , , 1
- Mark, Keane , 2000, 1000000 2
- , , , 3
- mark, keane , 1000, 100000 4
- , , , 5
- Sal, Paradiso , 30000, 1000 6

Monster & Leinster Inc. © 2011

The Problem starts here...

Ruby on Rails Guides: Layout × Google Image Result for http × localhost:3000/other/name ×

localhost:3000/other/name

Home&Abroad Finance Net&Functions MyCal Google Scholar ISI Health&Exer Popular Hacking Wikis Other Bookmarks 999+

The Monster & Leinster Bank



We are very pleased to welcome you our home mortgage page.

Please enter your first and last name below:

First Name:

Last Name:

Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are not responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

...look at the web server logs...

Log....

Started GET "/other/name" for 127.0.0.1 at 2011-10-18 12:36:09 +0100

Processing by OtherController#name as HTML

SQL (139.9ms) INSERT INTO "entries" ("address", "created_at", "first_name", "last_name", "loan", "loan_reason", "salary", "updated_at") VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?) [[{"address": nil}, {"created_at": "Tue, 18 Oct 2011 11:36:10 UTC +00:00"}, {"first_name": nil}, {"last_name": nil}, {"loan": nil}, {"loan_reason": nil}, {"salary": nil}, {"updated_at": "Tue, 18 Oct 2011 11:36:10 UTC +00:00"}]

Rendered other/name.html.erb within layouts/application (20.0ms)

Completed 200 OK in 445ms (Views: 159.8ms | ActiveRecord: 140.9ms)

...

Started GET "/other/name?utf8=%E2%9C%93&fst_name=Mark&lst_name=Keane&commit=send" for 127.0.0.1 at 2011-10-18 12:36:21 +0100

Processing by OtherController#name as HTML

Parameters: {"utf8": "✓", "fst_name": "Mark", "lst_name": "Keane", "commit": "send"}

SQL (1.5ms) INSERT INTO "entries" ("address", "created_at", "first_name", "last_name", "loan", "loan_reason", "salary", "updated_at") VALUES (?, ?, ?, ?, ?, ?, ?, ?) [[{"address": nil}, {"created_at": "Tue, 18 Oct 2011 11:36:21 UTC +00:00"}, {"first_name": "Mark"}, {"last_name": "Keane"}, {"loan": nil}, {"loan_reason": nil}, {"salary": nil}, {"updated_at": "Tue, 18 Oct 2011 11:36:21 UTC +00:00"}]

Rendered other/name.html.erb within layouts/application (1.3ms)

Completed 200 OK in 66ms (Views: 7.4ms | ActiveRecord: 2.2ms)

What's the problem...

- Well, remember what I said about the way Rails works by calling methods when you move pages
- If you look at the log, you can see that it creates one entry when you enter the name-page and another when you press the **send** tag
- So, how can we fix this ?

The Site: Screen IIa

The screenshot shows a web browser window with three tabs open: "Ruby on Rails Guides: Layout", "Google Image Result for http", and "localhost:3000/other/name". The main content area displays a large title "The Monster & Leinster Bank" above a "MONSTER ENERGY" logo. Below the logo, there is a welcome message and two input fields for "First Name" and "Last Name". A small note at the bottom states: "Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are not responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule."

Ruby on Rails Guides: Layout × Google Image Result for http × localhost:3000/other/name ×

localhost:3000/other/name

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

999+

The Monster & Leinster Bank



We are very pleased to welcome you our home mortgage page.

Please enter your first and last name below:

First Name:

Last Name:

Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are not responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

other#name Controller...

```
def name
  @fname = params[:fst_name]
  @lname = params[:lst_name]
  @entry = Entry.create({:first_name => @fname, :last_name => @lname})
end

def index
  @person = Entry.last
  @fname = @person.first_name
  @lname = @person.last_name
  @person.update_attributes({:address => params[:st_name],
    :salary => params[:salary], :loan => params[:loan],
    :loan_reason => params[:reason]})  
if !@person.address.nil? then render "show" end
end

def show
  @person = Entry.find(:last)
end

def change
  @person = Entry.last
  @fname = @person.first_name
  @lname = @person.last_name
  @entry = Entry.create({:first_name => @fname, :last_name => @lname, :salary => params[:salary], :loan => params[:loan]})  
end

def quote
  @person = Entry.last
  if !@person.salary.nil? then
    if (@person.salary * 3) < @person.loan
      then @message = "You goddam pauper, you asked us for a loan that is:  
more than three times your salary. \n Please go back  
and re-enter your correct salary."
    else @message = "Sure thing baby..."
    end
  else @message = "Seem to have an empty record??"
  end
end
```

other/name View...

The screenshot shows the JetBrains RubyMine IDE interface. The title bar indicates the project is named 'banko' and the current file is 'other/name.html.erb'. The left sidebar displays the project structure under 'Project' and 'Structure' tabs. The main editor area shows the content of the 'other/name.html.erb' file:

```
<p> We are very pleased to welcome you our home mortgage page. </p>
<p> Please enter your first and last name below: </p>

<%= form_tag("name", :method => 'get') do %>
  <p> First Name:
    <%= text_field_tag 'fst_name' %>
  <p> Last Name:
    <%= text_field_tag 'lst_name' %>
    <%= submit_tag 'send' %>
<% end %>

<% if @fname then %>
  <p> Welcome <%= h @entry.first_name %>, let's go to the
  <%= link_to 'next page', :controller => "other", :action => "index" %>.</p>
<% end %>
```

The code uses ERB syntax for generating HTML. It includes a form for entering a first and last name, and a conditional block that displays a welcome message and a link to the next page if a first name has been entered.

What's the problem...

- Well, remember what I said about the way Rails works by calling methods when you move pages
- If you look at the log, you can see that it creates one entry when you enter the name-page and another when you press the **send** tag
- So, how can we fix this ?

The Change...

```
class OtherController < ApplicationController
  def name
    @fname = params[:fst_name]
    @lname = params[:lst_name]
    if @fname then @entry = Entry.create({:first_name => @fname, :last_name => @lname}) end
  end

  def index
    @person = Entry.last
    @fname = @person.first_name
    @lname = @person.last_name
    @person.update_attributes({:address => params[:st_name],
      :salary => params[:salary], :loan => params[:loan],
      :loan_reason => params[:reason] })
    if !@person.address.nil? then render "show" end
  end

  def show
    @person = Entry.find(:last)
  end

  def change
    @person = Entry.last
    @fname = @person.first_name
    @lname = @person.last_name
    @entry = Entry.create({:first_name => @fname, :last_name => @lname, :salary => params[:salary], :loan => params[:loan] })
  end

  def quote
    @person = Entry.last
    if !@person.salary.nil? then
      if (@person.salary * 3) < @person.loan
        then @message = "You goddam pauper, you asked us for a loan that is:
          more than three times your salary. \n Please go back
          and re-enter your correct salary."
      else @message = "Sure thing baby..."
      end
    else @message = "Seem to have an empty record??"
    end
  end
end
```

...also should change Main

The screenshot shows a Java-based IDE interface with a dark theme. The top bar displays the project name "banko" and the file path "RubyWeek10 (Nov 15th).13/RubyLect10.progs/bankodb - .../app/controllers/main_controller.rb - JetB...". The toolbar includes standard icons for file operations like Open, Save, and Find.

The left sidebar is a "Project" view showing the directory structure of the "bankodb" project. It includes folders for app, bin, config, db, lib, log, public, test, vendor, and various files like .gitignore, config.ru, Gemfile, Gemfile.lock, Rakefile, and README.rdoc. The "app/controllers" folder contains files for entry.rb, schema.rb, seeds.rb, main/goodbye.html.erb, other/name.html.erb, other_controller.rb, main_controller.rb, and application_controller.rb. The "main_controller.rb" file is currently selected and shown in the main editor area.

The main editor area displays the following Ruby code:

```
class MainController < ApplicationController
  layout 'start_to_end'

  def welcome
    @message = "Welcome to DB version"
    @entries = Entry.all
    @entries.each { |entry| entry.destroy}
  end

  def goodbye
    @message = "Goodbye to the Db version"
    @entries = Entry.all
  end
end
```

A large blue callout bubble points from the bottom right towards the code, containing the text: "here we clear the db records; nb table structure still there".

Last Screen...BANG !



Goodbye to Monster & Leinster's

End Page



[click logo to re-enter site]

• mark, keane , 2000, 100000 9

Monster & Leinster Inc. © 2011

PC Problems: Path Var

To create a new rails website in Windows 8 follow the lecture instructions below:

1. Run command prompt
2. **gem install rails**
3. **gem update --system**
4. **gem install bundle**
5. **bundle install**
6. Go to your Ruby Project directory: **cd C:\RubyMine**
7. Make a new directory: **mkdir temp**
8. **cd temp**
9. **rails new newsite**
10. **cd newsite**
11. **rails generate controller test hello**
12. look at /newsite/app/controllers/test_controller.rb
13. add @message = "hello" to hello method
14. look at newsite/app/views/test/hello.html.erb
15. add <p> <%= @message %></p> to file
16. **rails server**
17. load page in browser: <http://localhost:3000/test/hello>

An error page occurs (sorry I have no screenshot - post it if you can). I think it has to do with a missing system path variable - but I am not entirely sure. To fix this:

1. Install NodeJs from <http://nodejs.org/dist/v0.10.22/node-v0.10.22-x86.msi>
2. Restart your computer.

To run the example code RubyLect10.progs in Windows 8:

1. Download the RubyLect10.progs from Moodle (<https://csimoodle.ucd.ie/moodle/mod/resource/view.php?id=20485>)
2. Extract the zip to your Ruby Project folder (e.g. C:\RubyMine\temp)
3. Run command prompt
4. Navigate to the extracted zip
- 5.

```
C:\RubyMine\temp>cd RubyLect10.progs
C:\RubyMine\temp\RbubyLect10.progs>cd hello
C:\RubyMine\temp\RbubyLect10.progs\hello>rails server
C:/Ruby200/lib/ruby/gems/2.0.0/gems/sqlite3-1.3.4-x86-mingw32/lib/sqlite3.rb:6:in `require': cannot load such file -- sqlite3/sqlite3_native (LoadError)
        from C:/Ruby200/lib/ruby/gems/2.0.0/gems/sqlite3-1.3.4-x86-mingw32/lib/sqlite3.rb:6:in `rescue in <top (required)>'
        from C:/Ruby200/lib/ruby/gems/2.0.0/gems/sqlite3-1.3.4-x86-mingw32/lib/sqlite3.rb:2:in '<top (required)>'
        from C:/Ruby200/lib/ruby/gems/2.0.0/gems/bundler-1.3.5/lib/bundler.rb:72:in `require'
        from C:/Ruby200/lib/ruby/gems/2.0.0/gems/bundler-1.3.5/lib/bundler.rb:72:in `block (2 levels) in require'
        from C:/Ruby200/lib/ruby/gems/2.0.0/gems/bundler-1.3.5/lib/bundler.rb:70:in `each'
        from C:/Ruby200/lib/ruby/gems/2.0.0/gems/bundler-1.3.5/lib/bundler.rb:70:in `block in require'
        from C:/Ruby200/lib/ruby/gems/2.0.0/gems/bundler-1.3.5/lib/bundler.rb:59:in `each'
        from C:/Ruby200/lib/ruby/gems/2.0.0/gems/bundler-1.3.5/lib/bundler.rb:59:in `require'
        from C:/Ruby200/lib/ruby/gems/2.0.0/gems/bundler-1.3.5/lib/bundler.rb:49:in `require'
        from C:/RubyMine/temp/RubyLect10.progs/hello/config/application.rb:52:in `require'
        from C:/Ruby200/lib/ruby/gems/2.0.0/gems/railties-3.1.0/lib/railties.rb:52:in `block in <top (required)>'
        from C:/Ruby200/lib/ruby/gems/2.0.0/gems/railties-3.1.0/lib/railties.rb:49:in `tap'
        from C:/Ruby200/lib/ruby/gems/2.0.0/gems/railties-3.1.0/lib/railties.rb:49:in `<top (required)>'
        from script/rails:6:in `require'
        from script/rails:6:in `<main>'
```

6.

```
C:\RubyMine\temp\RubyLect10.progs\hello>bundle update sqlite3
Fetching gem metadata from http://rubygems.org/.....
Fetching gem metadata from http://rubygems.org/..
Resolving dependencies...
Using rake <0.9.2>
Using multi_json <1.0.3>
Using activesupport <3.1.0>
Installing bcrypt-ruby <3.0.1>
Using builder <3.0.0>
Using i18n <0.6.0>
Using activemodel <3.1.0>
Using erubis <2.7.0>
Using rack <1.3.4>
Using rack-cache <1.0.3>
Using rack-mount <0.8.3>
Using rack-test <0.6.1>
Using hike <1.2.1>
Using tilt <1.3.3>
Using sprockets <2.0.2>
Using actionpack <3.1.0>
Using mime-types <1.16>
Using polyglot <0.3.2>
Using treeTop <1.4.10>
Using mail <2.3.0>
Using actionmailer <3.1.0>
Using arel <2.2.1>
Using tzinfo <0.3.30>
Using activerecord <3.1.0>
Using activeresource <3.1.0>
Using ansi <1.3.0>
Using bundler <1.3.5>
Using coffee-script-source <1.1.2>
Using execjs <1.2.9>
Using coffee-script <2.2.0>
Using rack-ssl <1.3.2>
Using json <1.6.1>
Using rdoc <3.10>
Using thor <0.14.6>
Using railties <3.1.0>
Using coffee-rails <3.1.1>
Using jquery-rails <1.0.14>
Using rails <3.1.0>
Using sass <3.1.10>
Using sass-rails <3.1.4>
Using sqlite3 <1.3.8>
Using turn <0.8.3>
Using uglifier <1.0.3>
Your bundle is updated!
```

7.

```
C:\RubyMine\temp\RbLect10.progs\hello>rails server
=> Booting WEBrick
=> Rails 3.1.0 application starting in development on http://0.0.0.0:30
=> Call with -d to detach
=> Ctrl-C to shutdown server
[2013-11-15 04:28:27] INFO  WEBrick 1.3.1
[2013-11-15 04:28:27] INFO  ruby 2.0.0 <2013-06-27> [i386-mingw32]
[2013-11-15 04:28:27] INFO  WEBrick::HTTPServer#start: pid=124 port=300
```

8. Now the server runs. Request the page in a browser:

<http://localhost:3000/other/name>

9. When you request the page in a browser you get an error because of a missing or incompatible version of the bcrypt-ruby gem (see screenshots below)

10.

```
[2013-11-15 04:28:44] ERROR LoadError: cannot load such file -- 2.0/bcrypt_ext
  C:/Ruby200/lib/ruby/gems/2.0.0/gems/activesupport-3.1.0/lib/activesupport/dependencies.rb:240:in `require'
    C:/Ruby200/lib/ruby/gems/2.0.0/gems/activesupport-3.1.0/lib/activesupport/dependencies.rb:240:in `block in require'
      C:/Ruby200/lib/ruby/gems/2.0.0/gems/activesupport-3.1.0/lib/activesupport/dependencies.rb:223:in `block in load_dependency'
        C:/Ruby200/lib/ruby/gems/2.0.0/gems/activesupport-3.1.0/lib/activesupport/dependencies.rb:640:in `new_constants_in'
          C:/Ruby200/lib/ruby/gems/2.0.0/gems/activesupport-3.1.0/lib/activesupport/dependencies.rb:223:in `load_dependency'
            C:/Ruby200/lib/ruby/gems/2.0.0/gems/activesupport-3.1.0/lib/activesupport/dependencies.rb:240:in `require'
              C:/Ruby200/lib/ruby/gems/2.0.0/gems/bcrypt-ruby-3.0.1-x86-mingw32/lib/bcrypt_ext.rb:2:in `<top (required)>'
                C:/Ruby200/lib/ruby/gems/2.0.0/gems/activesupport-3.1.0/lib/activesupport/dependencies.rb:240:in `require'
                  C:/Ruby200/lib/ruby/gems/2.0.0/gems/activesupport-3.1.0/lib/activesupport/dependencies.rb:240:in `block in require'
                    C:/Ruby200/lib/ruby/gems/2.0.0/gems/activesupport-3.1.0/lib/activesupport/dependencies.rb:223:in `block in load_dependency'
                      C:/Ruby200/lib/ruby/gems/2.0.0/gems/activesupport-3.1.0/lib/activesupport/dependencies.rb:640:in `new_constants_in'
                        C:/Ruby200/lib/ruby/gems/2.0.0/gems/activesupport-3.1.0/lib/activesupport/dependencies.rb:223:in `load_dependency'
                          C:/Ruby200/lib/ruby/gems/2.0.0/gems/activesupport-3.1.0/lib/activesupport/dependencies.rb:240:in `require'
                            C:/Ruby200/lib/ruby/gems/2.0.0/gems/bcrypt-ruby-3.0.1-x86-mingw32/lib/bcrypt.rb:12:in `<top (required)>'
                              C:/Ruby200/lib/ruby/gems/2.0.0/gems/activesupport-3.1.0/lib/activesupport/dependencies.rb:240:in `require'
```

```
C:/Ruby200/lib/ruby/gems/2.0.0/gems/rack-1.3.4/lib/rack/lock.rb:55:in `call'  
C:/Ruby200/lib/ruby/gems/2.0.0/gems/actionpack-3.1.0/lib/action_controller/middleware/static.rb:53:in `call'  
C:/Ruby200/lib/ruby/gems/2.0.0/gems/railties-3.1.0/lib/rails/engine.rb:55:in `call'
```

11. Press CTRL-C in the terminal to shut down the server. Press y and enter. (see screenshot below)

12.

```
[2013-11-15 04:33:45] INFO  going to shutdown ...  
[2013-11-15 04:33:45] INFO  WEBrick::HTTPServer#start done.  
Exiting  
Terminate batch job (Y/N)? y
```

13. The following step is not necessary but will make you understand the problem. Notice the **ruby x86-mingw32** post-fix to the gem.

14.

```
C:\RubyMine\temp\RyLect10.progs\hello>gem list bcrypt-ruby  
*** LOCAL GEMS ***  
bcrypt-ruby (3.0.1 ruby x86-mingw32)
```

15.

```
C:\RubyMine\temp\RyLect10.progs\hello>gem uninstall bcrypt-ruby  
You have requested to uninstall the gem:  
  bcrypt-ruby-3.0.1-x86-mingw32  
  
activemodel-3.1.0 depends on bcrypt-ruby (>= 3.0.0)  
If you remove this gem, these dependencies will not be met.  
Continue with Uninstall? [yN] y  
Successfully uninstalled bcrypt-ruby-3.0.1-x86-mingw32
```

16. Make sure you type the next command correctly, its: **gem install bcrypt-ruby -version=3.0.1 --platform=ruby**

17.

```
C:\RubyMine\temp\RybyLect10.progs\hello>gem install bcrypt-ruby --version=3.0.1 --platform=ruby
Fetching: bcrypt-ruby-3.0.1.gem (100%)
Temporarily enhancing PATH to include DevKit...
Building native extensions. This could take a while...
Successfully installed bcrypt-ruby-3.0.1
Parsing documentation for bcrypt-ruby-3.0.1
unable to convert "\x90" from ASCII-8BIT to UTF-8 for lib/bcrypt_ext.so
Installing ri documentation for bcrypt-ruby-3.0.1
1 gem installed
```

18. Again the following step is not necessary, compare the bcrypt-ruby gem description to the one in step 13. See the difference :)

19.

```
C:\RubyMine\temp\RybyLect10.progs\hello>gem list bcrypt-ruby
*** LOCAL GEMS ***
bcrypt-ruby (3.0.1)
```

20. Execute the server again: **rails server** and request the page in the browser.

(<http://localhost:3000/other/name>)

21.

The screenshot shows a web browser window with the following details:

- Title Bar:** Shows multiple tabs, with the active tab being "localhost:3000/other/name".
- Page Content:**
 - Title:** The Monster & Leinster Bank
 - Image:** A logo for "MONSTER ENERGY" featuring a stylized green monster face.
 - Text:** We are very pleased to welcome you our home mortgage page.
 - Form:** A text input field labeled "First Name:" followed by a "Last Name:" input field and a "send" button.
 - Text at Bottom:** Monster & Leinster Inc. accept no responsibility if you fall into arrears on your loan and are not responsible for the thugs that will call around to your house with baseball bats to help you with your repayment schedule.

22. If you see an error page (sry I have no screenshot - post it if you can) you might be missing a system path variable - but I am not entirely sure. To fix this:
23. Install NodeJs from <http://nodejs.org/dist/v0.10.22/node-v0.10.22-x86.msi>
24. Restart your computer.
25. Run command prompt, navigate to the hello folder, execute the **rails server**, and request the page.