



D06 - Training Ruby on Rails

Authentication and permissions

Staff 42 bocal@staff.42.fr

Stéphane Ballet balletstephane.pro@gmail.com

Summary: You will discover the wonderful world of rights, through cookies, flags, permissions and privileges.

Incidentally, Rails has a clear error handling, use. At this stage of your learning the 'error-driven development is a good option!

Contents

I	Preamble	2
II	instructions	3
III	Specific Rules of the day	5
IV	Exercise 00: It's me	6
V	Exercise 01: Add me in	8
VI	Exercise 02: Need an account	10
VII	Exercise 03: Peer edit	12
VIII	Exercise 04: UvDv	13
IX	Exercise 05: Can you?	14

Chapter I

Preamble

And speaking of permissions and privileges, a small letter Mr Gates dated but that is pretty funny in view of history.

-2-

February 3, 1976

An Open Letter to Hobbyists

To me, the most critical thing in the hobby market right now is the lack of good software courses, books and software itself. Without good software and an owner who understands programming, a hobby computer is wasted. Will quality software be written for the hobby market?

Almost a year ago, Paul Allen and myself, expecting the hobby market to expand, hired Monte Davidoff and developed Altair BASIC. Though the initial work took only two months, the three of us have spent most of the last year documenting, improving and adding features to BASIC. Now we have 4K, 8K, EXTENDED, ROM and DISK BASIC. The value of the computer time we have used exceeds \$40,000.

The feedback we have gotten from the hundreds of people who say they are using BASIC has all been positive. Two surprising things are apparent, however. 1) Most of these "users" never bought BASIC (less than 10% of all Altair owners have bought BASIC), and 2) The amount of royalties we have received from sales to hobbyists makes the time spent of Altair BASIC worth less than \$2 an hour.

Why is this? As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid?

Is this fair? One thing you don't do by stealing software is get back at MITS for some problem you may have had. MITS doesn't make money selling software. The royalty paid to us, the manual, the tape and the overhead make it a break-even operation. One thing you do do is prevent good software from being written. Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distribute for free? The fact is, no one besides us has invested a lot of money in hobby software. We have written 6800 BASIC, and are writing 8080 APL and 6800 APL, but there is very little incentive to make this software available to hobbyists. Most directly, the thing you do is theft.

What about the guys who re-sell Altair BASIC, aren't they making money on hobby software? Yes, but those who have been reported to us may lose in the end. They are the ones who give hobbyists a bad name, and should be kicked out of any club meeting they show up at.

I would appreciate letters from any one who wants to pay up, or has a suggestion or comment. Just write me at 1180 Alvarado SE, #114, Albuquerque, New Mexico, 87108. Nothing would please me more than being able to hire ten programmers and deluge the hobby market with good software.

Bill Gates
Bill Gates
General Partner, Micro-Soft

Chapter II

Instructions

- Only this page serve as a reference: Do not fi rm the hallway noise.
- The subject can change up to an hour before rendering.
- If no contrary information is explicitly present, you must assume the following language versions:
 - Ruby 2.3.0
 - for d09 Rails> 5
 - but for all the other days rails 4.2.6
 - HTML5
 - CSS 3
- We prohibit EXPRESSLY using keywords while, for, redo, break, retry and until in Ruby source code that you will make. Any use of these keywords is considered cheating (and / or unfit), giving you the score of -42.
- The exercises are precisely ordered from simple to more complex. In any case, we do not bear attention or take into account a complex exercise if a simpler exercise is not very successful.
- Attention to the rights of your fi les and your directories.
- You must follow the rendering process for all your exercise: only the present work on your GIT repository will be evaluated in defense.
- Your exercises will be evaluated by your pool mates.
- You should leave in your repertoire no other fi le than those explicitly speci fi ed by the forward drills.
- You have a question ? Ask your neighbor to the right. Otherwise, try your left neighbor.
- Your reference manual called Google / man / Internet /
- Think discuss on the forum of your pool Intra!
- Read the examples carefully. They might require things that

not otherwise specified in the subject ...

- Please, by Thor and Odin! Re fl échissez dammit!

chapter III


Specific Rules of the day

- All the work of the day will be improved versions of the same application rails.
- Any addition to the Gem fi provided in the resources is prohibited.
- Any global variable is prohibited.
- You must provide a seed, you're going to Implement the fi exercises of the day. For correction, no exercise will only be valid if imple- mented fonctionnalités are depicted within the data from the db.
- you must imperatively handle errors. No error page rails will be tolerated for correction.
- You must have a score awarded by rubycritic higher or equal to 90
- You must be NO 'warning' reported by rails_best_practice
- You must ALLOW intact fi le "rails_best_practices.yml" and place it in the root of "app / con fig /" (it contains con fi g softened to d06)

Tips: If you want your corrections are simple, fast and friendly, write tests! It simplifies life for everyone and it is a crucial habit, nay, MANDATORY make for your future.

chapter IV

Exercise 00: It's me

	Exercise: 00 Exercise
00: It's me	
Render Folder: ex 00 / Files to make: LifeProTips	
Authorized Functions: Notes: n	
/ A	

You must create an app brand new. She will complexification fi ed all day, so pity (for you), code properly and use git. If the authentication has been encoded with your seat, it will annoy you all day ...

Where was I ... Oh yes! The authentication. You must create a decent, with encryption, redirect and everything goes well. Serve you has_secure_password.

Creez for now a 'root page' and make a vacuum line user page header (you are not designers, I know, but a layout -peu- please '):

"Welcome Stéphane! Log_out"

If visteur is unknown, his name will consist of a random animal name followed by "_visiteur":

"! Welcome pony_visitor sign_in log_in" The animal name at

random is like this:

```
url = "https://www.randomlists.com/random-animals" doc = HTML :: Nokogiri (open
(uri)) doc.css name = ( 'li'). first.css ( 'span'). text
```


So you need to do that:

- An unregistered user is greeted with "Welcome <animal_name> _visitor" followed by a registration link and login.
- The anonymous user name stored in a cookie valid for one minute (easier for corrections). Thus, after a minute, his name will be changed to the next loading.
- After registering, the new user is automatically logged.
- The users must have a name and an email UNIQUE.
- The registration form will not let the apparent passwords.
- Passwords do not appear bright in the log server either.
- The registration form asks:
 - A name
 - An email
 - A password (min 8 char)
 - A password_confirmation
- The form has error checking with a message.
- You must create a seed with some users.
- **You should have tests.**
- The utility 'rails_best_practice' must not return any warning:

```
%> Rails_best_practices
Source Code: | ===== | Please go to
http://rails-bestpractices.com to see more Rails Best Practices. No warning found. Cool!
```


chapter V

Exercise 01: Add me in

	Exercise 01 Exercise 01:
Add me in	
Render Folder: ex 01 / files to make: LifeProTips	
Authorized Functions: Notes: n	
/ A	

Ok great! We have a DB and users of authentication. Sum- mayor is, but it's a start.

You will add a section admin in order to administer users. For safety it will be based on a design pattern that uses namespaces.

In short: the URL for the actions of requirment admins rights com- menceront by "/ admin / ...". This allows you to Implement a controller For users without conflict with the former.

The new controller will have prototype like this:

```
:: class Admin UsersController < ApplicationController

# =====
# code here
# ===== end
```

Thus, the URL `http://localhost:3000/admin/users` is available and leads to the created page.

I hope for you that you used good agreements and that you therefore `current_user` is the user logged or if the cookie.

Directors must be differentiated by `admin` true field or false (default false) in the users table.

Only they can access the list of all users and edit values (name and email), and remove them.

Each user can, however, change his name or email.


When a user is admin its connection, the message is followed by a link that leads to more page `admin / users` (the index of our new controller):

"Welcome Stéphane! `Administrate_users` `Log_out`" In addition to 'callback', you will need to

properly configure your **routes** .

chapter VI

Exercise 02: Need an account

	Exercise 02 Exercise 02:
Need an account	
Render Folder: <i>ex 02</i> / Files to make: LifeProTips	
Authorized Functions: Notes: n	
/ A	

The basic back-office is almost finished, we'll add some meaning to our app. In effect, the LifeProTips application is intended to receive and share "life pro tips" but it remains somewhat advantageous tricks we will restrict access to members, but not too ...

You must create using a scaffolded Post an object that contains:

- a 'user_id' (required)
- a 'title' (single and longueure min 3 char)
- a 'content' (which must contain a lot of text)

Obviously, the author does not put its id itself to the establishment is in control- them that the thing is done automatically with `current_user`

The index must list the titles and authors by name and sorted all time-high (youngest first).

Only the index is visible to visitors, the title should be a link to the action 'show post' and if a visitor clicks on it, it must be redirected to the root page 'with instructions like: "You need an account pour voir cette "

The new root page is index posts

A user must be able to edit all posts (for now).


Repeat the same operation as in FY01: an admin page for posts or administrator has all the rights.

The line in the header is (if one is admin):

"Welcome Stéphane! Administarte_users Administrate_posts Log_out" The link "Administrare_users" leads to a page listing all created posts. This one allows access to deletion and modification.

chapter VII

Exercise 03: Peer edit

	Exercise 03 Exercise 03:
Peer edit	
Render Folder: <i>ex 03</i> / Files to make: LifeProTips	
Authorized Functions: Notes: n	
/ A	

Although badly resigned one who believes to know everything.

The Life Pro Tips are editable, you have a ffi expensive on the show page for each post by whom and when the post was modified last time:

Original author: bob2

Title: How to tell if the water is too hot?

Content: Dip your baby in it and check its temperature with the inside of your wrist Edited by: Stéphane

Date of Modification: 2016-07-19 1:55:51 p.m. UTC


You need to Implement the functionality that is represented by the last two lines of the example

You are free to choose your solution. There are many but the good are rare ..

Warning no global variables or cookie-based solutions

chapter VIII

Exercise 04: UvDv

	Exercise 04 Exercise
04: UvDv	
Render Folder: ex 04 / Files to make: LifeProTips	
Authorized Functions: Notes: n	
/ A	

Add on the page "post show", the possibility of 'Upvote' and 'downvote' and an affichage of the total votes (possibly negative).

Vote on the same page redirects

Nb: no global variable or another of this style .The votes have to be an object that inherits from ActiveRecord.


Again seen must have an admin interface as for exerecie 01 (with the same design).

This interface will have a view that lists the votes grouped by posts indicating the title of the post in the header and the names of associated voting.

The administrator can remove votes.

chapter IX

Exercise 05: Can you?

	Exercise 05 Exercise 05:
Can you?	
Render Folder: ex 05 / Files to make: LifeProTips	
Authorized Functions: Notes: n	
/ A	

You need for this last exercise of the day, set up privileges based on the total votes.

- 0 to 3: no specific law
- 3 to 6: right to upvote
- 6 to 9: right to downvote
- above 9 right to edit posts

Now a user can edit his posts, unless it is above 9 upvote.

The detail of the user page (user show) affi che what right has the user in question, and these rights also fi gurent on page user administration.